Patterns for Person-Centered e-Learning

Dissertation
zur Erlangung des akademischen Grades
Doktor der Sozial- und Wirtschaftswissenschaften
(Dr. rer. soc. oec.)

Dissertationsgebiet: Wirtschaftsinformatik

eingereicht an der Universität Wien

von

Mag. Michael Derntl

unter Betreuung von

ao.Univ.Prof. Dipl.Ing. Dr. Renate Motschnig

Wien, im August 2005

ABSTRACT

Current advanced approaches to learning and teaching take advantage of the widespread potential of Information and Communication Technology to support learners in constructing knowledge. One promising, novel approach that builds upon the well founded theory of the Person-Centered Approach is Person-Centered e-Learning (PCeL). PCeL is based on the hypothesis that time and resources can be set free if administrative processes and pure transfer of information are deferred to the computer by employing elements of e-learning. Time and (organizational) resources allocated this way can be used in face-to-face sessions to spend more effort on immediate interactions with/among students and on facilitating activities that lead to cooperative and creative construction of knowledge.

Several case studies showed that introducing PCeL is more demanding with respect to time, competence, and facilitative skills than conventional teaching. This is the starting point for the basic research question underlying this thesis: How can successfully conducted PCeL scenarios be captured and disseminated to enable subsequent reuse across organizational boundaries? For this enterprise the thesis proposes a pattern-based approach to PCeL. Note that, in this context, the development of facilitative dispositions in educators is considered essential but is outside the scope of this thesis.

The pattern approach is rooted in the field of architecture, and today a widely accepted method to enable reuse of expert experience particularly in software design. Generally, a pattern generically describes the core of the solution to a problem that is frequently recurring within a specific context. This thesis introduces a methodology for PCeL pattern mining, description, scenario modeling using the Unified Modeling Language (UML), instantiation on learning platforms, and evaluation based on BLESS, a layered model of blended learning systems. To guide instructors in instantiating PCeL scenarios on e-learning platforms, the patterns provide sets of “Web templates” that employ and arrange basic functionality of current Web-based learning platforms to generically specify how to optimally support the underlying learning scenario. The whole PCeL pattern collection is stored in a repository that is organized into different pattern packages.

The underlying research method is grounded in a theory-guided, cyclic Action Research approach aiming to improve both theory and practice of PCeL and PCeL patterns. To show the feasibility and applicability of the proposed concept, the Web service-based CEWebS learning platform architecture is employed to present a prototypical implementation of some of the patterns’ Web templates: The prototype shows how the pattern-based, modular Web services are used to derive, construct, and to populate an online learning platform space for an exemplary PCeL course.

ZUSAMMENFASSUNG

Neue fortgeschrittene Ansätze des Lehrens und Lernens nutzen die Vorteile von aktuellen Informations- und Kommunikationstechnologien, um Lerner bei der Wissensaufnahme zu unterstützen. Ein neuer und vielversprechender unter diesen Ansätzen ist Personzentriertes e-Learning (PCeL). PCeL basiert auf den Grundsätzen des Personzentrierten Ansatzes und auf der Hypothese, dass Zeit und Ressourcen freigesetzt werden können, wenn pure administrative und Informationstransferaufgaben unter Einsatz von Methoden und Werkzeugen des e-Learning durch Computer unterstützt werden. So freigesetzte zeitliche und organisatorische Überschüsse können in den Präsenz- und Onlinephasen verwendet werden, um jene Aspekte der Lernprozesse zu verstärken und zu begleiten, die zu kooperativer und kreativer Aufnahme von Wissen und praktischen Fähigkeiten führen.

Studien zeigen, dass die Ein- und Durchführung von PCeL aufwändiger ist bezüglich Zeit und fördernde zwischenmenschliche Einstellungen und Kompetenzen, als konventionelle Lehre. Genau hier setzt die Forschung der vorliegenden Dissertation an, indem untersucht und präsentiert wird, wie erfolgreiche oder effektive PCeL-Szenarien festgehalten und verbreitet werden können, um Wiederverwendung zu fördern und Aufwand zu reduzieren. Um dies zu erreichen, schlägt die Arbeit einen „musterbasierten“ Ansatz vor („Patterns“). Man beachte dabei, dass die Entwicklung der für PCeL charakteristischen fördernden Einstellungen der Lehrenden als essenziell angesehen wird, jedoch nicht Gegenstand dieser Arbeit ist.

Patterns wurden ursprünglich in der Architektur gesucht und beschrieben und sind heute weit verbreitet als eine Methode der einheitlichen Beschreibung von Expertenwissen, speziell im Bereich des Softwareentwurfs. Ein Pattern stellt dabei eine generische Beschreibung eines wiederkehrenden Entwurfsproblems in einem bestimmten Kontext dar. Die vorliegende Arbeit präsentiert in Anlehnung daran eine Methode für PCeL Patterngewinnung, -beschreibung, -modellierung mit Hilfe der Unified Modeling Language (UML), -instanzierung auf Lernplattformen, und -evaluierung basierend auf „BLESS“, einem Schichtenmodell des Blended Learning. Um die Instanzierung von PCeL-Szenarien auf Lernplattformen zu unterstützen, beinhalten die Patterns eine Menge von „Web Templates“. Diese beschreiben, wie Werkzeuge und Technologien des Internet eingesetzt werden können, um die Aktivitäten des darunterliegenden Lernszenarios online zu unterstützen. Verwandte Patterns sind in Patternpaketen zusammengefasst und in einer Patternsammlung abgelegt.

Die eingesetzte Forschungsmethodik basiert auf theoriegeleiteter und zyklischer Aktionsforschung, die darauf abzielt, sowohl Theorie als auch Praxis von PCeL und PCeL-Patterns weiter zu entwickeln. Um die Möglichkeiten und die Anwendbarkeit des vorgeschlagenen Konzepts zu zeigen, wurde ein web-basierter Prototyp entwickelt, der einige der Web Templates der Patterns auf der CEWebS Lernplattform implementiert. Anhand des Prototypen wird gezeigt, wie die Onlineunterstützung für einen beispielhaften PCeL-Kurs abgeleitet, erzeugt, konfiguriert, und verwendet werden kann.

ACKNOWLEDGEMENTS

Every aspect of working on this thesis was challenging and interesting at the same time, especially the interdisciplinary nature of research that had to be done. Coming from computer science and business informatics, I was able to advance and integrate personal knowledge and experience in fields such as learning and e-learning theory and technology, humanistic psychology, qualitative and quantitative inquiry, Web technologies, pattern applications and research, conceptual modeling techniques, computer and Web application programming, and research methodology in general.

I realize that this was made feasible and achievable in the first place by my doctoral mentor and facilitator, Renate. I doubt that I would have completed this effort without the security of being able to fully rely on her professional and personal support.

Many moments in the process of writing the thesis were not so enjoyable. In this respect I would like to thank my significant other, Birgit. She showed understanding and acceptance even in the numerous months in the finalization phase of this thesis when I was sitting in front of the computer for 15, 17, and more hours per day. Equal thanks go to my parents. Despite of our local displacement, I was always sure of their unconfined support. Special thanks also go to my colleagues who contributed to my personal development process during my work at the department and lab, especially to Jürgen Mangler and Hannes Lischka, and (listed alphabetically) to Kathrin Figl, Alex Hampel, Monika Hofer, Erich Schikuta, Heinz and Kurt Stockinger (my favorite twins), Helmut Wanek, and Don Thomas Weishäupl. To Prof. Günther Vinek for fruitful discussions during the research seminars; to Prof. Dimitris Karagiannis for useful hints and suggestions as co-referee of this thesis; to Gerhard Specht and Harald Kühn at BOC Austria for providing me with help and assistance regarding the modeling toolkit ADONIS. Last but not least, thanks also go to numerous anonymous persons who attended my presentations related to the research of this thesis at conferences and meetings, for providing useful, constructive comments and suggestions.

A special, cordial *thank you* goes to my friends for the hang-out nights and chill-out days, especially to Hari and Mel, and to my playfellows Mike (bowling and cards), Gerald and Martin (cards). I was also always happy to see my cousin Christian and my former "boss" Reinhard when visiting my hometown Linz.

* * *

Just let this work be referenced frequently, instead of collecting dust in dark libraries!

CONTENTS

FIGURES

TABLES

ABBREVIATIONS

ACM .... Association for Computing Machinery (http://www.acm.org)

ADL .... Advanced Distributed Learning Initiative (http://www.adlnet.org)

AGR.................. AICC Guidelines and Recommendations

AICC................. Aviation Industry Computer Based Training Committee (http://www.aicc.org)

API .... Application Programming Interface

AR .... Action Research

ARIADNE .... Alliance of Remote Instructional Authoring and Distribution Networks for Europe (http://www.ariadne-eu.org)

BLESS .... Blended Learning Systems Structure

CAD .... Computer Aided Design

CAM .... Content Aggregation Model

CBL .... Computer-Based Learning

CBT .... Computer-Based Training

CD-ROM .... Compact Disc – Read Only Memory

CEWebS .... Cooperative Environment Web Services

cf. .... confer

CMC .... Computer-Mediated Communication

CMI................... Computer-Managed Instruction

CMS.................. Content Management System

CSCL .... Computer Supported Collaborative Learning

CSCW .... Computer Supported Collaborative Work

CSV .... Comma Separated Values (this is a common text-based file format for exchange of data between different applications)

DL...................... Distance Learning

DVD.................. Digital Video Disc

e.g. .... *exempli gratia* (Latin, means "for example")

EML.................. Educational Modeling Language

GoF .... "Gang of Four", or "Group of Four": Common abbreviation for the four authors of the seminal "Design Patterns" book: Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides

HCI .... Human-Computer Interaction

HTML................ Hypertext Markup Language

HTTP .... Hypertext Transfer Protocol

i.e. .... *id est* (Latin, means "that is")
ICT .... Information and Communication Technology
IEEE .... Institute of Electrical and Electronics Engineers (http://www.ieee.org)
IMS .... Instructional Managements Systems Project
ISO .... International Standardization Organization (http://www.iso.org)
IT .... Information Technology
LD .... Learning Design
LCMS .... Learning Content Management System
LMS .... Learning Management System
LOM .... Learning Object Metadata
LP .... Learning platform
LTSC .... Learning Technology Standards Committee (http://ltsc.ieee.org)
OMG .... Object Management Group (http://www.omg.org)
OMT .... Object Modeling Technique
OO .... Object-oriented
OOSE .... Object-Oriented Software Engineering
PBL .... Project-Based Learning (also Problem-Based Learning)
PC .... Personal Computer
PCA .... Person-Centered Approach
PCeL .... Person-Centered e-Learning
PCT .... Person-Centered Teaching
PDA .... Personal Digital Assistant
PLoP .... Pattern Languages of Programming
POSA .... Pattern-Oriented Software Architecture
PPP .... Pedagogical Patterns Project
QWAN .... Quality without a Name
RDF .... Resource Description Framework
RLET .... Research Lab for Educational Technologies (at the Faculty of Computer Science, University of Vienna)
RLO .... Reusable Learning Object
RSS .... RDF Site Summary
RTE .... Run-Time Environment
SCO .... Sharable Content Object
SCORM .... Sharable Content Object Reference Model
SIG .... Special Interest Group
SOAP .... Simple Object Access Protocol

UI .... User Interface
UML .... Unified Modeling Language
W3C.................. World Wide Web Consortium (see http://www.w3c.org)
WBL .... Web-Based Learning
WBT.................. Web-Based Training
WG .... Workgroup
WSDL................ Web Service Description Language
WWW .... World Wide Web
WYSIWYG........ What You See Is What You Get
XML .... Extensible Markup Language
ZPD .... Zone of Proximal Development

1 Introduction

1.1 Motivation

The power of the Internet as a resource of knowledge and its protocols as means of information transfer have made the ascent of new, technology-enriched theories and frameworks of learning and teaching just a matter of time. Current advanced e-learning theories take advantage of the widespread potentials of Information and Communication Technology (ICT) to support all aspects of learning, including, but not limited to the transfer of information. One promising approach is Person-Centered e-learning (PCeL), which aims to combine the Person-Centered Approach to teaching and learning as developed by the well-known American psychologist Carl Rogers[1] with elements of e-learning.[2] PCeL builds upon the hypothesis that if mere transfer of information is transferred from lectures to the Internet, face-to-face encounters can fruitfully be used to deepen learning experiences for students as well as for instructors. This can, for example, be achieved through project work, sharing and discussion of different perspectives, exchange of experiences, and collaborative work in teams. [3]

Rogers researched and specified the necessary and sufficient conditions, which have to be held by the instructor to facilitate these person-centered learning processes[4]: *realness* or transparency in the facilitator, *acceptance* or respect towards the learner, and the striving for *empathic understanding* of the learner. On the side of the learner, the application of person-centered principles is not intrinsically bound to specific target areas or age ranges: empirical research has substantiated the benefits of the Person-Centered Approach in educational environments[5] as well as in organizations[6].

In the literature, demands for different modes and qualities of teaching and learning in combining online with face-to-face sessions (also known as *Blended Learning*) have been observed and widely investigated. However, the transition of respective theories into everyday practice clearly lags behind. Apparently, one of the major reasons for the slow adaptation of recent learner-centered theories is their being by far more demanding with respect to several factors:

---

1 Rogers (1983)

2 Motschnig-Pitrik (2001), (2002b), Motschnig-Pitrik and Holzinger (2002)

3 Motschnig-Pitrik and Derntl (2002), (2003a), (2003b), Motschnig-Pitrik, Derntl and Mangler (2003), Motschnig-Pitrik and Mallich (2002)

4 Rogers (1961), (1983)

5 For example, Aspy (1972), Baxter and Gray (2001), Chase and Geldenhuys (2001), Cornelius-White (2003), Cornelius-White et al. (2004), Gamboa et al. (2001), Rogers (1961), Rogers and Freiberg (1994), Tausch and Tausch (1998)

6 Ryback (1998)

- They require instructors to have social skills and to hold facilitative attitudes.
- Higher organizational demand arises in course design and provision of resources, especially when introducing blended, person-centered learning practices for the first time.
- More communication – both organizational and subject-related – is necessary due to the more situated, flexible, and individual style. Communication is also intensified in the case that various specialized people may be involved, which is likely in instructional design efforts[7].
- Designing blended courses is inherently more complex than designing traditional courses due to virtually limitless design and learning activity arrangement options. Although there are numerous individual studies on employing new media in education, a coherent theory on which to hold on in designing blended learning courses is missing. The current state resembles rather a phase of experimentation[8]: reports are mostly descriptive, experience-based, and often lacking clues on how to generalize the employed scenarios to enable transfer to other domains and contexts.

Working Hypothesis

Based on the challenges presented above, the working hypothesis underlying the research and results presented in this thesis is an aggregation of the following basic principles:

- Person-Centered teaching and learning is the means by which ***significant, personally meaningful, lasting learning experiences*** can be facilitated. We view Person-Centered e-Learning as the most promising approach for blended learning. However, it ***requires additional efforts and qualifications*** in the planning, preparation, and conduct of courses.
- Creating a repository of learning design practices for the whole e-learning domain is a preposterous effort. The better guiding principle is to ***build upon a sound pedagogy and support/enrich its value system by means of learning technology***.
- The core of the value system is ***interpersonal relationships*** between learners and the facilitator. ***Learning addresses the whole person***, i.e., intellectual, social and personal aspects.
- Conceptual models, especially when expressed in terms of semi-formal visual models, can greatly ***reduce the cognitive load and complexity in the design of blended learning***. Moreover, through explication of implicit knowledge, these visual models can ***foster communication, knowledge preservation, and exchange*** related to learning design.

7 Cf. Gagnè and Briggs (1979, p. 21)

8 Cf. Nichols (2003)

- Based on object-oriented principles, modeling and describing learning scenarios at ***different levels of granularity and abstraction***, while adhering to a uniform template for description, ***supports (de-)composition, understanding, and reuse of learning designs***.
- Designing for technology-enhanced learning not only includes decisions on instructional methods to be employed, but also on the whole learning environment from the interpersonal, organizational, and technological points of view. Provision of ***clear guidance in designing and instantiating the learning process and its technology environment is considered essential*** for successful implementation. Enhancing the learning environment using tool support is in line with Rogers[9], who claims that the striving for actualization and enhancement "*involves the expansion of effectiveness through the use of tools*."

Starting from scratch, designing courses which utilize ICT is hard for instructors and trainers when lacking prior personal experience. This is particularly true without the presence of a common conceptual basis to build, discuss, and communicate upon. Therefore this thesis introduces a pattern approach to Person-Centered e-learning practices in higher education. Lending from the literature on the pattern approach in architecture[10], software engineering[11], and pedagogy, the term *patterns* is used here to refer to reusable templates for capturing successful practices of managing recurrent tasks. Initially, Alexander developed and used the pattern approach to capture his perceptions of the "timeless way" of designing towns and buildings[12]. His theory is grounded on the observation that each design problem is the result of a certain configuration of forces in a specific context. Describing a way to resolve these forces in general terms unveils one of the main benefits of a pattern, namely its *reusability*.

Combining a number of patterns in a way that allows for decomposing and solving complex problems for a whole target domain characterizes a *pattern language*. To achieve this, the patterns have to be described at different scopes and levels of detail, while conceptually guiding the pattern user through the network of patterns in a pattern language. This is usually accomplished by providing diagrams or textual information on inter-relations between patterns[13].

Patterns of successful Person-Centered e-Learning practices were specified and are presented in this work. Taking into account the major role of learning platforms in blended learning settings, extra value has been added by providing Web-based templates of patterns for online (inter)actions, e.g., the collection and provision of resources, submission and discussion of

9 Rogers (1959)

10 Alexander et al. (1977); see Section 2.2.1.1.1, p. 36

11 For example, Gamma et al. (1995); see Section 2.2.1.1.2, p. 37

12 Alexander (1979)

13 See Section 2.2.2, p. 42

learning contracts, knowledge construction in teams, Web forms for peer-discussion and evaluation, and other frequent interactions. More generally, the approach presented here is targeted at capturing and subsequently reusing effective Person-Centered e-learning patterns and making them widely adoptable and available.

The process of pattern mining is conducted iteratively, theory- and practice-driven in Action Research cycles. Briefly sketched, after a person-centered teaching activity or one of its constituent activities has (repeatedly) shown effective from an empirical, qualitative, and/or the instructor's point of view, a new pattern can be considered. At this point we start to model its structure and generic *learnflow* conceptually by using the Unified Modeling Language (UML)[14], as well as to verbally describe the pattern, its parameters, and other relevant aspects. The process of modeling the flow of activities in patterns has proven to be very instructive and supportive as it gradually increases the transparency of teaching activities by incrementally decomposing a number of activities into groups of frequently recurring activities (i.e., scenario patterns). For example, the dissemination of information is one central aspect in any e-learning setting, so one of the first patterns that emerged is called PUBLISH, a highly reusable pattern whose intent is to abstractly describe the disclosure of an item or a piece of information to a certain target person or target group.

As a result, with an initially sufficient number of patterns available, all specified patterns were stored in the *PCeL pattern repository* and published in a Web-based library[15] in order to provide for quick and easy access using different kinds of search criteria, such as expected number of participants or target skills addressed by a scenario. What is proposed here is a well-documented repository of Person-Centered teaching and learning scenarios, supported by Web page prototypes to manage learning materials, organizational aspects, cooperative learning, etc. This shall contribute to making organizational concerns of learning and teaching easier, more effective, more productive, and shall provide room and guidance for facilitators to make their courses more person-centered in a technology-enhanced environment.

1.2 Goals

Put concisely, the primary goals of this thesis as raised in the motivational context above are:

- ***Contribution***: Presenting a pattern-based approach to capturing, describing, modeling, disseminating, researching, and reusing successful Person-Centered e-learning (PCeL) practices based on an identified need and on a sound methodology.
- ***Elaboration***: Presenting the PCeL pattern repository. Note that thereby the goal is not to present the final version of the repository or a comprehensive approach covering all

[14] Object Management Group (2003)

[15] The *Pattern Web*; see http://elearn.pri.univie.ac.at/patterns

aspects of that domain, but rather a decent, preliminary version that is capable of conveying the usefulness and feasibility of the proposed approach.

- ***Integration***: Showing how pedagogical theory and practice can inform the technological solution to result in a fluent, flexible whole that constitutes more than the sum of its parts.
- ***Validation (proof of concept)***: Show the applicability of the PCeL pattern approach by presenting a pattern-based, prototypical implementation of a Web-based course platform building on the Web templates of the course's constituent patterns.

1.3 Thesis Structure

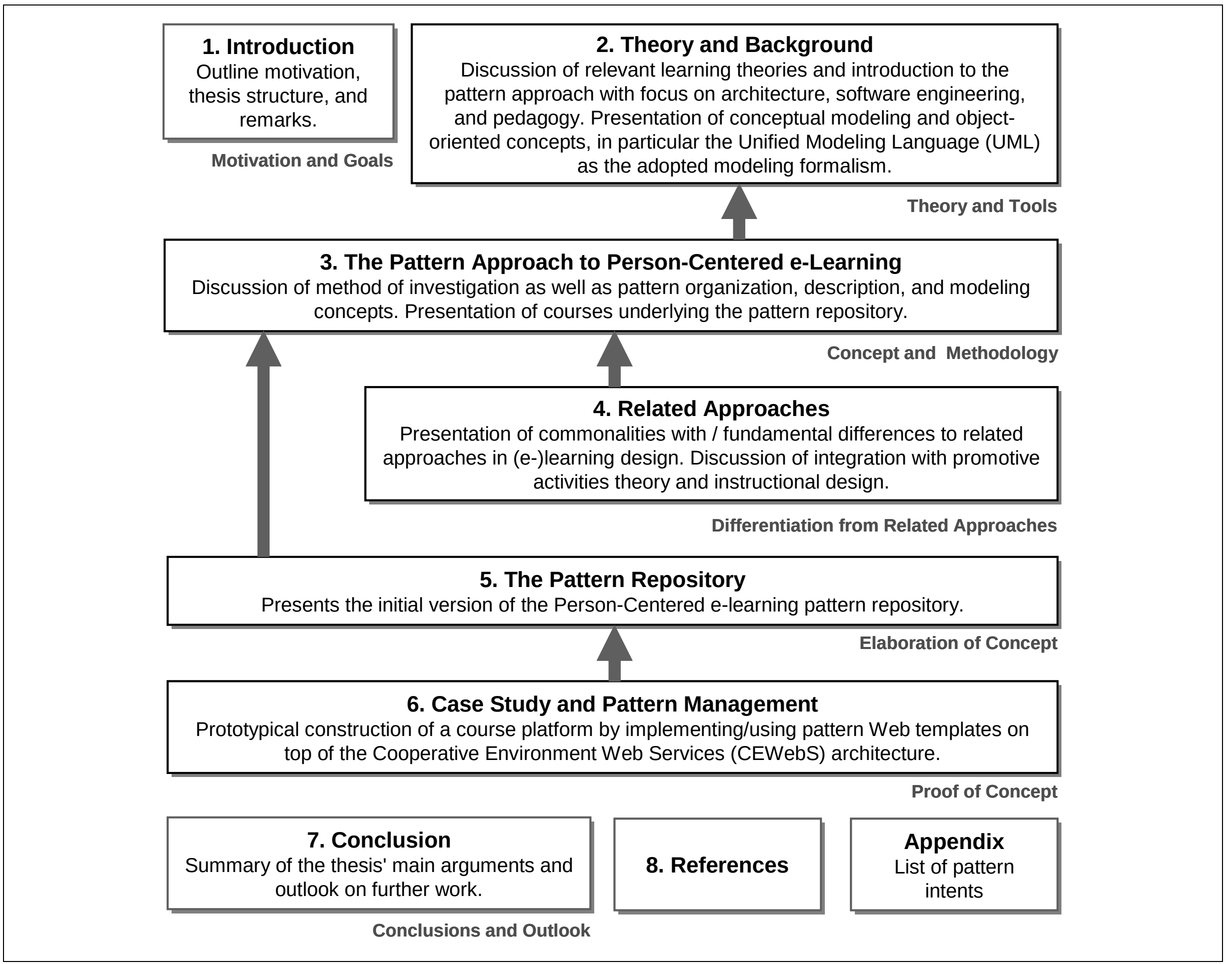


Figure 1: Thesis structure and dependencies among sections.

Figure 1 outlines the thesis structure with short abstracts of the respective Chapters. Arrows connect Chapters with other Chapters they depend on. The dependencies among Chapters are transitive, which means that Chapter 6 is not only dependent on Chapter 5, but also on Chapter 3. However, these dependencies are not intended to convey mandatory reading instructions, but rather to guide the reader through the work.

1.4 Remarks

- To keep the text of this thesis clear, references are cited in footnotes. Footnotes are also used for additional information on certain topics, concepts, or words, whereas none of the additional information in footnotes is required for following the discussion. This should contribute to making the text more readable. References are cited by author name(s) and year of publication. Ideas, block citations, sentences, or concepts originated by other authors are cited standalone. Concepts which are put in other words or in a more dense form here are referenced using "cf." (= confer), meaning that the cited source contains more details. Usually, such citations include page numbers (either single or range of pages) with the exact location of what is referenced in the source.
- Throughout the text, all headings at level 1 (top-level headings) are referred to as "Chapters", while all lower heading levels within Chapters are referred to as "Sections".
- Referencing the Unified Modeling Language (UML) specification[16] requires special attention, as its page numbers are compiled from chapter and page number within the respective chapter; e.g. "p. 3-17" does not reference a range of pages, but page 17 in chapter 3. The same applies to figures and tables in the UML specification.
- For every figure or table in this work information is provided regarding its source, except for figures or tables that were produced totally independently from any source; this is done in a footnote, which is attached to each figure or table caption.
- Words or phrases in *italics* indicate emphasis. Citations by word also are also printed in italic font.
- References to patterns in the text are emphasized by writing the pattern name in small capitals (e.g., ONLINE DISCUSSION).
- Mostly I prefer active wording instead of passive phrases to create a more vivid reading experience. For example, "in the next Section we will present a general introduction to this work" instead of, "in the next Section an introduction to this work is presented."
- Regarding the *he/she* problem I follow Carl Rogers[17]: "*In many cases, I refer to* him *or* her*; [...] sometimes I use* she *and* her *as the generic terms; occasionally I follow some compromise path. I am not completely consistent, but I am very much aware of the injustice done to women by the use of the generic* he*, and I have tried to remedy that situation as best as I can.*"

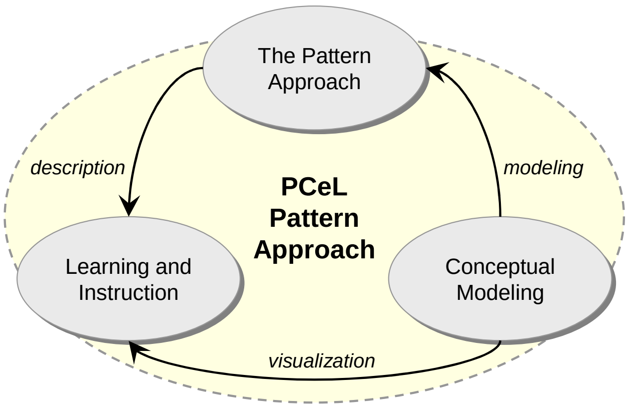
16 OMG (2003)

17 Rogers (1983, p. 4)

2 Theory and Background

This Chapter introduces and discusses relevant cornerstones of the PCeL pattern approach by elaborating relevant theories and background information on:

- ***Learning and Instruction*** (Section 2.1, p. 8): Basic notions and principles related to learning and instruction, including traditional conceptions on learning and instruction, the field of e-learning and blended learning, as well as Person-Centered e-Learning as the didactical baseline for PCeL patterns.
- ***The Pattern Approach*** (Section 2.2, p. 35): History and state-of-the-art in pattern research and practice from the viewpoint of the most relevant disciplines (software, architecture, and pedagogy/e-learning). Pattern organization and description concepts are elaborated, compared, and discussed.
- ***Conceptual Modeling*** (Section 2.3, p. 66): General discussion on the notions of models and concepts, as well as in-depth elaboration of basic object-oriented principles and the Unified Modeling Language as conceptual tools for PCeL pattern modeling and organization.



These three cornerstones, coming from completely different backgrounds and disciplines, build the interdisciplinary foundation of the research and results presented in this work. Note that this is not the usual "state-of-the-art" Chapter, which is due to the fact that the intersection (or better integration) of the three cornerstones constitutes a quite new terrain. In fact, it represents a novel, integrated toolbox and methodology for person-centered blended learning design. The "instruments" that the toolbox comprises are presented separately in this Chapter. Related approaches to learning design are presented and discussed in Chapter 4 *after* the introduction of the PCeL pattern approach.

2.1 Learning and Instruction

According to one of the most prominent definitions[18], "*learning is a change in human disposition or capability, which persists over a period of time, and which is not simply ascribable to the processes of growth.*" Naturally, over time many approaches and theories about learning have emerged and changed. Most of them define their own notions of what learning is and how learning can be initiated, facilitated, and/or supported. The following Sections aim to:

- Provide an overview of prominent traditional and current learning theories;
- Discuss the Person-Centered Approach to teaching and learning as the main theory behind Person-Centered e-Learning;
- Present e-learning and related technology-enhanced approaches to teaching and learning using new media;
- Introduce Person-Centered e-Learning (PCeL) as the didactic baseline of the pattern approach to PCeL, which is introduced later in Chapter 3.

2.1.1 Traditional Theories of Learning

Didactics as a field of science is concerned with all aspects learning and teaching processes. Any concrete didactic method builds upon a learning theory[19], whereas learning theories have their roots in many different sciences, such as in social sciences, neuroscience, or philosophy. Learning theories are also key requisites for understanding and creating specific learning designs (or instructional designs). Basically, most of the works on learning theories differentiate three classical main streams:

- ***Behaviorism*** was one of the dominating learning theories of the twentieth century. It has emerged when a growing group of psychologists broke with introspective approaches to psychological research: the main point of critique was the impracticality of substantiating and verifying its results.[20] So behaviorists turned to concentrate on *behavior*, something perceptible and measurable. Research focused on describing learning in terms of stimuli and corresponding reactions, based on conditioning. Learning was defined as change in behavior. Famous behaviorist contributors include B. F. Skinner, J. B. Watson, I. P. Pavlov, L. Wittgenstein, E. C. Tolman, and C. E. Hull, to mention a few.[21]
- ***Cognitivism*** began to displace behaviorism as the main theory of learning in the 1960's. It is based on the claim that internal, cognitive processes in humans cannot be

[18] Gagnè (1977, p. 3)

[19] Motschnig-Pitrik and Holzinger (2002, p. 163)

[20] Cf. Watson (1997, p. 38-39)

[21] Cf. O'Donohue and Kitchener (1999)

ignored when researching human learning, which was in turn one of the core assumptions of behaviorists. Cognitivists see the learner in a more interactive role within his or her learning environment, not just as a reactor. The role of the teacher also changes: As learning is seen as a process of knowledge acquisition by integrating new information into existing cognitive knowledge structures, the teacher is no more just the instructor or the expert, but more a tutor or enabler of the learning process.[22] Key players in cognitivism were J. Piaget, L. Vygotsky, or G. A. Miller.

- ***Constructivism*** is the most recent learning theory mainstream. Learning is not seen as a pure stimulus-reaction sequence as in behaviorism, nor is it considered to be solely driven by mental and cognitive processes as in cognitivism. Knowledge is not assimilated through training, nor is it held by the learner, but it is seen as being constructed as the learner's understanding of his or her environment based on prior experiences and reflection of current and past situations.[23] In that sense, teachers are not merely considered as trainers, but as facilitators who are coaching the learners in the process of knowledge construction.

Building on these mainstreams (mainly on constructivism, the most recent theory), a host of derived learning theories has evolved for teaching, learning, and development, such as:

- ***Anchored Instruction***: the primary principle of anchored instruction is the solving of complex, realistic problems by providing material (primarily media such as videotapes) that acts as an *anchor* for subsequent learning activities. The aim is to encourage a process of active knowledge construction in the learner[24]. For example, mathematic concepts are learned by using them for solving authentic problems (i.e., the anchors)[25].
- Vygotsky's ***Social Development*** theory[26] deals with the importance of social interaction in the learning process. Learning is a social process where the learner acquires knowledge and ability to perform tasks without assistance. A central concept in this theory is the so-called *zone of proximal development* (ZPD) that denotes the difference between what a person can do with and without help[27].
- The ***Situated Learning*** theory[28] stresses the role of activity, context and culture (i.e., situation) in which learning occurs. It shows links to Vygotsky's social development the-

[22] Cf. Mergel (1998)

[23] Cf. Forrester and Jantzie (1998)

[24] Cf. Cognition and Technology Group at Vanderbuilt (1993)

[25] Cf. Open Learning Technology Corporation (1996)

[26] Vygotsky (1978); the theory is also sometimes called *Social Cognition* theory or *Cognitive Development* theory.

[27] Cf. Vygotsky (1978, p. 86): The ZPD is characterized as, "*the distance between the actual developmental level as determined by independent problem solving and the level of potential development as determined through problem solving under adult guidance or in collaboration with more capable peers.*"

[28] Lave and Wenger (1991)

ory, as social interaction is a central component of situated learning. More recent works have further developed this theory, e.g., Wenger's *Communities of Practice*[29] dealing with groups of people or learners that tackle problems by collaboration, sharing, and information exchange. The social context also plays a major role in the *Cognitive Apprenticeship* theory[30] that stresses the benefits of a relationship between learner and expert (facilitator), where the focus is not on transmission of factual information but rather addresses the social and personal level in the learner by guiding the development of his or her skills and abilities in real-world environments[31] through observation, training, and practice[32].

Many of the newer theories, especially those derivates of constructivism, share some concepts with the main learning theory underlying this thesis (Person-Centered e-Learning), as will become clearer in the following Sections. Moreover, the majority of the technology-enhanced courses today clearly show the influence of constructivist educational principles as the primary instructional method[33].

But first we take a look at key issues and recent developments in the e-learning field.

2.1.2 E-Learning and its Current Key Issues

> *Technology integration requires teachers to alter their teaching processes, no longer being the sole distributor of information. This change in role requires support from many sources in order for the teacher to make the transition.*[34]

2.1.2.1 Opening Remarks

Generally, e-learning refers to teaching and learning activities involving the use of computers and electronic media[35]. Similar arguments apply to related paradigms like Computer-Based Learning, Web-Based Learning, and Online Learning, to mention a few. While there are significant differences, the common major leap in these technology-enhanced approaches was making *asynchronous* learning possible. Obviously, the most salient advantage is that this allows learners to proceed at their own pace and independent of location. On the other hand, the most insistent misconception introduced by the "e-learning hype" was that teaching would

---

29 Wenger (1998)

30 Brown, Collins and Duguid (1989)

31 Cf. Gruber, Mandl and Renkl (1995)

32 Cf. Enkenberg (2001, p. 502)

33 Cf. Bangert (2004)

34 Baylor and Ritchie (2002, p. 401)

35 See for example, Baumgartner, Häfele and Häfele (2002), Ewing and Miller (2002), Govindasamy (2002), Kruse (2002), Nichols (2003), Tsai and Machado (2002)

become more efficient through being less time- and resource intensive. Today, it is widely accepted that designing and conducting technology-enhanced courses and content is usually more expensive with respect to time and other resources, especially for new users and adopters. So, more recently, the focus has changed from mere organizational efficiency to *learning efficiency*. Research in the field has started to recognize the potential influence of these approaches on pedagogy and educational settings: The roles of both learners and instructors are changing in the new paradigm. Instruction and learning environments can become more learner-centered, while the instructor can take on the role of a coach or facilitator in the learning process. From a constructivist viewpoint, such settings lead to deeper and more persistent learning and knowledge construction.

In the history of e-learning, Computer-Based Training is the oldest approach, and perhaps still the most widely recognized one among non-experts. It is a way of delivering learning content via digital media that has emerged decades ago, when barely any computer was connected to another. When world-wide connectedness through the Internet started to spread significantly in the mid 1990's, researchers and practitioners started to recognize the potential of this new[36] medium for learners. Through the Internet, and especially the World-Wide Web (WWW), inter-connected learners were then capable of exchanging material, collaborating online, and even of consulting the instructor online. This gave birth to the then-novel paradigm Web-Based Training/Learning. Recently, the approach of employing personal computers for educational purposes became more and more interesting and important for companies, enterprises, and educational institutions. It has become big business, as professional training and vocational education is necessary for keeping employees up to date with rapidly changing technologies, applications, and environments. As such, e-learning and its satellite concepts have come by with a host of new facets, dimensions, and research directions. This has dispersed the original notion so much that today the term "e-learning" conveys everything and nothing at the same time. For e-learning research and especially for its progress it has become increasingly important that notions be used appropriately and, regrettably to have to say, correctly. Still, it seems that there exists not the sole appropriate notion for a given set of teaching, training, and learning activities involving computers. For example, delivering learning content via the WWW is common property of both Online Learning and Web-Based Training. Apparently, Tsai and Machado[37] were the first in the field to explicitly articulate and tackle this problem by trying to find common sense of frequently used terms and buzzwords[38] in current learning paradigms. In a discussion report, they stress the importance of using e-learning and related terms cautiously and correctly with respect to the addressed learning concepts and activities. In many research resources and reports (e.g.,

---

[36] To be precise, at that time the Internet was surely not new. What was new was the significant and fast-growing proportion of computers world-wide connected to the Internet.

[37] Cf. Tsai and Machado (2002)

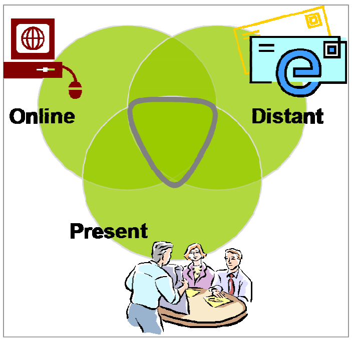
[38] According to Cambridge Advanced Learner's Dictionary, a buzzword is "*a word or expression from a particular subject area that has become fashionable by being used a lot, especially on television and in the newspapers*" (see http://dictionary.cambridge.org/define.asp?key=10609&dict=CALD).

papers, technical reports, articles, books, knowledge bases, or Web pages) covering the field of e-learning, terms are often not used with appropriate accuracy. This becomes evident to the reader, e.g., when very general terms, such as "Web-Based Learning", are used for describing or identifying very specific activities, such as "content delivery via the WWW".

Taking these considerations into account the following Section was designed to introduce terms and definitions related to e-learning in an alphabetically sorted, dictionary-like manner, in a way that allows for identifying commonalities and crucial differences of neighbor approaches.

2.1.2.2 (Towards an) E-Learning Dictionary

Blended Learning



Blended learning[39] is a form of education that mixes (= *blends*) online, distant, and face-to-face teaching and learning scenarios and delivery channels[40]. It is the most current "buzzword" in e-learning practice and research, presumably due to the fact that it covers most of the online and distant learning concepts in combination with face-to-face learning. In this respect, it is not itself a new concept, rather a new term combining existing concepts. The main factor of interest in blended learning is therefore the didactic approach employed to design the instruction and learning experience. The challenge is to find the *right blend*[41] among limitless feasible design possibilities, activity arrangement, and delivery channels, while keeping in mind that a blended learning design should represent significantly more than just adding on to its face-to-face and online components[42].

Computer-Based Learning

Computer-Based Learning (CBL)[43], which is better known as Computer-Based Training (CBT), describes scenarios where digital media (e.g., CD-ROM, DVD, etc.) are used as primary information sources and delivery channels in the learning or training process[44]. It mainly refers to situations where the learner uses standalone multimedia applications locally on her personal computer. Usually, no network connection (Internet or intranet)

---

39 Also referred to as *Hybrid Learning*

40 Cf. Baumgartner, Häfele and Häfele (2002)

41 For example, Schreurs, Moreau and Picart (2003)

42 Cf. Garrison and Kanuka (2004, p. 97), who state that, "*a blended learning design represents a significant departure from either of these [components].*"

43 Also referred to as *Computer-*{*Based*, *Assisted*, *Supported*, *Aided*} {*Training*, *Instruction*, *Learning*, *Education*, *Teaching*}

44 See for example, Baumgartner, Häfele and Häfele (2002), Issing and Klimsa (1997, p. 481-482)

is needed. Assessment of the learner's progress can be done through interactive quizzes and tests. Apart from computer-assisted distance learning approaches, CBL is one of the oldest forms of e-learning[45].

Multimedia functionality as well as computer performance is still constantly increasing, so CBL is big business today: Producing sophisticated multimedia applications, like any complex software product, is a highly resource- and time-intensive enterprise. For that reason, custom CBLs are usually very expensive, but for many large companies buying such CBLs is the only way to further education and training of their masses of employees.

Distance Learning

Distance learning[46] is often subject to the misconception that it is necessarily linked to ICT[47]. Rather, it is an educational approach that is over one century old[48] and is therefore not necessarily linked with computer technology[49], not to mention the Internet. It refers to any setting where teaching and learning occur at different places (i.e., *distant*). Moore and Kearsley[50] identify three generations of distance education paradigms, whereby the newest media employed in the newest generation of distance learning are CD-ROMs, audio conferencing, and video delivery! Passerini and Granger[51] claim that the vast opportunities that the Internet has opened up for distance learning justify giving birth to the *fourth* generation.

E-Learning

E-Learning[52] describes any form of education involving the use of electronic media for instruction and/or learning[53]. However, it is neither a pedagogical approach, nor does it imply a particular learning setting. The definition of e-learning excludes, for example, pure distance learning when no electronic media are involved. Through the emergence of new technology- and media-enabled digital devices such as cell phones, Tablet PCs,

45 Interestingly (at least to the author's perception), most non-experts still think only of CBL when confronted with the term e-learning, e.g., learning something from a multimedia disk.

46 Also referred to as *Distance Education*.

47 Cf. McGorry (2003, p. 159)

48 Passerini and Granger (2000, p. 2)

49 Cf. Tsai and Machado (2002)

50 Moore and Kearsley (1996)

51 Passerini and Granger (2000, p. 3-4)

52 There is no consensus on the writing style of the term e-learning. In the scope of this work, *e-learning* is used in the text, *E-Learning* in captions, and *e-Learning* in proper nouns. Alternative writing styles include *elearning*, *Elearning*, *e-Learning*, or *eLearning*.

53 See for example, Govindasamy (2002, p. 288)

Personal Digital Assistants (PDA), the focus today shifts away from *computer* support to *media* or software support of learning[54].

Mobile Learning

Mobile learning (or *m-learning*) combines the promises of two major fields: e-learning and mobile computing[55]. It refers to teaching and learning settings where media-enabled, mobile computing devices (e.g., PDAs, mobile phones, Tablet PCs, notebooks) are used to enable location-independent – and possibly context-aware[56] – learning of typically small chunks of information. This way, the learner achieves independence from the desktop computer[57]. As such, mobile learning imposes different restrictions and focuses on the delivery of learning content: As mobile devices such as PDAs or cell phones usually have small display sizes and low-bandwidth connectivity, content preparation (information size) and presentation (information layout) are major topics of interest.

Online Learning

According to Nichols, online learning "*describes education that occurs only through the Web.*" [58] There is no face-to-face contact and no physical learning materials are issued to students. In a similar sense[59], online learning is seen as supporting conventional teaching scenarios with Web-based learning environments. While many other definitions and discussions point in the same direction requiring the use of Internet and Web technologies to qualify for online learning[60], Tsai and Machado[61] stress that the concept of online learning has evolved *long before* the Web. In their conception the central requirement is that learning material has to be *readily accessible* to qualify for online learning. According to this definition, learning a software program (e.g., Microsoft Office) by browsing and reading through the readily accessible help contents would also qualify as online learning. Such a conception is in line with Weston et al.[62], who qualify the medium of delivery (e.g., Web or CD-ROM) as irrelevant. Concluding, we identify a significant gap in current perceptions of the essence of online learning. The simplest description is often the most useful: Online education can be seen as a form of distance

54 See also Baumgartner, Häfele and Häfele (2002, p. 5)
55 Cf. Trifonova and Ronchetti (2003, p. 635)
56 For example, Hummel (2003)
57 See e-Learning Centre (2004)
58 Nichols (2003, p. 2)
59 Cf. Seng and Mohamad (2002)
60 See Stewart (2004) for a more detailed account
61 Tsai and Machado (2002)
62 Weston et al. (1999, p. 35)

education taught via the Internet[63], whereas the share of online and face-to-face learning is left open.

Web-Based Learning

Web-based learning[64] (WBL) may be seen as Web-enhanced CBL. The main difference between these two is the method of delivery: While CBL is typically delivered via some hard disc media (e.g., CD-ROM), WBL is rather (but not exclusively) delivered via the Web. In addition, WBL adds another important aspect to the learning process: the ability to interact and cooperate with other learners through the Internet using the metaphor of a *virtual classroom*[65].

2.1.2.3 Technological Aspects

This Section concentrates on technological aspects of e-learning. To implement technology-supported learning scenarios different learning technology tools are available. Any complete e-learning solution has to comprise tools to manage electronic learning content (*e-content*) as well as the learning *process* (e.g., collaboration among learners, assignments, online tests, etc.). Basic notions of Web-enabled learning technology are presented in the following:

A ***learning platform*** (LP) is the entirety of tools and software to organize and facilitate Web-supported learning[66]. Most Web-enabled LPs are designed as client/server applications where the LP server process executes on a Web server that is capable of receiving and processing client requests. A client can be any kind of software program interacting with the server. Mostly, the client program runs in a standard Web browser (i.e., Mozilla Firefox, Internet Explorer, Netscape Navigator, or Opera). Such a solution is feasible only when the LP uses the Internet's standard transmission protocol for data exchange between client and server program (i.e., the Hypertext Transfer Protocol – HTTP[67]) and one of the standard protocols for Web content presentation (e.g., the Hypertext Markup Language – HTML[68]). However, there are other options of implementing learning platforms:

63 Cf. Hailey, Grant-Davie and Hult (2001, p. 387)

64 Also referred to as *Web-Based* {*Instruction*, *Training*, *Education*}

65 Cf. Cronje (2001)

66 Cf. Baumgartner, Häfele and Häfele (2002, p. 16)

67 HTTP is a standard defined by W3C (2003a). According to W3C's Request for Comments #2616 (ftp://ftp.isi.edu/in-notes/rfc2616.txt), "*HTTP is an application-level protocol for distributed, collaborative, hypermedia information systems. It is a generic, stateless, protocol which can be used for many tasks beyond its use for hypertext, such as name servers and distributed object management systems, through extension of its request methods, error codes and headers. A feature of HTTP is the typing and negotiation of data representation, allowing systems to be built independently of the data being transferred.*"

68 HTML is a standard defined by W3C (2004a): "*HTML is the lingua franca for publishing hypertext on the World Wide Web. It is a non-proprietary format [that] can be created and processed by a*

- Monolithic standalone applications that have to be installed locally on the PC, e.g., multimedia applications for CBT. In such solutions, no interaction or networking among learners is needed. Typically these applications are distributed via intranets or data carriers such as CD-ROMs. They bring about some major advantages over Web-based solutions: As no Internet connection is needed, there are no performance restrictions to be considered during development. Thus, development of powerful learning and training programs making extensive use of multimedia is feasible.
- Distributed learning platforms that use proprietary data transmission protocols. In such solutions, special client applications have to be installed locally on the PC, and a connection to an intranet or to the Internet is needed, depending on where the server program is executed.

However, almost all of the top selling and most widely deployed LPs today are Web-based systems using *the* Web standards HTTP and HTML, where just a Web browser and Internet connection are needed for access on the user side. Essentially used synonymously with the concept of Web-based learning platforms is the term ***Learning Management System (LMS)***: LMS are specialized learning technology systems based on Internet and Web technologies to provide education and training[69]. They usually provide tools for[70]:

- Information distribution
- Learning material management (e.g., authoring, content packaging)
- Multiple communication facilities (e.g., chat, discussion, conferencing)
- Course management (e.g., online assessment, learner activity tracking)

Table 1 gives an alphabetical overview of some of the popular LMS solutions used today.

Table 1: Excerpt of popular learning platforms.

Solution	Vendor	Homepage
Blackboard	Blackboard, Inc.	http://www.blackboard.com
CEWebS	University of Vienna (*Open source*)	http://www.cewebs.org
Dayta	tomcom GmbH	http://www.dayta.de
FirstClass	Open Text Corporation	http://www.firstclass.com
ILIAS	Universität Köln (*Open Source*)	http://www.ilias.uni-koeln.de/ios/index-e.html
LearningSpace	IBM (Lotus)	http://www.lotus.com/products/learnspace.nsf/wdocs/homepage?opendocument
Saba	Saba Software Inc.	http://www.saba.com

wide range of tools, from simple plain text editors - you type it in from scratch- to sophisticated WYSIWYG authoring tools. HTML uses tags such as <h1> and </h1> to structure text into headings, paragraphs, lists, hypertext links etc."

69 Cf. Avgeriou et al. (2003b, p. 11)

70 Cf. McCormack and Jones (1997) as cited in Avgeriou et al. (2003b, p. 13)

Solution	Vendor	Homepage
TopClass	WBT Systems	http://www.wbtsystems.com
WebCT	WebCT, Inc.	http://www.webct.com

Often confused with LMS are ***Learning Content Management Systems*** (LCMS) that combine LMS with features of conventional Content Management Systems (CMS): They are used to create, store, and manage ***Reusable Learning Objects*** (RLO) and to support the organization of Web-based courses[71]. This means that LCMS are used for collaborative creation and maintenance of learning content[72]. Thereby, the salient advantage of using RLOs is that they can be exchanged among different LCMS when adhering to supported metadata and content description standards (see the Section on *Standardization Issues*[73] below). Most of the current platforms support standardized import/export of RLOs.

For instructors and content providers the process of content and learning object development and organization in L(C)MS is supported by different kinds of ***authoring tools***[74]. As LMS rely on Web-based technology, the end-user language used for content presentation and linking in Web browsers is HTML. Most LMS providers offer so called *WYSIWYG* (What You See Is What You Get) editors for Web page authoring (see Figure 2).

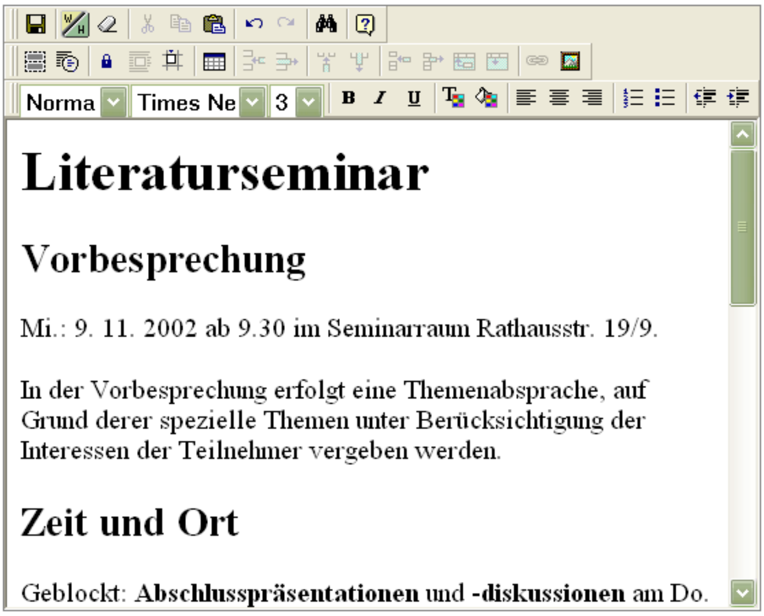


Figure 2: A What You See Is What You Get (WYSIWYG) editor.

71 Baumgartner, Häfele and Maier-Häfele (2003)

72 Ismail (2002)

73 Section 2.1.2.4, p. 19

74 For example, Baumgartner, Häfele and Maier-Häfele (2003, p. 32-34)

However, features of authoring and presenting content and learning objects are only one important aspect of e-learning technology. Other ***e-learning atoms***[75] provided by many LPs include[76]:

- ***Discussion forum***: enables asynchronous online communication among users. Usually, each discussion forum comprises a number of discussion threads that mark off the starting point for an online discussion. Users can subsequently post replies (*posting*) to threads and other postings. Such forums can be anchored to existing Web pages that are dedicated to specific learning activities, or they may be used for open discussions in standalone forums. As communication takes place asynchronously, many LPs provide a notification feature that allows for automatically notifying users of replies to their postings, e.g., through e-mail, RSS feeds[77], or instant messaging facilities provided within or outside of the LP. Figure 3 shows an example of a discussion forum in the dayta LP.

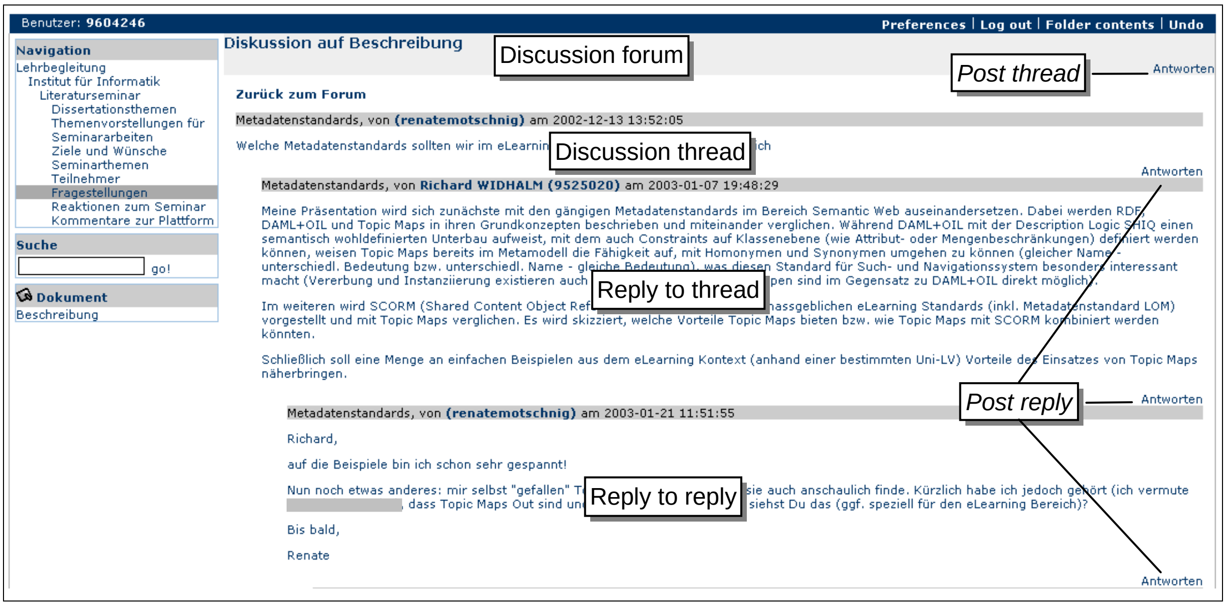


Figure 3: An example of a discussion forum in dayta.

- ***Chat***: used for synchronous online communication in so called *chat rooms*. Messages posted to chat rooms are immediately visible to all participating users, thus allowing for real-time online conversation and interaction.

- ***Questionnaires***: used to collect quantitative and qualitative feedback from users/participants. Platforms that include this feature offer tools for constructing and collecting data from questionnaires.

- ***Tests / Quizzes***: online tests or quizzes are a popular form of assessment in Web-based learning courses. Thereby, the instructor is provided with tools to construct tests

75 Derntl and Motschnig-Pitrik (2004c)

76 See also the list of LP evaluation criteria in Baumgartner, Häfele and Maier-Häfele (2003, p. 6-7)

77 RSS = RDF Site Summary (see http://web.resource.org/rss/1.0/)

by creating new or reusing existing test items. Tests can include open questions, multiple choice, or single choice questions.

- ***Workspaces***: for collaborative elaboration and for sharing of documents, learning platforms provide the option to assign private or shared workspaces to single participants or teams of participants. Workspaces can be used for document creation, storage, sharing, and management.
- ***Messaging***: most platforms feature facilities for instant messaging. Thereby users can send messages to other users that are currently online.
- ***Annotation***: sophisticated learning platforms offer learners tools for annotating learning content. Thereby, learners may highlight text or insert comments at arbitrary places in the text.
- ***Authorization***: restricting access to certain resources is a central element of any learning platform. Users are provided with authorization credentials that are used for logging in.
- ***Search***: facilities for searching content and text on the platform are definitively a "must". A text field for quickly searching desired pages is typically offered on each page (as is also the case in most state-of-the-art Web sites).
- ***Learner tracking***: this is a feature that allows instructors to generate reports of learning activities of any participant, providing an essential tool to monitor whether learners participate as required in online activities.

2.1.2.4 Standardization Issues

Standardization issues[78] are one of the hot topics in e-learning research today. As creating Web-based learning content and processes is a time- and often money-intensive task, a number of standardization bodies have begun to define open standards for different aspects of e-learning. The primary aim thereby is to enable reuse and exchange of learning content and learning processes among different learning platforms. In the following the currently most important standardization bodies in e-learning, along with their primary standards, are listed in alphabetical order. Note that most of these standardizations efforts concentrate on e-content description and exchange, and therefore not really address the core topic of this work, which concentrates on e-learning processes. This is substantiated by the fact that the process aspect in current standardization efforts is primarily addressed from the point of view of *content sequencing*. Contrary, the process emphasis in this thesis considers content as complementary (input/output) to the actual *learning process*. The following listing is intended to point to the lack of efforts in this area.

[78] I consider these also as "*acronym issues*", as will become clear to readers throughout the following pages.

Advanced Distributed Learning Initiative (ADL)

The ADL (http://www.adlnet.org) initiative "*is a collaborative effort between government, industry and academia to establish a new distributed learning environment that permits the interoperability of learning tools and course content on a global scale.*"[79]

The ADL is accountable for the Sharable Content Object Reference Model (SCORM), a widely implemented and accepted standard that is built upon the work of other standardization bodies such as AICC, IMS, IEEE, or ARIADNE[80].

SCORM defines[81]:

- A Web-based learning Content Aggregation Model (CAM) for assembling, labeling, and packaging of learning content. The basic units of interest in the CAM are Sharable Content Objects (SCO) and Content Packages that are used to bundle content (see Figure 4).

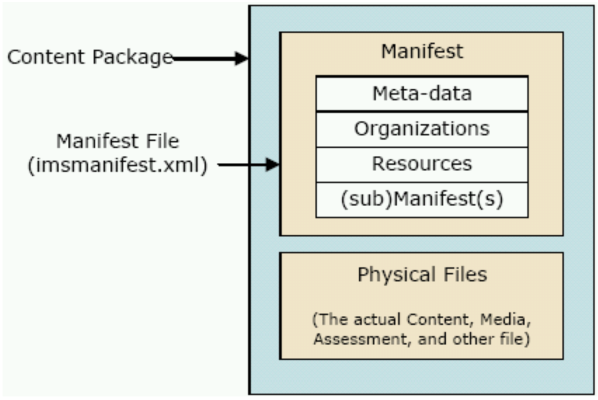


Figure 4: A SCORM content package.[82]

- A Run-Time Environment (RTE) which includes *Launch*, a content-to-LMS communication Application Programming Interface (API), tracking, data transfer and error handling.

- Sequencing and Navigation (SN) for sequencing and content navigation, which affects how the content is assembled, and consequently presented to and navigable by the learner.

Alliance of Remote Instructional Authoring and Distribution Networks for Europe (ARIADNE)

ARIADNE (http://www.ariadne-eu.org) aims "*to exploit and further develop the results of the ARIADNE and ARIADNE II European Projects, which created tools and methodologies for*

---

[79] ADL (2003b)

[80] Cf. ADL (2003a), (2004, p. 29-32)

[81] ADL (2004, p. 28)

[82] ADL (2004, p. 29)

producing, managing and reusing computer-based pedagogical elements and telematics supported training curricula."[83] ARIADNE's work in educational metadata in collaboration with the IMS Project (see below) had a major influence in the development of the IEEE Learning Objects Metadata (IEEE/LOM) standard (see also below).

Aviation Industry Computer Based Training Committee (AICC)

The AICC (http://www.aicc.org) "*is an international association of technology-based training professionals. The AICC develops guidelines for aviation industry in the development, delivery, and evaluation of CBT and related training technologies. The objectives of the AICC are as follows: (1) Assist airplane operators in development of guidelines which promote the economic and effective implementation of computer-based training (CBT), (2) Develop guidelines to enable interoperability, and (3) Provide an open forum for the discussion of CBT (and other) training technologies.*"[84]

The main artifacts of AICC are subsumed under the AICC Guidelines and Recommendations (AGRs). Relevant e-learning AGRs issued by the AICC include[85]:

- AGR-002 (*Courseware Delivery Stations*): Includes technical recommendations for the acquisition of CBT stations.
- AGR-006 (*Computer-Managed Instruction* – CMI): Recommends guidelines for the interoperability of CMI systems, enabling them to use CBTs from different origins.
- AGR-007 (*Courseware Interchange*): Includes guidelines for interchange of CBT courseware elements such as text, graphic, audio, etc.
- AGR-010 (*Web-Based Computer Managed Instruction*): Adapts the AGR-006 interoperability guidelines particularly for Web-based CMI systems.

IEEE Learning Technology Standards Committee (LTSC)

The LTSC (http://ltsc.ieee.org) "*is chartered by the IEEE Computer Society Standards Activity Board to develop accredited technical standards, recommended practices, and guides for learning technology.*"[86]

The IEEE/LTSC is organized into 20 workgroups (WGs) elaborating on different aspects of learning technology. Among the currently most often cited in the field are:

- WG1 (Architecture and Reference Model): WG1 has issued the Learning Technology Systems Architecture (LTSA), a pedagogically neutral standard that "*specifies a high level architecture for information technology-supported learning, education, and training*

83 ARIADNE (2002)

84 AICC (2001a)

85 Cf. AICC (2001b)

86 IEEE LTSC (2004a)

systems that describes the high-level system design and the components of these systems."[87]

- WG12 (Learning Object Metadata): WG12 is working on specifying the syntax and semantics of Learning Object Metadata (LOM), which is defined as the attributes required to fully/adequately describing a learning object.[88]

Instructional Managements Systems Project (IMS)

The IMS (http://www.imsproject.org) "*develops and promotes the adoption of open technical specifications for interoperable learning technology. Several IMS specifications have become worldwide de facto standards for delivering learning products and services. [...] IMS is a worldwide non-profit organization that includes more than 50 Contributing Members and affiliates. These members come from every sector of the global e-learning community.*"[89]

The core deliverables of the IMS are specifications. Currently, the IMS is working on the following relevant specifications[90]: Accessibility, Competency Definitions, Content Packaging, Digital Repositories, Enterprise, Learner Information, Learning Design, Meta-data, Question and Test Interoperability, Simple Sequencing, and Vocabulary Definition Exchange. The most interesting specification located in the scope of this work is the IMS Learning Design (IMS/LD) specification that is discussed in more detail in the Chapter on related approaches, as it also includes parts dealing with formal descriptions of learning processes.

International Standardization Organization (ISO)

A subcommittee of the world-wide operating standardization body ISO (http://www.iso.org), the JTC 1 / SC 36 committee, is working on standardization issues in information technology for learning, education and training in liaison with the IEEE LTSC[91].

The ISO/JTC1/SC36 committee is organized in five workgroups on

- Vocabulary
- Collaborative technology
- Learner information
- Management and delivery of learning, education, and training
- Quality assurance and descriptive frameworks

Under direct responsibility of this committee, no standards have been published yet, but it seems likely that the ISO as *the* major standardization body will be a key player in the de-

87 IEEE LTSC (2004b)

88 IEEE LTSC (2004c)

89 IMS Global Learning Consortium (2003a)

90 IMS Global Learning Consortium (2004)

91 Cf. ISO (no date)

velopment of a general e-learning standards bundle integrating the diverse efforts existing to date[92].

2.1.2.5 Current Research Directions

At the beginning of the e-learning hype, standalone solutions like CBT with focus on technical realization and presentation of content using means of multimedia were prevalent. The Internet brought about new opportunities and challenges: massive amounts of hyperlinked documents called for appropriate structuring approaches, while allowing for more dynamic, interactive scenarios involving networks of content and learners. New didactical considerations regarding Web-supported processes of information transfer and knowledge construction began to emerge. Meanwhile, we seem to have reached the point where ICT is serving instead of dictating as a means both for content-related and learning process aspects.

Today, major research threads in e-learning are:

- ***Blended learning***, focusing on the alignment of online and face-to-face phases[93].
- ***Mobile learning***, promising to enable ubiquitous, pervasive, and context-aware learning scenarios[94].
- ***Networked learning***, focusing on collaborative aspects of interconnected learners. Coaching, mentoring[95], tutoring, and moderating[96] are current buzzwords for supporting and facilitating networked learners.
- ***Standardization issues***, whereby the focus is currently on content metadata standardization, and slowly shifting to standardized descriptions of learning processes involving e-content.
- Additionally, we perceive a research drift away from monolithic platforms solutions to component-based platforms employing modularized, reusable Web services[97]. The period in which centralized functionality was the central subject seems to give way to a period of investigation on distribution and situated use of existing functionality. The focus shifts from structure to ***process***.

Summarizing, we are currently experiencing a strong movement from mainly technological e–learning research (how can new media be used for existing learning scenarios?) to learning

---

92 Cf. the standardization bodies cooperation network structure in Baumgartner, Häfele and Maier-Häfele (2003)

93 See p. 12

94 See p. 14

95 Ensher, Heun and Blanchard (2003)

96 Salmon (2000)

97 Cf. Derntl and Mangler (2004), Mangler and Derntl (2004), Sampson and Kastradas (2004), Sowe et al. (2004)

process awareness and chances of new, technology-enriched application scenarios (how can existing media be used for *new* learning scenarios?).

2.1.3 Person-Centered Teaching and Learning

> *When we put together in one scheme such elements as a prescribed curriculum, similar assignments for all students, lecturing as almost the only mode of instruction, standard tests by which all students are externally evaluated, and instructor-chosen grades as the measure of learning, then we can almost guarantee that meaningful learning will be at an absolute minimum.*[98]

2.1.3.1 The Person-Centered Approach

The Person-Centered Approach (PCA) to teaching and learning has been developed by the American psychologist Carl R. Rogers (1902–1987). It has its roots in counseling and psychotherapy: In addition to his extensive practice, Rogers dedicated his lifetime to researching the conditions that characterize the most constructive atmosphere in the relationship and communication between counselor or therapist and client. He established that there are three attitudinal dispositions that the counselor or therapist has to live, and which the other person has to perceive, to improve personally[99]:

1) ***Congruence*** (realness, transparency, genuineness): honesty with the client, not putting on any facades.
2) ***Acceptance*** (respect, prizing, positive regard): unconditional positive regard toward the client.
3) ***Understanding*** (empathy): the ability to feel what the client feels.

Throughout Rogers' theory the main hypothesis was the existence of a directive tendency in any living organism: the *actualizing tendency*, a constructive tendency that drives the organism to strive towards making the best out of the own existence and developing its potentials to the fullest extent possible[100]. Rogers stressed that this tendency unfolds best in an atmosphere that is characterized by the above three conditions (also called *Rogers variables*). Rogers and his colleagues soon recognized that their findings were not constrained to application in client-centered therapy, but were also applicable in other domains where different people interact and communicate. Thereby, the ideas from client-centered counseling/therapy flow into the more general PCA, for example as applied in educational settings: In Rogers' theory of Person-Centered Teaching, the role of the instructor becomes that of a *facilitator* of

[98] Rogers (1983, p. 21)
[99] Cf. Rogers (1961)
[100] Cf. Boeree (1998)

learning who provides a learning climate that supports the student in his or her own striving for solutions. In general, Person-Centered teaching aims toward[101]:

- a climate of trust in which curiosity and the natural desire to learn can be nourished and enhanced;
- a participatory mode of decision-making in all aspects of learning in which students, teachers, and administrators have a part;
- helping students to prize themselves, to build their confidence and self-esteem;
- uncovering the excitement in intellectual and emotional discovery, which leads students to become life-long learners;
- developing in teachers the attitudes that research has shown to be most effective in facilitating learning;
- helping teachers to grow as persons, finding rich satisfaction in their interaction with learners;
- an awareness that the good life is within, not something which is dependent on outside sources.

Rogers stressed that the conventional approach of purely cognitive instruction by assignment, today still the most widely utilized approach in higher education, is incapable of producing significant learning effects, as in such settings students are solely passive information recipients. Real significant, meaningful learning (also referred *experiential learning* or *whole-person learning*) is in contrast characterized by[102]:

- Personal *involvement* of the student as a whole person
- *Self-initiated* learning that is driven by desire and coming from within
- Being *pervasive* by addressing intellect, skills, and feelings
- It is *evaluated* by the learner in a sense that the learner knows what she *wants* to know
- Personal *meaning* is the essence of the learning event

Compatible with these characterizations, Barrett-Lennard identifies six major aims of person-centered education:[103]

1) Development of functional knowledge (*knowledge of* instead of *knowledge about*)
2) Learning that is purposeful on the learner's part (compare Rogers' point on personal *meaning* above)
3) Nourishing the individual's curiosity
4) Learning is enlivening and releasing, facilitating the development towards a well-functioning person
5) Fostering *meta-learning*, i.e., learning how to learn
6) Furthering self-responsibility for learning and assessment

To enable such meaningful learning processes, a climate that is characterized by the three Rogers Variables is considered most effective. These specify the *relationship* elements necessary to achieve the aforementioned *aims*[104]. Rogers recognizes that it might be hard for in-

[101] Slightly adapted from Rogers (1983, p. 3)

[102] Cf. Rogers (1983, p. 20)

[103] Cf. Barrett-Lennard (1998)

[104] Cf. Barrett-Lennard (1998)

structors to put person-centered theory into practice, and provides a number of practical examples of ways that are capable of providing students with the freedom to learn experientially, e.g.:

- Providing the students with *problems perceived as real*, meaningful, and relevant[105]. Ways of doing so are presented in a discussion of the attitude of *realness* in the context of Person-Centered e-Learning (see p. 33).
- Provision of all kinds of *resources* in a way that makes them easily and readily accessible to students[106]. This includes not only material resources like books, papers, or e-content[107], but also human resources, e.g., the facilitator herself[108], external experts or guests.
- Using different teaching and learning options from the *instructional continuum* (Figure 5) instead of sticking to only one single approach[109]. Some of the more student-centered options are subsequently described in more detail.

Teacher-centered	Lecture
↑	Questioning
	Drill and practice
	Demonstration
	Discussion
	Cooperative groups
	Guided discovery
	Contracts
	Role-play
↓	Projects
	Inquiry
Student-centered	Self-evaluation

Figure 5: The instructional continuum.[110]

- *Learning contracts*[111] are a way of dealing with the uncertainties inherent in a climate of freedom. They allow students to define and follow their own learning plans and targets, which has the additional advantage of enabling higher transparency in the evaluation process: If targets defined in a learning contract are achieved, the negotiated grade is assigned. Otherwise, it is justified and agreed upon that the facilitator adjusts the grade.

---

105 Cf. Rogers (1983, p. 148)

106 Cf. Rogers (1983, p. 148-149)

107 Cf. the PUBLISH pattern (p. 287)

108 Compare the CONSULTATION pattern (p. 319)

109 Cf. Rogers and Freiberg (1994, p. 190)

110 Source: reproduced with minor changes from Rogers and Freiberg (1994, p. 190)

111 Cf. Rogers (1983, p. 149-153); see also the LEARNING CONTRACTS pattern (p. 381)

- *Peer teaching*[112] is a way to establish a relationship among peer students that is beneficial for both tutees and tutor. Involving students in teaching activities additionally has the advantage of saving the time of the "primary" facilitator.
- Imposing freedom on everyone is not reasonable if the students themselves do not desire it. In such circumstances, *division of group*[113] into more and less directed parts may be useful (but not always feasible due to time restrictions or organizational constraints such as available infrastructure).
- Especially in large classes the formation of facilitator-learning (FL) groups[114] may be effective to further learning activities in small groups. Thereby, FL group meetings should not be imposed on the students by the facilitator, but rather offered and attended as needed. Such meetings may be based on functional criteria, e.g., to cluster students that elaborate a similar topic or subject area.[115]
- Developing an *inquiring state of mind* in the learners by posing problems and providing assistance in the problem-solving process such that self-directed and self-initiated discovery can take place.[116]
- *Encounter groups* are another example of fostering significant learning. Rogers, who has written hundreds of pages on encounter groups, admits that it is difficult to describe them in brief[117]: "*The group usually begins with little imposed structure; the situation and the purposes are up to the group members to decide*." Being facilitated, expressiveness tends to increase, defenses are lowered, and deep, fruitful personal experiences can take place.
- Providing the learners with the option to *evaluate their own learning*[118] (*self-evaluation*) requires taking the self-responsibility of searching and defining own criteria and own measures as well as assessing the degree to which these personally relevant criteria are fulfilled.

Rogers admits that introducing and developing such a climate of freedom in one's own teaching activities requires a lot of risk-taking and spiritedness on the side of the facilitator. It has to follow a process where the facilitator builds inner confidence incrementally from successful, pleasing experiences. Anyway, a look at the characteristic elements of two totally opposite

---

112 Cf. the TUTORIAL pattern (p. 373)

113 Cf. the CONSIDER CONVENTIONAL STYLE pattern (p. 316)

114 Cf. Rogers (1983, p. 154-156)

115 Such FL groups may be implemented in meetings of, e.g., the EXCHANGE OF CONTRIBUTIONS pattern (p. 326) and in some of the INFORMATION GATHERING patterns (p. 329) such as BRAINSTORMING (p. 305) or THEORY ELABORATION (p. 369).

116 Cf. Rogers (1983, p. 156-157)

117 Cf. Rogers (1983, p. 158)

118 Cf. Rogers (1983, p. 158-159); see also the SELF-EVALUATION pattern (p. 215)

approaches as depicted in Table 2 should be sufficiently convincing and encouraging to give the person-centered mode a try.

Table 2: Comparison of a traditional and a person-centered mode of education.[119]

Traditional Mode	Person-Centered Mode
The instructor is the possessor of knowledge, the student the expected recipient. Lectures, books and other means of intellectual instruction are the major means of getting knowledge into the recipient. The instructor is the possessor of power, the student the one who obeys ("rule by authority"). Trust is at a minimum. Students are best governed by keeping them in a constant state of fear. Students' participation in all aspects of decision-making and other democratic values are ignored. The intellect is the only level addressed in the learner, suppressing whole-person learning.	The facilitator puts basic trust in the learners and considers them capable of thinking and learning for themselves. The facilitator shares responsibility in the learning process with students, and provides different kinds of resources, material as well as human. Students develop their own program of learning, and the evaluation of the learning progress is primarily up to the learner. The only discipline necessary is students' *self*-discipline. A facilitative learning climate is provided, one which promotes growth on all levels of learning: on the intellectual, social, and personal levels.

Speaking from our own experience, it was soon clear that switching to a person-centered mode was a good choice. While the students at the University of Vienna are not used to being granted such a high degree of freedom and responsibility for their own learning, they strongly tend to appreciate this mode. This becomes clear when we frequently receive reactions like the following:

> "[The course] showed that one can teach even a rather conservative lab subject matter like programming in a new style. It does not depend on the subject, but only on attitudes. In the beginning I was convinced that this is only possible with diffuse subjects, but it turned out that instructors can even provide enough freedom (not only through self-chosen project topics) despite the exact nature of the requirements."

However, granting such freedom and allowing individuals to unfold and to find their own ways also requires more effort and flexibility on the side of the facilitator, as the individual "learning paths" are not as predetermined and congruent as they are in conventional teaching settings.

---

119 Source: compiled from Rogers (1983, p. 185-190)

2.1.3.2 Person-Centered e-Learning

Person-Centered e-learning (PCeL) is one of the foundational theories underlying this work's contribution. PCeL and its contemporary as well as antecedent research theoretically and practically delivers the learning scenarios on which the PCeL patterns developed as part of this thesis are based upon. PCeL was developed by Renate Motschnig at the University of Vienna and is rooted in Carl Rogers' Person-Centered Approach (PCA) to teaching and learning. It aims to combine the benefits of the PCA with the opportunities that e-learning has to offer. The main hypothesis of PCeL is that if mere transfer of information is deferred to the computer (e.g., hyperlinked e-content) time and resources are set free that can fruitfully be used to enrich face-to-face learning phases in terms of depth and scope[120]. Thereby the facilitator is required to hold as well as to be able to communicate the three Person-Centered attitudes of acceptance, transparency, and empathy toward the students. Allowing students to direct their own learning by providing them with sufficient amounts of learning resources as well as with a significant degree of freedom to activate their inner curiosity, personal interests, and self-actualizing tendency is one of the main propositions of PCeL. Following this, Person-Centered courses are not prearranged to a degree as high as conventional courses, so the PCA may introduce significant extra effort, in particular initially, with respect to time for preparation and resource provision on the side of the facilitator. E-learning elements thereby have the potential to reduce a major share of this overhead, while retaining the benefits of the whole approach[121]. For example:

- Making learning material and resources electronically available as well as linking to resources on the Web has several benefits: The instructor does not have to prepare comprehensive collections of material, but rather an anchor for the students to start with. Students, on the other side, are subsequently able to explore the material in their own ways and independent of time and location.
- Many activities of Person-Centered teaching are suitable to be conducted online, e.g., self-evaluation, peer-evaluation, providing feedback, reflective activities, providing different options, student proposals, learning contracts, communication with peers and facilitators, etc. Employing learning technology for such activities can significantly reduce administrative and organizational overhead.

2.1.3.2.1 Three Levels of Learning in PCeL

Taking the above considerations into account, PCeL primarily calls for a blended learning approach with the goal to find an effective blend of face-to-face encounters and online/distant activities. This allows addressing not only the intellectual level in the learner (as is the case in conventional scenarios and in media-centered e-learning settings) but all

[120] Cf. Motschnig-Pitrik and Holzinger (2002), Motschnig-Pitrik and Mallich (2002)

[121] Motschnig-Pitrik (2002b)

three levels of learning: *intellect*, (social) *skills*, and *personality*, which is in line with Rogers' conception of whole-person or experiential learning, combining "*[...] the logical and the intuitive, the intellect and the feelings, the concept and the experience, the idea and the meaning.*"[122] The three levels of learning[123] are depicted in Figure 6, along with a comparison of the levels addressed by conventional teaching and by PCeL:

- In conventional scenarios, especially in pure lecturing sessions, teaching activities focus on transmission of factual, procedural, and/or theoretical information. In the learner, this addresses primarily the intellectual level.
- In PCeL the transmission of intellectual information is preferably deferred to the computer (e.g., e-content, hypermedia, multimedia, Web resources, etc.) and other information resources such as printed books, chapters, or articles. Contrary to conventional settings, the task of the instructor is not to take over major parts of information transmission, but information *preparation* to aid the learners in finding relevant resources. As this is less time- and resource intensive than lecturing sessions, there is more time left for engaging in more meaningful learning scenarios that actively involve participants, further collaboration, communication, problem-solving, interactivity, and discussions. Such activities primarily address the level of social and practical *skills* as well as *personality*. The primary task of the instructor/facilitator is to provide a positive learning climate as well as to facilitate learners. However, such scenarios need not necessarily be conducted face-to-face: Means of computer-mediated communication (e.g., chat, conferencing, online discussion forums) and information/resource exchange (e.g., online workspaces, messaging, publishing) allow such scenarios to be as well conducted online, which means that to a certain degree ICTs also penetrate into deeper learning levels.

[122] Rogers (1983, p. 20)

[123] For information on the original roots of the three levels of learning see Nykl and Motschnig-Pitrik (2002) and Bühler (1907), Vygotsky (1992)

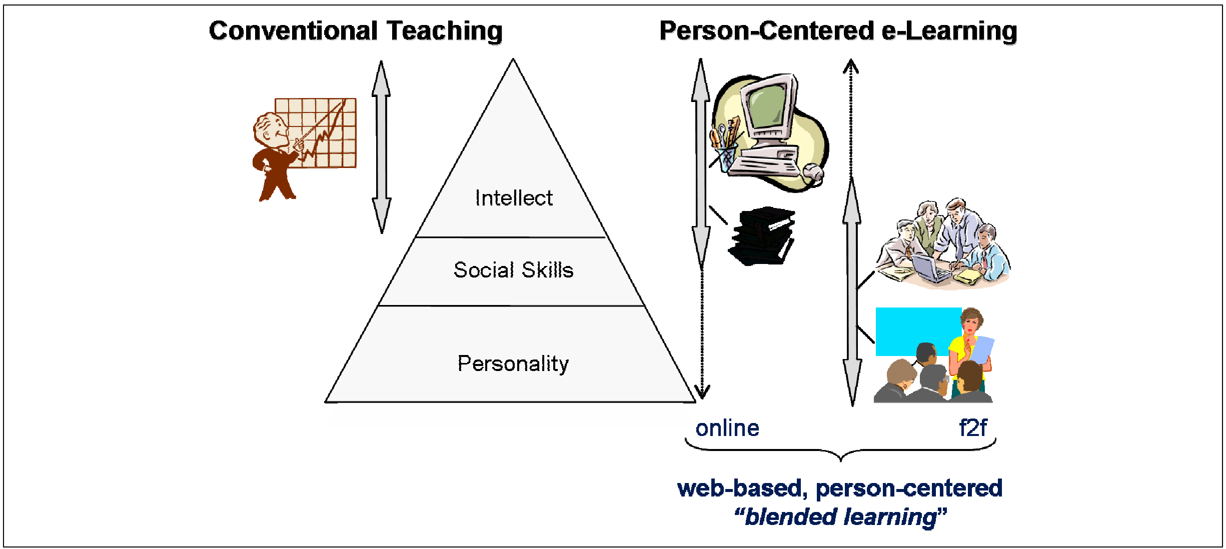


Figure 6: The three levels of learning in conventional teaching and in PCeL.[124]

A recent study[125] has shown that the primary goals of PCeL, namely promoting learning on all three levels – intellect, social skills, and personality – matches perfectly with company managers' requirements and expectations towards graduates of the business informatics study. The top-ranked requirements are:

1) *Social competence*
2) *Teamwork abilities*
3) *Analytical thinking*

The first occurrence of an item associated purely with intellectual assets is ranked far behind at position 11: *state-of-the-art IT knowledge*. Interestingly, managers' actual perceptions of these skills in graduates turn out to be completely contrary: While intellectual assets are perceived as well-developed, managers see a huge lack of education in social and interpersonal, practically relevant skills. This further substantiates the proposition that current teaching activities focus mainly on information transmission (lecture) and on assignment-based learning scenarios, missing out the levels of social skills and personality.

On the side of the students, evaluations of feedback and especially the data collected from their completed questionnaires have shown that their primary a-posteriori motivations to participate in a PCeL course are[126]:

- Improving professional skills.

  This resembles the *intellectual* level of learning. In comparison to an a-priori surveyed, hypothetical conventional course, this factor is not as important in PCeL courses, but still a top motivational factor.

---

[124] This figure is an extended version of that in Motschnig-Pitrik and Mallich (2002, p. 4)

[125] Motschnig-Pitrik (2002a)

[126] Cf. Motschnig-Pitrik (2004b); see also the *Evaluation* section of the COURSE pattern (p. 260)

- Collegial cooperation with peers.

  This resembles the level of *social skills*.

- Experiencing a positive atmosphere and learning climate in the course.

  This supports the level of *personality* and *dispositions* in learning. It is significantly more a motivational factor in a PCeL course than in a conventional course only when students highly perceive Person-Centered attitudes in their facilitators.

Note that the three motivational aspects are equally important to students, meaning there was no statistically significant difference among these aspects.

Following the above considerations, a desirable blend for PCeL should be based upon the following considerations:

1) Employing ICTs as much as possible for provision of resources, learning materials, and information.
2) Employing ICTs as appropriate to conduct tasks online that primarily address internal processes in the student (e.g., reflections, feedback, evaluations, and elaborations) as a result from some face-to-face experience or in preparation for it. For such tasks there is no imperative need to proceed in the group. Additionally, ICTs may help to ease organization and administration of such tasks (e.g., collecting and tracking peer-evaluations offline is very time- and data-intensive).
3) Use face-to-face encounters for tasks that can be enriched by interpersonal contact and multiple perspectives (e.g., sharing of goals, presentations, elaborations in small teams, discussions, encounter groups, etc.) as well as for tasks that cannot be reasonably conducted online.
4) Face-to-face meetings and online phases alternate[127]. Meetings can provide the context for the following online phase or conclude a previous online phase. On the other hand, online phases can be used for preparation of meetings or for providing a means of continuing them in the virtual space. This way, the phases can complement each other effectively.

2.1.3.2.2 The Rogers Variables in PCeL

In spite of all opportunities e-learning technology offers, it seems important to mention that recent studies[128] have shown that, when employing e-learning elements in combination with the PCA, interpersonal values and attitudes of facilitators still have a highly significant influence on most aspects of students' learning motivations, and even on their attitudes towards the use of e-learning technology in a course.

---

[127] See the ALTERNATING PHASES pattern on p. 249

[128] See for example Derntl and Motschnig-Pitrik (2004c), Motschnig-Pitrik (2004b), Motschnig-Pitrik and Derntl (2003b), Motschnig-Pitrik, Derntl and Mangler (2003)

On page 24, we have already briefly outlined the three Rogers variables in their original meaning, which is rooted in the context of counselor-client relationships. In the following, we will discuss ways of being acceptant, transparent, and understanding towards students particularly in PCeL settings.

Realness

Realness or authenticity requires that students are provided with the opportunity to solve real, authentic problems that are in line with their personal interests[129]. In conventional, assignment-based settings, most tasks and topics are preset by the instructor. This naturally leads to situations where students are confronted with problems or assignments that do not at all match their personal interests or prior experiences, thus repelling their motivation to learn. In PCeL scenarios, students are to a certain degree free to propose[130] their own topics and problems of interest within a certain context that is supplied by the instructor (or by curricular requirements). Provided with such freedom, students tend to afford more time for elaborations[131], tend to achieve higher academic results[132], and also tend to perceive the course mode as more appealing[133].

Furthering constructive, open feedback from participants is another way of transporting realness and transparency in a course[134]. There are several options of collecting feedback[135], e.g., verbally after presentations performed by students or in the form of written reaction sheets[136], which may be collected for single course units, learning activities, and/or for a whole course. This way, students are offered the opportunity to reflect on and to transparently convey their personal experiences to the instructor and/or to their peers. This can even be extended in scenarios where participants are involved in the evaluation of their own as well as their peers' contributions, thus achieving a high degree of transparency in the evaluation process[137].

Acceptance

There are several options of being acceptant towards students, which can actually be derived almost straightly from the central characteristics of person-centered teaching, e.g.:

---

129 Motschnig-Pitrik (2004a)

130 Cf. the PROPOSAL (p. 359) and APPROVAL (p. 302) patterns

131 Motschnig-Pitrik, Derntl and Mangler (2003)

132 Aspy (1972), Motschnig-Pitrik (2001), Rogers and Freiberg (1994, p. 254)

133 Derntl and Motschnig-Pitrik (2004c), (2005)

134 Motschnig-Pitrik (2004a)

135 Cf. the COLLECT FEEDBACK pattern (p. 221)

136 Cf. the REACTION SHEETS pattern (p. 238)

137 Cf. the patterns in the *Evaluation* package (p. 186)

- The claim for a participatory mode of decision making implies involving participants in elaborating and setting overall and specific course goals and expectations[138] as well as personal learning targets.[139]
- Letting participants engage in personally relevant learning processes, which can be achieved, e.g., by a soliciting concrete proposals within a facilitator-supplied topic/subject frame. Engaging in solving such problems that students perceive as real and meaningful is capable of inducing self-initiated learning, even when the stimulus comes from outside[140].
- Employing project-based learning scenarios or learning contracts[141] are examples of viable tools to provide the students with a significant amount of freedom to work and learn while preserving the ability to comply with curricular requirements[142]. Thereby, the facilitator can be acceptant by having basic trust in students and by allowing them to take self-responsibility in their learning efforts. Project-based learning scenarios foster co-construction of knowledge in a constructivist manner, for example allowing students' active involvement and collaboration in small teams[143].

In a broader sense, acceptance means facilitating students in their own constructive tendency, letting them participate in decision-making and learning process/goal design[144], providing them with a high degree of freedom (but not imposing freedom on them), and taking into account the diversity of their perspectives and contributions.[145]

Understanding

Showing empathy or empathic understanding requires the facilitator to try at her best to be aware of the perspectives of students without evaluating or judging them. While it is an essential facilitative attitude in *any* kind of facilitator-student interaction/communication, Rogers acknowledges that being empathically understanding sometimes needs to be traded for being real: "*if one has little understanding of the student's inner world and a dislike for the students or their behavior, it is almost certainly more constructive to be real than to be pseudo-empathic [...]*"[146]

[138] Cf. the ELABORATE GOALS AND EXPECTATIONS pattern (p. 322)

[139] Motschnig-Pitrik (2004a)

[140] Rogers (1983, p. 20)

[141] Cf. the PROJECT-BASED LEARNING (p. 387) and LEARNING CONTRACTS (p. 381) patterns

[142] Cf. Motschnig-Pitrik (2004a), Rogers (1983, p. 149), Rogers and Freiberg (1994, p. 195-201)

[143] Cf. Chou (2004, p. 13)

[144] Cf. the ELABORATE GOALS AND EXPECTATIONS pattern (p. 322)

[145] Cf. Motschnig-Pitrik (2004a)

[146] Rogers (1983, p. 126)

2.2 The Pattern Approach

The intent of this Section is to provide a compact, yet sufficiently detailed introduction to current pattern research and its history with respect to pattern definitions, organization of patterns in pattern collections, pattern format, and the pattern life-cycle. The special focus in doing so will be on three disciplines where the pattern approach is well-established today:

- ***Architecture*** is a central discipline in pattern research, which is simply due to the fact that architecture was the field where the pattern approach initially emerged in the late 1970s.
- After a decade of "silence", researchers in the field of ***software engineering***, and particularly in software design, were the first to adopt the pattern approach, initiating a hype that brought world-wide acceptance for the pattern community.
- Though the pattern approach was adopted by many other disciplines, the third point of focus here is on ***e-learning*** and ***pedagogy***, as it may most likely provide valuable input and concepts for developing patterns for Person-Centered e-learning.

The introduction to pattern approach topics is covered in the following three major Sections:

- Section 2.2.1: "Pattern Basics". This Section presents an introduction to basic terms and definitions of the pattern approach from the viewpoints of the three focal disciplines: Architecture, software engineering, and pedagogy. The question addressed is: "*What is a pattern?*"
- Section 2.2.2: "Combining Patterns". As single patterns are mostly being disseminated as parts of collections of related patterns, this Section introduces current methods of organizing pattern collections. The question addressed is: "*How can patterns be combined?*"
- Section 2.2.3: "Inside Patterns". This Section gives an overview of different pattern formats and descriptions. It also includes a discussion on the measurability of pattern quality, and a description of the main phases of the pattern life-cycle, such as writing, dissemination, application, and maintenance of patterns and pattern collections. The question addressed is: "*What does a pattern comprise and how does it develop over time?*"

2.2.1 Pattern Basics

2.2.1.1 Pattern Definitions

In the following, an overview of various definitions of the term *pattern* is given, with special respect to the fields relevant to this thesis, i.e. software engineering and pedagogy, whereas architecture, the discipline where the pattern approach is rooted, also deserves attention.

2.2.1.1.1 Patterns in Architecture

The field of architecture was the initial point of the pattern movement[147]. Accountable for this is Christopher Alexander with a series of books, among others "A Pattern Language"[148], one of the most cited and best-known, in which he characterizes a pattern as follows:

> "*Each pattern describes a problem which occurs over and over again in our environment and then describes the core of the solution to that problem, in such a way that you can use this solution a million times over without ever doing it the same way twice.*"[149]

One central aspect of the given definition are recurring problems in our environment, whose solution core is described in a way that is both concrete enough to allow for resolving concrete problems, and abstract enough as to avoid adhering to specific observed problems only. Going more into the problem environment and the forces effective inside, Alexander describes a pattern in his book "The Timeless Way of Building" as "*a relationship between a certain context, a certain system of forces which occurs repeatedly in that context, and a certain spatial configuration which allows these forces to resolve themselves.*"[150] This force-resolving spatial configuration that a pattern shall describe is often referred to as a rule, which led to the concept of "pattern as a rule".[151] The rule shows the pattern user in an easily comprehensible way which steps he or she has to take to transform an initial situation into the desired situation in a certain problem context, or, "*in short, to generate the pattern itself, in the world.*"[152] These relationships are depicted schematically in Figure 7.

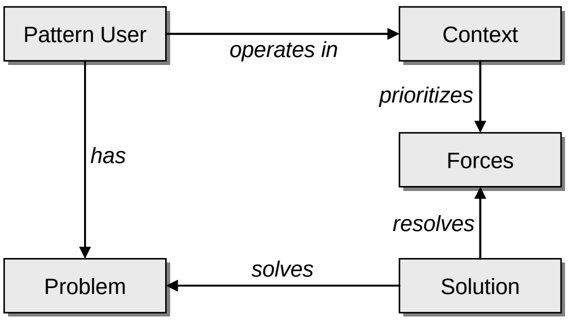


Figure 7: Relationships between key elements in Alexander's pattern definition.[153]

[147] There is no real consensus on that, but most of the authors primarily refer to architect Christopher Alexander and his work when outlining the early days of pattern history.

[148] Alexander et al. (1977)

[149] Alexander et al. (1977, p. x)

[150] Alexander (1979, p. 247)

[151] Cf. Riehle and Züllighoven (1996), or Alexander et al. (1977, p. 247): "*Each pattern is a three-part rule, which expresses a relation between a certain context, a problem, and a solution.*"

[152] Alexander (1979, p. 183)

[153] Source: Reproduced according to Arnold and Podehl (1999, p. 143), Meszaros and Doble (1998)

2.2.1.1.2 Patterns in Software Engineering

At the beginning of the 1990s the pattern idea was picked up by software engineers to support reuse[154] und documentation[155] of software design, which are still the most predominant approaches associated with patterns today, by specifying *Design Patterns*. Nevertheless, there exist numerous pattern approaches in the software engineering discipline, e.g., for the analysis phase[156], for software engineering processes[157], for the implementation phase in form of object-oriented "Coding Idioms"[158] in the C++ programming language[159], or for user interface and interaction design in Human-Computer-Interaction[160] (HCI) research.

The following sub-Sections present basic pattern concepts grouped by different sub-disciplines of software engineering, and thereby introduce the most influential protagonists in the pattern community today.

Design Patterns

Erich Gamma, Richard Helm, Ralph Johnson und John Vlissides, nick-named "The Gang of Four" or "GoF", describe Design Patterns as "*easily applicable, systematic descriptions of a named, relevant and recurring design problem in object-oriented software systems.*"[161] GoF establish the connection to Alexander's architectural patterns via an analogy of the walls and doors of a building with the objects and interfaces in object-oriented software systems[162]. More unreservedly than GoF and Alexander, Riehle and Züllighoven define a pattern simply as an "*abstraction from a concrete form which keeps recurring in specific non-arbitrary contexts.*"[163] Peter Coad, one of the pattern pioneers[164] in object-oriented software design, just referred to the definition of the term pattern in Webster's Dictionary: "*A fully realized form,*

[154] Cf. Buschmann (1993), Coad (1992), Coplien (1994), Gamma et al. (1995), (1993), Johnson (1992), Pree (1994), Shaw (1989), or Pena-Mora and Vadhavkar (1996)

[155] Beck and Johnson (1994)

[156] Fowler (1998)

[157] Ambler (1998)

[158] Coplien (1998)

[159] Stroustroup (1986)

[160] For an introduction to HCI research, see Dix et al. (1998). Important representatives of the pattern approach in HCI are, among others, Jan Borchers with *Interaction Design Patterns* (Borchers (2001)), or Jenifer Tidwell with her HCI pattern languages *Common Ground* (Tidwell (1999)) and *UI Patterns and Techniques* (Tidwell (2002)). See also van Welie and van der Veer (2003).

[161] Gamma et al. (1995, p. 3)

[162] Gamma et al. (1995, p. 3)

[163] Riehle and Züllighoven (1996, p. 3)

[164] Cf. Pree (1994, p. 62)

original, or model [...] for imitation: something regarded as a normative example to be copied."[165]

Analysis Patterns

Martin Fowler's focus is located in the phase of gathering and modeling knowledge about associations in the problem domain – in the analysis phase of the software engineering process. He defines an Analysis Pattern as, "*an idea that has been useful in one practical context and will probably be useful in others.*"[166] Two concepts are prominent in Fowler's conception:

- Idea: The intent behind the usage of such a general term is to manifest his conviction that there is no pattern "uniform".[167]
- ***Practical context***: Patterns are no inventions in the common sense; instead they emerge from practical experience, in his case from work in complex projects.

Geyer-Schulz and Hahsler[168] specify their own set of Analysis Patterns dealing with collaborative information filtering and sharing, and knowledge management, captured and applied in project work. They do provide an own definition of patterns, but they underline the focal point of Analysis Patterns: *"[I]n contrast to Design Patterns, [Analysis Patterns] focus on organizational, social and economical aspects of a system, since these aspects are central for the requirements analysis and the acceptance and usability of the final system.*"[169]

Process Patterns

Scott Ambler, in a whitepaper on Process Patterns, describes Process Patterns as "*a collection of general techniques, actions, and/or tasks (activities) for developing object-oriented software*" and divides them according to their scale into *Task* Process Patterns, *Stage* Process Patterns, and *Phase* Process Patterns[170]. This approach focuses on dynamic aspects by arranging techniques, actions, and activities in the process of software construction and software management. In the summary Section of the same work he characterizes Process Patterns as "*reusable building blocks from which [an] organization can tailor a mature software process.*"[171] And he gives yet another, more general definition of a pattern in the glossary of the paper: "*The description of a general solution to a common problem or issue from which a detailed solution to a specific problem may be determined.*"[172]

165 Coad (1992, p. 152)

166 Fowler (1998, p. 8)

167 See Section 2.2.3.1 for a discussion on pattern forms

168 Geyer-Schulz and Hahsler (2001)

169 Geyer-Schulz and Hahsler (2001, p. 2)

170 Ambler (1998, p. 1-2)

171 Ambler (1998, p. 11)

172 Ambler (1998, p. 13)

Idioms

James Coplien described and categorized a set of Coding Idioms[173] in C++[174]. Though Coplien's Coding Idioms do not appear like being related to patterns at a first glance, they are commonly[175] considered as the counterpart of patterns in the world of programming languages. In this respect, Riehle and Züllighoven explicitly call Coding Idioms "Programming Patterns" that are described with means of programming language constructs[176]. Coplien's perception of the relation between patterns and Coding Idioms is slightly different: "*[I]dioms are special kinds of patterns that are tied to a specific programming language.*"[177] Appleton[178] more concretely refers to these special kinds of patterns as "*low-level patterns*", as they are language specific and may lack some levels of abstraction.

2.2.1.1.3 Patterns in Pedagogy

According to the "Pedagogical Patterns Project" (PPP), a world-wide network of pattern authors employed in academic teaching, patterns in general, "*are designed to capture best practices in a specific domain. [...] In essence a pattern solves a problem [...] that recurs in different contexts*," whereas pedagogical patterns in particular "*capture expert knowledge of the practice of teaching and learning.*"[179] Generally, the PPP authors are not as exhaustive in theorizing on patterns as authors from other fields.

Nevertheless, the e-learning and pedagogy pattern proposals/projects today are almost entirely based on Alexander's philosophy and conception of patterns.[180]

2.2.1.2 Discussion on Pattern Definitions

Table 3 gives a summary of the presented definitions and statements, along with the respective key figures presented so far.

---

173 According to SIL International (1999b), an "*idiom is a construction whose meanings cannot be deduced from the meanings of its constituents.*"

174 Coplien (1992)

175 Cf. Buschmann et al. (1996), Coplien (1992), Riehle and Züllighoven (1996)

176 Riehle and Züllighoven (1996, p. 8)

177 Coplien (1998)

178 Appleton (2000)

179 Pedagogical Patterns Project (2002)

180 See also Caeiro, Llamas-Nistal and Anido (2004), and Section 4.1 later in this work.

Table 3: Overview of pattern definitions from different perspectives.

Field	Protagonists	Statements
Architecture	Alexander	– *"Each pattern describes a problem which occurs over and over again in our environment and then describes the core of the solution to that problem, in such a way that you can use this solution a million times over without ever doing it the same way twice"*
Software Design	Buschmann, Coad, Coplien, Gamma, Helm, Johnson, Riehle, Vlissides, Züllighoven, etc.	– *Design Patterns are "easily applicable, systematic descriptions of a named, relevant and recurring design problem in object-oriented software systems"* – *Design Patterns are an "abstraction from a concrete form which keeps recurring in specific non-arbitrary contexts"* – *A pattern is "a fully realized form, original, or model [...] for imitation: something regarded as a normative example to be copied"*
Software Analysis	Fowler	– *"A pattern is an idea that has been useful in one practical context and will probably be useful in others"*
Software Processes	Ambler	– *Process Patterns are "a collection of general techniques, actions, and/or tasks (activities) for developing object-oriented software"* – *"The description of a general solution to a common problem or issue from which a detailed solution to a specific problem may be determined"*
Coding Idioms	Coplien	– *"Idioms are special kinds of patterns that are tied to a specific programming language"*
Pedagogy	Bergin, Eckstein, Fricke, Sharp, Völter	– *Pedagogical Patterns "are designed to capture best practices in a specific domain. [...] In essence a pattern solves a problem [...] that recurs in different contexts"*

The most central aspects of a pattern, widely independent of buildings and construction, were delivered by Alexander:

- Point of origin are frequently ***recurring problems*** in our environment, whereas the problems are never completely identical, but share parts of their basic structure and immanent forces.
- The target of describing a pattern is to specify the ***core of the solution*** to a set of problems sharing certain properties (e.g., forces) in such a way, that the solution is not tied to a certain problem instance and to support ***reusability*** in problem solving.
- Using the relation of ***spatial configuration*** that breaks up a ***system of forces*** in a given ***context***, Alexander shows an illustrative description of a pattern; this makes clear that a pattern is not only the solution, but also guidance on when to apply it (e.g., description of forces).

Other disciplines which have adopted the pattern approach for their purposes also provide important aspects about patterns:

- The GoF Design Pattern approach underlines the importance of ***naming*** and ***evaluating*** a ***relevant design***. Naming relevant design concepts has created a ***shared vocabulary*** which largely improves the effectiveness of communication on previously known, but unnamed design concepts. This is a key aspect in all pattern approaches.
- Riehle and Züllighoven place emphasis on the ***abstraction*** of a concrete, recurring form.
- Peter Coad points out the ***normative value*** of patterns, which justifies the ***imitation*** of patterns; this aligns with the Alexandrian claim for reusability.
- Scott Ambler accentuates on abstraction in a similar way as Riehle and Züllighoven, but in a reversed form: the abstract description of problem and solution can be tailored and ***specialized*** to a concrete situation.
- The Pedagogical Pattern Project emphasizes on the quality of patterns in postulating that patterns should describe ***best practices***[181].

While some of the more recent pattern approaches introduce specific aspects in their patterns and pattern definitions, they primarily rely on the theoretical and philosophical foundation laid by Alexander's pioneer works.[182]

An effective method to communicate components and properties of a pattern can be found in Buschmann, Rohnert, Sommerlad and Stal[183] in the form of a hierarchic structure, which is depicted in Figure 8. This structure comprises important pattern elements like context, problem, and solution, each with a short descriptive statement.

---

181 It may be mentioned that the PPP make a slightly different use of the problem context: They claim to use patterns for a problem that occurs in *different* contexts, while the other approaches previously mentioned argue on problems in a *certain* context.

182 [Author's note: Even though the usefulness of a pattern definition that comprises most of the mentioned aspects is doubtable, it might look like the following: *A pattern is a named description and evaluation of a relevant, context-dependent recurring problem, as well as of the core of the solution to that problem in the form of a configuration of the respective discipline's design options, so that the resulting problem-solving template may be reused for solving similar problems.* I discovered the term "problem-solving template" independently; not surprisingly, I found it again later in Gardner et al. (1998, p. 35): "*[E]ach pattern is reflected by a problem-solving template.*"]

183 Buschmann et al. (1996)

- **Pattern**
  - ***Context***
    - Design situation giving rise to a design problem
  - ***Problem***
    - Set of forces repeatedly arising in the context
  - ***Solution***
    - Configuration to balance the forces
      - Structure with components and relationships
      - Run-time behavior

Figure 8: Components of a pattern common to most approaches.[184]

The kind of rigid division of a pattern into three parts following the simplified Alexandrian tradition "*solution* to a *problem* in a *context*" has to be handled carefully: Vlissides[185] even qualifies it as one of the top ten misconceptions[186] about patterns, as three major aspects of a pattern are missing: *recurrence*, *teaching* (i.e., consequences of application), and a *name* to refer to.

Taking this concern into account, we have now reached common sense of the term pattern in the scope of this work, so it is possible to give way to the discussion of pattern combinations and to looking deeper inside a pattern.

2.2.2 Combining Patterns

Usually, it is not useful to specify patterns isolated from each other, as problems seldom occur isolated from each other in reality. There are different approaches to relate a number of patterns: The prevalent distinction today includes pattern languages, pattern catalogs, pattern systems, and pattern handbooks[187]. The usefulness of providing combinations of patterns generally lies in dividing problems and solutions into sub-problems and their respective solu-

[184] Source: Reproduced from Buschmann et al. (1996, p. 11)

[185] Cf. Vlissides (1998, p. 3-4)

[186] Vlissides (1998, p. 3-11) defines the *Top Ten Misconceptions* about patterns: A pattern is a solution to a problem in a context; Patterns are just jargon, rules, programming tricks, data structures...; Seen one, seen them all; Patterns need tool or methodological support to be effective; Patterns guarantee reusable software, higher productivity, world peace, etc.; Patterns 'generate' whole architectures; Patterns are for (object-oriented) design or implementation; There's no evidence that patterns help anybody; The pattern community is a clique of elites; The pattern community is self-serving, even conspiratorial.

[187] It needs to be mentioned here that there seems to be no real consensus in the literature on what these terms *exactly* refer to. Some of them are used synonymously by different authors; some of them are defined slightly different among different authors. This work tries to rely on the most commonly cited, leading figures of the pattern movement, such as Alexander, Coplien, GoF, Riehle and Züllighoven, Buschmann, and others.

tions. The resulting clarity and reduction of complexity at the level of sub-problems entails some major advantages, such as easier recognition of a pattern fitting to a specific problem on the side of the user, or better adaptability due to different levels of description granularity, to mention just a few. Generally, the more inter-related patterns at smaller scale are available, the higher will be the number of possible combinations of patterns, which increases the potential number of problems that can be solved.

2.2.2.1 Pattern Languages

The term *pattern language* was coined by Christopher Alexander[188]. It describes a (finite) set of patterns which enables the creation of an "infinite" number of combinations of single patterns. To support this, patterns in a language must not stand isolated from each other. Instead, they have to complement one another in a synergetic way, and at the same time provide instructions and guidance on possible combinations, just like natural languages not only offer syllables and words; in order to form a sentence, you will also need the respective language's grammar to align the words. Generally, pattern languages are expected to be one of the major design knowledge management tools in the future[189]. Put visually, in a pattern language every pattern is located in the center of a network which connects it with other patterns. Especially these inter-connections are as important for the language as the patterns themselves, as they show dependency and refinement relations, among others. Table 4 shows analogies between natural languages and pattern languages as stated by Christopher Alexander.

Table 4: Analogies between natural languages and pattern languages.[190]

Natural Language	Pattern Language
Words	Patterns
Rules of grammar and meaning which give connections	Patterns which specify connections between patterns
Sentences	Buildings and places

Due to the fact that a smaller number of patterns are also potentially capable of producing a large number of solutions[191], even a subset of the patterns of a pattern language can be considered a pattern language itself, though targeted at a smaller application area.

[188] Alexander (1979)
[189] See van Welie and van der Veer (2003)
[190] Source: Reproduced from Alexander (1979, p. 187)
[191] Alexander et al. (1977, p. xxxv)

2.2.2.1.1 Generativity of Pattern Languages

Patterns in mind are different from patterns in the real world, even though they are mental images thereof[192]. The mental image of a pattern is enriched by the knowledge of how to generate the pattern in the real world. So telling what to do in order to build the desired system[193], generative patterns are active and dynamic, as opposed to non-generative[194] patterns, which describe "*recurring phenomena without necessarily saying how to reproduce them.*"[195] In this sense generative patterns, as well as the connections and dependencies between these patterns in a pattern language, have to be seen as rules for generating the desirable pattern, or system of patterns in the real world.

Example

Figure 9 shows an example of a network of related patterns from a Client/Server Frameworks pattern language.

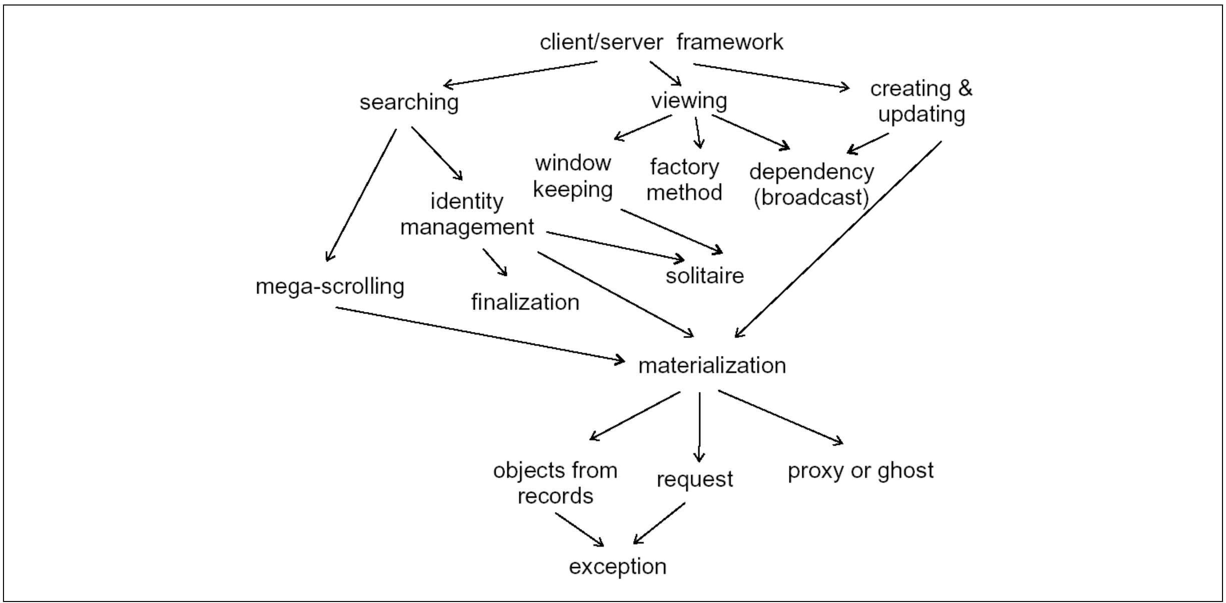


Figure 9: Relationships between patterns in a Client/Server Frameworks pattern language.[196]

---

[192] Cf. Alexander (1979)

[193] Alexander (1979, p. 182-183) additionally underlines the imperative aspects of generativity: Such a pattern not only shows how to do it, but also that it has to be done, "*in order to maintain a stable and healthy world.*"

[194] Coplien (1998) introduces the term "*Gamma patterns*" as a synonym for non-generative patterns, as the pattern catalog in Gamma et al. (1995) stems from observing structures in existing systems and frameworks.

[195] Appleton (2000)

[196] Source: Wolf and Liu (1995)

The arrows in Figure 9 show which lower-level patterns a higher-level pattern depends on. This concept introduces a hierarchical organization into the pattern collection: The patterns at the lowest level are more concrete design patterns, while the higher-level patterns are more abstract architectural or analysis patterns[197].

2.2.2.2 Pattern Catalogs

A pattern catalog comprises a loosely or informally related collection of patterns[198], whereas each of the patterns can also be used stand-alone[199]. Additionally, the patterns may be broadly categorized[200] to create families of patterns[201], and be presented uniformly (i.e., using a consistent form[202]). Figure 10 depicts the well-known GoF Design Pattern catalog with its structurally inter-related patterns.

[197] Kendall (1998)

[198] Cf. Appleton (2000), Arnold and Podehl (1999), Buschmann et al. (1996), Gamma et al. (1995), Riehle and Züllighoven (1996), Zimmer (1995)

[199] However, this does not mean that a pattern catalog exclusively comprises completely independent patterns.

[200] This is almost always the case, such as in Gamma et al. (1995). Fowler (1998), for example, does not adhere to uniformity. For a general discussion on pattern layout see Section 2.2.2.5.6.

[201] Gamma et al. (1993)

[202] See Section 2.2.3.1

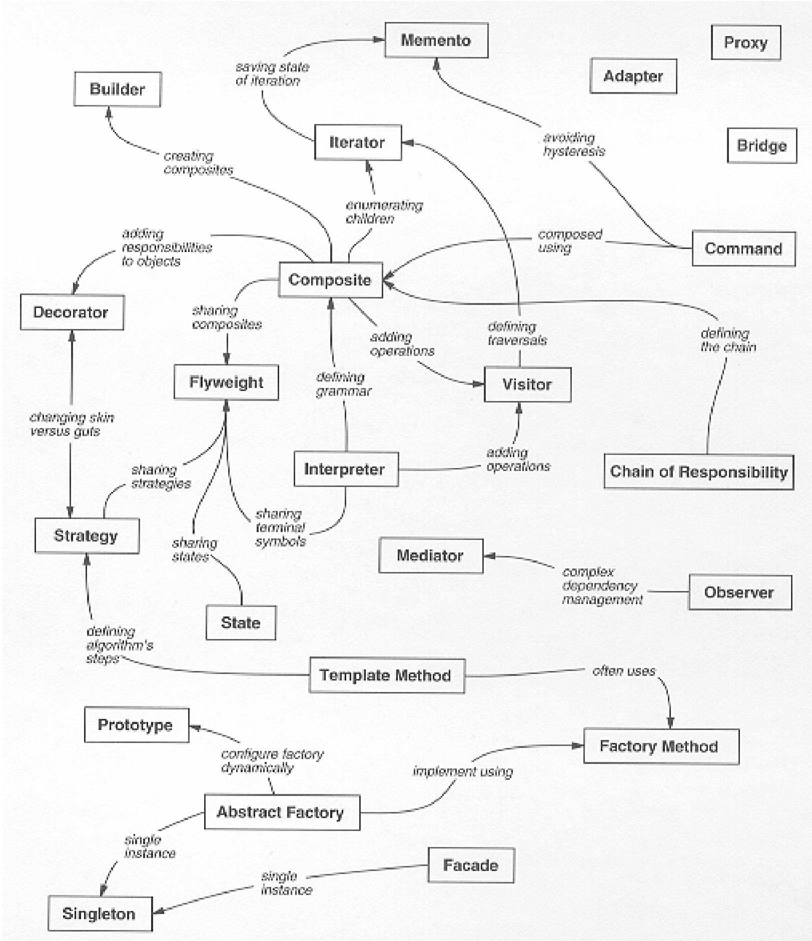


Figure 10: GoF Design Pattern catalog.[203]

The relations in a catalog are mostly based on similarity or on complementary aspects between related patterns, and do not explicitly address solving more complex super-ordinate problems[204]. This can also be found in Zimmer's work[205] on design pattern relationships after investigating and classifying the relations verbally described within the GoF pattern catalog. He identified three relationship categories between pairs (X, Y) of design patterns:

1) ***X uses Y in its solution***: the solution of Y represents one part of X's solution.

2) ***X is similar to Y***: The kind of problem addressed by X and Y is similar.

3) ***X can be combined with Y***: as opposed to X uses Y, this relation shows a typical combination of two patterns, e.g., "X traverses Y".

[203] Source: Gamma et al. (1995, p. 12)

[204] Cf. Klose (2002)

[205] Cf. Zimmer (1995)

2.2.2.3 Pattern Systems

A Pattern system ties its constituent patterns together much more than pattern catalogs do. It is designed "*as a collection of patterns [...], together with guidelines for their implementation, combination and practical use [...]*"[206] In this respect it is very similar to a pattern language, with an important difference: The pattern system does not claim to cover every important aspect in a given domain, so it does not provide *computational completeness*, as required for pattern languages[207]. In the POSA[208] book, the first work to theoretically deal with the notion and implications of a pattern system, the authors propose six requirements that a pattern system has to meet in order to be capable of producing a system that fulfills both its functional and non-functional requirements[209]:

- ***Sufficient base of patterns*** supporting specification and refinement of the basic system architecture, as well as implementing the architecture in a specific language.
- ***Uniformity of pattern description***.
- ***Exposition of pattern relationships***: Which other pattern a pattern refines, which others it exposes, with which others it can be combined, and what alternatives are available.
- ***Organization of constituent patterns*** to support quick finding of the appropriate pattern for the problem at hand.
- ***Support for the construction of systems***, i.e. support for applying and implementing the patterns.
- ***Supporting its own evolution***, which is necessary in frequently changing environments: Patterns are added, they may change, improve, or be removed.

To fulfill the above requirements, the patterns in a system have to be adequately organized by adding deeper structure through categorization and rich pattern interaction[210] to a catalog of patterns. The attentive reader might have noticed that there are only minor differences in the assumptions about pattern catalogs and pattern systems. In a nutshell, pattern systems tie their constituents together more interactively than pattern catalogs do.

---

[206] Buschmann et al. (1996, p. 361)

[207] Cf. Alexander (1979), Appleton (2000)

[208] POSA is the recognized acronym for the work of Buschmann et al. (1996): "Pattern-Oriented Software Architecture"

[209] Cf. Buschmann et al. (1996, p. 361f)

[210] Arnold and Podehl (1999)

2.2.2.4 Pattern Handbooks

Riehle and Züllighoven[211] propose to reintroduce the notion of handbook[212] as a handy summary of relevant concepts of a domain[213], being structured as follows:

- ***Introduction*** and overview presenting leitmotif as well as the background needed to understand the patterns.
- Outline of the ***application domain***, containing typical workplaces, co-operations, problems, and solutions.
- A ***structured set of patterns*** at different levels of abstraction (as appropriate): conceptual patterns, followed by design patterns, followed by programming patterns.

2.2.2.5 Discussion of Pattern Organization Concepts

The previous Sections have made clear that it is essential for pattern writers to provide collections of patterns in a way that supports the pattern user in selecting and applying the patterns. There are different ways to organize pattern collections, and most of the notions required to do so have already been mentioned in previous Sections. Still there exist different approaches to pattern organization, yet serving a common goal, which lies in helping the pattern reader to find the pattern or family of patterns appropriate to the problem at hand[214]. Usually, patterns are organized by defining two criteria, yielding a two-dimensional *pattern space* that can simply be depicted in matrix form. The following sub-Sections outline the method of organization in selected, representative pattern approaches.

2.2.2.5.1 Pattern Organization in the Alexandrian Patterns

Alexander's architectural patterns are divided into three levels[215]:

1) ***Towns***: comprises 94 global or large patterns, describing the layout of whole towns and communities

[211] Riehle and Züllighoven (1996)

[212] Anderson (1993)

[213] Conceptually, a pattern handbook is the same as a pattern language, but the authors tried to avoid the term "pattern language", as they saw no relation to a linguistic or computer science sense of the term "language". Additionally, they felt that other researchers were also reluctant to use that term. However, six years later it can be stated that "pattern language" is the prevalent term in use today.

[214] The pattern collection as a whole is useless if pattern readers have to read, analyze, and understand every pattern in detail to find the one they need (cf. Buschmann et al. (1996, p. 362)).

[215] Alexander et al. (1977)

2) ***Buildings***: consists of 110 patterns which can be applied on groups of buildings and individual buildings.
3) ***Construction***: comprises 49 patterns which show in detail how to build the buildings.

Within the levels there are additional layers, reaching from larger and more comprehensive patters, to smaller and narrower patterns. Different from more recent approaches, these levels and layers are not depicted in a tabular or diagrammed form, but sequentially, as shown below in Figure 11.

1. INDEPENDENT REGIONS
[W]ithin each region work toward those regional policies which will protect the land and mark the limits of the cities[:]
2. THE DISTRIBUTION OF TOWNS
3. CITY COUNTRY FINGERS
[...]
7. THE COUNTRYSIDE

Figure 11: Sequential arrangement of Alexandrian patterns. [216]

2.2.2.5.2 Pattern Organization in the GoF Design Pattern Catalog

The GoF pattern catalog is organized by using two top-level criteria to divide the pattern space, supporting fast lookup and comparability of the patterns[217]:

- ***Purpose***, which reflects what the pattern does. Any pattern serves one of the following three purposes: *creational* (object creation), *structural* (composition of classes or objects), or *behavioral* (interaction and responsibility distribution between classes or objects).
- ***Scope***, which tells the reader whether the pattern applies to *classes*, or *objects*.

Using these criteria to organize the catalog's 23 design patterns yields the pattern matrix as depicted in Table 5.

216 Source: Reproduced from Alexander et al. (1977, p. xix)
217 Gamma et al. (1995, p. 9-11)

Table 5: Organization of the GoF pattern catalog.[218]

		Purpose		
		Creational	**Structural**	**Behavioral**
Scope	**Class**	Factory Method	Adapter (class)	Interpreter Template Method
	Object	Abstract Factory Builder Prototype Singleton	Adapter (object) Bridge Composite Decorator Facade Flyweight Proxy	Chain of Responsibility Command Iterator Mediator Memento Observer State Strategy Visitor

2.2.2.5.3 Pattern Organization in the POSA System of Patterns

The POSA system of patterns[219] defines two criteria for pattern organization[220]:

- ***Pattern category***: Three different pattern categories are distinguished: *architectural patterns*, *design patterns*, and *idioms* (these categories are closely related to important phases in software development: analysis, design, and coding).
- ***Problem category***: Different problems can be categorized into groups. As depicted on the vertical axis of Table 6, ten problem categories are defined.

Assigning each pattern a *pattern category* and one or more *problem categories* yields the pattern space as shown in Table 6.

[218] Source: Reproduced from Gamma et al. (1995, p. 10)

[219] Buschmann et al. (1996)

[220] Buschmann et al. (1996, p. 362-366) use the term *classification* instead of *organization.*

Table 6: Pattern categorization in the POSA system.[221]

	Architectural Patterns	Design Patterns	Idioms
From Mud to Structure	Layers Pipes and Filters Blackboard		
Distributed Systems	Broker Pipes and Filters Microkernel		
Interactive Systems	MVC PAC		
Adaptable Systems	Microkernel Reflection		
Structural Decomposition		Whole-Part	
Organization of Work		Master-Slave	
Access Control		Proxy	
Management		Command Processor View Handler	
Communication		Publisher-Subscriber Forwarder-Receiver Client-Dispatcher-Server	
Resource Handling			Counted Pointer

2.2.2.5.4 Pattern Organization in Analysis Patterns

Fowler uses only one criterion to organize the patterns in his book, which just splits the book in two major Sections. Patterns fall into two categories[222]:

- ***Analysis Patterns*** as proven constructs in business modeling.
- ***Support Patterns*** that show how to apply the Analysis Patterns.

2.2.2.5.5 Pattern Organization in the Pedagogical Patterns Project

The Pedagogical Pattern Project (PPP) consists of several pattern sub-collections of patterns written by different authors, as listed below[223]:

---

[221] Source: Reproduced from Buschmann et al. (1996, p. 366)

[222] Fowler (1998, p. 8)

[223] Cf. Pedagogical Patterns Project (2002)

- ***Patterns for Active Learning***[224] is a pattern collection which focuses on engaging the students and keeping them active in learning settings.
- ***Feedback Patterns***[225] is a pattern collection which focuses on providing feedback to and obtaining feedback from participating students.
- ***Patterns for Experiential Learning***[226]: This collection focuses on many aspects of experiential learning, mostly on what is needed to learn by experimentation and by drawing on the students' own experiences.
- ***Patterns for Gaining Different Perspectives***[227] is a collection of patterns that deals with the diversity of instructional techniques. Different learners learn differently, and so the effective instructor must be able to help students encounter material in different ways.
- ***Seminars***[228] is a pedagogical pattern language about teaching seminars effectively. This pattern language is intended for those instructors in the industry or academia, who are not studied educators.
- ***Learning to Teach and Learning to Learn***[229]: The intended audience of this pattern collection is educators who care not only about what they are teaching but also about how they are mediating the topics. It addresses the learning on the students' side as well as the teaching on the educator's side.
- ***Fourteen Pedagogical Patterns***[230] comprise patterns from the initial approaches to a pattern language for Computer Science course development.
- ***Teaching from different Perspectives***[231] proposes successful techniques for teaching and learning. These patterns are primarily intended for novices.

The PPP authors do not explicitly categorize the patterns. Instead, there exist different options of browsing through patterns, as shown in the homepage in Figure 12: by *subject*, by *learning objective*, by *teaching/learning element*, by *name*, and by *author*.

[224] Eckstein, Bergin and Sharp (2002b)
[225] Eckstein, Bergin and Sharp (2002a)
[226] Eckstein et al. (2001)
[227] Bergin et al. (2001)
[228] Fricke and Völter (2000)
[229] Eckstein (2000)
[230] Bergin (2002)
[231] Eckstein et al. (2003)

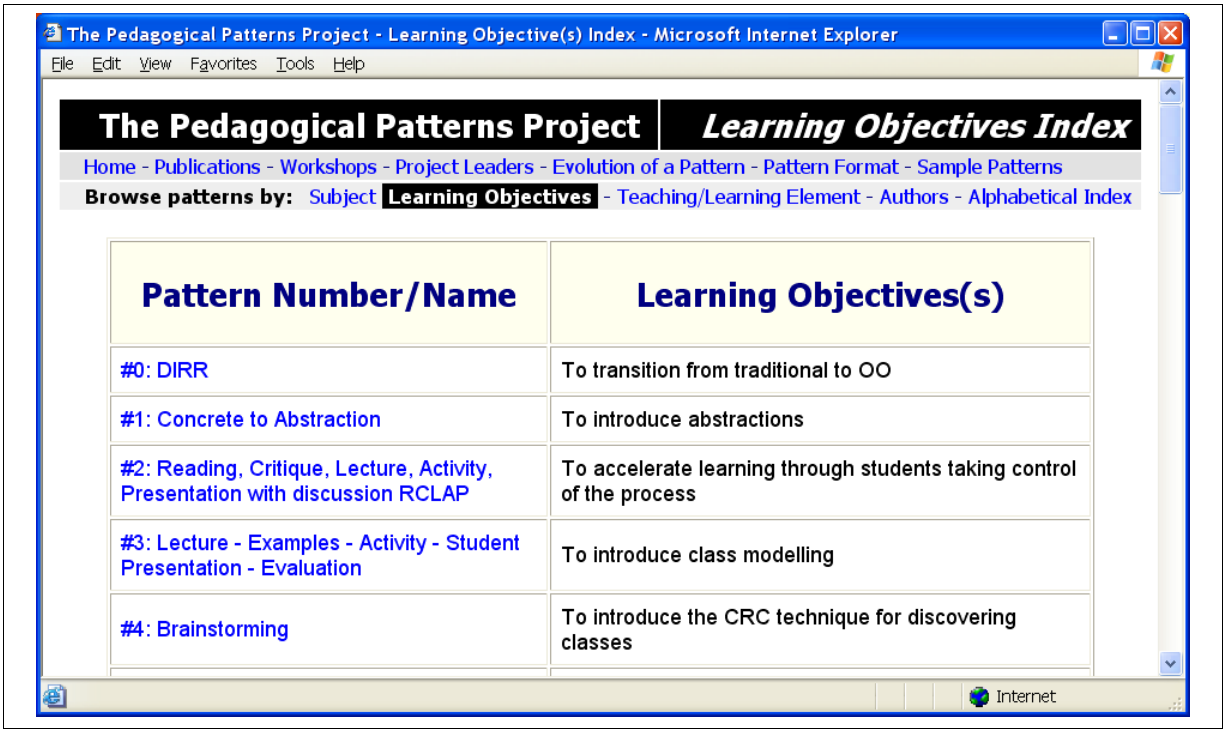


Figure 12: Index of the Pedagogical Pattern Project, browsed by learning objectives. [232]

Some sub-languages of the PPP offer pattern maps[233] showing how the patterns relate to each other, and most of them offer a *Quick Access Table*, which shows the problem in the left column and the patterns suitable for the respective problem in the right column. This can be seen in Table 7.

Table 7: Quick Access Table for "Feedback Patterns." [234]

Problem	Patterns
You want to ensure that the participants understood the topic.	Feedback, Differentiated Feedback, Try It Yourself, Kind of Exam
Participants might have understood the theory, but the have never applied it.	Try It Yourself, Self Test
Participants don't trust in their own knowledge.	Explain It Yourself, Peer Feedback, Embrace Correction, Peer Grading, Student Online Portfolio
You want to provide feedback.	Positive Feedback First, Early Warning
You want to make the participants less dependent on yourself.	Peer Grading, Embrace Correction, Student Online Portfolio

[232] Source: Screenshot of the PPP homepage (Pedagogical Patterns Project (2000))

[233] For example, Fricke and Völter (2000)

[234] Source: Reproduced from Eckstein, Bergin and Sharp (2002b)

Problem	Patterns
You want to ensure that participants learn from their own experience.	Embrace Correction, Grade It Again Sam
You want to make the gained knowledge visible.	Student Online Portfolio, Grade It Again Sam, Gold Star
Participants don't know how to prepare for the exam.	Self Test, Mock Exam
You want to ensure fair (individual) grading.	Fair Grading, Fair Project Grading, Key Ideas Dominate Grading, Grade It Again Sam
You want to grade teams fairly.	One Grade For All, Peer Grading, Fair Team Grading
You want to know if you and the course were useful for the students.	Acquire Participants' Feedback, Anonymous Feedback

2.2.2.5.6 Pattern Organization in the E-LEN Project

The E-LEN project, which is presented in more detail in the Section on closely related approaches[235], created patterns within four Special Interest Groups (SIGs). These SIGs provide the main structuring category. Within these SIGs, patterns are grouped by additional categories. The SIGs are[236]:

1) *Learning resources and Learning Management Systems*. Includes 10 patterns organized by categories Access patterns, Learning patterns, Instructional patterns, Informational patterns, and Administrational patterns. Some of these patterns have previously been published in a journal[237] and at a Pattern Languages of Programs (PLoP) conference[238].

2) *Lifelong learning*: Includes 22 patterns. No additional categories were specified for that SIG.

3) *Collaborative learning*: Includes 5 patterns. No additional categories were specified for that SIG.

4) *Adaptive learning*: Includes 6 patterns. No additional categories were specified for that SIG.

[235] See Section 4.1.2, p. 150

[236] See http://www2.tisip.no/E-LEN/patterns_info.php

[237] See Avgeriou et al. (2003b)

[238] See Avgeriou, Papasalouros and Retalis (2003a)

2.2.3 Inside Patterns

2.2.3.1 Pattern Form

What is still left open is how patterns are represented. As already mentioned earlier, patterns belonging to collections are typically represented uniformly, using a certain *pattern form*[239], which can be defined as "*a finite number of visible and distinguishable components and their relationships.*"[240] Different problem areas afford different ways of describing patterns, thus different forms. Many of the pattern approaches existing today have developed their own form of representation, and yet they all adhere to a certain minimal core of components, namely the elements a pattern consists of per definition[241]: problem, context, solution, and a well-chosen name. Generally, two basic styles of pattern forms can be distinguished:

- *Prose style*, a form introduced by Alexander; adopted by some pedagogical pattern sub-collections.
- *Template style*, a form predominant in software patterns, using named sections.

2.2.3.1.1 Alexandrian Form

One of the most widely used and adapted forms, is the Alexandrian form[242], which is organized as follows[243]:

- The first line consists of the *pattern number*, the capitalized *pattern name*, and a number of *asterisks* indicating how certain the authors are about the given pattern: Two asterisks indicate a true invariant, one asterisk indicates that there is progress in finding the invariant, and zero asterisks show failure in providing a true invariant.
- Following the first line is usually a *picture* from the real world that relates to the pattern.
- The following paragraph *relates* the given pattern *to previous patterns.*
- The next short section outlines the *problem* and is given in bold typeface.
- A detailed description of *forces* relevant in the given problem context along with hints on how to resolve the forces.

---

239 There exist many synonyms for the term pattern form, such as *pattern template* (e.g., Brown, McCormick III and Thomas (1999)), *descriptive framework* (e.g., Motschnig-Pitrik, Randa and Vinek (2002)), or *pattern format* (e.g., Gamma et al. (1995)). The term preferred in this paper is *pattern form.*

240 Riehle and Züllighoven (1996, p. 4)

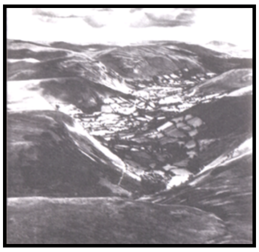
241 Cf. Figure 8

242 Alexander et al. (1977)

243 See the example in Figure 13

- A bold faced paragraph showing the core of the *solution* to the problem, along with one or more *diagrams* and pictures.
- The last paragraph gives an outlook on the *following patterns* and their relations to the given pattern.

4 AGRICULTURAL VALLEYS *



. . . this pattern helps maintain the INDEPENDENT REGIONS (1) by making regions more self-sufficient agriculturally; and it will create CITY COUNTRY FINGERS (3) almost automatically by preserving agricultural land in urban areas. But just exactly which land ought to be preserved, and which land built upon?

* * *

The land which is best for agriculture happens to be best for building too. But it is limited – and once destroyed, it cannot be regained for centuries.

In the last few years, suburban growth has been spreading over all land, agricultural or not. It eats up this limited resource and, worse still, destroys the possibility of farming close to cities once and for all. But we know, from the arguments of CITY COUNTRY FINGERS (3), that it is important to have open farmland near the places where people live. Since the arable land which can be used for farming lies mainly in the valleys, it is essential that the valley floors within our urban regions be left untouchen and kept for farming.

The most complete study of this problem that we know, comes from Ian McHarg (*Design With Nature*, New York: Natural History Press, 1969). In his "Plan for the Valleys" (Wallace-McHarg Associates, Philadelphia, 1963), he shows how town development can be diverted to the hillsides and plateaus, leaving the valleys clear. The pattern is supported, also, by the fact that there are several possible practical approaches to the task of implementation (McHarg, pp. 79-93).

Therefore:

Preserve all agricultural valleys as farmland and protect this land from any development which would destroy or lock up the unique fertility of the soil. Even when valley are not cultivated now, protect them: keep them for farms and parks and wilds.

* * *

Keep town and city development along the hilltops and hillsides – CITY COUNTRY FINGERS (3). And in the valleys, treat the ownership of the land as a form of stewardship, embracing basic ecological responsibilities – THE COUNTRYSIDE (7) . . . .

Figure 13: The "Agricultural Valleys" pattern from Alexander's pattern language.[244]

[244] Source: Reproduced from Alexander et al. (1977, p. 26-28)

Generally, patterns vary noticeably in length, some need less than one page as the one in Figure 13, some occupy up to 15 and more letter-sized pages.

2.2.3.1.2 Gamma Form

Unlike the prose-style Alexandrian patterns, the Gamma Design Patterns are described using a template which comprises 13 named sections to give the patterns a uniform structure[245]:

- ***Pattern Name*** and ***Classification***: The name succinctly conveys the essence of a pattern. The classification gives a purpose/scope value pair according to Table 5.
- ***Intent***: A short statement conveying the rationale and intent of the pattern, and what design problem it addresses.
- ***Also Known As***: Other names for the pattern, if any.
- ***Motivation***: A scenario illustrating a design problem and how the pattern solves it.
- ***Applicability***: Gives situations in which the pattern can be applied and how to recognize these situations.
- ***Structure***: A diagram of the classes in the pattern using a notation based on the *Object Modeling Technique (OMT)*[246]; interaction diagrams[247] are used to illustrate sequences and collaborations.
- ***Participants***: Participating objects and classes and their responsibilities.
- ***Collaborations***: How participants collaborate to carry out their responsibilities.
- ***Consequences***: Trade-offs and results of using the pattern.
- ***Implementation***: Pitfalls, hints, techniques, and language-specific issues regarding the implementation of the pattern.
- ***Sample Code***: Code fragments that illustrate a possible implementation of the pattern.
- ***Known Uses***: At least two examples of known uses in real systems.
- ***Related Patterns***: Shows patterns closely related to this one and important differences.

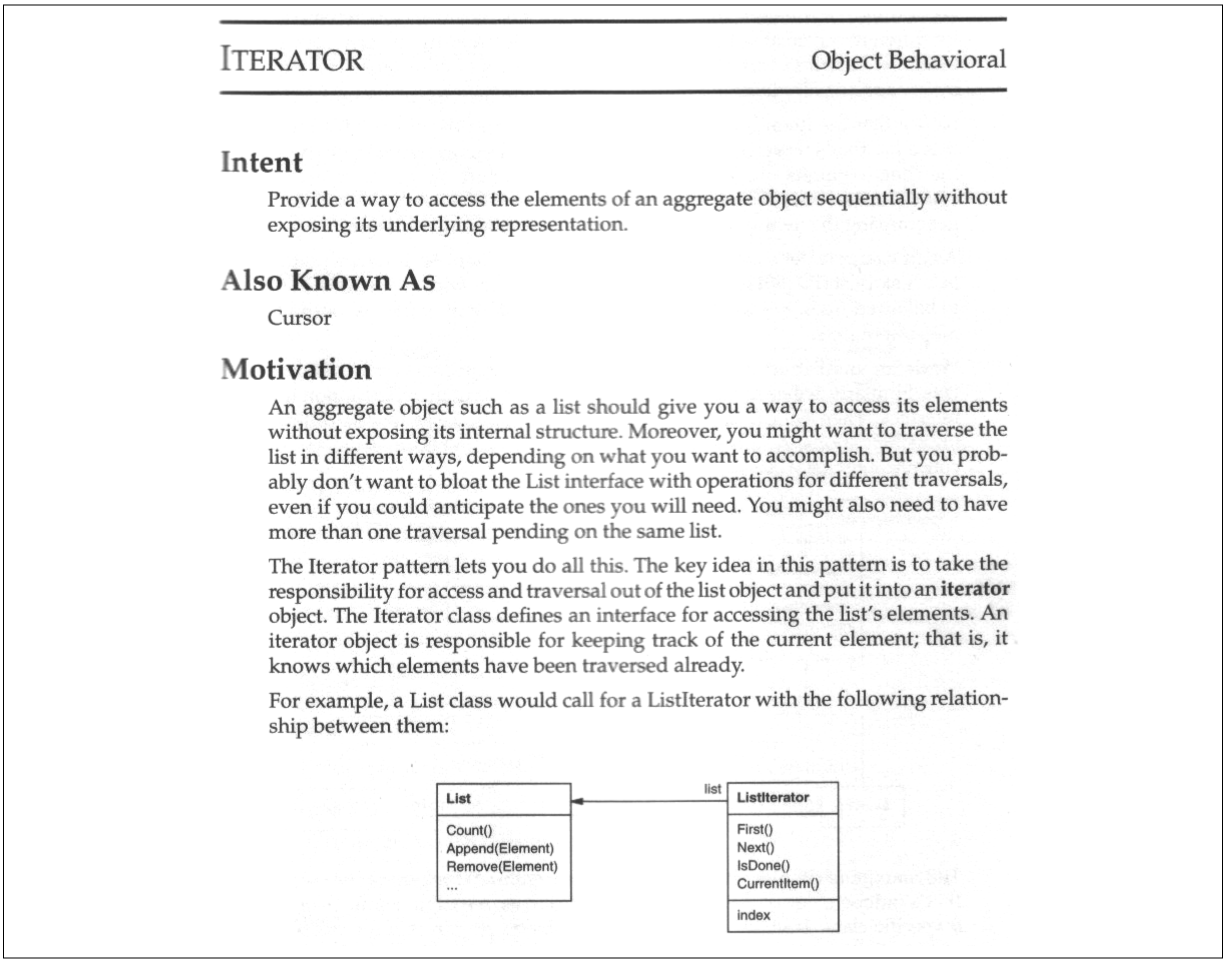
As shown in Figure 14, each of these sections comprises text, and, when appropriate, diagrams or source code snippets.

---

[245] Cf. Gamma et al. (1995, p. 6-7)

[246] Rumbaugh et al. (1991); cf. Section 2.3.3.1

[247] Booch (1994), Jacobson et al. (1992)



ITERATOR — Object Behavioral

Intent

Provide a way to access the elements of an aggregate object sequentially without exposing its underlying representation.

Also Known As

Cursor

Motivation

An aggregate object such as a list should give you a way to access its elements without exposing its internal structure. Moreover, you might want to traverse the list in different ways, depending on what you want to accomplish. But you probably don't want to bloat the List interface with operations for different traversals, even if you could anticipate the ones you will need. You might also need to have more than one traversal pending on the same list.

The Iterator pattern lets you do all this. The key idea in this pattern is to take the responsibility for access and traversal out of the list object and put it into an **iterator** object. The Iterator class defines an interface for accessing the list's elements. An iterator object is responsible for keeping track of the current element; that is, it knows which elements have been traversed already.

For example, a List class would call for a ListIterator with the following relationship between them:

Figure 14: Clipping from the first page of the "*Iterator*" pattern.[248]

2.2.3.2 Pattern Quality

Alexander initiated a highly philosophical discussion about the quality of patterns and its measurability. He introduced the concept of the *quality without a name* (widely referred to by the acronym QWAN) which imparts incommunicable beauty and value to a structure and, universally recognizable, represents an objective measure of beauty, perceived by individuals independent of culture and history[249]. Alexander's conception of architectural design was based on three basic notions: the *quality*, the *gate*, and the (timeless) *way*, whereas the gate is represented by a pattern language that describes the timeless way of incrementally designing a structure that possesses the quality without a name. Reaching the QWAN is the ultimate goal in supplying architectural patterns.

[248] Source: Gamma et al. (1995, p. 257)

[249] Cf. Alexander (1979)

Appleton states[250] that there are many who feel that the concept of QWAN is too "*whimsical and metaphysical*," lacking scientific foundation or tangibility required in engineering disciplines; but he recognizes that...

> ...in many respects, an individual's sense of the QWAN is also about cognitive [judgment]. Every master designer develops their own highly honed intuition which is borne from extensive experience. Although this "intuition" may be subjective, it can be uncannily accurate and give the designer an almost instinctive sense of what will work and what [will not] (even before the measures are brought to bear to try and verify it). This stems from the designer being intimate with the design and internalizing it at a visceral level, almost to the point of becoming an inhabitant whose sensory network is "plugged-in" to the system. If a pattern can impart to its readers and users, this same "plugged-in" feeling of being connected to the design and deeply comprehending it, then in theory it will impart to the reader the same cognitive feeling of its aptness that the designer experienced. If a pattern succeeds in this attempt, then all who see and use it will supposedly experience the resonant feeling of beauty and harmony that the QWAN is supposed to evoke.

In a more tangible way than Alexander, Lea[251] proposes six ideal properties or qualities that well-written patterns should exhibit:

- ***Encapsulation***: Each pattern is independent and precisely formulated, and addresses a well-defined problem and solution, respectively.
- ***Generativity***: Each pattern describes a self-standing process of constructing realizations[252].
- ***Equilibrium***: Each pattern identifies an invariant and the equilibrium provides the reason for each step.
- ***Abstraction***: Each pattern is an abstraction of practical experience or theoretical knowledge; general within the given context, but not necessarily universal.
- ***Openness***: Pattern hierarchies are open as they have no top or bottom; they may be refined or extended to more detailed levels.
- ***Composibility***: Patterns are hierarchically related, whereas coarse-grained patterns are layered on top of, relate, and constrain fine-grained ones.

Generally, there seems to exist no universal, objective measure of pattern quality, as it not only depends on the objective quality of the pattern itself, but heavily on the skills and experiences of the pattern users in the respective discipline, as well as in employing patterns. So the human factor in the usage of patterns, both on the side of the user and the author, is important throughout the whole pattern life-cycle, which is subject to discussion in the following Section.

[250] Appleton (2000)

[251] Lea (1994)

[252] Cf. Section 2.2.2.1

2.2.3.3 The Pattern Life-Cycle

Each pattern or pattern collection goes through a number of phases throughout its *life-cycle*[253]. Beginning from experience and expert knowledge in a certain problem domain, problems, solutions, and common situations are abstracted in a more or less formalized way (depending on the approach). After finding appropriate ways of storing and publishing the patterns, they are ready for selection and application through pattern users. From employing the patterns and, through constantly changing environments and preconditions inside the problem context, some patterns need to be updated or replaced to reflect new experiences, findings, and variations in the environment.

A generic model of the pattern life-cycle is depicted by Figure 15. The arrows connecting the phases show how patterns capture experience on the one hand, and influence experience and knowledge on the other hand: Writing a pattern, for example, not only captures experience and knowledge, but also *contributes* to it[254].

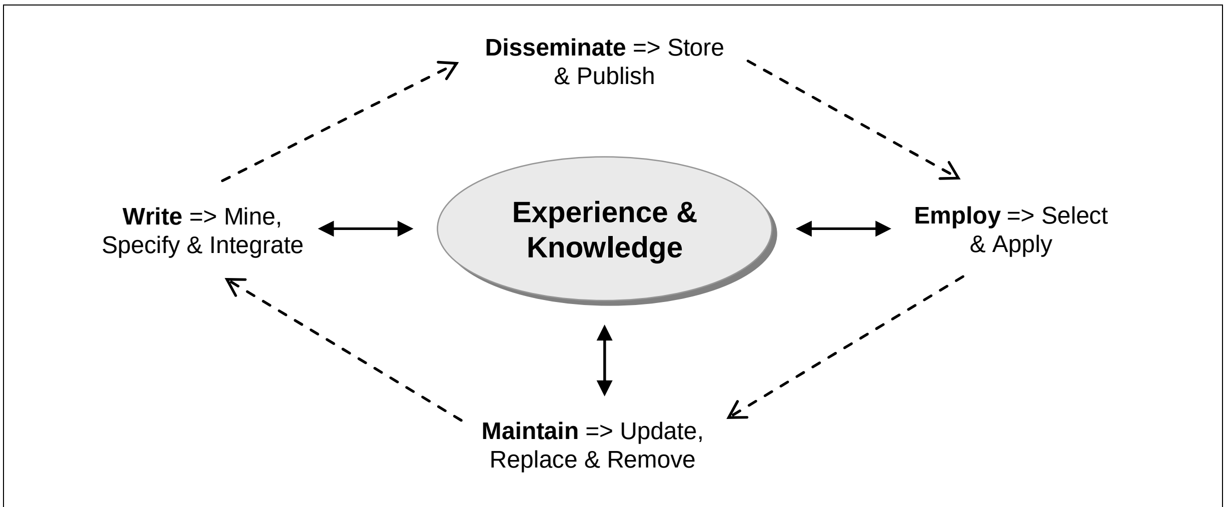


Figure 15: A model of the pattern life-cycle.

2.2.3.3.1 Experience

Repeatedly applying good and bad solutions to various kinds of problems in a domain or discipline is what makes an expert out of a novice. Among other factors like domain-specific knowledge, it is the experience which experts have internalized, and which makes them capable of solving complex problems by relating new situations to situations previously successfully solved. As explicit representations of experiences and knowledge, patterns go through a

[253] This Section relies partly on discussions found in Buschmann et al. (1996), partly on own experiences in finding and writing patterns, and partly on other sources referenced. The term "pattern life-cycle" is obviously a novel concept. Generally, there seem to be few explicit discussions on this important aspect of patterns in the literature.

[254] Derntl and Motschnig-Pitrik (2003a), (2003b)

number of phases in their life-cycle: After repeatedly applying successful solutions, a pattern forms itself in mind. Intending to write down a pattern, the next task is to abstract and decompose the experiences in order to specify a single pattern or a number of related patterns. Also, experience is involved in the employment and maintenance of patterns, both actively (e.g., as input provider) and passively (e.g., as feedback receptor).

2.2.3.3.2 Specification

As already mentioned above, each discipline has its own methods and tools, and a pattern has to employ these methods to facilitate the realization of desirable structures (e.g., in design disciplines) or procedures and interactions (e.g., in pedagogy). To give an example, Riehle and Züllighoven[255] differentiate three levels of pattern concepts, whereas each level has its own specific methods of description. The following levels are layered top-down:

- The form of ***conceptual patterns*** is described by means of the terms and concepts of the application domain.
- The form of ***design patterns*** is described by means of software design constructs.
- Finally, ***programming patterns*** are described by means of programming language constructs.

Writing down experience abstractly in a more or less formalized way[256] as a pattern not only captures experience, but actively contributes to understanding and *arranging* experience, particularly when decomposing complex structures, procedures, and interactions into smaller, more comprehensible units[257]. Specifying patterns is commonly considered to be a hard task, which is confirmed by the fact that there even exist pattern languages for pattern writing[258]. Equally important, newly discovered patterns have to be integrated with existing patterns of a pattern collection by defining relationships and classifying the pattern according to the organization scheme of the approach[259], so pattern writing is also part of the maintenance of pattern collections.

2.2.3.3.3 Dissemination

After specifying a collection of patterns which represent the toolkit necessary to solve real-world problems in the respective domain, the patterns have to be disseminated to allow for application in various contexts by different users. In the literature, there does not seem to be

---

[255] Riehle and Züllighoven (1996, p. 7-8)

[256] See Section 2.2.3.1 on pattern forms. Currently, there are numerous efforts on investigating the use of highly formalized design patterns to allow for automatic application and tool-support in building software systems, e.g., as reported in Budinsky et al. (1996), Eden, Hirshfeld and Yehudai (1998), Eden, Yehudai and Gil (1997).

[257] Derntl and Motschnig-Pitrik (2003a), (2003b)

[258] Meszaros and Doble (1998)

[259] See Section 2.2.3.3.5

excessive investigation on this phase of the life-cycle. Looking at different pattern collections, many of them are accessible online only[260], many others in printed form only[261], some are accessible both online and in printed form[262]. In any approach, it seems vital to supply the pattern users with effective means of retrieving the appropriate pattern(s) for their problem(s). In order to do so, patterns are mostly organized in matrix-like structures[263] to allow for quick retrieval.

2.2.3.3.4 Application

According to Christopher Alexander, the application of patterns follows a process called *piecemeal growth*[264], whereby a problem is solved by successively applying patterns along the hierarchy from higher-level to lower-level patterns. Instantiating a pattern generates a new context which itself may match with the initial context of another pattern. This iterative and incremental process is repeated recursively until either the problem is solved, or until there are no more applicable patterns at hand. There is no uniform guide on the process of instantiating a pattern, i.e. putting a pattern in use, as it largely depends on the application domain: Modeling concepts in the analysis phase of a software project by using analysis patterns is certainly different from constructing a building by applying architectural patterns.

Therefore, a minimal set of prerequisites has to be met by pattern authors in order to support effective application by pattern users:

- The provision of a ***general intent*** statement for the pattern approach at hand and how it makes use of the concept of pattern[265] helps users to find the pattern collection that is capable of solving their problems.
- Description of pattern ***form*** and concept of ***organization*** along with guides to notations, if appropriate[266]. As an example, see Section 2.2.3.1.2. This supports the users in understanding each pattern and how the patterns are organized.
- Guidance on ***selecting*** a pattern that is capable of solving the current problem[267]. To achieve this, pattern authors usually supply tables where the patterns are listed in a

[260] Visit http://www.hillside.net/patterns for a comprehensive list of patterns and pattern languages.

[261] For example, Buschmann et al. (1996), Fowler (1998), Gamma et al. (1995), Schmidt et al. (2000)

[262] E.g., selected papers from the PLoP (Pattern Languages of Programs) Conference series are published in the "PLoP Design" books: Coplien and Schmidt (1995), Harrison, Foote and Rohnert (1999), Martin, Riehle and Buschmann (1998), Vlissides, Coplien and Kerth (1996). For a list of PLoP conferences visit http://www.hillside.net/conferences.

[263] For example, Quick Access Tables or categorized pattern spaces (cf. Section 2.2.2.5)

[264] Alexander et al. (1977)

[265] For example, Section 1.1 in Gamma et al. (1995): "What is a Design Pattern?"

[266] For example, Section 1.5 in Buschmann et al. (1996): "Pattern Description"

[267] For example, Section 1.7 in Gamma et al. (1995): "How to Select a Design Pattern"

categorized form. Additionally, some authors supply step-by-step procedures to pattern selection, e.g. in the POSA pattern system[268]:

1) Specify the problem
2) Select the pattern category
3) Select the problem category
4) Compare the problem descriptions
5) Compare benefits and liabilities
6) Select the variant that best implements the solution to the design problem
7) Select an alternative problem category

- Guidance on ***using*** a pattern, once selected[269]. Authors often provide step-by-step approaches to using patterns, e.g. in using Design Patterns[270]:
  1) Read the pattern once through for an overview
  2) Go back and study the Structure, Participants, and Collaborations sections
  3) Look at the Sample Code section to see a concrete example of the pattern in code
  4) Choose names for pattern participants that are meaningful in the application context
  5) Define the classes
  6) Define application-specific names for operations in the pattern
  7) Implement the operations to carry out the responsibilities and collaborations in the pattern.

One of the most important aspects in pattern application is the provision of concrete examples, where users can see how to apply the patterns. Most pattern approaches supply examples as part of the patterns themselves[271], others separate patterns and reference applications[272].

2.2.3.3.5 Maintenance and Evolution

Generally, four basic activities to keep a pattern collection up-to-date can be identified[273]:

268 Cf. Buschmann et al. (1996, p. 368-370) and Table 6 in this paper

269 For example, Section 1.6 in Fowler (1998): “Using the Patterns”

270 Cf. Gamma et al. (1995, p. 29-31)

271 For example, the Gamma form invented by Gamma et al. (1995) with the “Known Uses” and “Sample Code” sections

272 For example, Alexander’s pattern language was published separately from project reports where the patterns were applied – cf. Alexander (1981), (1983).

273 Cf. Buschmann et al. (1996)

1) ***Updating an existing pattern.*** Application of a pattern may show that the pattern description is incomplete or partly wrong. Or, changes in the problem environment (e.g. changes in technology) may require rewriting a pattern to keep it up-to-date.

2) ***Adding a new pattern***. Experience and pattern application may show that there exists no pattern matching the current problem configuration. In this case, after repeated observation of such a gap in the currently provided pattern space, a new pattern can be considered to be written. After a pattern is written, it needs to be integrated into the existing organization schema by connecting/relating it with other patterns, and by classifying it according to the schema's categories. New patterns may also require reconsidering of pattern form or organization.

3) ***Removing an outdated pattern***. A pattern gets outdated when the problem it addresses disappears or when there are better alternatives available. Completely removing a pattern from a collection can be dangerous, e.g. if legacy systems incorporating the pattern have to be maintained. It seems to be a better solution to mark the pattern as outdated.

4) ***Modify form and/or organization***. Certain developments may require the reconsideration of the pattern form and/or organization schema used to describe and classify the patterns; e.g., adding a family of new patterns may justify the creation of a new pattern category. Also, when a pattern collection grows rapidly it may be wise to split existing categories into sub-categories to prevent certain spots in the pattern space from getting too obscure and "over-populated."

2.2.4 Summary and Discussion on Patterns

Patterns have become a valuable tool in a variety of disciplines. The most important role today patterns have in software engineering, and particularly in software design, which is most closely related to the field where patterns are rooted, i.e. architecture. As patterns capture proven solutions to common problems, it was only a matter of time that this successful approach was adopted by frontiers in other disciplines, whereas most of them are related to software[274]. Patterns, in the literal meaning of the word, can be found everywhere: Patterns of thought, patterns of behavior, patterns of development, and many more. But Christopher Alexander was the first one to explicitly write useful patterns down on paper, seeking to enable the reuse of effective designs and practices and the establishment of a shared vocabulary thereof.

Alexander wrote his patterns in easily understandable, formatted prose style. Other researchers and practitioners who later adopted the pattern approach proposed different ways of

[274] For a comprehensive list of software patterns and pattern languages, visit the online catalog of the Hillside Group under http://www.hillside.net/patterns/onlinepatterncatalog.htm), or the Portland Pattern Repository under http://c2.com/ppr/index.html.

describing patterns. The *Gang of Four*, authors of the seminal work "Design Patterns," have defined a structured description template which enables the pattern users to compare and complement different patterns more easily. The content of a pattern not only includes text paragraphs, but also diagrams, tables, figures, and pictures as appropriate. Nevertheless, regardless of the form of a pattern, there is consensus on what a pattern has to contain: A description of a problem, of the context in which the problem occurs, and of a way to resolve the problem. Additionally, the pattern author should provide a meaningful name for the pattern and guidance on how to apply the pattern.

Alexander's intention was not only to solve a number of specific problems, but to provide a comprehensive collection of patterns in order to cover all potential design problems and useful techniques in architecture. Therefore, lending from the principles of natural languages, he introduced the concept of *pattern language*, acknowledging that patterns on their own may be useless without providing a "grammar" to align related patterns in order to solve complex problems. However, recognizing that achieving completeness is nearly impossible on the one hand, and doubting the utility of such an effort on the other hand, other pattern authors have proposed different ways of combining patterns: Loosely related patterns in pattern catalogs which cover certain problem categories in a field on the one hand, and more comprehensive pattern systems as collections of hierarchically structured and tightly interrelated patterns on the other hand.

Depending on the complexity of the current problem and the way a pattern collection is organized, the application of patterns mostly follows the so-called *piecemeal growth* approach by successively applying related patterns beginning with general higher-level patterns to more detailed lower-level patterns. To support the user in this process, pattern collections have to be organized and disseminated in a way that allows for effective and situated selection of the appropriate pattern(s) capable of solving the problem. The users applying the patterns in turn supply valuable feedback which helps the pattern authors in validating and improving their patterns. This dynamic process of continual evolution of pattern collections keeps them up-to-date with their changing environments and target domains.

The world-wide interest in the pattern approach is still steadily growing. In 1993, the *Pattern Languages of Programs* (PLoP) Conference was to be the first conference exclusively dedicated to patterns and pattern languages. Today, pattern authors frequently meet at a number of PLoP conferences taking place all over the world, among them EuroPLoP, ChiliPLoP, KoalaPLoP, MensorePLoP, SugarLoafPLoP, and VikingPLoP. However, pattern research has by far not reached the final destination yet. The current and past focus is mainly on development of new patterns[275], so there is a lot of research work left to be done regarding the dissemination, indexing, organization, improvement, description, application, and validation of patterns and pattern collections. According to interdisciplinary and visionary protagonists in the pattern movement, the pattern community in the long-run has to work towards jointly

[275] Cf. Buschmann et al. (1996, p. 423)

developing the *pattern universe*, a combination of pattern languages addressing any imaginable problem domain, accessible to everyone.

In this respect, this thesis aims to provide a pattern collection for a small, yet important aspect of a possible pattern universe, namely Person-Centered e-Learning.

2.3 Conceptual Modeling

A picture shows me at a glance
what it takes dozens of pages of
a book to expound.

(Ivan Turgenev, 1862)

2.3.1 General Issues and Terminology

Many of the problems we face, and systems we interact with today are too complicated to be comprehended and processed in their entirety by single persons. The fast growing complexity and diversity of technologies and knowledge has made ways of structuring and representing knowledge indispensable. One way of knowledge representation is conceptual modeling, which allows real, complex problems or situations to be represented visually in terms of related concepts. Such an approach not only enables documentation of systems but also exchange of system models given the understanding of the modeling technique's underlying notation or formalism. This applies not only to exchange between human individuals but also allows for exchange and automatic processing of models by computers.

There are many definitions of the term model, many of them very intricate, long and philosophical. Generally, as we want to discuss in concise terms, a model[276] may be seen as a useful representation or abstraction of some aspect of an original[277]. One of the primary aspects of building a model is to reduce the complexity of relationships and properties of the original while considering different external and internal viewpoints[278]. For example, in

[276] The discourse on modeling within the scope of this work is restricted to two-dimensional visual modeling aspects, e.g. drawings on paper. Plastic modeling and building realistic small-scale models of real objects will be ignored here.

[277] The term *original* is used purposely here, because it does not make any implications about properties of the modeled object(s). In similar definitions of a model the term *reality* is often used, which seems too specific when talking about representing concepts in general.

[278] Cf. Sommerville (1992, p. 66-71)

mathematics models are used to describe real phenomena with means of mathematic expression such as formulas and equations. This makes clear that one central aspect of a model is the language or technique it uses. The language may be very formal, like Algebra in mathematics, but may also be compiled of arbitrary figures and connectors. A verbal (natural language) representation may also serve as a model, but natural language descriptions are often too cumbersome and ambiguous, especially when complex concepts and relations have to be explained.

Firesmith[279] proposes a set of principles which characterize the building of "good" models. Even their focus is on software models, the concepts may seamlessly be transferred to modeling in general. The principles are:

- ***Abstraction***, representing essential characteristics and behaviors of modeled entities.
- ***Completeness***, ensuring that *all* essential abstractions are drawn.
- ***Confirmability*** of correctly building correct models.
- ***Independence***, assuring low *coupling*[280] among modeled entities and fostering hiding of extraneous information (***Information Hiding***), which increases understandability.
- ***Localization***, ensuring that all essential characteristics of each entity are used to model that entity.
- ***Modularity***, keeping complexity and size of models in check.
- ***Uniformity***, assuring consistency in modeling among different models.

What became very important to engineering disciplines is conceptual modeling, which aims to create graphical representations of concepts[281], entities[282], and behaviors in the real world, e.g., models of electric circuits, models of software systems, models of business processes, and many more thinkable. In order to draw benefit from the model and to make it interchangeable and computable, conceptual models are formal models using a closed, finite graphical notation with an unambiguous mapping between a concept and its graphical representation[283]. Conceptual models are usually defined in terms of a graph structure[284], linking nodes with edges.

---

[279] Cf. Firesmith (1993, p. 103)

[280] Cf. Firesmith (1993, p. 27): "*Coupling refers to the number of relationships and the amount of information flow between entities.*"

[281] The term *concept* in the domain of modeling was first mentioned by Ross Quillian in 1966 in his PhD thesis on semantic networks: concepts are inter-connected through associations – cf. Simons (1994)

[282] The term *entity* modeling was introduced by Chen (1976) within the scope of the Entity-Relationship Model (ERM), today still one of the most widely-used approaches in data modeling.

[283] Cf. Simons (1994). For example, in the UML a typical user interaction with the target system (i.e., a *use case*) is modeled by an ellipse, with the name of the use case drawn in the center of the el-

Basically, models can be divided into static and dynamic models. Static models represent the structure of the modeling domain, while dynamic models show the flow of events and interactions within this structure. [285]

2.3.2 Object-Orientation

This Section acts as a brief excurse into the world of object-oriented (OO) concepts. These concepts and their related terminology are crucial assets for the understanding of subsequent discussions and uses of object-oriented modeling, which is one key factor in modeling PCeL patterns.

Generally, object-orientation is a well-researched approach to constructing models of virtually any type of complex system. Particularly in software technology, OO concepts have been in use since the 1960s, when the programming language Simula has got equipped with most of the object-oriented features of today's programming languages. However, object-orientation did not attract much attention until about a decade ago.[286]

2.3.2.1 Basic Notions

The central concepts of object-orientation are *abstraction* and *encapsulation*, from which emerge operative building blocks of a system, called *objects*[287]. Each object is an "*abstraction that models all relevant aspects of a singe tangible or conceptual entity or thing from the […] solution space*."[288] This way, an object[289] encapsulates all relevant characteristics and behaviors of a certain aspect of a system. Each object also has a unique identity.

The main mechanism of human thinking is classification[290]. As for modeling, it would not be effective to model just concrete objects in a system modeling effort. In that sense, a *class*[291] is an abstract form to represent the type of an object and to provide a useful generic description for a set (= class) of objects. This generic description includes the class's relevant *attributes*, i.e., the structural and behavioral elements each *instance* (i.e., object of type class) of

lipse. The unambiguous mapping is that there exist no concepts or entities represented by ellipses, while on the other hand a use case is always represented by an ellipse.

284 Mylopoulos and Easterbrook (2003)

285 Cf. Jacobson, Ericsson and Jacobson (1995, p. 28); this distinction is especially important from a software system perspective.

286 Cf. Jacobson, Ericsson and Jacobson (1995, p. 45-46)

287 Neumann (1998, p. 1)

288 Firesmith (1993, p. 29)

289 The notion of *object* may be misleading for novices, because it covers not only *tangible* objects like cars or persons, but also *intangible* objects such as bank accounts or progress.

290 Cf. Motschnig-Pitrik and Mylopoulos (1992, p. 61-62)

291 *Class* is often used synonymously with *concept*, *frame*, *unit*, *entity*, *template*, etc.

the class possesses[292]. In the process of *instantiation*, the newly created object is assigned a concrete value for all of its attributes while complying with each attribute's type as defined by the object's class. The behavioral elements of an object – its operations – may perform inspections and alterations[293] on the object's attributes when they are invoked.

Example: John's car is a black Volvo with a 170 horsepower (HP) engine, built in 1985. Michael's car is a red 1996 Peugeot with a 90 HP engine. From the description of these two cars some common attributes of each car (at least with respect to this example's complexity and scope[294]) can be identified: *Make*, *color*, *engine power*, and *year of construction*. When describing John's car, we can think of it like instantiating an object called "John's car" with exactly the set of attributes defined by the class *car*, and assigning each attribute a value. These considerations are sketched in Figure 16.

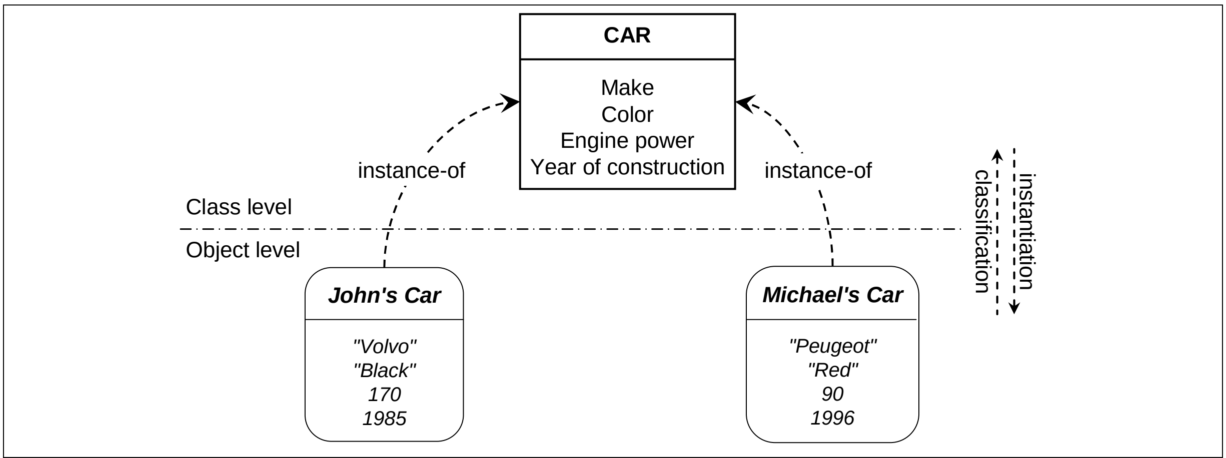


Figure 16: Example of a class and object space.

2.3.2.2 OO Concepts

To understand object-oriented artifacts such as models, it is essential to have basic knowledge in key object-oriented concepts. Even though discussions of OO concepts always bear a touch of OO programming languages and their features, the following discussion tries to avoid adhering to programming language features as far as possible. This should be feasible, as object-orientation, to use the words of Ivar Jacobson[295], "*is a universal technique that can be applied to many types of systems: artificial, man-made systems as well as natural systems.*"

292 According to Motschnig-Pitrik and Mylopoulos (1992, p. 71), the process of propagating attributes from a class to a concrete instance is called *instance-inheritance*, as the instance inherits exactly the attributes of its corresponding class.

293 Jacobson, Ericsson and Jacobson (1995, p. 68)

294 Traceably, it is common practice to consider and model only relevant object properties.

295 Jacobson, Ericsson and Jacobson (1995, p. 69)

2.3.2.2.1 Abstraction

The abstraction concept was already introduced in the general discussion above[296]. It is intended to disengage observations from concrete occurrences of things and events, but to identify similar occurrences which share a certain set of essential, inherent properties and operations[297]. This way, abstraction is a concept similar to *classification*, as classification supports abstraction.

2.3.2.2.2 Encapsulation

In the process of abstraction and classification, private attributes of objects are identified and hidden from collaborating, co-existing objects[298]. An object cannot access the encapsulated information of another object directly[299], but through its public operations. So at object level, the main intent behind encapsulation is to separate the public interface of an object (i.e., a protocol of its capabilities) from its internal representation, thus making interoperation, reuse, changes, and coexistence of objects easier to manage. A major benefit of this concept is that client objects of a certain server object cannot become dependent on the internal structure of the server object. This is why encapsulation is often mentioned synonymously with *information hiding*[300]. And as long as it complies with its published interface (which may basically be assumed), client objects will be able to use it for their purposes, regardless of internal representation.

2.3.2.2.3 Inheritance

Inheritance comes by with a host of problems and general issues[301], but the following will be sufficient in the scope of this work: Inheritance is a mechanism that defines the propagation of a concept's properties to its children. Child classes are considered specializations[302] of their parent classes, as they inherit all the properties and behaviors of their parents and may add their own or overwrite (parts of) their parents' set of properties and behaviors.

Example: Imagine a class `Rectangle`. Each rectangle has a `Width` and a `Height` attribute, and additionally offers an operation to `Draw` itself on a drawing canvas. Now, if we derive a class `RectangleWithRoundEdges` from rectangle, it inherits both attributes. As a rectangle with round edges has to draw itself differently than a normal rectangle, it redefines (i.e.,

[296] See Section 2.3.2.1, p. 68

[297] Cf. Rumbaugh et al. (1991, p. 7)

[298] Firesmith (1993, p. 7)

[299] However, many OO programming languages such as C++ provide mechanisms to declare certain objects as friends, thus allowing private object properties to be directly accessed by friend objects.

[300] See for example, Rumbaugh et al. (1991, p. 7)

[301] Cf. Nierstrasz (1989, p. 6-10)

[302] The inverse notion of specialization is *generalization*.

overwrite in OO programming terminology) the `Draw` operation of its parent. Additionally, to provide for different edge sizes, the `RectangleWithRoundEdges` may add the attribute `EdgeSize` to its inherited set of attributes.

2.3.2.2.4 Polymorphism

Speaking generally, a polymorphic object is able to take on many different forms. In OO, polymorphism is very closely related to programming language concepts: An object may be used in any circumstance where an object of its class or an object of any super-class of the object's class is expected. An operation which is defined in a specific class context may result in different behavior in any compliant context, e.g., in objects of different sub-classes. [303]

Example: A procedure which manipulates the dimensions of a rectangle (compare the example above) expects an object of class `Rectangle` as input parameter. However, it does not matter what kind of rectangle it manipulates, as long as it exposes `Width` and `Height` as attributes. As `RectangleWithRoundEdges` is derived from `Rectangle`, an object of this sub-class may also be passed to the procedure, and so, figuratively, acts like a casual rectangle. In this case it does not matter if it is a casual rectangle or a rectangle with round edges, as both of them contain at least the needed and expected set of properties and operations as defined by `Rectangle`.

2.3.3 The Unified Modeling Language (UML)

The Unified Modeling Language (UML) is a visual modeling language for specification, visualization, construction, and documentation of virtually any kind of system where objects play a major role, whereas the focus of the UML lies clearly on supporting the software engineering process by modeling of object-oriented software systems[304]. As the UML plays an important role in the pattern modeling approach presented later in this work, general aspects as well as crucial details are considered accordingly in the following Sections.

2.3.3.1 History

Before going into detail with relevant aspects of the UML, it may be worthwhile to take a brief look at the history of the emergence of this modeling language. Three persons are inextricably connected to the history of the UML:

---

[303] For more detailed discussions on polymorphism see, e.g., Booch, Rumbaugh and Jacobson (1999, p. 64), Eriksson and Penker (1998, p. 98), Graham (1995, p. 23-29), Jacobson et al. (1992, p. 99), Nierstrasz (1989, p. 10-11)

[304] Cf. OMG (2003)

Grady Booch, who developed the so called *Booch* method for object-oriented software development[305] in the early 1990's. The *Booch* method entails an iterative and incremental process for analyzing both micro- and macro-views of the target system[306], whereas the method is focused on dynamic modeling of real-time, concurrent systems, and to programming language concepts[307].

James Rumbaugh was to join Grady Booch to start work on the UML in 1994. Rumbaugh came up with another system modeling method, which became well-known as the *Object Modeling Technique (OMT)* used for describing software systems by employing use case models, object models, dynamic models, and functional models[308]. OMT is closely tied to data modeling techniques, as it was initially used to model complex design objects in *Computer Aided Design (CAD)* systems[309]. Booch and Jacobson were seeking to create a modeling method unifying the *Booch* and *OMT* approaches.

About a year later, in 1995 ***Ivar Jacobson*** joined Booch and Rumbaugh with yet another different object-oriented method, namely *Object-Oriented Software Engineering (OOSE)*[310], a method for modeling and simulating real-world processes[311], as well as the *Objectory*[312] method for modeling and applying business reengineering on software engineering processes[313]. So the UML is mainly based on the *Booch*, *OMT* and *OOSE* methods.

The Object Management Group (OMG), *the* standardization organization in object-oriented systems development, called for a modeling standard in 1996. In reply, Booch, Rumbaugh, and Jacobson submitted the UML version 1.0 in early 1997. Developers of major rival approaches soon recognized the potential impact of the UML, and joined efforts on helping to revise and improve UML 1.0, so that the follow-up – consolidated version UML 1.1 – was accepted as an object-oriented modeling standard by the OMG in November 1997. Today, the UML can be considered the *lingua franca* of object-oriented software engineering[314].

The current version relevant to this work is UML 1.5. Most of the available books and articles on the UML are based on version 1.4 or lower. However, the differences between these versions do not fall into the scope of UML usage in this work.

305 Booch (1994)

306 Cf. Eriksson and Penker (1998, p. 3)

307 Hitz and Kappel (2003, p. 5)

308 Rumbaugh et al. (1991)

309 Cf. Hitz and Kappel (2003, p. 4-5)

310 Jacobson et al. (1992)

311 Cf. Hitz and Kappel (2003, p. 5), OOSE was initially used to model telecommunication systems.

312 Jacobson, Ericsson and Jacobson (1995)

313 Jacobson, Ericsson and Jacobson (1995, p. 200-201)

314 Hitz and Kappel (2003, p. 6-7)

2.3.3.2 Basic Concepts

The UML is mostly independent of particular software life-cycle concepts[315], as the language itself does not come by with a mandatory or inherent process to follow[316]. It is a very flexible modeling language, as for almost any task there is, basically, more than one way to accomplish it. This is achieved by providing partly redundant modeling elements and views, each with a slightly different focus[317].

However, most development processes rely on a general scheme of stepwise system building, starting with analysis and design prior to implementation and realization[318]. Even though the UML as a language is process-independent, it is particularly conceived to support development processes in the requirements definition, analysis, and in the design phases[319].

Generally, most UML diagrams are relatively easy to understand without having to possess deeper knowledge about notational obligations and semantic restrictions. The shape of the notational elements is very intuitive and uncomplicated, which greatly reduces problems with comprehending models while increasing their readability. Such a rather intuitive diagram is shown in Figure 18, which outlines the use cases of an (extremely) simplified bank account system. No special UML knowledge is needed to grasp the essence. On the other hand, even if the diagrams can be kept clear and readable, it is also possible to exploit the huge number of features and "hidden" details of the language which may be necessary in very complex, detailed aspects of a model. An example of such a diagram is shown in Figure 17, which shows the semantics of the UML extension mechanisms. It is certainly hard for any viewer to grasp this diagram without thorough understanding of class diagrams and their constituent elements.

---

315 Booch, Rumbaugh and Jacobson (1999, p. 33)

316 However, the Rational Software Corporation (http://www.rational.com), Inc., the employer of the UML designers, have created a software development process particularly tailored to supplement the UML as its primary modeling language. This customizable, powerful process is called *Unified Software Development Process*, *Rational Unified Process*, or simply *RUP*. It lends significantly from prior work of Jacobson, Ericsson and Jacobson (1995) on the *Objectory* method. For details on the *RUP*, cf. Jacobson, Booch and Rumbaugh (1999), or visit the *RUP* website at http://www.rational.com/products/rup.

317 See Section 2.3.3.2, p. 73

318 Cf. Sommerville (1992, p. 5-6). Almost any software engineering process model follows this pattern, e.g., the waterfall model by Royce (1970) is starting off with 'requirements analysis and definition', followed by 'system and software design'; later, more sophisticated process models added incremental and iterative steps to this rigid generic model, e.g. the risk-centered spiral model as proposed by Boehm (1988). Either way, analysis and design are major phases in any (software) engineering process.

319 OMG (2003)

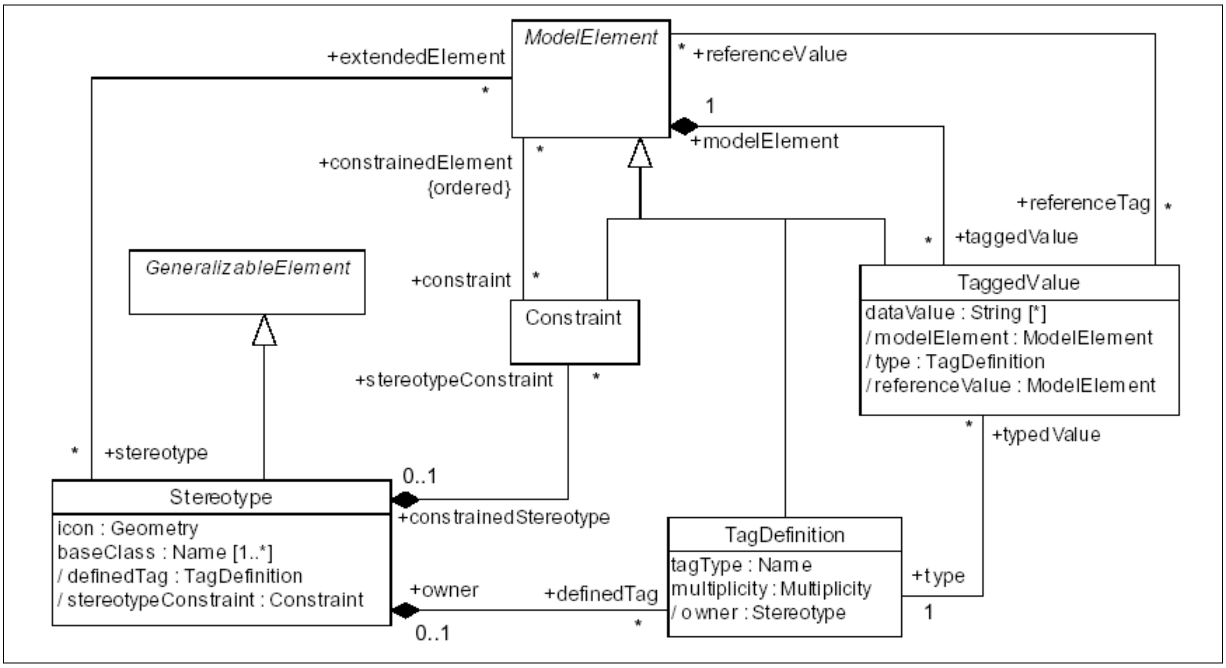


Figure 17: A semantic model of the UML extension mechanisms in the form of a UML class diagram.[320]

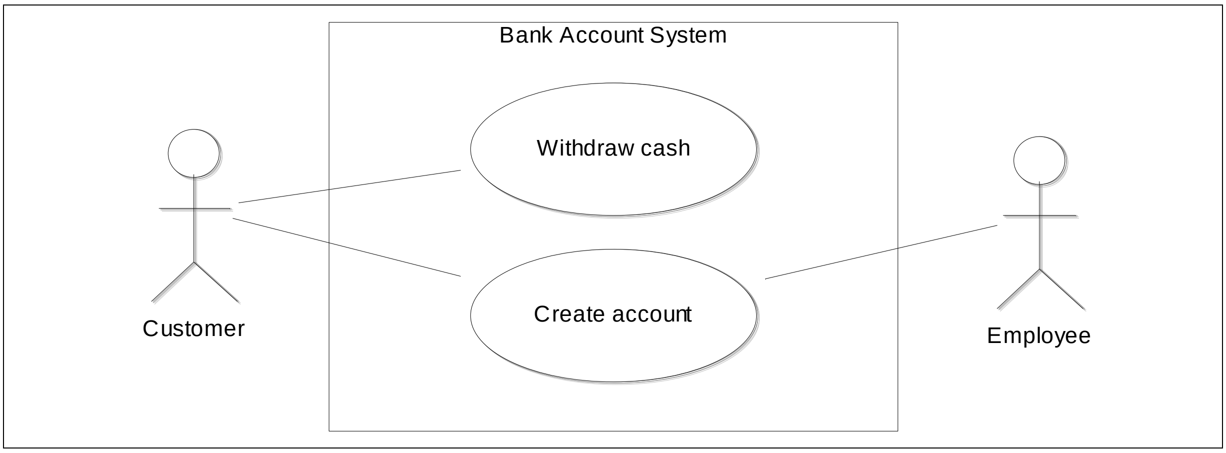


Figure 18: A use case diagram of a simplified bank account system.

2.3.3.3 Modeling Concepts

In the UML, the notions of *model* and *diagram* are differentiated[321]. Models represent complete abstractions of a certain aspect of the target system. Depending on the complexity and on the size of the system, models can become very complex which certainly does not contribute to an increased understanding of the real system. To allow for both modeling of general concepts and complex, detailed aspects, diagrams act as projections of certain aspects of

[320] Source: Figure 5–7 in OMG (2003, p. 5–9)

[321] Cf. Hitz and Kappel (2003, p. 7-8)

models. This way, a complex model can consist of a number of diagrams, each providing a *different view on the same model*.

To allow for a comprehensive set of modeling options, the UML provides eight types of diagrams which are the primary artifacts that a modeler sees and produces[322]:

- ***Class Diagram***: A class diagram represents aspects of the static structure of the model[323] and its constituent entities (*classes*), along with internals of and relationships among these entities. A special form of a class diagram is an object diagram, which models concrete instances of the classes of the system and thus creates a prototypical view on the real system. As class diagrams play a major role in the structural modeling of e-learning patterns, they are discussed in more detail in Section 2.3.3.3.1 on page 78.
- ***Use Case Diagram***: Use Case Diagrams model a system's use cases and its actors. A use case represents some coherent functionality provided by the system, a subsystem, or a class. The functionality is manifested by interaction of external actors with actions performed by the system. Use cases may be represented in a granular way, such that use cases include or extend other use cases. Each actor represents a set of roles system users play when interacting with the system. This is not constrained to human actors; for example, another system interacting with the modeled system may also be represented by an actor. For an example of a simple use case diagram see Figure 18 on page 74. Use case diagrams are particularly important in the requirements definition phase of the software life-cycle, as user requirements and the gross system functionality are negotiated at that point. These diagrams (due to their notational simplicity) may be used as a common 'language' for exchange between technical engineers and end-users to reach a shared perception of the system.
- ***Statechart Diagram***: A statechart diagram describes the behavior of a certain model element instance, e.g., the actions and states an object handles and goes through during its lifetime. The states in the statechart graph are modeled as transitions which are usually event triggered. For an example see Figure 19.

322 Cf. OMG (2003)

323 This is why the class diagram is also called *static structure diagram*

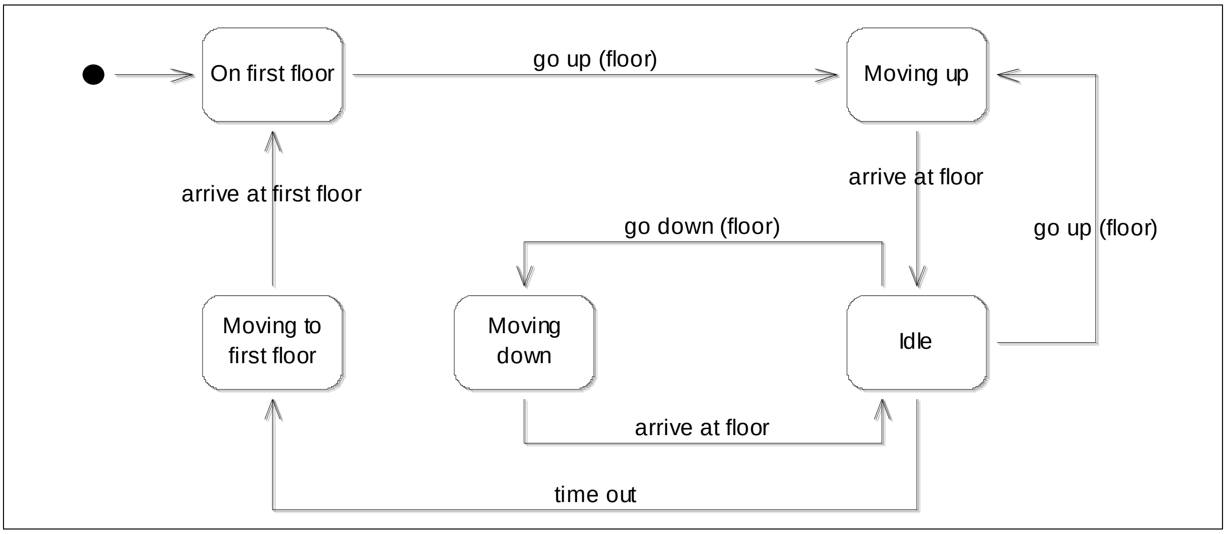


Figure 19: A state diagram for an elevator.[324]

- ***Activity Diagram***: An activity diagram models the activities (also called *action states*) of an element instance, such as of a use case. This way, activity diagrams are variations of state machines, where transitions are usually triggered by the completion of an activity. An activity diagram is a directed graph that focuses on situations where most of the transitions between activities are based on the completion of internal events, as opposed to statecharts which focus on modeling of asynchronous events. This type of diagram is discussed in more detail in Section 2.3.3.3.2 on page 82.
- ***Sequence Diagram***: Sequence diagrams model interactions between certain instances of model elements. It is arranged in two dimensions. The vertical dimension represents time and the horizontal dimension represents the different interacting systems. The lifetime of each instance is represented by a vertical bar (the *life line*). Generally, a sequence diagram models the sequence of operations sent between different instances. See Figure 20 for a sample sequence diagram.

[324] Source: Reproduced from Figure 2.5 in Eriksson and Penker (1998, p. 20)

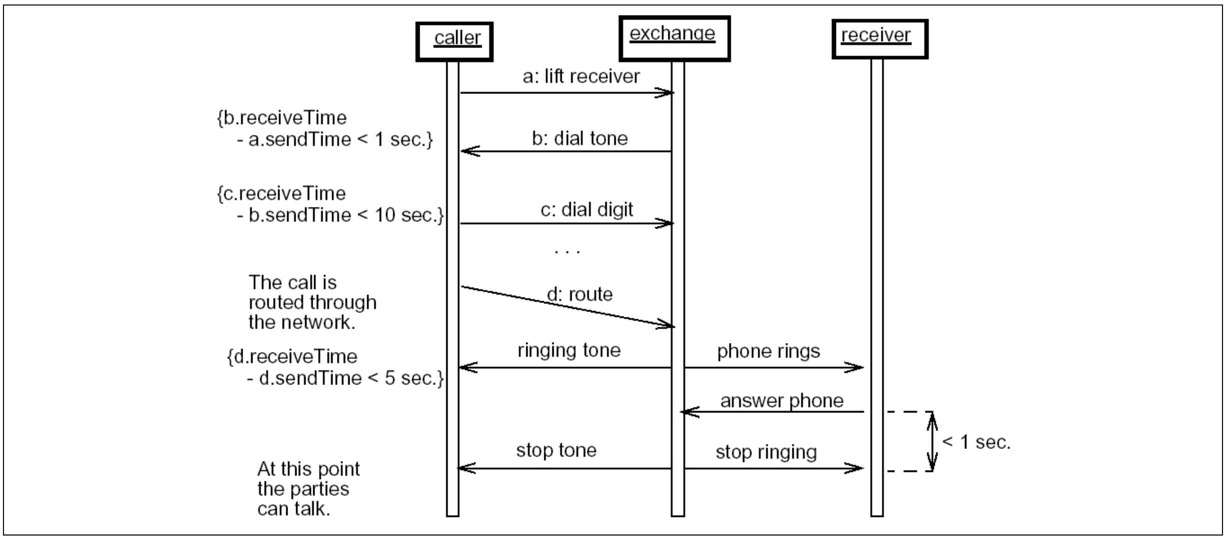


Figure 20: A sequence diagram modeling how a phone call is established.[325]

- ***Collaboration Diagram***: A collaboration diagram models a set of collaborations between certain model element instances. Collaboration diagrams are closely related to sequence diagrams, as they may represent the same flow of operations with different presentation: collaboration diagrams do not have an explicit time dimension. The modeling of sequence in the collaborations is optionally achieved by assigning consecutive numbers to the connectors (i.e., operation calls) between instances. See Figure 21 for an example of a collaboration diagram.

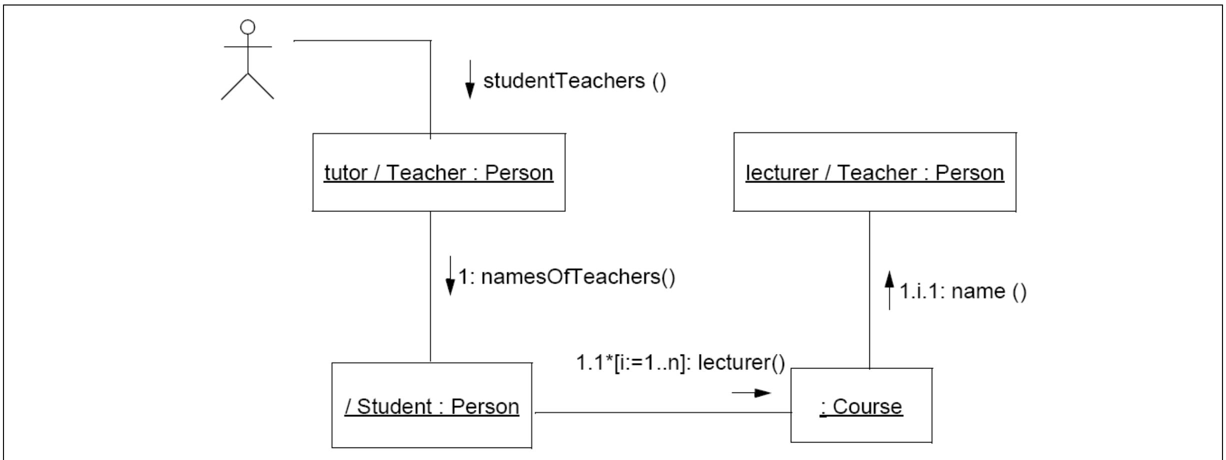


Figure 21: Example of a collaboration diagram.[326]
The examples shows a nested sequence of calls intended to retrieve all teachers of a student.

- ***Component Diagram***: A component diagram is an implementation diagram as it is closely related to the implementation of the software system by visually describing the dependencies among the components of the software system. The artifacts contained in a component diagram are, for example, source code files, binaries, executables, and others.

[325] Source: Figure 3–55 in OMG (2003, p. 3–104)

[326] Source: Figure 3–61 in OMG (2003, p. 3–117)

- ***Deployment Diagram***: Deployment diagrams are also implementation diagrams, like component diagrams. The difference is such that deployment elements are modeled as concrete run-time instances of software components, as opposed to component diagrams which model the components at the generic type level. Deployment diagrams are graph structures, where the nodes represent processing resources (e.g., a server computer), along with the run-time elements executing on them (e.g., a server process).

As the description of the different diagram types shows, there is not always one single option of modeling certain system aspects. Interactions, for example, may be represented by a state-chart or activity diagram. In this respect, the UML provides very flexible techniques to build models of a system. The following Sections cover more detailed aspects of modeling class and activity diagrams[327], focusing on model elements and their inter-connections, as this is an important prerequisite for modeling and comprehending PCeL patterns. The other diagram types – which will not play major roles later in this work – will not be discussed any further. The most important UML techniques for this work's contribution are class diagrams and activity diagrams, along with the powerful UML extension mechanisms discussed in Section 2.3.3.4 on page 85.

2.3.3.3.1 Class Diagrams

A class diagram is a static model type used to build the static structure of the system's analysis or design model[328] by primarily modeling classes and their relationships.

Class

As already pointed out in the Section on object-orientation, a class is a structural element that represents an abstract or concrete concept of the target area as it models a set of objects with shared attributes (representing the state of an object) and operations (representing its behavior). As sketched in Figure 22, in the UML a class is graphically represented by a solid-outline rectangle which is horizontally divided into (up to) three *compartments*: [329]

1) The top compartment contains the ***name*** of the class in bold letters and other general class properties such as constraints and stereotypes[330]. In case the class is abstract, the name is given in italics[331].

[327] However, due to the fact that the UML 1.5 specification covers more than 700 pages, it is clear that this work cannot deliver complete coverage of all the details and options the UML has to offer.

[328] Eriksson and Penker (1998, p. 67), Hitz and Kappel (2003, p. 18-19)

[329] Cf. Eriksson and Penker (1998, p. 69-76)

[330] For details on constraints and stereotypes see Section 2.3.3.4, p. 85

[331] This applies to any model element which can be abstract. For an example see the abstract class `ModelElement` in the class diagram in Figure 17.

2) The middle compartment contains the ***attributes*** of the class. Each attribute in that compartment is represented by a single line of text. The name of the attribute is mandatory; optionally, the modeler may provide its visibility[332] for other classes, its type[333], and/or its default value[334].

3) The bottom compartment contains the ***operations*** of the class, which the class uses to manipulate the attributes and to perform other actions[335].

The three compartments are separated by horizontal lines, whereas any of the lower two compartments (attributes, operations) may be left out when not needed.

Figure 22: A class in UML.[336]

The class `Figure` as shown in the Figure above gives a simplified representation of a graphical figure with `Size` and `Position` as attributes, and the operation `Draw` which may be used to draw a figure object. The left-hand side representation shows all three compartments (*detailed, design view*), whereas the right-hand views the same class without the attribute and operation compartments (*simplified, analytical view*).

Relationships

Different types of relationships between classes in class diagrams may be presented on the basis of a simple example which is given in Figure 23.

---

[332] The basic visibility distinction is public (+) and private (-). Public attributes are visible to any other class, while private visibility restricts visibility to the containing class. Depending on programming languages used to implement the models, additional visibility modifiers may be used, e.g., protected (#), restricting visibility to the containing class and to its child classes.

[333] Attribute types may be scalar types (e.g., integers, strings) or composite types (e.g., arrays, objects). Using types depends heavily on the level of detail of the class diagram.

[334] A default value is the value that the attribute is automatically assigned when a new object of the class is created.

[335] Cf. Section 2.3.2.1, p. 68, and Section 2.3.2.2.2, p. 70

[336] Source: Own illustration based on Figure 4.11 in Eriksson and Penker (1998, p. 73)

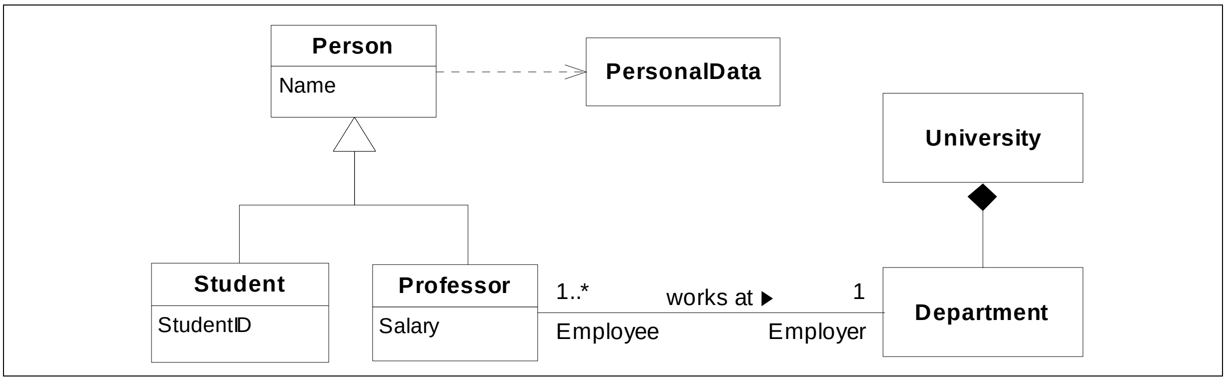


Figure 23: A UML class diagram.

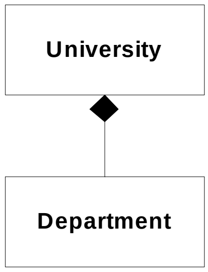
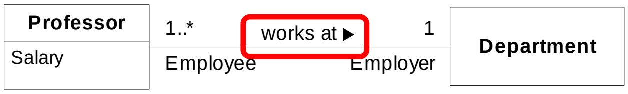


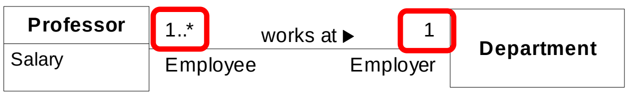
Figure 23 shows a simplified view on the static structure of a university. A university has a number of departments. This is represented by a ***composition*** relationship, which models a part-of relationship, whereas the existence of the parts (departments) is constrained by the lifetime of the composite structure (university). Conceptually, this means that as soon as a department is assigned to a university it is dependent on the existence of the university, i.e. the university is responsible for its departments. A weak form of a composition is an ***aggregation*** relationship where the parts are independent of the existence of the composite element.[337] Graphically, the diamond at the composite element of an aggregation is hollow, unlike in the composition relationship.

Departments are connected to professors through a (binary) ***association***. An association relationship describes the shared structure of a set of relationships between objects of two (binary association) or more classes[338]. Graphically, an association is represented by a solid-line path connecting the associated classes. Optionally, an association relationship may carry one or more of the following adornments:

- ***Name***: Conceptually, the name in the middle of the association connector supports readability, along with an optional filled triangle leading the reading direction[339]. In addition to the name, a stereotype for the association may be placed above or in front of the association name[340].



- ***Cardinalities***: Each association end may carry a cardinality (also multiplicity) which identifies the number or the allowed interval of numbers of objects participating in each occurrence of the as-



---

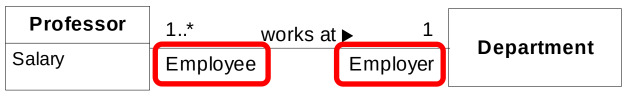
[337] Cf. Eriksson and Penker (1998, p. 88-90), Hitz and Kappel (2003, p. 35-42)

[338] Hitz and Kappel (2003, p. 27)

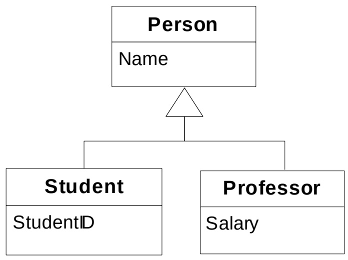
[339] From a technical point of view, the association name plays a second important role: As the relationship is part of the structure of the participating classes, it has to bear a unique name for proper identification.

[340] OMG (2003, p. 3–69)

sociation at the given end. Typical cardinality values include `1` (exactly one object), `x..y` (at least `x` objects, with a maximum of `y` objects, e.g. `1..3`). A star (`*`) indicates an arbitrary positive number of objects. This way, it may be used as an upper bound together with a concrete number as lower bound (e.g., `2..*`), or standalone, indicating an arbitrary interval of positive integers (i.e., `*`). In the example in Figure 23 any instance of the class professor is associated with exactly one department. In the other direction, any department has at least one professor.



- ***Roles***: The association ends may also carry the names of the roles the participating objects play in the association. In the concrete example, a professor is an employee of exactly one department (`1`), which in turn acts as an employer for at least one professor (`1..*`) .

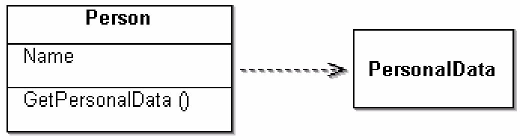


Students and professors are persons. In this sense, they inherit[341] the attributes, operations, and relationships of the class `Person`. This kind of relationship is represented by a ***generalization*** in UML. Semantically, a generalization is "*[...] the taxonomic relationship between a more general element (the parent) and a more specific element (the child) that is fully consistent with the [parent] and that adds additional information.*"[342] Graphically, a generalization is shown as a solid-line path from the child to the parent with a hollow triangle at the end of the path which connects to the more general element. In the concrete case, the class `Person` is the parent element of the child elements `Student` and `Professor`. Thus, students and professors inherit the `Name` attribute and the `GetPersonalData` operation from `Person`. Students additionally have a `StudentID` and professors additionally draw a `Salary` from their department.

---

341 Cf. Section 2.3.2.2.3, p. 70

342 OMG (2003, p. 3–86). This is not restricted to classes only. Generalization may also be applied to packages, use cases, and other classifiers.



Another important connector is the ***dependency*** relationship, which indicates a semantic relationship between two model elements of any type, meaning that a change to the source element of the dependence may require a change to the target element[343]. However, dependency relationships need not only relate to change dependencies, but may also be used to model access, binding, derivation, import, refinement, trace, and usage dependencies[344]. To be precise in diagrams, it may be necessary to add the respective stereotype to the dependency indicating the intended type. In our concrete university example, class `Person` is dependent on class `PersonalData` because it uses the `PersonalData` type in its operation `GetPersonalData`. Graphically, this is shown by a dashed arrow, whereas the model element at the tail of the arrow (the *client*) depends on the model element at the arrowhead (the *supplier*).

2.3.3.3.2 Activity Diagrams

Activity diagrams, along with state, sequence, and collaboration diagrams are used to model the behavior of a system, demonstrating how the objects modeled in the structural model interact dynamically[345]. More specifically, an activity diagram may serve one of four purposes:[346]

- To model a ***realization of an operation***: Realizing an operation is executing a sequence of directives, which is directly represented through a sequence of activities in the activity diagram.
- To ***describe a use case***: The execution of a user interaction with the system (i.e., a use case) is a sequence of steps which can be modeled 1-to-1 as a sequence of activities in an activity diagram.
- To describe a ***cooperation of use cases***: Executing a single, complex use case may imply the execution of a number of other use cases at certain steps within its flow of events. This way, the interconnections and networking among a number of use cases may be represented by nested activity diagrams. Certain complex activities can be refined in more detailed activity diagrams, thus perfectly supporting the concept of abstraction.
- To describe a ***business process***: A business process is a sequence of activities which is arranged and conducted to achieve certain business goals[347]. Through considerable se-

[343] OMG (2003, p. 3–90)

[344] Cf. OMG (2003, p. 3–91)

[345] Eriksson and Penker (1998, p. 119-120)

[346] Cf. Hitz and Kappel (2003, p. 160-161)

[347] See also Karagiannis, Junginger and Kühn (1999)

mantic equivalence with the respective model elements in business graphs, activity diagrams may well be used to describe business processes.

To explain the model elements of activity diagrams, a simple example will provide guidance:

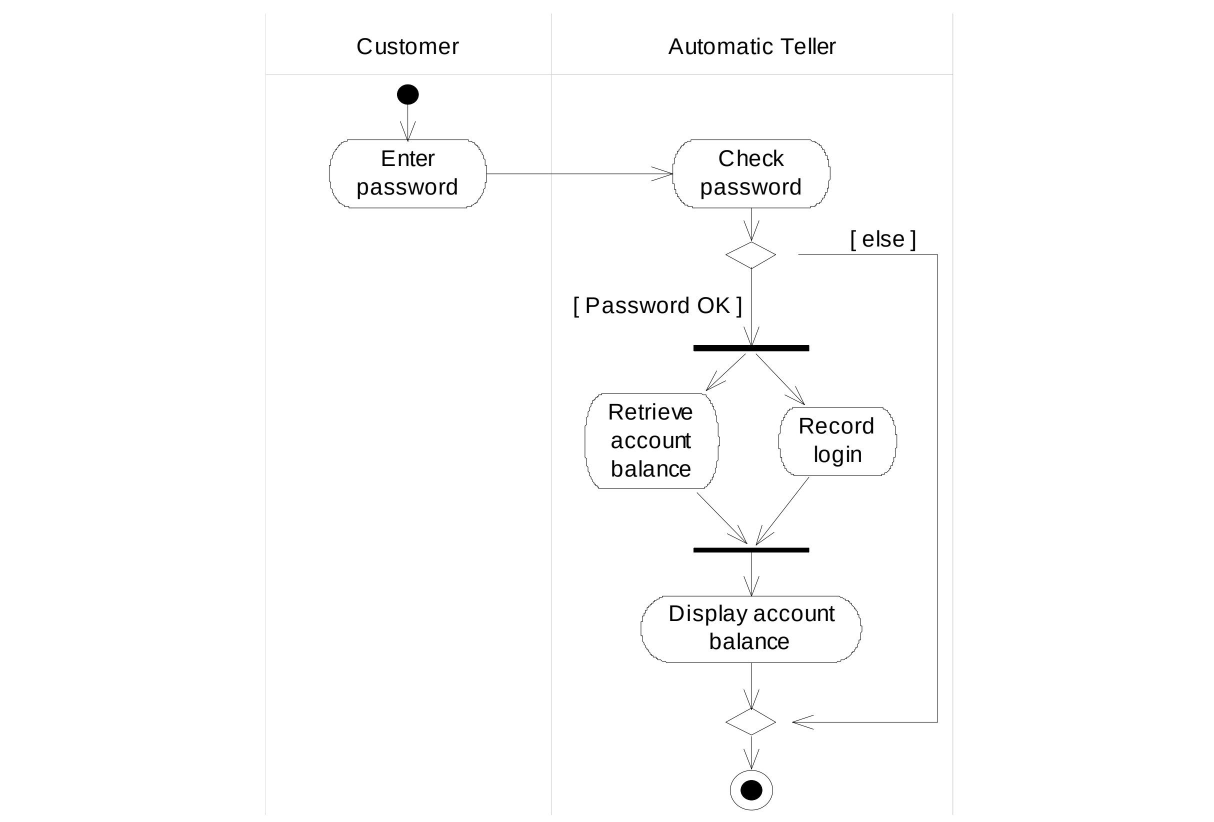


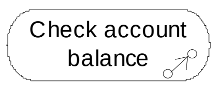
Figure 24: A simple activity diagram.
The diagram describes the process of retrieving and displaying a customer's bank account balance on an automatic teller machine.

Action state

An ***action state*** (precise denomination of an *activity*) describes the state of the system in which certain actions (or activities) are performed. For example, the action state `Enter password` in the diagram in Figure 24 describes the typing of the password on the teller machine's keyboard as well as the confirmation of the input by the user. Graphically, an action state is modeled by a hollow rectangle with rounded corners, bearing a succinct designator of the performed action(s) in the center of the shape.

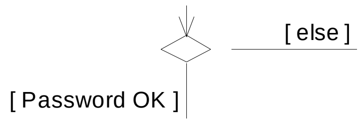
Additionally, each activity diagram has exactly one start state (solid filled circle) and an unrestricted number of end states (circle surrounding a smaller solid, filled circle) depending on the complexity of the graph.

Subactivity state



A ***subactivity state*** is a special state which invokes an activity diagram. When a subactivity state is entered, the nested activity diagram takes over control until this invoked diagram has reached an end state.[348] A subactivity state looks the same as an action state with an additional icon in the lower right corner depicting a nested activity diagram. In our concrete example, the activity diagram in Figure 24 may be such a nested diagram in a `Check account balance` subactivity state.

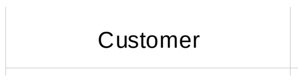
Decision



There are points in activity diagrams where different possible transitions need to be split up according to certain guard conditions. This is modeled by a ***decision*** element (graphically a large hollow diamond). Each outgoing transition of the decision should carry one possible outcome of the decision, in order to allow the control flow to continue. Usually, one of the outgoing transition carries the guard condition `[else]` which makes it fire if all the other outgoing transitions' conditions are not satisfied[349].

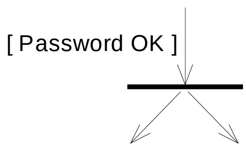
In our concrete example, after the password check a decision has to be reached whether the password entered was correct (`[ Password OK ]`) or not (`[ else ]`). If the password was entered correctly, the balance retrieval may proceed; otherwise, the control flow is redirected to the decision object just before the end of the diagram (at this point, the decision symbol acts as a merging of flows previously split by the other decision).

Swimlane



A ***swimlane*** represents a certain area of responsibility within an activity diagram. This way, activities may be grouped to show where actions are performed or by whom an action is performed, respectively.[350] Our sample diagram in Figure 24 contains two swimlanes: one for the actions and activities performed by the `Customer`, and the second lane for any action performed within the control of the `Automatic Teller`.

Transition



As soon as all actions in an action state finished, or when the nested activity graph of a subactivity state has terminated, all outgoing ***transitions*** of the respective element fire, which means that the flow of control in the system passes on to model elements connected via transition connectors. Graphically, transitions are solid arrows which may carry a trigger event, a guard condition, and/or an action name, in the form `event [ guard ] / name`. [351]

[348] Cf. OMG (2003, p. 3–159)

[349] Cf. Hitz and Kappel (2003, p. 165-166)

[350] Cf. Eriksson and Penker (1998, p. 153)

[351] Cf. OMG (2003, p. 3–145)

It may be necessary to have concurrent flows of control from a certain point. This can be achieved through a ***concurrent transition***, which may have multiple source and target states. It is modeled by a vertical or horizontal *synchronization bar*. In such a case, the synchronization bar yields control to the outgoing transition(s) only if all of its incoming transitions have fired.

2.3.3.4 Extension Mechanisms

To keep the notational and semantic aspects of the UML clear, there is only a limited set of predefined model elements and modeling constructs. However, more complex systems or specialized applications may require definition of frequently needed elements which the UML does include. For this reason, the UML offers three basic, standardized extension mechanisms: *constraints*, *tagged values*, and *stereotypes*. These mechanisms are discussed briefly in the following sub-Sections, as they are used later in this work for extending the UML metamodel in order to model Person-Centered e-learning patterns.

2.3.3.4.1 Constraints

A constraint limits the usage or semantic meaning of the model element to which it is attached[352]. It is a semantic relationship that specifies conditions which must be maintained as true[353]. Constraints are shown as text strings in braces ({), whereas the text language is not predetermined by the UML[354].

There are different options of graphically representing constraints, as depicted in Figure 25. A constraint is usually *(1)* placed near the symbol or near the name of a model element, if it applies to a single element. Other options include *(2)* placing the constraint in a standard note symbol and attaching the note to the constrained model elements or, *(3)* if exactly two model elements are involved, they may be connected by a dashed arrow bearing the constraint in braces (in this case the direction of the arrow is relevant information).

[352] Eriksson and Penker (1998, p. 30-31), Hitz and Kappel (2003, p. 76-77)

[353] OMG (2003, p. 3–26)

[354] Nevertheless, the UML specification includes a predefined language for constraints: the *Object Constraint Language (OCL)*. Its use is optional, but recommended for the sake of interchangeability.

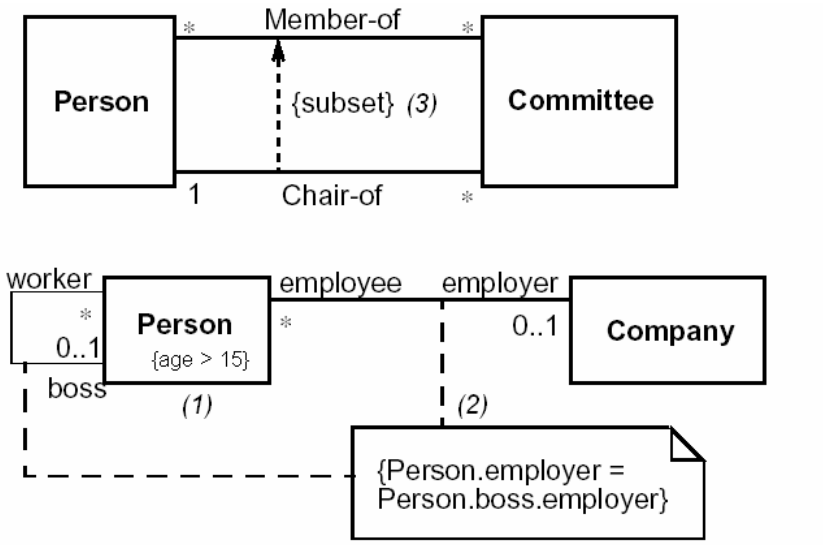


Figure 25: Presentation options for constraints in UML.[355]

Note: the different presentation options as described in the above paragraph are referred to in the figure through the respective option number in parentheses: *(1)*, *(2)*, and *(3)*

2.3.3.4.2 Tagged Values

Elements often have properties which do not have a standard representation in the UML. Any number of such properties may be assigned to any kind of model elements using tagged values, which are presented as keyword-value pairs[356] (the keyword is also called a *tag*). Generally, property specifications are of the form `name = value`, where `name` is the name of a property or a tag, and `value` is an arbitrary string denoting a concrete value. Tagged values, like constraints, are enclosed in braces, e.g. `{author = ''Michael Derntl''}`, or `{isAbstract = true}` which may be presented in simplified form[357] as `{abstract}`. The graphical presentation of tagged values is similar to the presentation of constraints.

2.3.3.4.3 Stereotypes

The stereotyping mechanism allows the modeler to extend the UML metamodel directly at modeling time[358] by introducing new model elements based on existing ones. Thereby the existing element is placed in the diagram and the stereotype is placed near (usually left or above) the name of the element. Basically, a stereotype is presented within matched guillemets (« »), e.g. `«stereotype»`. Additionally, it is possible to define a graphical representation for each stereotype which is introduced. This graphical representation (an *icon*) may be

[355] Source: Modified Figure 3–17 from OMG (2003, p. 3–28)

[356] Cf. OMG (2003, p. 3–29)

[357] This simplification is possible for boolean (true/false) tags. The default assumption for the value then is true.

[358] Hitz and Kappel (2003, p. 72), OMG (2003, p. 3–31)

used instead of, or together with the textual stereotype. Table 8 depicts the different options for the predefined «control» stereotype.

Table 8: Graphical presentation options for stereotypes in the UML[359]

Presentation option	Description
	Textual stereotype presentation above element name within the standard class symbol.
	Textual and iconic stereotype presentation within the standard class symbol.
	Iconic stereotype presentation within the standard class symbol.
	Icon-only stereotype presentation replacing the standard class symbol.

The above discussion shows that stereotypes are the most powerful extension mechanism the UML offers, as they allow the modeler to define his or her set of model elements, thus being able to tailor the UML to different methods, organizations, or users.[360]

---

359 Source: Own compilation based on Figure 3–19 in OMG (2003, p. 3–33)

360 Eriksson and Penker (1998, p. 28)

3 The Pattern Approach to Person-Centered e-Learning

> *I need to recognize that my conscious thought is full of fixed constructs which may interfere with the perception of an underlying pattern. [...] If I can lay aside rigidly held preconceptions, and forget for a moment the "truth," or the clear-cut constructs already known, then the pattern may shine through more clearly.*[361]
>
> *On their own, without the guidance of images, actions would not take us far. Good actions need the company of good images. Images allow us to choose among repertoires of previously available patterns of action and optimize the delivery of the chosen action – we can, more or less deliberately, more or less automatically, review mentally the images which represent different options of action. We can pick and choose the most appropriate and reject the bad ones. Images also allow us to invent new actions to be applied to novel situations and to construct plans for future actions – the ability to transform and combine images of actions and scenarios is the wellspring of creativity.*[362]

This Chapter presents the Person-Centered e-Learning (PCeL) pattern approach:

- First, the ***methodology*** underlying pattern mining, description, and evolution is presented (Section 3.1, p. 90).
- The largest part (Section 3.2, p. 100, and Section 3.3, p. 108) concentrates on describing the ***pattern organization, description, and modeling*** in the pattern repository, which are essential prerequisites for reading and understanding the patterns.
- A concise ***dictionary*** of frequently used terms in the patterns is presented, based on the structural model of the COURSE pattern (Section 3.4, p. 127).
- Finally, the ***underlying courses*** from which the patterns in the repository were mined are presented (Section 3.5, p. 129). Each course is described verbally and by using the modeling approach introduced in this Chapter.

Note that the state of the pattern repository as presented in this thesis is explicitly not intended to capture the "ultimate wisdom" of PCeL practice. Its degree of elaboration and completeness are chosen such that substantial elements in the sense of facilitative and supportive activities of the PCeL approach are conveyed in terms of pattern description and models. The repository in its current form aims to provide a starting point for discussion, exchange, research, and dissemination of PCeL practices.

---

[361] Carl Rogers, in Kirschenbaum and Henderson (1989, p. 270)

[362] Damasio (1999, p. 23-24)

3.1 Methodology

It seems clear from the previous discussions on learning theories that employing learning technology without any *didactical or pedagogical baseline* is a futile effort[363]. However, only few models or frameworks of making *situated use* of learning technology in learning scenarios are available and most of the experience reports and case studies in e-learning lack *systematic evaluation*[364]: research and reports on effective use of learning technology are often traded for using or implementing improved technical features and platforms. The approach presented in this thesis is based on the belief that one of the most critical factors of successfully blending online with face-to-face learning is making *situated* and *targeted*, thus *deliberate* use of learning technology. For example, using means of Computer-Mediated Communication[365] such as online discussion forums for preparation of meetings or workshops is certainly situated and targeted[366]. However, acquiring comprehensive knowledge, experience, and a sense of which activities are suitable for what kind of online interaction, or which activities are preferably conducted face-to-face, is impossible to achieve within one or two application cycles. Therefore, the researcher/practitioner needs to build awareness towards two basic dimensions of blended learning:

- First, the ***structural*** (vertical) dimension addresses the gap between didactical considerations and the employment of technology for teaching and learning purposes. What lies *between* these two? How can we project and support learning activities on a learning platform in a *situated* way? How can technology be employed to *enrich* learning processes? The current state of blended learning research resembles rather a phase of experimentation[367]: reports are mostly descriptive, experience-based, deductive in reasoning, and often lacking clues on how to generalize employed scenarios and findings to enable transfer to other domains. The fundamental question seems to be forgotten: What is the added value for learning and what is the means to achieve that added value? The point is that an integrated, blended approach needs to create a significant departure[368] from its face-to-face and online constituents.
- Second, the ***dynamic*** (horizontal) dimension addresses (a) the change that technology brings about for learning processes over time as well as (b) the research methodology that is employed to mine, describe, evaluate, and refine PCeL practice and patterns. In-

363 This is underlined by Goodyear et al. (2004, p. 451), who identifies a "*naïve and unsustainable belief that guidance about networked learning should be pedagogically (and even morally) neutral.*"

364 Cf. Reeves (1997)

365 See the COMPUTER-MEDIATED COMMUNICATION pattern (p. 313)

366 Cf. Dietz-Uhler and Bishop-Clark (2001)

367 See also Nichols (2003)

368 Cf. Garrison and Kanuka (2004, p. 97), who state that, "*the real test of blended learning is the effective integration of the two main components (face-to-face and Internet technology) such that we are not just adding on to the existing dominant approach or method.*"

troducing e-learning and especially PCeL is not a one-time effort. Rather, it follows an iterative, incremental process where technology should act as the *enabler*, with technology-enhanced learning theories and a sound methodological approach acting as the primary *drivers* of change[369].

The Blended Learning Systems Structure (BLESS) model of blended learning as presented in the following Section addresses precisely these issues, as it aims to provide a structural and dynamic framework for both blended learning practice and research. The dynamical dimension is addressed in particular in Section 3.1.2, presenting a theory-guided Action Research approach for PCeL patterns.

3.1.1 Blended Learning Systems Structure (BLESS) Model

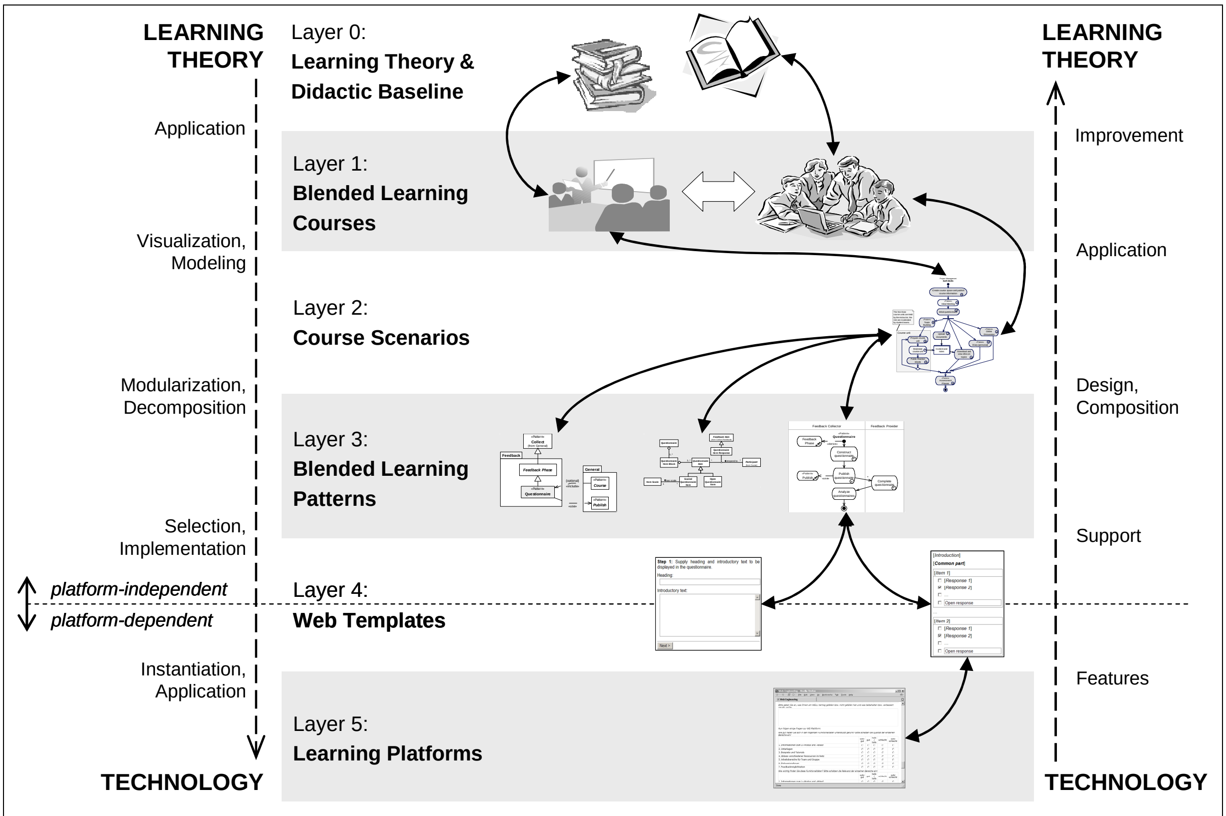


Figure 26: The layered Blended Learning Systems Structure (BLESS) model.

[369] See also Hamid (2002, p. 312-313), who argues in a similar direction: "*Unfortunately, […] the emphasis on e-learning in the past has been on the […] technology. […] There is a need to shift the emphasis […] to the learning.*"

In the Person-Centered e-Learning patterns initiative the BLESS[370] model as depicted in Figure 26 was developed and used as a framework for mining, applying, evaluating, and improving blended person-centered learning scenarios. Its layered structure evolved from the necessity to decompose the complexity inherent in the transition from actual course practice and design to the implementation of Web-based services for learning support. Note that the layers are designed to stay platform-independent as far as possible. Further note that each layer intertwines didactic and technical issues such that both aspects are co-developed. The layers and transitions of the BLESS model are described in the following:

Layer 0: Learning Theory and Didactic Baseline

The topmost layer is the driving one in that it provides the philosophy or value-orientation of the whole enterprise. It sets the overall educational targets and resulting interpersonal attitudes and, as a consequence, delivers requirements and constraints on the technological solution. Undoubtedly, the majority of the Web-based courses today are designed using constructivist educational principles[371]. The blended learning strategy employed in the PCeL project is consistent with these principles, yet it focuses on the provision of a facilitative learning atmosphere based on person-centered interpersonal attitudes[372].

Layer 1: Blended Learning Courses

The next layer represents *concrete* blended learning courses. It realizes and applies the didactic orientation flowing in from the topmost level and integrates technology-enhanced elements from layer 4 into the basic educational philosophy. To be fully effective, both didactic and technological elements need to match smoothly. Learning technology features have to be selected and arranged such as to enhance learning processes by supporting the underlying didactic baseline.

Layer 2: Course Scenarios

This layer constitutes the first level of abstraction from reality. This layer aims at semi-formal, conceptual modeling and visualization of concrete scenarios by modeling their sequence as activity diagrams in the standard Unified Modeling Language (UML)[373] notation and by documenting the activities in accompanying textual descriptions. This is the first step of pattern mining: course activities are combined and generalized into self-contained learning activity patterns. For example, in a PBL course the participants engage in an iterative problem-solving process, whereby in terms of patterns, PROJECT-BASED LEARNING can be ar-

[370] The current Section relies mainly on work published in Derntl and Motschnig-Pitrik (2004a), (2004b).

[371] Cf. Bangert (2004)

[372] Rogers (1983)

[373] OMG (2003)

ranged as a sequence of PROJECT MILESTONES. Such a modularization and abstraction process (link to layer 3) entails substantial advantages for the analyst and course designer, as it enables reuse of these patterns for both course scenario description (link from layer 1) and application (link to layer 1).

Layer 3: Blended Learning Patterns

Just as architect Christopher Alexander[374] has employed patterns to deal with the construction of towns and buildings using architectural design and arrangement techniques, the pattern approach to PCeL employs patterns for capturing and guiding course- and learning activity design. Course activities – basically fragments of scenarios – considered effective in following the learning targets are decomposed and generalized into self-contained learning activity patterns. Examples of patterns include publishing of electronic content, online knowledge gathering and construction in teams or groups, interactive elements like online or face-to-face brainstorming, discussion, several forms of feedback, evaluation and assessment, as well as other blended learning techniques. The modularization transition from layer 2 to layer 3 enables more tightly focused and selective implementation (link to layer 4) and evaluation of patterns. Vice versa, by compiling and combining single patterns, a new course or learning activity scenario can be formed (link to layer 2) and subsequently applied and evaluated in concrete courses (layer 1).

Layer 4: Web Templates

The Web templates[375] at layer 4 are derived from the patterns and show parameterized, interactive Web pages that describe how learning platform utilities (*atoms*) can be arranged and combined such as to build *molecules* in a way that optimally maps the underlying process pattern onto the learning platform (link to layer 5).

Layer 5: Learning Platform

To support a pattern's learning scenario on a learning platform, the respective Web templates as well as those of dependent and included patterns have to be implemented on that learning platform. For prototyping purposes the Web templates were implemented on top of the *CEWebS architecture*[376], which provides a Web-service-based architecture for cooperative environments. For example, in the ONLINE DISCUSSION[377] pattern, the Web template instantiation process is relatively simple: The instructor/administrator only has to specify the location of the discussion forum by selecting a learning activity or Web page to which the

---

[374] Alexander et al. (1977)

[375] For details on Web templates see Section 3.3.8, p. 119

[376] *CEWebS* is the short form of *Cooperative Environment Web Services*; see Mangler and Derntl (2004)

[377] See p. 347

forum shall be anchored, as well as some optional (and platform-dependent) parameters regarding its usage: e.g., are users allowed to initiate their own discussion threads?

Concluding, the BLESS model provides an integrated structural and dynamical framework for dealing with the complexity inherent in blended learning scenarios. In this respect, conceptual modeling is a powerful tool to describe and enable pattern-based reuse of successful blended learning scenarios. The BLESS model is the scaffold on which the pattern approach to PCeL is built upon. While its structural components have been introduced, the following Section aims at outlining in more detail the methodological base of the dynamical dimension (i.e., the transitions between the BLESS layers).

3.1.2 Action Research – The Driving Force

Teaching and learning environments are complex, ill-defined social systems[378]. As such they are hard to investigate, and it is difficult to determine effects of changes systematically in a conventional, experimental approach by extracting variables of interest while keeping others constant[379]. Also in educational environments it often seems questionable whether it is possible or justified to build control groups to test against. The highly dynamic, rich, practice-oriented nature of learning environments calls for an approach that aims at building a more holistic, participative view on the dynamics effective inside the environment.

Due to its distinguishing characteristics, ***Action Research***[380] (AR) appears appropriate for investigating and improving such an environment:

- It is based on active *participation* of the researcher[381]: the researcher is acting in the same social environment as the research targets, and both influence the dynamics inside the environment. This characteristic could equally be attributed to educational environments, with the instructor being at the same time the researcher who is interacting with students.
- In AR, the actively participating researcher wishes to *immediately apply* any new knowledge obtained based on an explicit, clear *conceptual framework*[382]. In our approach the BLESS model provides the explicit, clear conceptual framework of proceeding.

---

[378] Cf. Naidu, Cunnington and Jasen (2002, p. 24-25)

[379] See also Baylor and Ritchie (2002, p. 396), who point out that, "*the intertwining of complex variables in such a rich environment [...] precludes the pure isolation necessary to determine cause and effect.*"

[380] For anyone who has deeper interest in Action Research history and current research directions, an excellent article by Baskerville (1999) is available online.

[381] Ottosson (2003)

[382] Cf. Baskerville (1999, p. 11)

- It follows a *cyclic* approach linking *theory and practice*[383]. The main phases in each cycle are diagnosing, action planning, action taking, evaluating, and specifying learning[384]. These typically are followed in academic settings and courses, which need to be initiated, prepared, conducted, evaluated, and reflected upon each semester or year.
- AR is fundamentally about incrementally *improving practice*, whereas the improvement is based on learnings from previous iterations[385]. However, as theories are tested and evaluated through practice and actions in each cycle, AR also aims at contributing to building *living theory*[386].

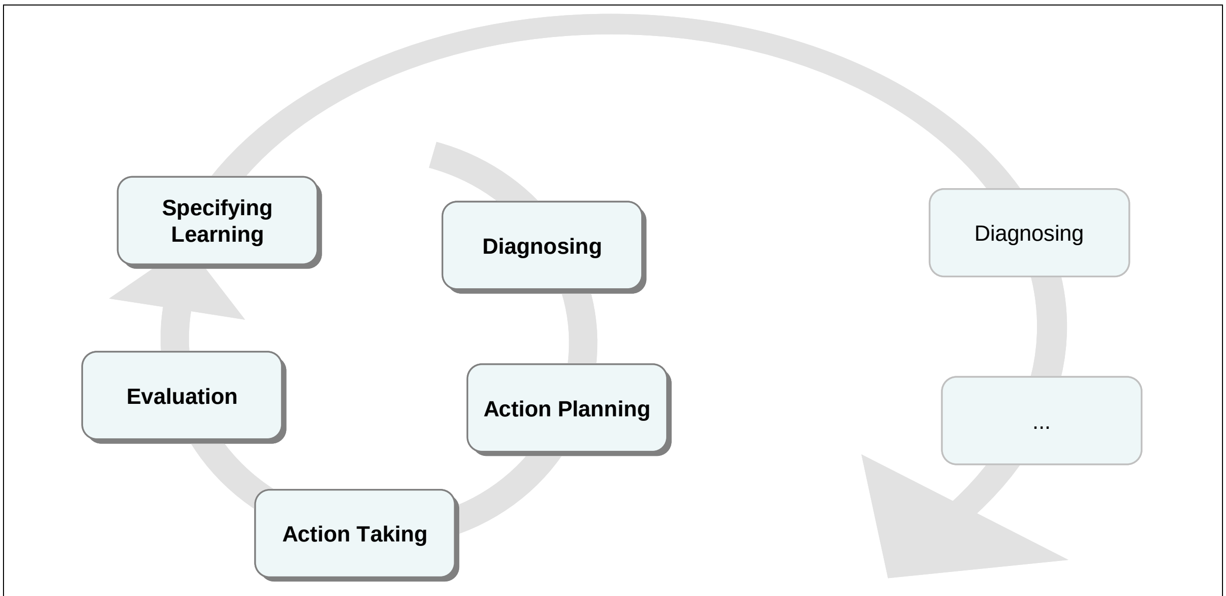


Figure 27: Main phases in each Action Research cycle.

In the literature, the main phases of an AR cycle as depicted in Figure 27 are described as follows[387]:

- ***Diagnosing***: Identification of the primary problems, questions, and desired changes in a holistic fashion. The desired product of this phase is a "working hypothesis" about the problem domain to be researched.
- ***Action Planning***: In this phase, any action that is suited to improve the situation of primary problem areas is planned.
- ***Action Taking***: Planned actions are implemented, whereas the researchers are actively involved in making the desired changes.

---

383 Baskerville and Wood-Harper (1996)

384 Susman and Evered (1978)

385 Naidu, Cunnington and Jasen (2002)

386 Cf. Levy (2002, p. 101)

387 Cf. Baskerville (1999), Susman and Evered (1978)

- ***Evaluation***: After action taking, outcomes are evaluated and changes are analyzed. Thereby it is investigated if identified changes are caused mainly by the actions that were taken, which requires careful investigation of cause and effect.
- ***Specifying Learning***: This final phase aims at specifying and formulating new norms according to the results obtained, new questions that have arisen during the current AR cycle, and dissemination of generalizable knowledge/results to practitioners and to the scientific community.

However, as a research methodology AR has also some serious drawbacks that need to be faced. It is for example criticized for frequently entailing the following threats[388]:

- The ***uncontrollability*** threat: The researcher usually does not have full control over the target environment, especially in complex application contexts. This threat is frequently reported for organizational environments where the researcher is usually not entitled to “overrule” the executives in charge. In higher education environments, however, this threat need not turn out to be that eminent as the instructor usually has full control within the context set by curricula and organization.
- The ***contingency*** threat: AR typically produces a broad body of data as the research problem and questions are not as rigorously and narrowly defined as in conventional experimental research settings. In such broad and “shallow” data, it may be hard to find evidence for particular effects, especially as AR does usually not include control groups. For educational environments, collecting verbal feedback from participants may be an initial antidote to such analysis problems, e.g., reaction sheets collected from participants often point to crucial issues and problems that can subsequently be used as anchors to analyze quantitative data. Additionally, as AR is mostly conducted in a number of consecutive cycles, the contingency problem almost “automatically” vanishes over time.
- The ***subjectivity*** threat is derived from the fact that the researcher is actively involved in, and influences the target environment by her actions. First, the researcher may draw biased and even wrong conclusions by interpreting collected data in subjective ways. Second, persons generally have difficulty in assigning negative effects that were found in the data to themselves; in such a situation, they rather mine for external factors[389].

It is certainly true that AR does not describe a method that can be readily applied. Rather, it provides a *framework* where specific objectives, actions, and research hypotheses have to be formulated by the researcher for the target problem or environment. In the AR approach to PCeL patterns, we try to overcome the shortcomings reported above while exploiting the potentials the AR framework offers. The specific approach, whose phases’ descriptions are given below, is highly guided by and intertwined with the structural dimension of the BLESS model.

---

[388] Kock (2003, p. 3-5)

[389] Kock (2003, p. 5)

Diagnosing

There is a theory behind PCeL (i.e., the PCA) whose application to blended learning and effective transfer into practice we pursue to practice and research. Hence, the basic hypotheses of Person-Centered e-Learning remain mostly unchanged among consecutive iterations. As for the pattern approach to PCeL, the main goals in each AR cycle are clearly stated:

- Improvement and extension of the patterns in the pattern repository according to results obtained in each iteration
- Optimizing generalization and enabling transfer of PCeL practices and scenarios to different contexts and domains.
- Targeted evaluation of Person-Centered e-Learning activities and incorporating results and experience reports into the respective patterns.
- Situated support of learning activities on the learning platform by incrementally developing and improving Web-service-based modules that implement the patterns.

In this initial phase of each cycle the learnings from previous cycles are reconsidered and reflected upon to provide relevant input for the following action planning phase.

Action Planning

In this phase, the course scenario to be employed is planned. Thereby, the pattern repository provides a number of predefined scenarios that have already been applied and evaluated in previous iterations. The patterns are used to guide the learning activity design of the course, as described for the transition from layer 3 to layer 2 in the BLESS model[390]. Single patterns are combined and arranged chronologically and logically to build a comprehensive model of learning activities to be conducted in the action taking phase. Additionally, the employed patterns guide the administrator or instructor in initializing the learning platform according to the course's requirements.

To allow for empirical analysis of the learning activities as well as of hypotheses and targets stated in the diagnosing phase, an online questionnaire is constructed. The questionnaire has to be adapted to the current course, so the previous version is revised according to the learning scenario patterns employed in the course. This proceeding allows for comprehensive course evaluation as well as for targeted evaluation of single scenarios that are described by the patterns. Currently, the following core aspects are subject to analysis in the overall course questionnaires[391]:

- *General attitudes* of the participant regarding course participation.

[390] See Section 3.1.1, p. 91

[391] The current core of the questionnaire that is used in each iteration can viewed in the *Evaluation* section of the COURSE pattern (p. 260)

- *Motivational orientations* to participate in the course, such as influence of course style, personal interest, competition, competence orientation, success orientation, and obligation towards the instructor.
- Perception of *learning aspects* in the course such as collaboration, practical work, provided resources, and discussions.
- Perception of *skills* the course has imparted and transported, such as problem-solving skills, factual knowledge, communication skills, etc.
- Support provided by the *learning platform* with respect to several aspects such as provision of resources, exchange with colleagues, and other learning activities that are mainly conducted online (e.g., online PEER-EVALUATION[392]).
- *Person-Centered attitudes* of instructors as perceived by participants. This is one of the core parts of the questionnaire, as in PCeL it is particularly interesting which other factors and activities in the course are influenced by these attitudes.[393]

Usually, a meeting with all involved teaching staff (instructors, tutors, assistants, and administrators) concludes the action planning phase to ensure that everyone is equally well prepared for action taking.

Action Taking

Before the actual learning activities in the courses start, each participant has to complete the initial course questionnaire to allow for collecting quantitative data. The same questionnaire is then provided again at the end of the course, mostly with additional questions that emerged during the course, or with other items that can only be surveyed a posteriori (e.g., usability issues regarding the learning platform.) Additionally, to collect qualitative responses, online reaction sheets[394] are solicited as feedback in written form.

In the action taking phase the patterns' sequences guide the conduct of the course and its learning activities. Thereby, the patterns do not purport a rigorous sequence or script to follow (e.g., as is the case with CSCL scripts[395]), but rather a *generic flow* of activities that has to be put into practice by the instructor with careful consideration of the current course context. It may even be beneficial to try variations of the pattern sequence in the application phase, as resulting observations and evaluations may be incorporated into the respective patterns as advice if successful, or as warning if the results are undesired.

---

[392] See the pattern on p. 206

[393] The positive influence of the Person-Centered attitudes has long ago been shown for person-centered classroom environments, e.g., by Aspy (1972), Rogers (1983). More recent studies in the PCeL context involving blended learning activities report similar results (see Derntl and Motschnig-Pitrik (2004c), Motschnig-Pitrik (2004b), Motschnig-Pitrik, Derntl and Mangler (2003)).

[394] See the REACTION SHEETS pattern on p. 238

[395] Dillenbourg (2002)

Evaluation

The main activity in this phase is evaluation/analysis of questionnaire data and feedback results. Through the employed "dual questionnaire" mode (i.e., pre- and post-survey), change in opinions and dispositions of participants as well as factors influencing the learning experience for students can be researched. However, it is worthwhile to mention that it may be better to pose comparative items only in the end, as it is acknowledged that students' motivation and attitudes tend to decrease at the end of the term[396]. The collected reaction sheets additionally give initial hints to identify critical positive and negative issues that can subsequently be empirically investigated in the questionnaire data. Also, reaction sheets provide valuable feedback regarding any aspects of the course that were not itemized in the questionnaires.

The evaluation phase is often open-ended, as it is impossible to comprehensively investigate and analyze every interesting facet that is "hidden" in the data body. For example, if new conclusions can be drawn from current data, it might be worthwhile to analyze whether the same conclusions can be drawn from data of previous iterations.

Specifying Learning

To make results of previous phases flow into the subsequent cycle, lessons learned and central findings are identified and captured:

- If a new scenario has been applied, its activities are modeled conceptually for visualization. Subsequently, it is investigated whether the new scenario can be modeled in terms of existing patterns. With the current state of the pattern repository this is almost always possible, at least to a certain degree of appropriateness. However, it may turn out that the scenario at hand cannot be modeled correctly by using existing patterns without obscuring or blurring the original intent behind the scenario. In such a case, it is either feasible to derive a new pattern from an existing one, or to adapt existing patterns to new findings and/or variations. In the majority of cases, a combination of these two options is appropriate[397].

---

[396] Cf. Rogers and Freiberg (1994)

[397] *Example*: The original generalization hierarchy of the EVALUATION pattern (see p. 189) just included EVALUATION as the generic parent pattern, with SELF-, PEER-, and INSTRUCTOR-EVALUATION as concrete child patterns. This was sufficient for application in the original environment, in which these patterns were mined. Subsequently we came across a course that was using online self-tests during one course. Such tests are clearly a kind of EVALUATION. But the question remains, what kind? In favor of SELF-EVALUATION stands the fact that the student herself is the active person in this evaluation process. However, it is not the student herself who evaluates the performance, which is definitely required for SELF-EVALUATION. This in turn would favor INSTRUCTOR-EVALUATION, as the instructor provides the questions and correct responses. This is still not a completely satisfactory solution, as the instructor is not the primary actor in the self-testing process. So the EVALUATION patterns had to be reconsidered accordingly: An abstract pattern EXAMINATION was derived from EVALUATION, representing a kind of structured evaluation employing a question-and-answer process. Two concrete child patterns were defined for

- The main results of the evaluation phase are attached to the *Evaluation* sections[398] of the respective patterns.
- Possible as well as concrete uses of learning technology particularly for employed or newly mined patterns are identified and described in the *Web Template* sections[399] of the respective patterns.
- A short report of the application case and optionally some screenshots and reactions are included in the *Examples* sections[400] of the employed patterns.
- Other relevant aspects and sections of the patterns that were involved in the conduct of the course are reconsidered and revised according to instructors' experience reports, participants' feedback, and new findings from pertinent literature (e.g., Motivation, Sequence, Structure, etc.[401]).

3.2 Pattern Organization

> *The lesson is part of the larger design involved in the presentation of a topic (a course segment), and this topic in turn makes up part of a still more comprehensive design of the course or curriculum.*[402]

3.2.1 Structural Organization

The patterns in this repository are arranged at different levels of detail and abstraction. Unlike many other pattern approaches that specify pattern inter-relations only textually, this work provides a conceptual model of the pattern repository and the relations among the patterns using static structure diagrams of the standard UML notation. Generally, families of related patterns are organized in packages, which contain the pattern definitions, e.g., the ALTERNATING PHASES pattern is located in the *General* package. Inside the packages, the patterns themselves are modeled using classes which are stereotyped with the custom key-

EXAMINATION: SELF-EXAMINATION, where the student is the initiator of the examination (e.g., in the aforementioned online self-test), and INSTRUCTOR-EXAMINATION, where the instructor is the initiator of the examination process (e.g., in oral examinations). This way, a new facet was incorportated in the EVALUATION patterns: In addition to the *evaluator* and the *evaluation target* we have identified the *initiator* and the *degree of structuring* as further crucial attributes of evaluation scenarios.

[398] Cf. Section 3.3.10, p. 126

[399] Cf. Section 3.3.8, p. 119

[400] Cf. Section 3.3.9, p. 125

[401] See a list of pattern sections and relevant aspects in Section 3.3, p. 108

[402] Gagnè and Briggs (1979, p. 3)

word «Pattern». Many of the patterns are defined as abstract in the object-oriented sense, which in the UML is represented by the name of the abstract element being italicized. An abstract pattern just describes basic activity flows which are not intended to be put into practice right away, but have to be specialized by more concrete patterns. An example of an abstract pattern is ALTERNATING PHASES, which models the core idea of blended learning, which is alternating online with face-to-face phases. However, as there are no additional assumptions made about the nature of learning activities in these phases, the pattern has to be declared abstract.

Note that each pattern is defined and located in exactly one pattern package. Further note that package models are confined to show generalization relationships. In particular, dependencies among patterns are not shown. In some cases it is quite difficult to define the host package for a pattern. The final decision is always made by comparing the intended purpose of a package's patterns with the primary intent of the pattern.

In the following, seven different packages are defined and described in alphabetical order.

Assessment

The *Assessment* package defines patterns that show different ways of assessing participants' achievement with the ultimate goal of determining a grade for each participant. The patterns in this package describe composed scenarios, using other patterns that define concrete ways of evaluation. Currently, this pattern hosts only one single pattern, which is ASSESSMENT PHASES.

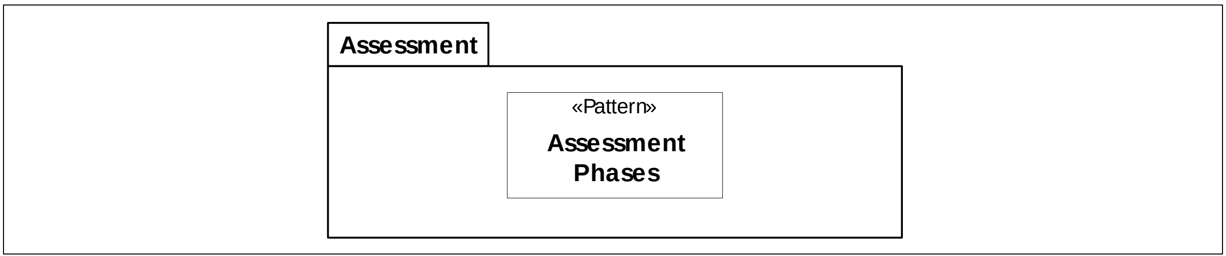


Figure 28: The *Assessment* package.

Course Types

The *Course Types* package supports the design of familiar course types (e.g., lab courses) in terms of the Person-Centered e-Learning pattern repository. Thus, some of the sequences described in these patterns may seem a bit unusual at the first glance.

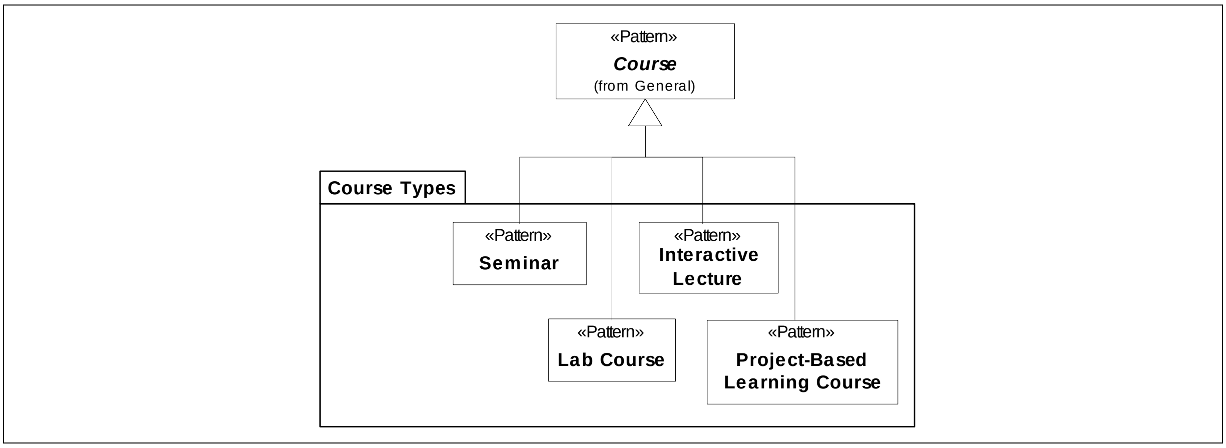


Figure 29: The *Course Types* package.

Evaluation

The *Evaluation* package describes different methods for evaluating participants. Thereby, evaluation means valuing judgment on the performance of participants. Also, some combined scenarios for mixing different evaluation methods are presented. There is a significant difference to the *Assessment* package: The primary aim of evaluation patterns is not to assign a grade but rather to lay the foundation (e.g., producing evaluation reports) for grading.

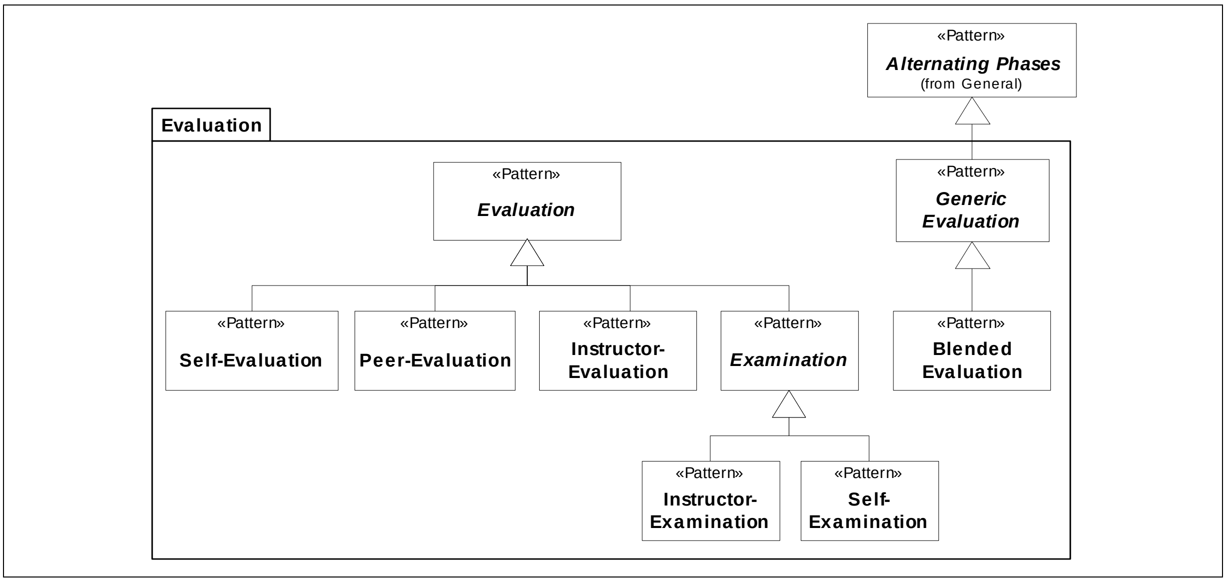


Figure 30: The *Evaluation* package.

Feedback

The *Feedback* package contains patterns that describe different ways of collecting feedback on any aspect of the course or learning activities.

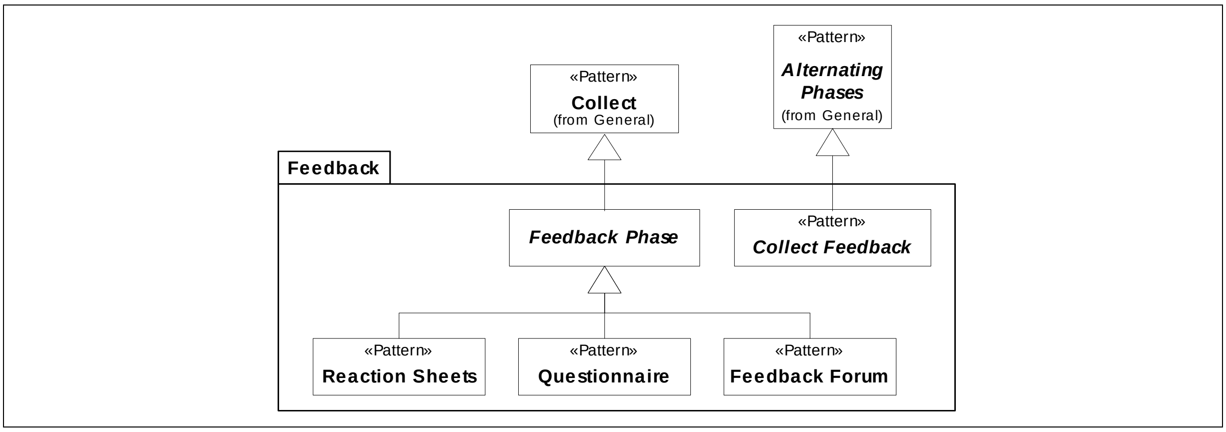


Figure 31: The *Feedback* package.

General

The *General* package hosts patterns that are generally reusable or do not perfectly match any one of the specific purposes defined for other packages.

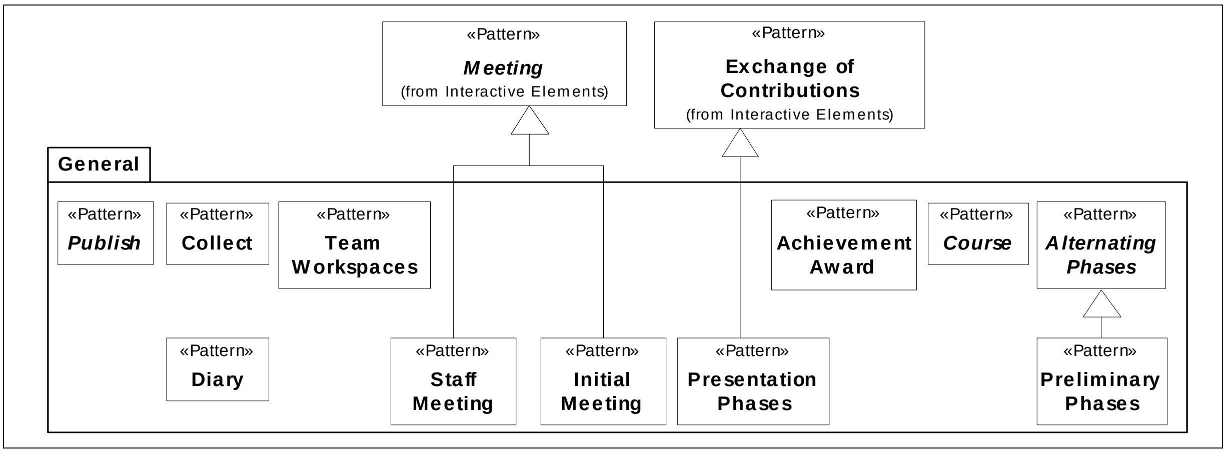


Figure 32: The *General* package.

Interactive Elements

The *Interactive Elements* package is by far the largest package, defining a number of patterns that aim to foster interaction and interactivity among participants, instructor, tutors, and/or external guests. The central proposition is that at least two actors must be involved to constitute an interactive element.

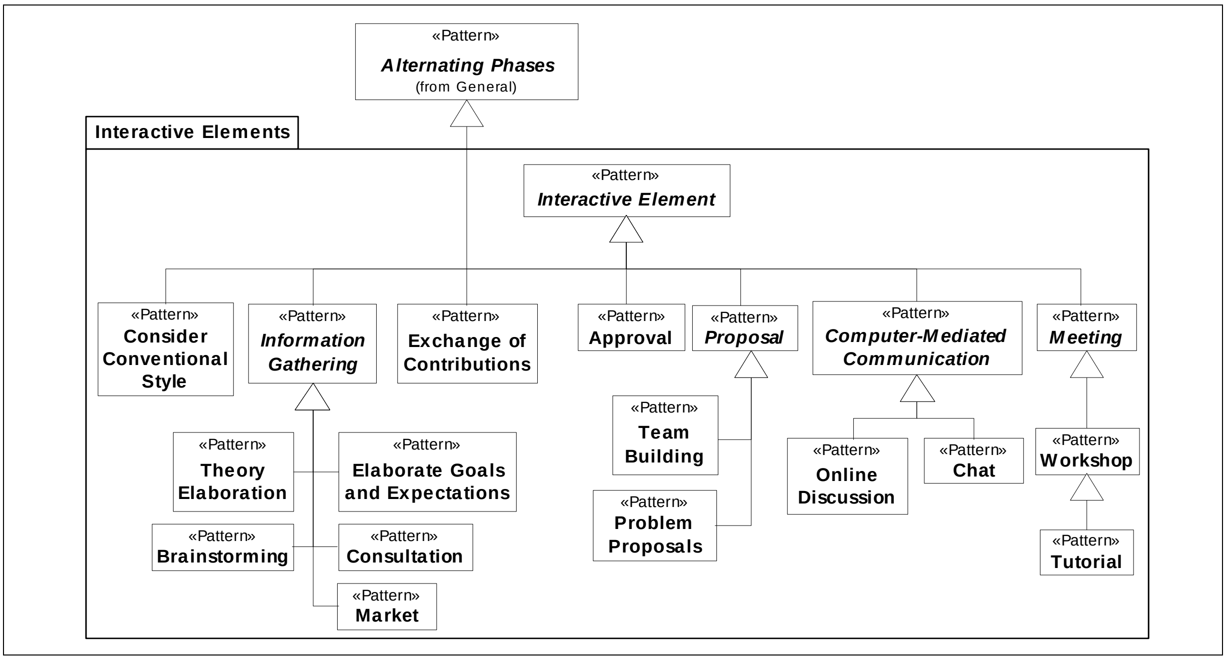


Figure 33: The *Interactive Elements* package.

Project-Based Learning

The *Project-Based Learning* package defines patterns that describe some sort of iterative and/or incremental learning process which can be expressed in terms of (project) milestones.

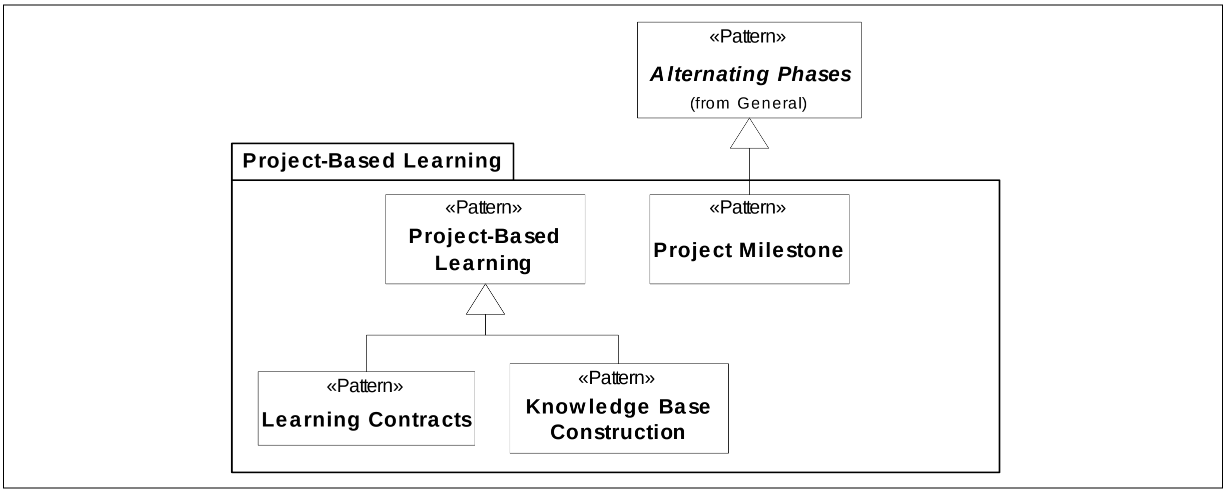


Figure 34: The *Project-Based Learning* package.

Finally, there is a list of patterns (or pattern candidates) that are not yet included in the pattern repository, but foreseen for future extensions:

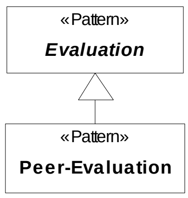
- **ENCOUNTER GROUP**: A particular kind of MEETING, where participants meet to share and express their personal feelings. Encounter groups are usually "led" by an experienced facilitator. In an encounter group, participants are facilitated such that deep, lasting personal experiences and significant learning can take place.[403]

- **Interview**: A possible form of Collect Feedback, which may be supported by means of Computer-Mediated Communication. The aim is to pose a set of questions and to record the resulting conversation.
- **Role Play**: An Interactive Element that allows participants to assume roles in playing certain situations that are covered by or based on some theory. For example, students play different roles in simulating a negotiation process. In such situations, role playing allows participants to experience the underlying theory and to reflect and communicate about the theory and their experiences.
- **Survey**: Aims to Collect opinions and reflections on certain topics. Surveys can be conducted on the Web by providing means of accessing the underlying material and submitting a written opinion. Subsequently, opinions and views can be shared and exchanged online or in a subsequent Meeting.

3.2.2 Pattern Relationships

Structural relationships between patterns can take on one of two different types:

1) Generalization / Specialization



This relationship interconnects a more concrete lower-level pattern with a more abstract higher-level pattern. For example, the Evaluation pattern is (among others) specialized by the Peer-Evaluation pattern, as the latter specifies that the evaluation is done by participants' peers. A derived pattern inherits all the sections (including section content) from its parent pattern. But it may override any of the inherited sections by specifying an own – more dedicated and differentiated – description of the section. In such a case, propositions made in super-sections are still valid, but the current section may refine any aspect of the super-section, and may add additional aspects. If any section is left out in a derived pattern, arguments for doing so are presented in an additional section called *Remarks*.

Note that the *Sequence* section of a pattern deserves special attention with respect to inheritance: In statecharts (such as activity diagrams), inheritance is a combination of *extension* and *refinement*[404]: Extending an activity diagram means adding activities and control flows, while refinement of an activity diagram means decomposing activities into subactivities. There are two types of behavior-consistent inheritance[405]:

403 See also p. 27

404 Cf. Stumptner and Schrefl (2000)

405 Simplified from Stumptner and Schrefl (2000, p. 528)

- *Observation consistency*, based on observation of concrete instances of the flows described by the diagrams. A specialized activity diagram `X` constitutes an observation-consistent inheritance of another activity diagram `Y`, when each possible observation that is a valid instance of `X` is also a valid instance of `Y`, given all extensions and refinements by `X` are ignored for the observation.
- *Invocation consistency*, meaning that an instance of a specialized activity diagram X can be used in the same way as an instance of the "parent" activity diagram Y.

Invocation consistency seems more tightly related to the object inheritance structure underlying the activity diagrams. This viewpoint may not strictly apply to our "learnflows" as they rather resemble workflow or organizational activities, and not object state transitions. So the inheritance relationship of the *Sequence* sections of the PCeL patterns is primarily based on ***observation-consistent inheritance***.

Note that observation-consistency is often not easy to recognize at a first glance. As an example, consider Figure 35: The base activity diagram `Y` just includes a decision whether to execute activity `A` or `B`. Activity diagram `X`, which inherits from `Y`, refines `Y`'s activity `A` into `A1` and `A2`, and adds the activity `C` as successor to `A`. So even if `B` is completely missing in `X`, `X` is still an observation-consistent specialization of `Y`, as each observation of `X` excluding the refinement of `A` and the extension through `C` is also a valid observation of `Y` (or can be expressed in terms of `Y`'s activity flow such that always the decision thread including activity `A` is executed in `Y`). In the PCeL pattern repository this can, for example, be observed for patterns that derive from ALTERNATING PHASES. While this pattern's flow of activities looks quite complex, its essence is simple, namely alternating online and face-to-face phases, regardless with which phase to start. Thus, activity diagrams of patterns inheriting from ALTERNATING PHASES may look simpler than their parent pattern.

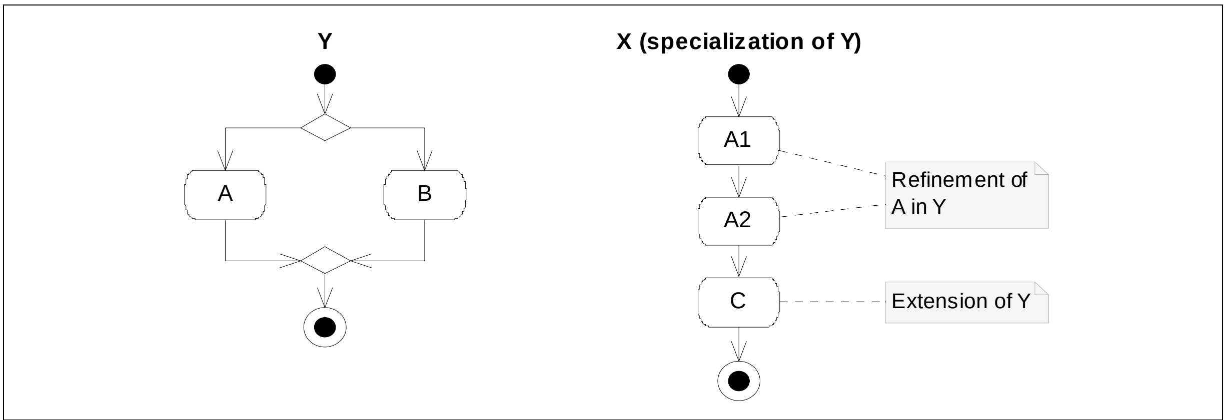
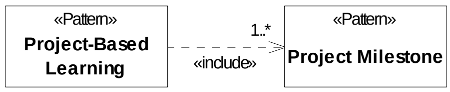


Figure 35: Example of observation-consistent inheritance of activity diagrams.

Generally, to keep the pattern repository clear and understandable, no more than three levels of generalization are used. By adhering to this, all advantages of this type of hierarchical, object-oriented organization can be utilized while not making the repository too over-structured and too complex.[406]

2) Dependency

The dependency relation is used to model the usage, inclusion, or refinement of another pattern. Usage of this relationship is based on the sequence diagrams of the patterns: If a pattern uses another pattern in its sequence, it is *dependent* on the other pattern. For example, the PROJECT-BASED LEARNING pattern includes (`«include»`) at least one (`1..*`) instance of the PROJECT MILESTONE pattern in its sequence, however without refining the activities modeled in PROJECT MILESTONE.



Basically, using the dependency relationship without any stereotype is possible but not very meaningful, as no cue is available on the type of dependency. Therefore, each dependency relationship used in the pattern repository carries one of the following stereotypes:

- `«include»` is used in cases where the *client* pattern (the source of the dependency) includes the *supplier* pattern (the target of the dependency) in its sequence. Such a dependency may be interpreted as one kind of dynamic `part-of` relationship. If the inclusion in the sequence is optional, the dependency relations additionally shows a tagged value `{optional}`.
- `«derive»` is used in sequence diagrams to model that the current pattern is derived from another pattern. In the structural dependency models the equivalent relationship is generalization.
- `«successor-of»` is used when the client pattern depends on prior execution of the supplier pattern in any part of its sequence. For example, TEAM WORKSPACES depends on a completed TEAM BUILDING scenario for its sequence, as team workspaces can not be initialized before any teams are built.
- `«use»` is a stereotype that is assigned when one of the activities in the *client* pattern uses another pattern (the *supplier*) for its purposes, without including it en bloc (see the description of `«include»`). For example, LAB COURSE uses GENERIC EVALUATION as its activity “Blended project evaluation” uses a form of GENERIC EVALUATION.

Of course, usage of many other stereotypes would be feasible and possibly useful, but the primary aim in pattern modeling was clarity and understandability even for users who are “newcomers” to the UML and its formal concepts. So in some cases, formal details were traded for better understandability.

[406] The Depth of Inheritance Tree (DOI) as well as the Number of Children (NOC) per class are relevant object-oriented design metrics that have a huge impact on the comprehensibility and complexity of the structural design, whereby high DOI or NOC values introduce higher complexity. See for example Chidamber and Kemerer (1994), Henderson-Sellers (1995), Rosenberg (1998).

3.3 Pattern Description and Modeling

The following sub-Sections list and describe the *sections* that are used to uniformly describe and model each single pattern. Usually, each pattern includes all of the sections described below, but there is one defined exception: Due to distinct intentions and categories of patterns (administrative patterns, learning activity patterns, etc.) one or more of the sections may be inapplicable or irrelevant for certain patterns. In such a case, the respective section is not just omitted, but rather the reason for the section being empty is depicted. This can be one of the following:

- *Inapplicable*: The section is inapplicable or irrelevant to the pattern at hand.
- *Inherited*: The section is inherited from the parent pattern as is, without refinements.
- *Not available*: The section is not available. The reason for this may be presented when it is not obvious.
- *To be done*: The section has not yet been elaborated.

The main reason for not just omitting sections is supporting pattern readers by providing each section (regardless of actual pattern and content) in the place they expect it[407]. There are only two exceptions to the non-omission rule: the *Remarks* section is omitted if there are no additional remarks to the pattern, and the *References* section is actually omitted if there are no references cited in the pattern text, as the reader would not expect this section to be present in such a case.

Anyway, the *Pattern Name*, *Intent*, *Motivation*, *Taxonomy/Dependencies*, and *Parameters* sections are completed in each pattern.

3.3.1 Pattern Name

The first words that hit the eyes of users are those of the pattern name. Pattern names have been selected with uttermost attention to the capability of succinctly conveying the essence of the pattern.

3.3.2 Intent

This Section includes a short statement about the situation or scenario the pattern addresses. Usually, the pattern intent does not cover more than one or two sentences. Additionally, the

[407] One may argue that it could be reasonable to define different sets of sections for different categories or types of patterns (e.g., different sets for abstract and for concrete patterns). If there were just two distinct section sets it would be acceptable to do so. However, more than two sets of sections would be required even for patterns at the same level of abstraction.

intent of the patterns is one of the central criteria for the organization of the repository: A new pattern is only created when there is significant difference to otherwise similar patterns. If two or more patterns share large parts of their intent, the creation of a more general pattern holding the shared parts is considered.

The *Intent* section lends from Gamma et al.[408], as it supports quickly locating and comparing patterns depending on the problem at hand.

3.3.3 Motivation

The *Motivation* section aims to justify the existence of the pattern. The content of this section cannot be uniformly explained for all patterns, as there are many different categories of patterns, each requiring different motivational considerations. The following list shows possible issues that may be discussed:

- Motivational aspects of the pattern scenario are outlined along with a discussion on relations to learning theory and other patterns.
- Characteristics, problems and/or shortcomings of similar conventional scenarios are discussed.
- Technical aspects or technical necessity of the pattern in the context of Person-Centered e-Learning are presented.
- In generic, higher-level patterns the motivational foundation is laid for more concrete sub-patterns.
- When appropriate, thoughts and suggestions are presented on how Person-Centered attitudes and characteristics of Person-Centered Teaching are involved in the scenario.
- Implications and/or variations are considered for putting the pattern into practice.
- Crucial parts of the pattern sequence are described verbally, as a sort of introduction to the following *Scenario* section.

3.3.4 Sequence

This section presents the sequence of the pattern as a UML activity diagram. Additionally a table with descriptions of the activities may be presented below the diagram (only if the diagram is not self-explanatory).

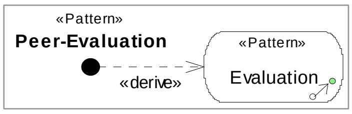
The *Sequence* section is one of the most important pattern sections: Patterns are not only identified from taxonomical or ontological viewpoints but for a pattern to evolve a (possibly generic) meaningful sequence must exist. For example, a *learning activity* is not described as

[408] Gamma et al. (1995); see also Section 2.2.3.1.2, p. 57

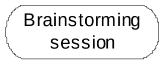
a pattern as not even a default, generic sequence can be provided. Instead it is a term that is described in the dictionary[409] and used subsequently by patterns.

Intending to implement a pattern's activities on a learning platform, one important consideration regards the responsibility of memorizing and controlling the current state and position in the sequence of activities. As there is intentionally no automatic processing mechanism for pattern instantiation, the instructor has to keep track of the current state and position in the sequence. While this might introduce administrative overhead (which seems to be required anyway), such proceeding is independent of learning platform options and constraints, as the *instructor or administrator triggers the initialization and execution of a pattern and its sequence of activities*. This way, the patterns provide general guidance for designing and conducting the learning process and offer a substantial degree of flexibility for instructors while avoiding being too prescriptive[410].

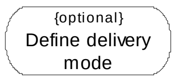
The following UML model elements are used in the *Sequence* sections[411]:



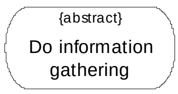
- ***Start node***: Each sequence has exactly one start node. It carries the name of the sequence or pattern and, if the sequence describes a pattern sequence it is stereotyped with `«Pattern»`. If the sequence derives from another sequence, the start node has an outgoing dependency relation (with stereotype `«derive»`) that points to the parent sequence, which is modeled as a subactivity.



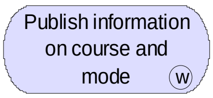
- ***Activity (action state)***: Activities are the main elements of the pattern sequences. They are used to describe actions and interactions of any type, be it a learning activity or an administrative activity. The name activity may be misleading, as it does not tell anything about the actual timeframe its execution will occupy. It just describes a coherent set of actions. Each activity has at least one incoming transition and exactly one outgoing transition. Additionally each activity may carry one or more of the following tagged values and stereotypes:



  - The tagged value `{optional}` means that the activity is not explicitly needed for this pattern and thus can be left out in an instantiation of the pattern, or can be provided as voluntary, extra effort.



  - The tagged value `{abstract}` means that the activity describes an abstract flow of actions which has to be specified by sub-patterns or by the user who instantiates the pattern. Abstract activities frequently occur in generic higher-level patterns.



  - The stereotype `«web-based»` means that the activity primarily proceeds online, with Web support. If an activity stereotyped this

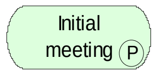
[409] See Section 3.4, p. 127

[410] This is in line with current opinions in pattern research, e.g., as discussed by Goodyear et al. (2004)

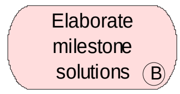
[411] For a general overview of model elements used in activity diagrams refer to Section 2.3.3.3.2, p. 82.

way occurs in a pattern's activity diagram, the pattern typically provides a Web template for online-support of the activity. Note that this stereotype is not presented textually, but as an icon (a circle containing the letter `W`) at the right-hand side of the activity. Additionally, the activity may be filled with light blue color to increase visual effect and recognition.

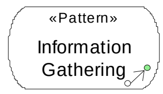
- The stereotype `«present»` indicates that the activity primarily takes place in a face-to-face setting. The icon for this stereotype is a circle containing the letter `P` at the right-hand side of the activity. The activity carrying this stereotype may additionally be filled in light green.



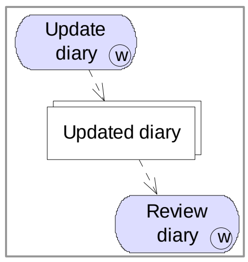
- Finally, the stereotype `«blended»` indicates a mix of the former two stereotypes: the activity is conducted in a blended style, mixing online and face-to-face modes. The icon for this stereotype is a circle with the letter `B` at the right-hand side of the activity. An activity carrying this stereotype may additionally be filled in light red.



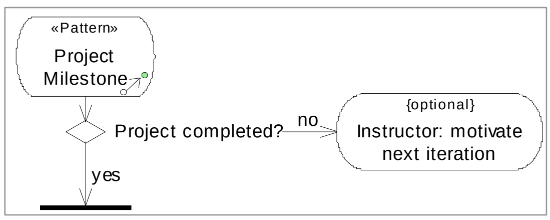
- ***Subactivity (subactivity state)***: A subactivity (also known as compound activity) points to another activity diagram that shows a more detailed flow of the activities. This is very useful to avoid overloading diagrams with too many activities. If the subactivity points to a pattern sequence, it carries the stereotype `«Pattern»`. Additionally, a subactivity can carry the same tagged values and stereotypes as a normal activity. Note that subactivities are denoted with a small arrow icon in the lower right corner of the model element.



- ***Objects and object flow***: Objects are used to model documents and content, which are output from or input to activities. A situation where an object is direct output of an activity and immediate input to some subsequent activity resembles what is defined as "object flow" in UML specification[412]. Thereby, a dashed arrow (much like the visual representation of a *dependency*) is drawn from the activity to the output object and from the input object to another activity. If output and input object are the same, no transition needs to be drawn between the involved activities.

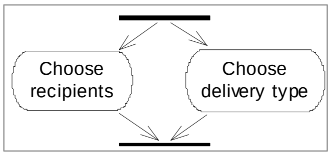


- ***Decision***: Decision elements (depicted as diamonds) are capable of diverting the flow of control with respect to a certain condition. To increase readability of the diagrams in the pattern sequences, conditions are mostly displayed as questions with possible responses on the outgoing transitions (as opposed to formal guard conditions in square brackets proposed by the UML specification.)

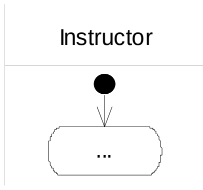


[412] OMG (2003, p. 3-163)

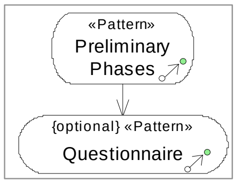
- ***Synchronization bar***: This element splits up the flow of control into two or more *concurrent* flows. Usually, the flows are merged again later in the sequence by another synchronization bar.



- ***Swimlane***: Swimlanes mark off areas of responsibilities of different roles involved in a pattern sequence. They are only used when clear arrangement of activities remains possible. Otherwise the name of the responsible role is depicted within the respective activity followed by a colon and the activity's name.



- ***Transition***: A transition is depicted as a solid arrow that connects activities, subactivities, decisions, and/or synchronization bars with each other. A transition fires when the source element yields flow control to connected elements. This is the case, e.g., when the source activity has completed its flow of actions. If the transition end carries a star (`*`), it means that the source activity may be executed an arbitrary number of times. Or, when it carries a number, the activity is executed that number of times.

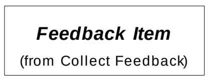


- ***End node***: Each flow of control in a pattern sequence ends with an end node, so there may be multiple end nodes in a sequence, e.g., when the flow is split up by a decision element.

3.3.5 Structure

This section shows structural relations among entities and concepts involved in the pattern as a UML class diagram[413], whereby each structural element, concept, or entity is modeled as a UML class. Only the most essential classes are modeled, with focus on direct relevance for the actual pattern. This not only supports understandability of terms used in the pattern but also shows how certain structural elements of other patterns are involved. No UML extension mechanisms are used in the structure diagrams.

If any structural element defined in another pattern is involved in the current pattern, there is no redefinition of that element, but it is reused. To make this visible to the reader, the host pattern of the element is depicted in the lower part of the respective class in the form "`(from` *Pattern name*`)`", e.g., "`(from Collect Feedback)`".



[413] See Section 2.3.3.3.1, p. 78, for a general introduction to UML class diagrams.

3.3.6 Taxonomy/Dependencies

This section embeds the pattern into a network of related patterns by using UML class diagrams. Only the most adjacent patterns, to which the current pattern maintains relationships, are depicted. This is valid except for generalization relationships where the whole inheritance path for the current pattern up to its inheritance root is always depicted.

So this section presents a constrained view on the overall pattern network, including the packages in which the patterns are located. The core package containing the currently discussed pattern is always depicted, even if the constrained view contains only one pattern (the one at hand). Another pattern's package is only (but not necessarily[414]) explicitly placed in the dependency overview if *more* than one related pattern in this package is related to the pattern at hand. In any case where only a single pattern of a related package is needed, the package where the pattern is located is placed as textual information in the lower part of the respective pattern's graphical model element (see the example in Figure 36 below) .

The pattern in the current context is always highlighted by a slight gray shade. If this pattern is used by all patterns of another package (or vice versa) for the sake of simplicity only one dependency relation will be connected with the respective package instead of connecting each pattern of that package.

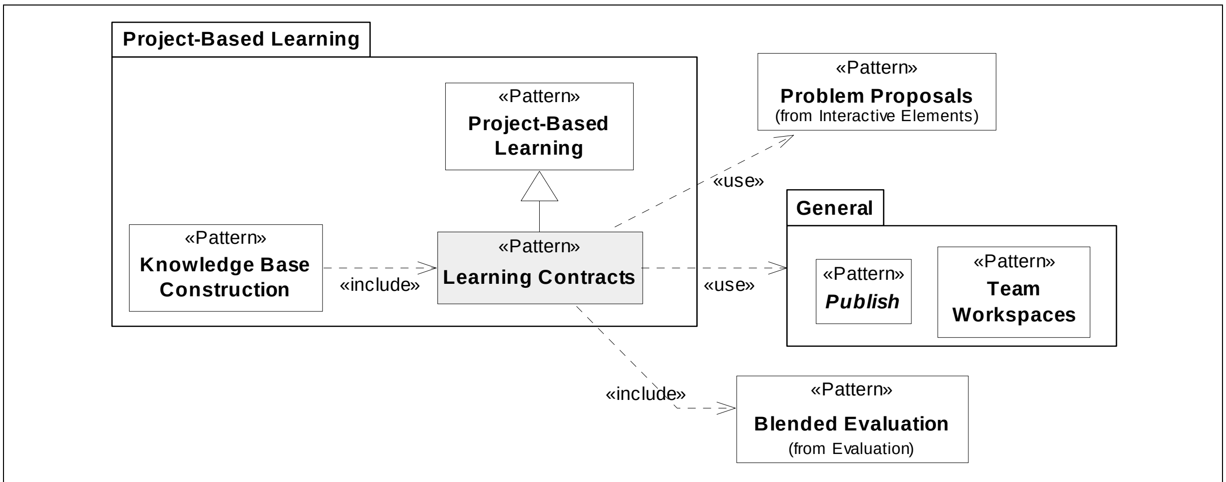


Figure 36: Taxonomy/Dependencies section of the LEARNING CONTRACTS pattern.

3.3.7 Parameters

Parameters help to categorize, identify, relate, and distinguish different patterns. Additionally, the parameter values provide a concise list of common properties of each pattern. In a

[414] For example, when clear arrangement of patterns is infeasible using this rule (i.e., two patterns from one single package have to be dislocated to produce a clear diagram). This is the case in the *Taxonomy/Dependencies* section of EXCHANGE OF CONTRIBUTIONS.

data-persistent pattern repository these parameters may be used for querying patterns by providing desired features and their values.

If a parameter cannot be attributed an appropriate value within the current pattern, or if a parameter is just inapplicable to the current pattern, it is omitted from the list rather than showing an empty value. Each pattern inherits the *parameters as well as their respective values* from each of its parent patterns, and may add or redefine parameters and values. To sustain overview and clarity of parameter lists scattered over so many pages in the text, any inherited parameters are reprinted in each pattern, to avoid the necessity of looking up parameters in parent patterns.

The following is a commented, complete list of the parameters used in the repository, along with descriptions of possible values for each parameter:

Primary pattern author

This parameter lists the primary pattern author(s). In this initial version of the pattern repository most of the patterns have been written by the author of this thesis. However, intended subsequent population of the repository by external contributors and colleagues justifies the specification of this administrative parameter.

Primary pattern source

This parameter depicts the institutional or material source (i.e., the original context) of the pattern, which will in most cases be the primary pattern author's institution, where the pattern was mined or initially applied. Additionally, some patterns may rely and/or lend significantly from literature and/or online material.

Pattern categories

Used for attribution of one or more of the following descriptive categories:

- *Administrative*: This category is assigned to patterns that mainly support organizational or administrative tasks. Such a pattern is usually completely supported by Web templates. TEAM WORKSPACES is a typical administrative pattern.
- *Collateral*: Collateral patterns are linked to a specific learning activity, whereas their flow is synchronized with that of the 'sibling' learning activity. Using ACHIEVEMENT AWARD in a LEARNING CONTRACT scenario would be a typical example, where ACHIEVEMENT AWARD is the collateral pattern.
- *Composite*: Composite patterns are mainly or completely composed of other patterns, and do not define many activities on their own. A typical example is ASSESSMENT PHASES. The implication is that patters assigned to this category mostly defined no own *Web Template*, *Example*, and *Evaluation* sections, but rather point to the respective sections of the included patterns.
- *Course type*: This category is used for patterns that describe scenarios for typical course types, e.g., for SEMINAR.

- *Generic*: This type of pattern provides a generic template to be refined or specialized by more concrete patterns or by the pattern user. Generic patterns are usually not readily applicable, as they just arrange a number of other (possibly abstract) patterns and activities in a fashion that has to be concretized by sub-patterns. Typical examples include COURSE (describing a generic sequence of phases within a course) or GENERIC EVALUATION (describing possible combinations of EVALUATION scenarios in the ASSESSMENT PHASES of a COURSE).
- *Motivational*: Motivational patterns are used to leverage learning motivation or activity of the participants, e.g. by providing incentives. Often, patterns of this category are also collateral with a certain learning activity to which they are anchored. A typical example is ACHIEVEMENT AWARD.
- *Supplementary*: Supplementary patterns are typically based on other patterns and may be used to achieve an optional, yet useful aim (e.g., KNOWLEDGE BASE CONSTRUCTION).
- *Traditional*: This category is used for patterns and scenarios that are also widely used in conventional scenarios or that it contains aspects of traditional teaching scenarios in major parts, e.g., for INSTRUCTOR-EXAMINATION.
- *Utility*: A utility pattern just provides a basic flow of activities and/or a basic structural model of entities involved. Often, utility patterns have no child patterns, but they are used frequently with minor variations. PUBLISH is a typical utility pattern. Also, such patterns can potentially be used in many different scenarios, so that they do not define their own *Web Template* and *Evaluation* sections.

Level of abstraction

Each pattern is embedded into a generalization/specialization hierarchy. Three distinct levels of abstraction within that hierarchy are specified:

- *Low*: Patterns which are located nearer to or at the leaves of the repository hierarchy are usually – but not necessarily – lower in abstraction. Low-level patterns are easier to apply because their description is close to the terms of concrete activities.
- *Medium*: Medium abstract patterns are located somewhere between lowly and highly abstract patterns. At this level, both reusability and direct applicability are usually fairly high.
- *High*: Patterns which are located nearer to or at the roots of the repository hierarchy are usually higher in abstraction. High-level patterns aim at providing a generic template to derive from. They are normally not directly applicable as significant parts are either specified in a very abstract way or some activities are purposely left to be defined by lower-level patterns.

Scope

The scope identifies the 'normal' duration of the pattern. As time measures seem inappropriate and in many cases unreliable, the following measures are used instead:

- *Activity*: Activities typically occupy a timeframe of some minutes or a couple of hours, and in rare cases even some days. Examples include presentation blocks in a seminars (MEETING) or PUBLISHING documents.
- *Phase*: A phase may occupy a couple of days or even weeks, e.g., a PROJECT MILESTONE.
- *Course*: Patterns attributed with scope 'course' occupy all or at least a major share of the duration of the whole course, e.g. PROJECT-BASED LEARNING or INTERACTIVE LECTURE.

Note that some patterns may have an ambiguous scope. For example, a face-to-face discussion is certainly an 'activity', while an online discussion may even occupy a couple of weeks and thus justify the attribution of 'phase' or even 'course.'

Primary presence type

This parameter can hold one of the following values:

- *Online*: Major parts of the pattern sequence are supported by Web templates and thus proceed online.
- *Present*: Major parts of the pattern sequence take place face-to-face.
- *Blended*: The essence of the pattern is a mix of online and face-to-face activities.
- *Concurrent*: The essence is two complementary streams of learning activities, one proceeding online, the other face-to-face, e.g., WORKSHOPS may be held concurrently with ONLINE DISCUSSIONS.

Flexibility

This parameter specifies the estimated order of magnitude of possible, reasonable variations of the pattern without mutilating its intent:

- *High*: There are numerous points in the pattern that may be interpreted, specified, or applied differently.
- *Low*: The pattern describes a rather invariant scenario, not giving much space for alternatives.

It is mostly useless to specify this parameter for higher-level patterns, as these typically describe highly flexible scenarios. It is important to note that this parameter addresses the flexibility of the *pattern*, not the flexibility of the underlying scenario. For example, usage of ONLINE DISCUSSIONS is certainly highly flexible as there are manifold application scenarios;

however, the pattern (which is creating a forum at a certain location and subsequent posting of messages) is rather inflexible.

Level of confidence

This parameter specifies the level of confidence the author has in the pattern:

- *1*: This reflects a low level of confidence in the pattern, which is mostly due to insufficient application reports/evaluations or early research cycles/stages. The pattern is intended as a proposal or idea to be used, researched and/or evaluated.
- *2 – 4*: Between low and high.
- *5*: The pattern is considered to promote well-substantiated advice, because it (A) has already been successfully applied in a number of circumstances, and/or (B) has been thoroughly evaluated both practically and theoretically, and/or (C) is grounded on primary research reports and results. However, such a high level of confidence does *not* imply that every pattern user or reader must perfectly agree with parts or all of the content/description of the pattern.

This parameter is subject to changes over time as it is dynamically determined by research results or application experiences. It is mostly inapplicable to higher-level patterns, as these patterns cannot be readily put into practice and are typically not intended to describe concrete scenarios.

Number of participants

This parameter provides information about the number of participants which allow for reasonable application of the pattern by one single instructor. It is mostly defined in terms of ranges (e.g., up to 20, 10 to 15, etc.). The range boundaries are not intended to be interpreted rigorously, but shall give an approximation of what is reasonable.

However, number of participants does not apply to all patterns, e.g. it makes no sense to supply a value for this parameter in PUBLISH or other pure utility patterns, as they highly vary depending on the patterns that use or incorporate them. In such a case this parameter is omitted.

In other cases the value 'unrestricted' is applied which means that the number of participants is not implicitly restricted by the pattern itself but just affects the application effort positively (fewer participants) or adversely (more participants).

Person-Centered variables addressed

Lists the Person-Centered variables which are addressed or involved when the pattern is applied, whereby the focus is on supporting the instructor in his or her expression of an attitude by putting the pattern into practice. Possible values are[415]:

[415] For a description of these attitudes please refer to Section 2.1.3.1, p. 24

- *Acceptance*
- *Transparency*
- *Understanding*

Application effort

The application effort required to execute a pattern heavily depends on values of other parameters, such as 'number of participants' or level of expertise of the instructor and/or participants, whereas this is not measured in time units but rather in terms of personal effort required. Anyway, the following relative values try to at least give a hint:

- *Low*: Most of the activities in the pattern are either simple or largely supported by administrative or user Web templates.
- *Medium*: Some of the activities in the pattern are more complex, and it requires significant additional thought, content and/or personal resources to be supplied by instructor and administrative personnel.
- *High*: When many aspects of the pattern are abstractly and generically described or when complex activities are involved it requires specific interpersonal or technical capabilities on the side of the pattern user to execute and employ the pattern in a situated way.

Level of expertise required

This parameter tries to specify the level of expertise required to put a pattern into practice both from the perspective of teaching as well as from a technical viewpoint, e.g., required level of interpersonal skills, technical expertise, etc. Three relative values are possible:

- *Low*: No special skill, technique, or assistance is required for this pattern.
- *Medium*: Some special technical or social skills are required expedient application. It may be advisable to consult assistance.
- *High*: The pattern requires special expert skills and/or substantial administrative or expert assistance.

Suggested assistance

For many administrative and teaching activities it is advisable or even inevitable to have assistance. This parameter lists suggested assistance staff for the instructor from three categories:

- *Expert*: Person with high practical experience and/or technical expertise.
- *Tutor*: Student or assistant who is able to technically assist primarily students, but also instructors in administration.

- *Administrator*: Capable of doing administrative and/or technical tasks as well as solving technical problems. This type of assistance is suggested when technology is intensely used in the administration or teaching process or when complex online procedures are employed.

Target skills

Gives a list of skills addressed in the participant (student) by taking part in a scenario of this pattern.

- *Practical skills* and *technical skills*: being able to apply theories or techniques, e.g., programming languages, report writing, solution frameworks, problem tackling, etc.
- *Intellectual skills*: e.g., factual knowledge processing, learning improvement ("*meta learning*"), technical know-how, etc.
- *Interpersonal skills*: e.g., social skills, Person-Centered variables, etc.
- *Communication*: e.g., presentation style, speech, dialog, etc.
- *Collaboration*: e.g., teamwork, distributed work, resource sharing, collaborative construction/elaboration, etc.
- *Reflective thinking*: addresses drawing experience and improvement from critical reflection, e.g. from evaluating one's own learning process
- *Problem solving*: e.g., project work, authentic examples, complex tasks, etc.
- Others as appropriate.

Input

This informal parameter lists artifacts, documents, and/or activities to be supplied as input to the pattern by involved persons or technology.

Output

This informal parameter lists anything that is produced or modified by involved persons or technology in the scope of the pattern.

3.3.8 Web Template

The flow of activities described in the *Sequence* section of a pattern is *not* intended to represent automatically executable flow semantics. Rather, it is intended to serve as a guideline or generic model for conducting, documenting, modularizing, and tracking learning activities as well as to enable targeted arrangement and support on learning platforms. Thereby, each *Web Template* section provides a generic ***visual specification*** of how to arrange Web technology features for Web-based support of learning activities (in particular those activities that are stereotyped as `«Web Template»` in the pattern's activity diagram). As such, specific

Web templates cannot be instantiated on a learning platform right away as they do not represent executable code or concrete implementations of Web pages. The Web template specification only shows *one feasible* option of supporting a pattern online, as there is no single objective measure of optimal arrangement of Web pages and functionality. Concrete Web template *implementations* may be – or better, need to be – tailored to specific target learning platforms (cf. the transition from layer 4 to layer 5 in the BLESS model[416]).

Web templates are organized in three distinct, yet complementary views: Participant view, administrator view, and report view. These views are discussed in the following sub-Sections. Generally, the Web templates are restricted to utilization of basic hypermedia[417] technologies on the Web, i.e.:

- ***Hypertext***: Textual information that may show links (*hyperlinks*) to other hypertexts to allow the reader to quickly jump to related information[418]. The Hypertext Markup Language (HTML) is the most widely used language for creating hypertexts.
- ***Multimedia***: In addition to textual information, Web pages may include different types of media, such as audio, animated graphics, still images, video, and interactivity via keyboard or mouse input[419] (e.g., client scripts or Web forms). However, Web template multimedia is restricted to using text, images, and Web form interactivity (see next bullet) only.
- ***Web forms***: Allow for server-side processing of user input and dynamic building of Web page content. Thereby, different types of input controls are available in HTML and used by Web templates (see Figure 37).

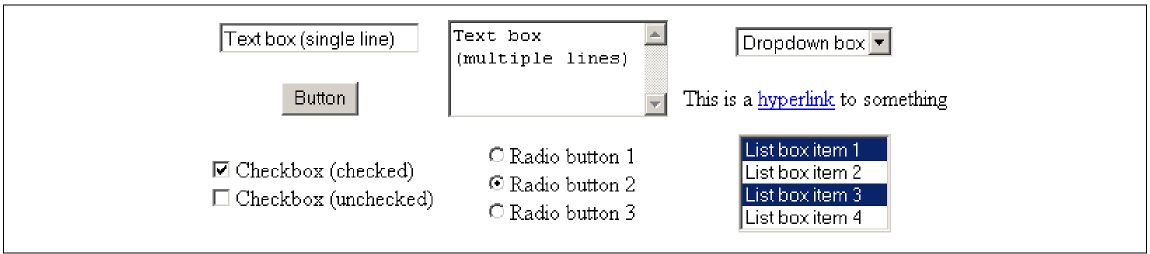


Figure 37: Overview of common Web form controls.

- o *Text box*: Allows the user to enter plain text. Text boxes may be restricted to hold only one single line or to hold multiple lines of text.
- o *List box*: Holds a vertically arranged list of predefined text items. One or more of these items may be selected (highlighted) by the user.

416 See Section 3.1.1, p. 93

417 Hypermedia is a combined term: Hypertext and multimedia

418 Cf. Nielsen (1995)

419 See for example, Holzinger (2001)

- *Dropdown box*: Is a special form of a list box that shows only the selected list item. It also comprises a number of text items in a list, but the list of available items is shown only when the user clicks the dropdown arrow that is usually located at the right-hand side of the control. After the dropdown list appears, the user may select one item, which in turn will be shown in the visible portion (single line) of the dropdown box.
- *Button*: Initiates some action for the form. Usually, each Web form includes one button that triggers the submission of the form to the Web server application, which receives the form control values. Subsequently, these values are processed and a Web page is built and returned to the user in response. Buttons may also trigger the execution of client-side scripts that are typically used to manipulate the Web form content.
- *Radio buttons*: These are always used in groups. For each radio button group, only one bullet may be set by the user, i.e., radio buttons offer choices where the user may select exactly one choice ("*single choice*" option). This is often used in questionnaires, for example when the user has to select exactly one value for some item on a nominal scale[420].
- *Checkbox*: In contrast to radio buttons, checkboxes allow for *multiple choices*. The user may place a check mark in any checkbox by clicking it. This is usually used to offer non-exclusive options to the user (e.g., enabling some application feature, selecting multiple preferences, etc.)
- Web forms may also hold simple *hypertext* and *hyperlinks*. Usually, hyperlinks are printed underlined in blue letters.

3.3.8.1 Participant View

The participant view shows a number of generic, interactive Web page specifications that are connected with each other (through hyperlinks, Web form submission, etc.) In combination, the Web pages described in that view should be capable of allowing for accessing and using all the functionality that is needed to support a pattern from the viewpoint of the course participant.

Note that some or all of the visual presentation in the participant view may be dependent on settings that are defined by the administrator/instructor. For example, textual information on the submission of REACTION SHEETS may be different from course to course. Therefore, the submission Web page of REACTION SHEETS (see Figure 38) does not show a concrete instance of the submission information, but rather a *placeholder* that indicates that this part is computed from administration view settings or from user input. Such placeholders are presented in ***italic type and enclosed in square brackets***, e.g., [*Submission*

420 For details about the nominal scale refer to p. 127

`information`]. As each Web page template is annotated with a description of the page items, the intent and source of these placeholders are made clear to the reader.

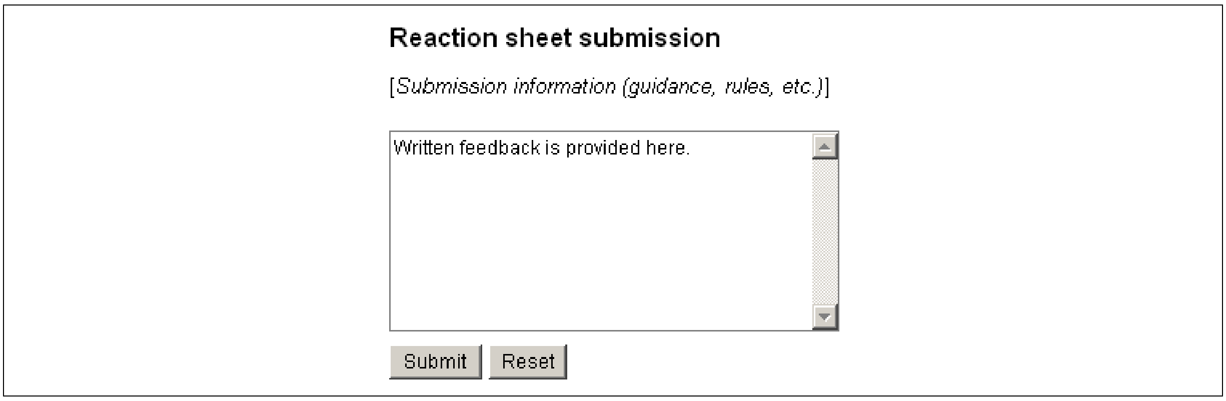


Figure 38: Participant view of the submission page Web template of REACTION SHEETS.

Figure 38 makes obvious that each page Web template holds only the minimal information required, to allow for keeping the templates as generic as possible. There are no visual adornments or other special formatting.

Note that some patterns require multiple Web page templates in the participant view that are hyperlinked through text or interactive Web form items. In such cases, to support the reader, the *Web template* section includes a use case model that shows the basic use cases and the involved actors supported by the Web templates pages.

3.3.8.2 Administration View

Each pattern, when supported on a learning platform, is dependent on a set of parameters that influence its appearance and behavior in the participant view. In order to instantiate a pattern on a learning platform, an *administrator*[421] has to supply values for the pattern's input parameters and options. For example, an ONLINE DISCUSSION forum[422] may allow ordinary users (i.e., course participants) to post replies to existing threads or messages, but may deny them to create their own discussion threads. Such an arrangement has to be specified by an administrator in the administration view.

Therefore, the administration view shows Web page templates that allow for configuration of the pattern's appearance and behavior in the participant view. Note that, as depicted in Figure 39, the Web templates in the administration view are confined to show ***only the actual pattern configuration activities*** in the online instantiation process. The activity of *selecting* a pattern for instantiation, as well as storing the pattern configuration for later

[421] The administrator represents a *role* that may be impersonated by instructors, administrators, tutors, etc.

[422] See the pattern on p. 347

reuse, is not included in any of the Web template views, as this is not a distinguishing action for a given pattern.

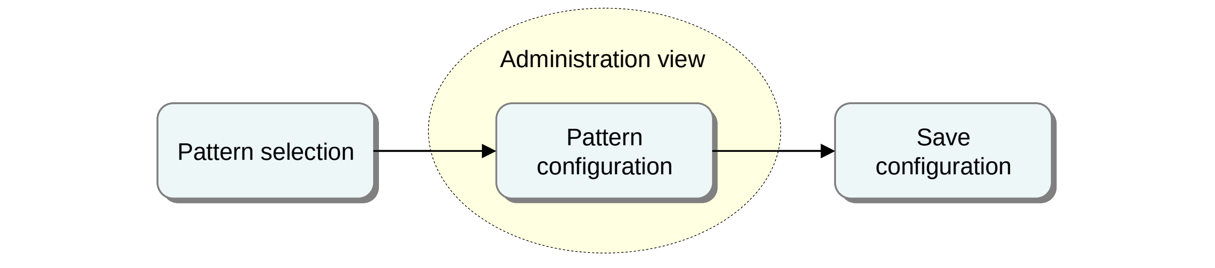


Figure 39: The scope of the administration view in the pattern instantiation process.

The visual presentation of the Web page templates in the administration view is similar to that in the participant view. In addition to the visual Web page templates, the different options and inputs are explained in the supplementary description text. Note that the configuration options required for pattern instantiation may be too manifold or too diverse to be reasonably combined into one single Web page. In such cases, the configuration settings are grouped into different *configuration steps* and presented to the administrator in a sequential, ***wizard***-like manner. This should contribute to ease of use in terms of learnability and efficiency[423] of the configuration process.

Providing a separate view on the administration of online pattern instantiation has one salient benefit: Any of the settings configured by administrators can be stored as templates for reuse in subsequent configuration processes. This mechanism may be employed for complete configuration sets (i.e., the whole pattern instantiation configuration), or for single inputs such as the general information text for REACTION SHEETS (see the participant view above). As a prerequisite, the values entered by administrators in each online pattern instantiation process have to be stored permanently for later retrieval (see the final activity in Figure 39) using a unique, human-readable identifier to be supplied by the administrator. Coming back to the reaction sheets example, the overall configuration of the REACTION SHEETS pattern for the final course reaction sheets in the Web Engineering course in the year 2003 could be stored using the identifier "*Web Engineering (2003), Final course reaction sheets*" (see Figure 40).

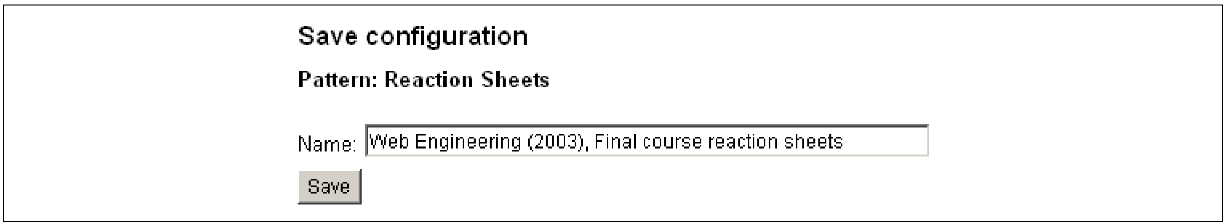


Figure 40: Web page for saving the current pattern configuration.

[423] Cf. Holzinger (2005, p. 72)

For later retrieval of stored configurations, the administration wizard has to include buttons or hyperlinks to *select an existing configuration* where appropriate. To keep the templates in the administration view clear, these buttons/links are ***not included***, even though an implementation should consider them. The configuration selection page referred to by these buttons/links is also not specified, as it is generically reusable by different administration Web templates. For example, the Web page for selecting existing configuration settings for reuse may look like the one depicted in Figure 41, independent of the pattern that is actually configured.

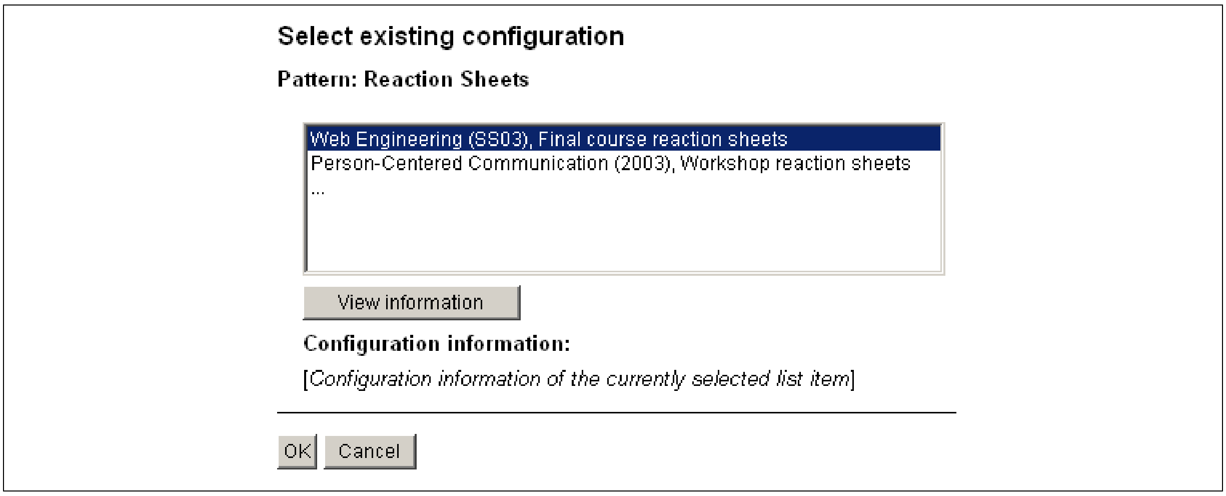


Figure 41: Web page for selecting an existing pattern configuration.
The screenshot shows a simple Web page layout for selecting an existing (i.e., previously stored) configuration of the REACTION SHEETS pattern. By clicking the *View information* button, the administrator may review the configuration information for the currently selected item. This may either be a summary of a complete pattern configuration, or just one single configuration item of a complete configuration, depending on the current context. Note that the list box only contains items relevant to the current pattern instantiation context (REACTION SHEETS) that is supplied by the referring administration Web page.

3.3.8.3 Report View

One frequently recurring task of any online course instructor/facilitator is collecting information about (online) learner activities. Using standard learning platforms, this can turn out to be very annoying, as you have to browse through various pages and either print each desired page or copy and paste desired sections into a separate report document.

The report view aims at alleviating the reporting process by providing tailored, situated report views on patterns, where appropriate. Thereby, the report application (typically a Web application that is restricted to be used by teaching staff) collects relevant information from the data that was collected by the online pattern instances during the learning activities. Thereby, many patterns are capable of producing various different reports. For example, the report view of the REACTION SHEET Web template may provide reports for a list of participants that have not yet submitted a reaction sheet or for a compilation of already submitted reaction sheets (possibly grouped by or constrained to specific participants). As this example shows, even the report view offers various options to be requested before reports can be generated.

Figure 42 shows the process of generating a report. Note that the report view is constrained to present Web page templates for customizing the generation of each report type supported by the current pattern as well as a generic layout of the resulting report. The first two steps in the report generation process (i.e., selecting a pattern instance and one of its supported report types) are independent of the actual pattern for which a report is generated.

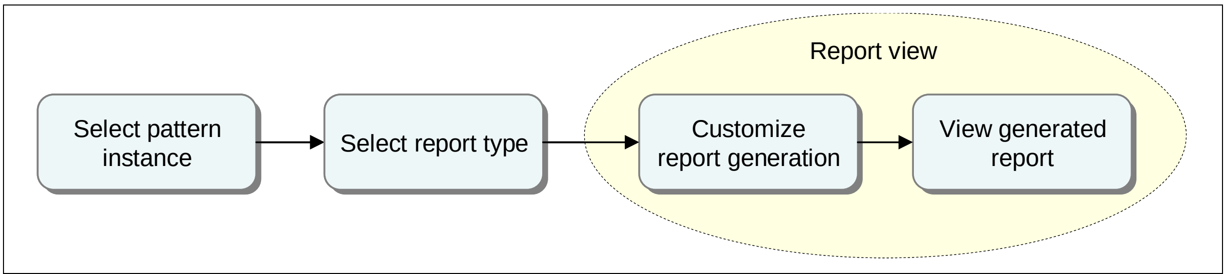


Figure 42: The scope of the report view in the report generation process.

Note that in some cases Web page templates for reports are depicted, while in some cases only verbal descriptions of the reports are given.

3.3.8.4 "Pass-Through" Web Templates

Some of the patterns, especially those that are mainly or completely composed of other patterns[424], deserve special attention regarding the Web template. These patterns usually do not provide an own Web template. They rather act as "pass-through" patterns, meaning that their administration view just leads the administrator through to the administration views of the included patterns. Consequently, the participant and report views are also given by the included patterns. Examples for such patterns are ASSESSMENT PHASES, GENERIC EVALUATION, COURSE, etc.

3.3.9 Examples

This section shows examples where the current pattern was employed or potential scenarios where this pattern is considered to be useful. Note that for generic higher-level patterns no examples can be provided.

If any concrete course is mentioned in this section without reference to the institution where it was applied, the default institution is the one mentioned as the *Primary pattern source* in the *Parameters*[425] section, which in the current version is mostly the Research Lab for Edu-

[424] See the pattern parameter "*pattern category*" for details on p. 114

[425] See p. 114

cational Technologies (RLET) at the Faculty of Computer Science of the University of Vienna[426].

3.3.10 Evaluation

Modularizing learning processes by describing them as patterns allows for targeted, systematic evaluation of many aspects, qualitative as well as empirical:

- General survey of the eligibility of a pattern: Is it capable of supporting learning targets set by instructors as well as curricula?
- Comparison to conventional scenarios: Does the scenario described by a pattern yield substantial benefits/advantages?
- Researching critical success factors for e-learning scenarios: What are the keys to successful implementation from the instructor's, students', and from the organizational viewpoint?
- Stressing the importance of Person-Centered teaching and learning styles and attitudes: Evaluations in the scope of the Person-Centered e-Learning efforts at the RLET have substantiated that Person-Centered attitudes held by instructors increase participants' motivation to participate more actively as well as to draw more significant learning results from the course.[427]
- Formative evaluation, providing direct feedback for optimizing/enhancing the employed scenario(s).

This section is intended to present one or more of the following with respect to the current pattern:

- Questions that have already been used in questionnaires or interviews in one or more courses along with respective results.
- Issues and questions of interest which have not yet been evaluated: these could be target to investigation in subsequent applications of the pattern.

---

[426] The University of Vienna was completely and rigorously reorganized as of January, 2005. I have done most of the work on which this thesis is based in the former organization scheme, where I was affiliated with the Business Information Systems group at the Department of Computer Science and Business Informatics (of the former Faculty of Social and Economic Sciences.) Within the new organization scheme, I am now affiliated with the Institute of Knowledge and Business Engineering and with the Research Lab for Educational Technologies of the newly founded Faculty of Computer Science. However, my original work group remained untouched, and its research work is now primarily allocated to the ***Research Lab for Educational Technologies***. For this reason, this research lab will be used as the pattern source organization in this thesis.

[427] See for example Derntl and Motschnig-Pitrik (2004c), Motschnig-Pitrik (2004a), Motschnig-Pitrik, Derntl and Mangler (2003)

- Students' reaction sheets and/or a qualitative evaluation thereof and a transcript of the students' statements from the final course meetings.

Any questionnaire item (or block of items) that is depicted in this section is annotated with an appropriate scaling suggestion for possible responses. This is based on one of the following levels of measurement[428]:

- ***Nominal***: Allows for differentiating objects with respect to certain properties, whereas the degree of difference remains undefined (example scale: *yes*, *no*)
- ***Ordinal***: Scale values are additionally arranged in a certain order, whereas the distance between scale values remains undefined (example scale: *excellent*, *satisfactory*, *miserable*)
- ***Interval***: Additionally, distances between scale values are defined, e.g., the difference between a value of 1 and 2 is equal to the difference between a value of 4 and 5 (example scale: 1, 2, 3, 4, 5). Note that interval scales are not restricted to numerical scale values, e.g., the values *low*, *medium*, and *high* may also be interval-scaled if the test designer asserts that the difference between *low* and *medium* is equal to the difference between *medium* and *high*. Usually, for calculation purposes such values are mapped to numerical values (e.g., *low* = 1, *medium* = 2, *high* = 3).

3.3.11 Remarks

This section may be used to provide comments or other useful remarks regarding the current pattern.

3.3.12 References

This final section lists references cited in the current pattern.

3.4 Dictionary

The dictionary describes frequently used terms in the pattern repository along with their relationships. The central concept used in educational environments is a *course*. All concepts relevant to the pattern repository are defined in the *Structure* section of COURSE. Therefore, this section may also serve as our general dictionary, with some extensions:

[428] See Trochim (2000)

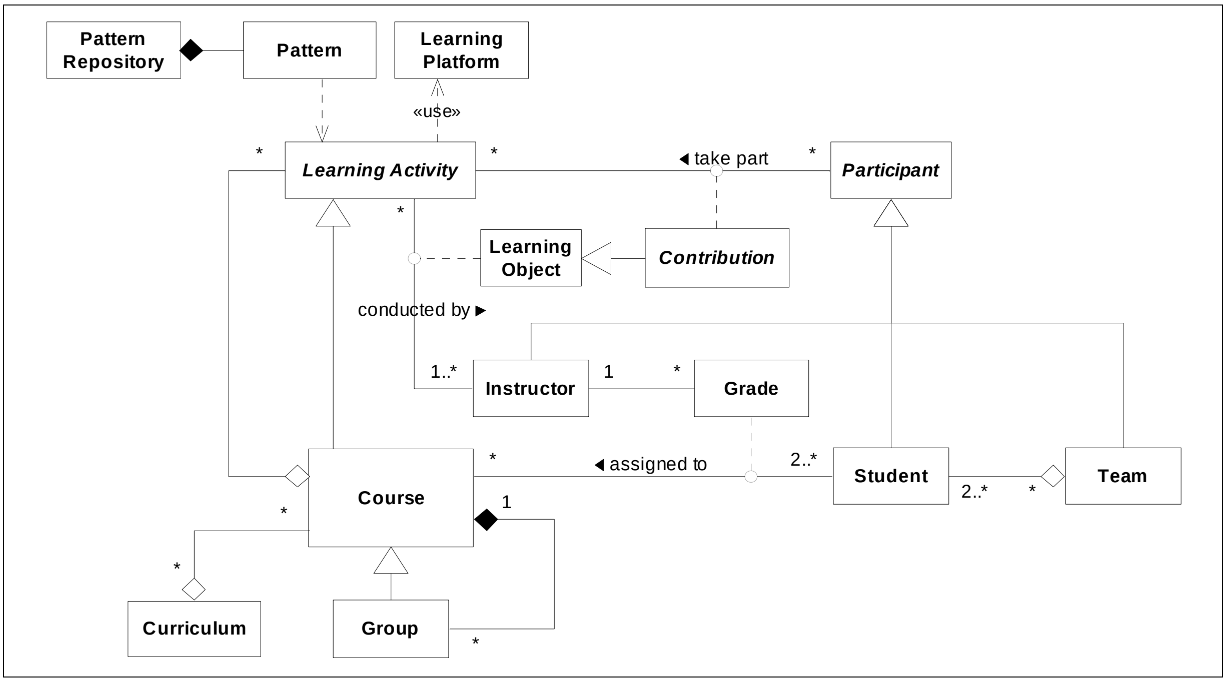


Figure 43: Structural model of key educational concepts in the scope of this work.

The structural model shows basic relationships among general entities involved in *courses*. In many educational environments, courses are divided into a number of *groups*. A course consists of a number of *learning activities*, while it itself is a certain form of a (compound) learning activity. At least one *instructor* is associated with a learning activity. Additionally, at least two *students* participate in a course. Each instance of this relationship has an associated *grade*, which is assigned by the instructor. At least two students may be joined to form one *team*. Usually, teams consist of 2 to 5 members, depending on the respective learning activity. Within the scope of this work, participating students, tutors, as well as teams are considered as *participants* of the course; thus, a participant is an abstract concept which is often used here to describe someone who takes part in learning activities and courses, regardless of his or her actual role. Participants elaborate, deliver, or perform a number of *contributions* during taking part in a learning activity.

The structural model of the COURSE pattern was extended by the following entities that create a more complete view on the educational model underlying the pattern repository: The *pattern repository* comprises a number of *patterns* and each pattern describes some *learning activity*. A learning activity may proceed on, or use, a *learning platform*. In the scope of conducting a learning activity, the instructor provides a number of *learning objects*. A learning object is the more general concept of a *contribution* (which is elaborated by participants when taking part in a learning activity.) Finally, a *curriculum* is constituted by number of courses.

3.5 Courses Underlying the Pattern Repository

This Section presents the courses from which the patterns in the PCeL pattern repository were mined. Unless otherwise indicated, each pattern's sequence of activities[429] is a generalization of activities from one or more courses. To explicate this relation, each pattern describes in its *Example* section[430] in which courses it was previously applied. Note that in the following, the repository's primary underlying course (i.e., "Web Engineering") is discussed in full detail beginning from a very low level of abstraction (=high level of detail), while the subsequent courses are handled in a more succinct, aggregate style.

It should also be mentioned that *not* every series of activities referring to some pattern in the following course diagrams formally constitutes an instance of the respective pattern's activity sequence. This is due to the fact that modeling the diagrams at such a fine-grained level of detail would exceed the space available on paper and would inadequately bloat the diagrams. Note that, anyway, this applies only to a few patterns where formal correctness was traded for a constant level of detail. However, each pattern instance is represented by at least one activity indicating the pattern's use.

3.5.1 Web Engineering

General Information

The Web Engineering module is part of the *Business Informatics* Bachelor's degree curriculum[431] of the Faculty of Computer Science at the University of Vienna. It is held in each academic year's summer term, for the first time in 2003. The discussion in this Section relies on the course sequence of the 2004 instance, because the technological capabilities of the learning platform that was employed in 2004 were by far more sophisticated than the year before. Nevertheless, the basic blueprint underlying the course was almost the same. The curriculum prescribes that the 4 hours[432] of the Web Engineering module are to be divided into 2 hours lectures and 2 hours lab practice per week. Learning target and content of the module are defined in the curriculum as follows:

- *Learning target*: Methods and processes for planning, modeling, and developing Web Information Systems.
- *Content*:
  - Planning of Web Information Systems (e.g. in categories Informational Systems, Interactive Systems, Transactional Systems, Workflow Systems, etc.)

[429] See Section 3.3.4, p. 109

[430] See Section 3.3.9, p. 125

[431] See http://winf.at/files/studienplan-winf-2001.pdf

[432] In the following, *hour* is used as short form for *academic hour*, which lasts 45 Minutes.

- Modeling of Web Information Systems at different system layers (e.g., content layer, structural or navigational layer, presentation layer, etc.)
- Implementation of Web Information Systems

Structure

As sketched in Figure 44, the Web Engineering module includes three major threads: Lab practice, lectures, and the WELL[433] (*Web Engineering Learning Contracts*) project. The module's overall activity diagram is presented in Figure 45. Note that the diagrams employ the advanced presence type stereotypes for activities (`P` for presence activities, `B` for blended activities, `W` for Web-based activities)[434].

- The *lectures* serve as means for presenting relevant subject matter and theory (see Figure 48)
- The *lab practice* thread aims at applying the theories presented in the lectures in the scope of small team projects (see Figure 46). The projects are completed in three consecutive milestones. Each milestone follows the activities given in Figure 47.
- The *WELL project* (Figure 49) serves as an alternative for participants to engage in constructive teamwork rather than passing the written lecture examination.

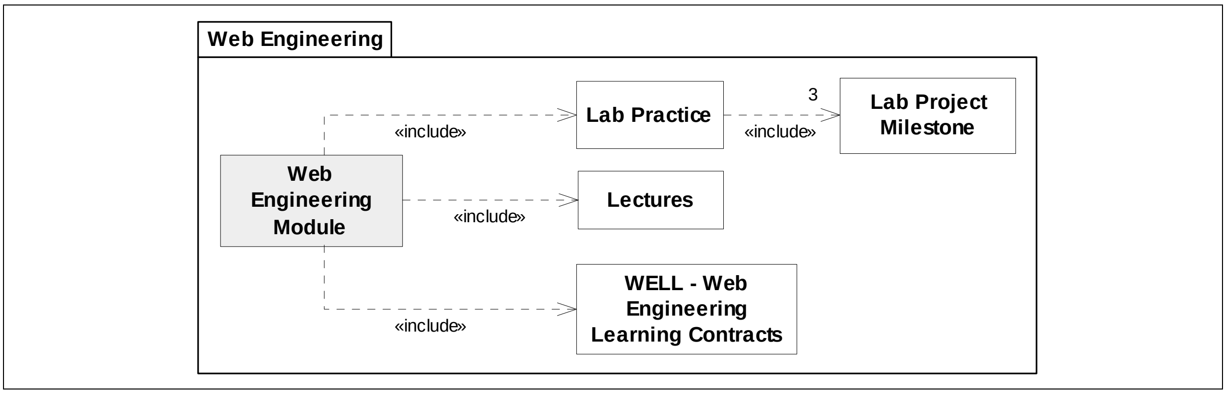


Figure 44: The Web Engineering diagram package.

433 German acronym for "Web Engineering Lern-Leistungsvertrag"

434 See a detailed description of these stereotypes in Section 3.3.4 on p. 110.

Activity Model

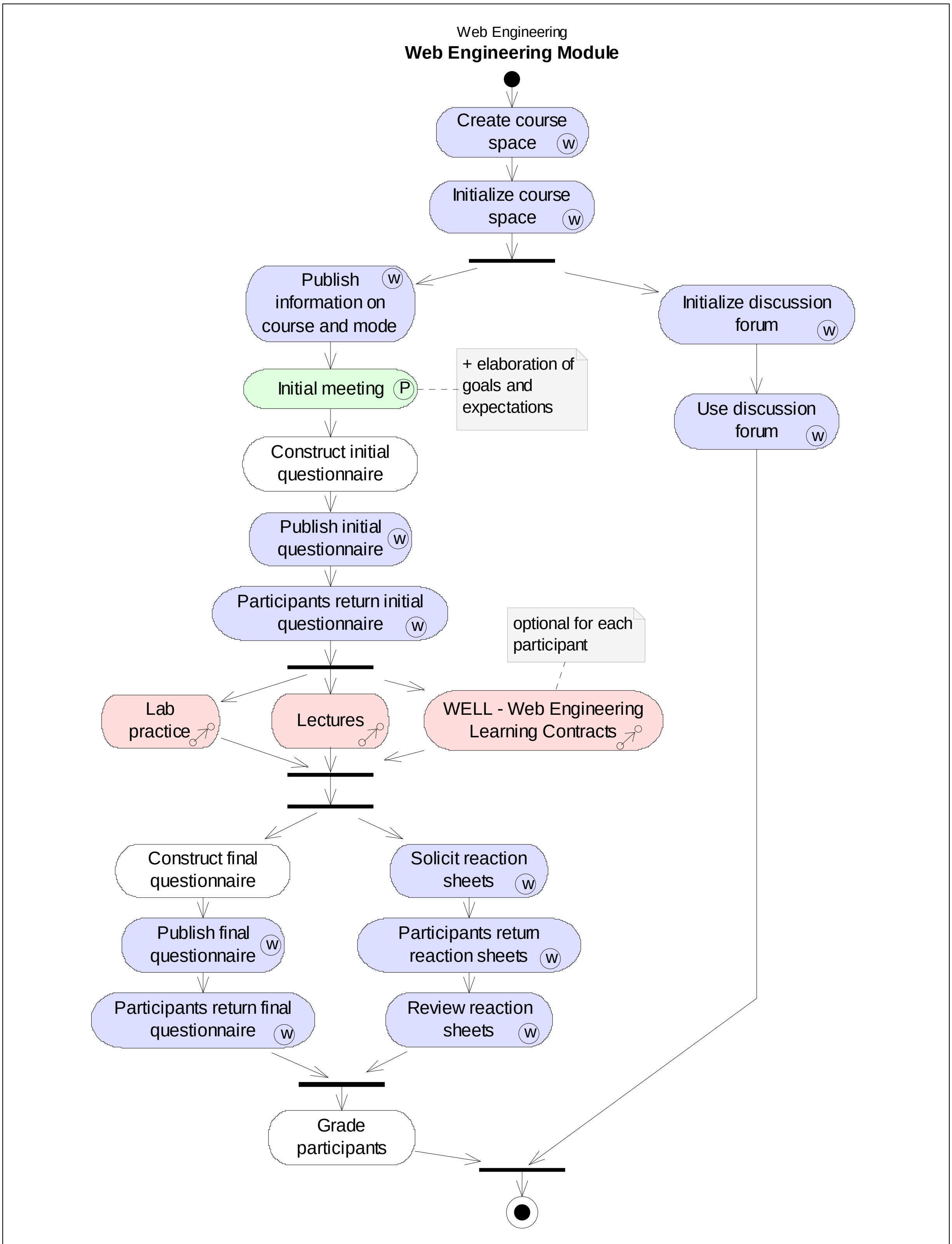


Figure 45: Top-level overview activity diagram of the Web Engineering module.

Note that three compound activities in the diagram link to more detailed diagrams: *Lab practice* is detailed in Figure 46, *Lectures* in Figure 48, and *WELL – Web Engineering Learning Contracts* in Figure 49.

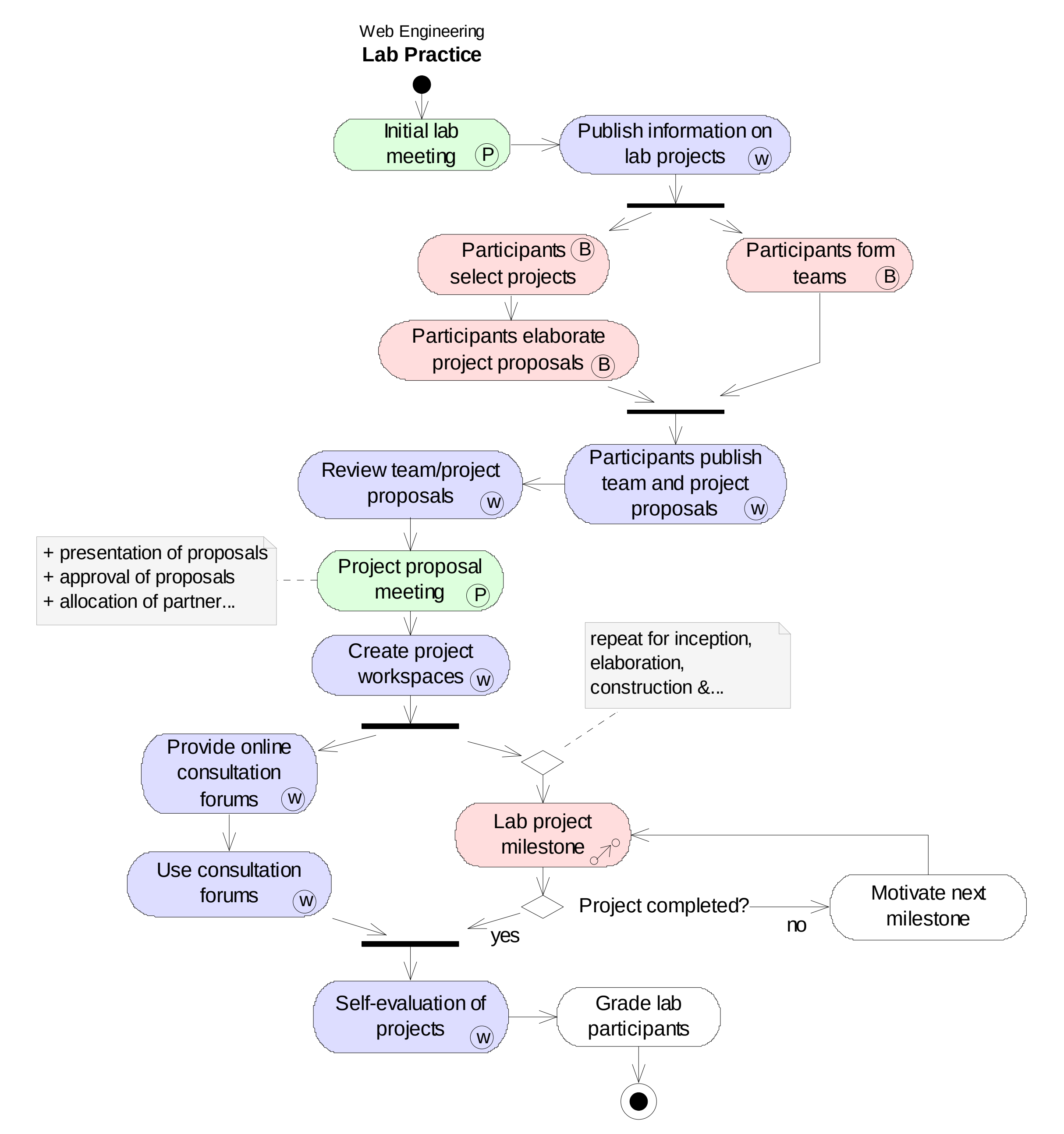


Figure 46: Lab practice activities in the main courses phases of the Web Engineering module.

This diagram is a detailed model the compound activity *Lab practice* in Figure 45. Note that the diagram includes the compound activity *Lab project milestone* that links to detailed models of each milestone in Figure 47.

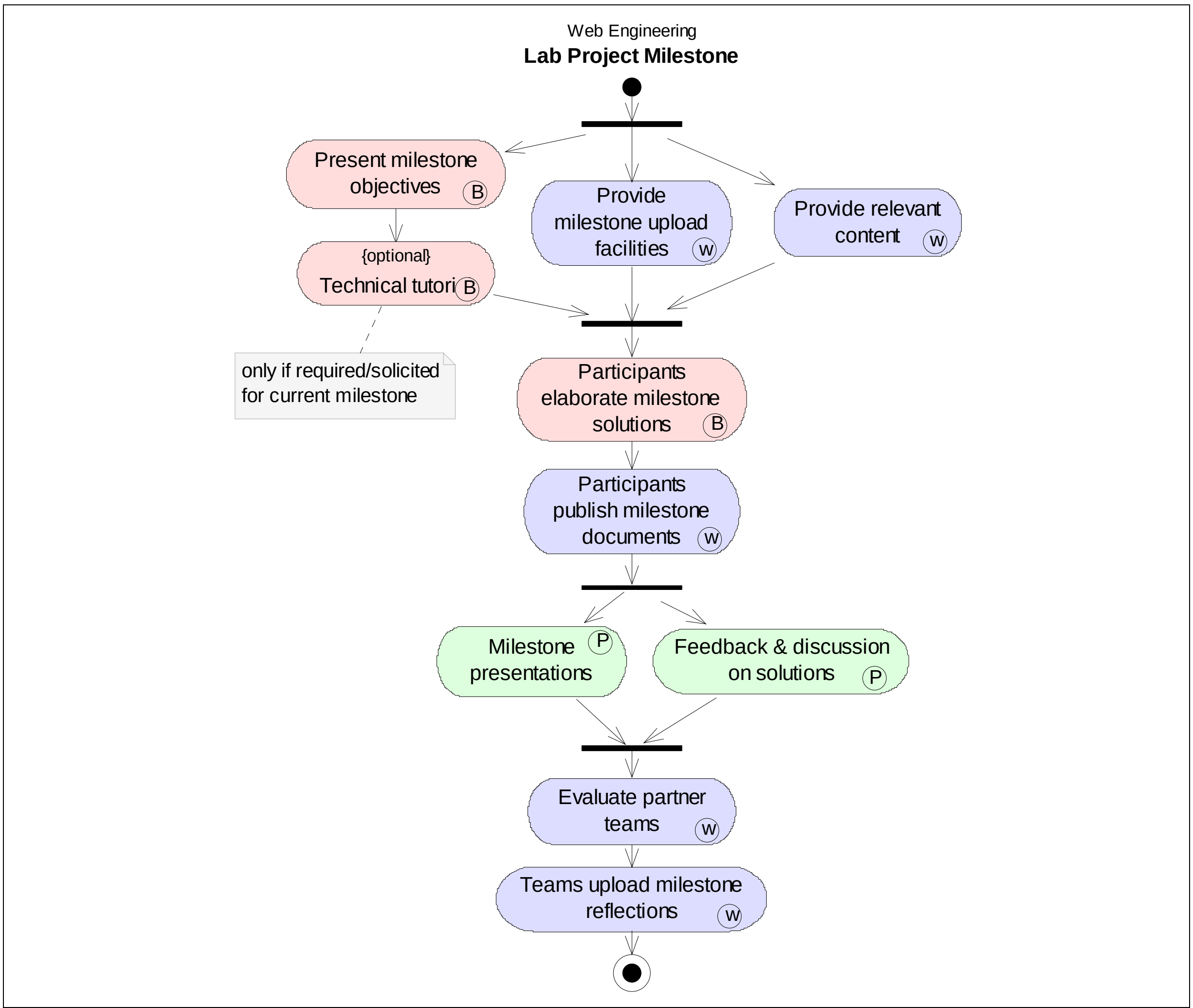


Figure 47: Activities for each of the three milestones in the Web Engineering lab projects.

This diagram is a detailed model of the compound activity *Lab project milestone* in Figure 46.

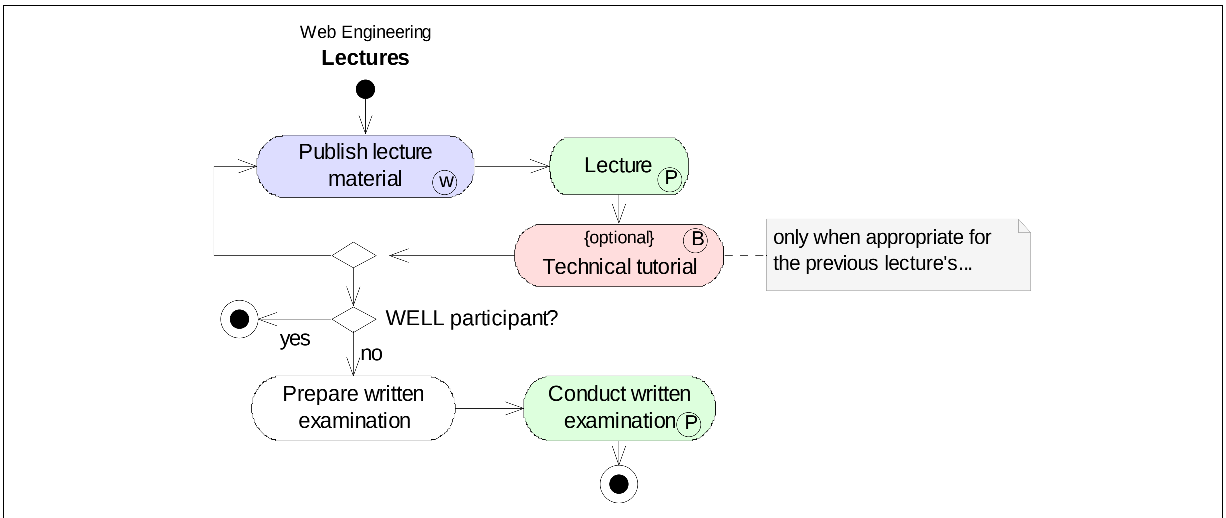


Figure 48: Activities in the lecture thread of the Web Engineering module.

This diagram is a detailed model of the compound activity *Lectures* in Figure 45.

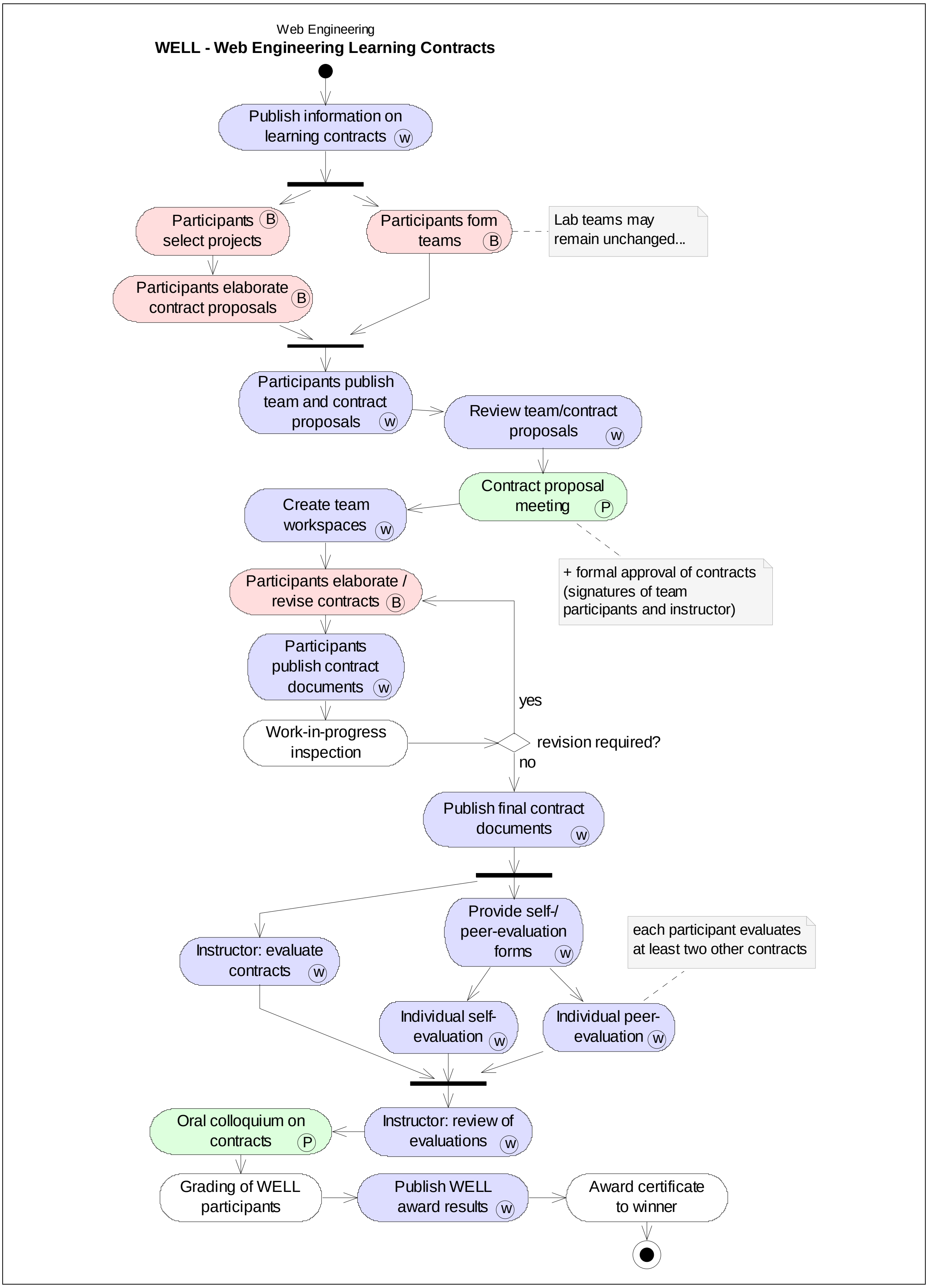


Figure 49: Web Engineering learning contracts (WELL) activities.
This diagram is a detailed model of the compound activity *WELL – Web Engineering Learning Contracts* in Figure 45.

Patterns Identified

The following list shows which patterns have been identified from the activity diagrams of the Web Engineering module.[435] Note that items (1) – (15) implicitly refer to Figure 45, unless otherwise explicitly indicated.

(1) Course. The whole Web Engineering module is an instance of the abstract COURSE pattern. The PRELIMINARY PHASES (2) of COURSE are covered from *Initialize course space* to *Initial meeting*. The following three activities (*Construct initial questionnaire* to *Participants return initial questionnaire*) map to the QUESTIONNAIRE (12) pattern activity in COURSE. The following *Main course phases* of COURSE are covered by *Lab practice*, *Lecture*, and *WELL* activities. The remaining activities are part of the ASSESSMENT PHASES (15) of COURSE.

(2) Preliminary Phases. This pattern, which is part of any COURSE (1) instance, covers the activities from *Initialize course space* to *Initial meeting*.

(3) Alternating Phases. This pattern is the base pattern of PRELIMINARY PHASES (2). It shows online phases alternating with face-to-face MEETINGS and thus constitutes the primary arrangement principle for blended learning activities. Many other patterns also derive from ALTERNATING PHASES, such as GENERIC EVALUATION (32), COLLECT FEEDBACK (14), PROJECT MILESTONE (22), and others that do not occur in the Web Engineering module.

(4) Publish. This most frequently employed pattern occurs first at *Publish information on course and mode*. Almost any Web-based activity includes some kind of publishing process. For this reason, more instances of PUBLISH are not reported explicitly here, although it occurs in all Web Engineering activity diagrams.

(5) Initial Meeting. Even if this pattern is covered only by one activity in the Web Engineering module, it was identified as a pattern for a simple reason: The initial meeting is a crucial point in a learning activity, constituting the first physical meeting between instructor(s) and participants. Additionally, it is part of the PRELIMINARY PHASES (2) of each COURSE (1).

(6) Meeting. A more generalized form of meeting than INITIAL MEETING (5). It can also be found as *Initial lab meeting* or *Project proposal meeting* in the Lab Practice diagram (Figure 46).

(7) Interactive Element. This is an even more generalized activity pattern, acting as the base pattern for MEETING (6). Its only assumption is interaction between two or more actors in a course or learning activity. It is the abstract base pattern for more than a dozen other concrete patterns in the *Interactive Elements*[436] pattern package.

[435] Note that the order of the patterns listed originates from using a *depth first* search paradigm.

[436] See the *Interactive Elements* package description on p. 103

(8) Elaborate Goals and Expectations. This pattern is included in each INITIAL MEETING (5), and is not modeled explicitly in the diagram (it is depicted as a note linked to the *Initial meeting* activity).

(9) Information Gathering. This is the abstract base pattern of ELABORATE GOALS AND EXPECTATIONS (8). Other concrete INFORMATION GATHERING sub-patterns include CONSULTATION (21), BRAINSTORMING (37), and THEORY ELABORATION (38).

(10) Online Discussion. The concurrent discussion stream including the *Initialize discussion forum* and *Use discussion forum* activities constitute an instance of the ONLINE DISCUSSION pattern.

(11) Computer-Mediated Communication (CMC). This pattern is a generalized, abstract form of ONLINE DISCUSSION (10). It is designed to subsume various known forms of CMC, as found in current e-learning literature and practice. In the current version of the PCeL pattern repository, this includes CHAT (not yet used in Web Engineering) and ONLINE DISCUSSION.

(12) Questionnaire. This pattern summarizes the three activities following the *Initial meeting*, from *Construct initial questionnaire* to *Participants return initial questionnaire*; and the final questionnaire at the end of the Web Engineering module, as part of the COLLECT FEEDBACK (14) pattern within the ASSESSMENT PHASES (15).

(13) Reaction Sheets. The three activities arranged concurrently to the final QUESTIONNAIRE activities (*Solicit reaction sheets* to *Review reaction sheets*) are abstracted in the REACTION SHEETS pattern.

(14) Collect Feedback. This pattern includes a generalization of QUESTIONNAIRE (12) and REACTION SHEETS (13) in the actual *feedback phase*, and describes the basic intent of collecting various forms of feedback. In addition to the two forms already mentioned, the PCeL repository also hosts FEEDBACK FORUM (not used in Web Engineering, but in the PhD Seminar[437]) as a third form of COLLECT FEEDBACK. Note that this pattern is an optional component of the ASSESSMENT PHASES (15) pattern. It also occurs at the end of the *Lab Project Milestone* diagram in Figure 47.

(15) Assessment Phases. This pattern is arranged in the final phases of each COURSE (1). It subsumes COLLECTING FEEDBACK (14), evaluating participants' contributions through various forms of EVALUATION (24), and finally grading participants. It also occurs in the *WELL contracts* thread depicted in Figure 49, after the *Publish final contract documents* activity, ending with *Grading of WELL participants* by the instructor.

The following items (16) – (24), unless otherwise indicated, refer to the *Lab Practice* diagram in Figure 46.

(16) Proposal. In the *Lab Practice* diagram, two instances of PROPOSAL occur: The first one for project proposals or, more generally, PROBLEM PROPOSALS (17), the second one

[437] See Section 3.5.4, p. 145

for team proposals in the course of TEAM BUILDING (18).[438] Beginning with *Publish information on lab projects*, which aims to set out the proposal guidelines, and concluding with an instance of the APPROVAL (19) pattern beginning at *Review team/project proposals*. PROPOSAL also occurs for the WELL contracts in Figure 49, from *Publish information on learning contracts* to *Contract proposal meeting*.

(17) Problem Proposals. This pattern is derived from PROPOSAL (16), and begins at the *Publish information on lab projects* activity and concludes with *Project proposal meeting*, the final activity of the included APPROVAL (19) instance. PROBLEM PROPOSALS also occur in Figure 49 as a specialized form of PROPOSAL.

(18) Team Building. This pattern is also derived from PROPOSAL (16), and begins and ends at the same activities as PROBLEM PROPOSALS (17). As the problem/project proposals and team building for projects belong together logically, their activities have been merged in the diagram. TEAM BUILDING also occurs in Figure 49 as a specialized form of PROPOSAL.

(19) Approval. This pattern concludes every PROPOSAL (16) scenario. Participant's proposals for projects (17) and teams (18) are reviewed, and approved in the *Project proposal meeting* if the published requirements were met. Another instance of APPROVAL occurs in Figure 49 for WELL contract and team proposals, beginning with *Review team/contract proposals* and ending with the *Contract proposal meeting*.

(20) Team Workspaces. This pattern is covered through approval of the team proposals in TEAM BUILDING (18) and the *Create project workspaces* activity. It also occurs in the *WELL contracts* thread following the *Contract proposal meeting* in Figure 49.

(21) Consultation. This pattern is covered through the *Provide online consultation forums* and *Use consultation forums* activities accompanying the PROJECT MILESTONES (22). Participants are provided with facilities for consulting their instructor through ONLINE DISCUSSION (10) forums.

(22) Project Milestone. This pattern is referred to by the *Lab project milestone* compound activity, and is detailed in Figure 47. It is the core pattern within each PROJECT-BASED LEARNING (25) thread of a course. As such it also occurs in the *WELL contracts* diagram (Figure 49) following the TEAM WORKSPACES (20) instance. The final contract milestone there is concluded by the *Publish final contract documents* activity.

(23) Self-Evaluation. The *Self-evaluation of projects* activity is a condensed representation of the SELF-EVALUATION pattern, where participants evaluate their own contributions. This pattern also occurs in Figure 49 in the form of an *Individual self-evaluation* of WELL contracts.

[438] These two PROPOSAL instances could also be considered one single, combined instance for projects and project teams.

(24) Evaluation. This pattern is a more generalized form of the SELF-EVALUATION (23) pattern as well as of a number of other patterns aiming at valuing judgment of contributions, such as PEER-EVALUATION (27), INSTRUCTOR-EVALUATION (30), and EXAMINATION (29) along with its derivates[439].

(25) Project-Based Learning. The whole *Lab practice* thread of the Web Engineering module follows the PROJECT-BASED LEARNING pattern, including PUBLISHING (4) of relevant information, TEAM BUILDING (18) and PROJECT PROPOSALS (17), and a number of PROJECT MILESTONES (22). Also, the *WELL contracts* thread in Figure 49 constitutes an instance of PROJECT-BASED LEARNING in the form of LEARNING CONTACTS (34).

The following items (26) – (27) refer to the *Lab Project Milestone* diagram Figure 47, unless otherwise indicated.

(26) Tutorial. The *Lab Project Milestone* diagram includes TUTORIAL, depicted in condensed form through the optional *Technical tutorial* activity. The same applies to the *Lectures* diagram in Figure 48. Activities for preparing/selecting material and publishing of tutorial details were omitted in these diagrams for the sake of clarity.

(27) Peer-Evaluation. This pattern is manifested in the *Evaluate partner teams* activity, where assigned partner teams evaluate each other's milestone contributions online. PEER-EVALUATION also occurs in Figure 49 in the form of Web-based *Individual peer-evaluation* of WELL contracts.

The following items (28) – (29) refer to the *Lectures* thread Figure 48, unless otherwise indicated.

(28) Instructor-Examination. This pattern occurs as *Prepare* and *Conduct written examination* at the end of the *Lectures* thread in Web Engineering, as well as in the form of *Oral colloquium on contracts* in the *WELL contracts* thread in Figure 49.

(29) Examination. This pattern is a more generalized form of INSTRUCTOR-EXAMINATION (28), constituting the base pattern also for SELF-EXAMINATION, which was not used in Web Engineering.

The following items (30) – (34) refer to the *WELL contracts* diagram in Figure 49, unless otherwise indicated.

(30) Instructor-Evaluation. This specialized form of EVALUATION (24) occurs as *Instructor: evaluate contracts.*

(31) Blended Evaluation. This pattern includes INSTRUCTOR-EVALUATION (30), SELF-EVALUATION (23) and PEER-EVALUATION (27) of the WELL contracts succeeding the *Publish final contract documents* activity, ending with *review of evaluations* by the instructor.

---

[439] For more information on the genesis of the EVALUATION pattern hierarchy see footnote #397, p. 99.

(32) Generic Evaluation. This pattern is the parent pattern of BLENDED EVALUATION (31), not specifying the particular number and kinds of EVALUATION (24) actually taking place.

(33) Achievement Award. This pattern is covered by the final two activities in the *WELL contracts* thread. Note that the award criteria were implicitly set by the assignment of bonus points in the scope of the PEER-EVALUATION (27) of contracts. The provision of candidate contributions is completed implicitly through *Publish final contract documents*.

(34) Learning Contracts. The whole WELL contracts diagram constitutes an instance of the LEARNING CONTRACTS pattern.

Finally, the list is concluded by patterns which are not explicitly modeled in the Web Engineering diagrams:

(35) Diary. Online diaries for teams/participants were not used consistently by the Web Engineering instructors. Some used it not at all, while others employed it for the whole course, or for the *WELL contracts*, and/or for the *Lab practice*.

(36) Staff Meeting. Staff meetings were employed periodically throughout the course to discuss current issues. Thus, these meetings were scheduled on demand.

(37) Brainstorming. Was used in many situations, primarily in the *Lectures* to increase active participation, and also frequently for project work in the *Lab practice* sessions.

(38) Theory Elaboration. See BRAINSTORMING (37).

3.5.2 Person-Centered Communication

Note: This course (and the subsequent ones as well) is depicted in less detail than the Web Engineering course.

General Information

The Person-Centered Communication course is part of the Business Informatics Master's degree curriculum at the University of Vienna, and is held every semester. Enrolling in the course is optional for students in the context of the advanced module "*Wirtschaftsinformatik*." Its core structure includes three workshops (about 4 hours each) for presentation of relevant theory and material as well as six encounter group[440] sessions (about 6 hours each).

- *Global learning targets*:
  - Furthering the transparent and acceptant development of participants towards facilitative interpersonal relationships and dispositions, improvement of the general

[440] See the Section on methods of building freedom on p. 27

problem-solving competence, as well as comprehensive, holistic perception of situations, based on the principles of the Person-Centered Approach.
  - Communication and mutual understanding
- *Content*:
  - Three Rogers' variables: congruence, acceptance, and empathic understanding
  - Active listening
  - Development direction in the Person-Centered Approach
  - Self-structure and role of experience
  - Significant learning and learning on three levels
  - Theory of encounter groups and encounter group processes; basics of facilitation

Activity Model

The activity model in Figure 50 is based on the course's activity sequence from winter term 2003. The naming of activities lends more strongly from the names of patterns in the repository. The diagram is a more condensed representation of the courses activities than in the Web Engineering diagrams.

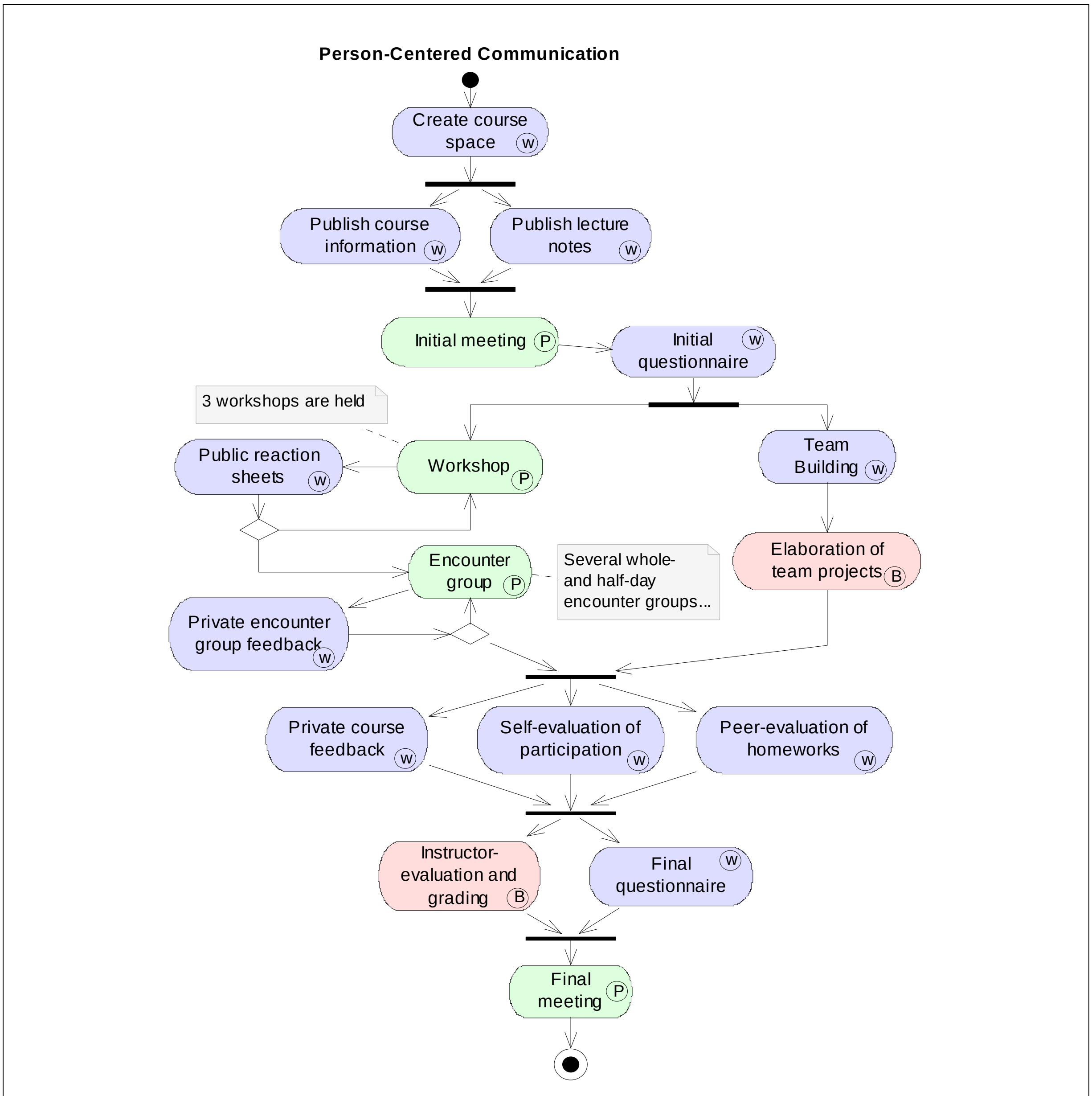


Figure 50: The Person-Centered Communication course.

Patterns Identified

Most of the *Person-Centered Communication* course can be described with patterns already presented in the scope of the Web Engineering module: The course starts with PRELIMINARY PHASES concluding with an INITIAL MEETING. Following the initial QUESTIONNAIRE are the main phases of the course which consists of a series of workshops and encounter groups with a concurrent thread in which participants elaborate their contributions (papers on theory related to the course) in teams. Each workshop and encounter group is followed by private feedback to the instructors via online REACTION SHEETS. In the ASSESSMENT PHASES, feedback on the whole course experience is collected again through REACTION SHEETS and a final QUESTIONNAIRE. BLENDED EVALUATION of participation and homework is used to assess final grades. In a final MEETING between participants and instructor(s), the whole course and the BLENDED EVALUATION results are discussed. The only new pattern introduced here is **WORKSHOP**, as a specialized form of MEETING, where theory is presented and elaborated interactively.

3.5.3 Project Management

General Information

Project Management (PM) is a *combination of courses* ("*KFK – Kernfachkombination*") in the Business Informatics Master's program at the University of Vienna[441]. For completion of the master's degree, taking that particular combination of courses is optional. Anyway, it consists of **4 course modules**, each including 4 hours of combined lectures and practice:

1) ***PM/Basics and Techniques***: Students acquire better understanding of their role in the realization of ICT projects; they get to know basic technological, organizational, management-specific, social, psychological, and personal factors influencing the project work within organizations; they acquire skills in effectively working on projects in teams, learn how to identify, define, seek solutions, choose, plan, execute, and guide, as well to reflect on problems, solutions and processes.
2) ***PM/Soft Skills***: Gathering of personal experiences in typical project situations such as presentations, teamwork, meetings, counseling. The content includes theory on interpersonal communication, conflict management, negotiation, doing presentations, moderation, and rhetoric.
3) ***PM/Business Processes and Organizational Development***
4) ***PM/Human Factors in the Context of ICT***

The former two, which are held every winter term, are conducted in a style following Person-Centered e-Learning. The latter two, which are held every summer term, are conducted in a conventional style, usually by external lecturers.

[441] For a detailed (German) description of the KFK see http://www.cs.univie.ac.at/institute/index.html?subject-2=2 (accessed January 25, 2005)

Activity Model: *Basics and Techniques*

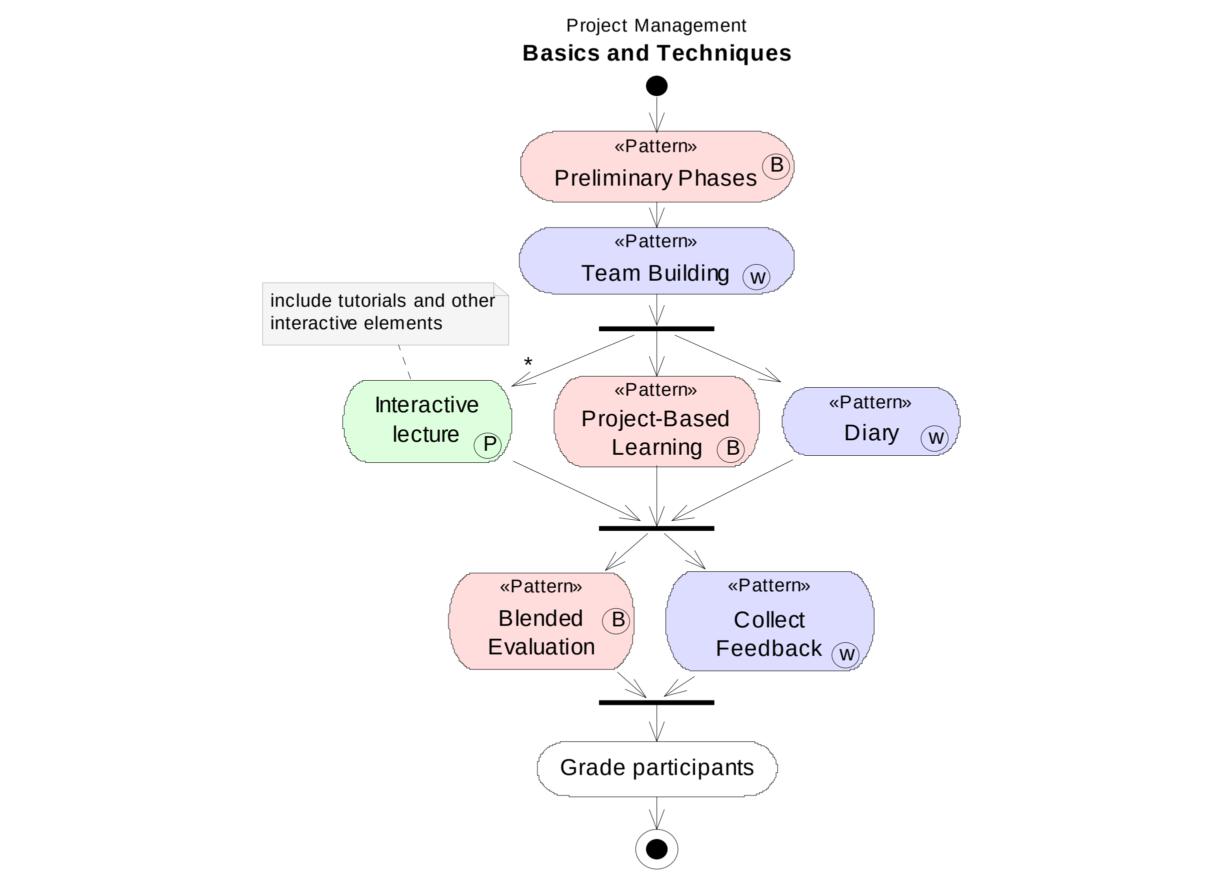


Figure 51: Basics and Techniques course in the Project Management module.

As evident from Figure 51, the *PM/Basics and Techniques* course can be almost completely modeled by using patterns already presented in the scope of the Web Engineering module. Even though the concrete course is quite complex, the diagram nicely shows how even complex flows of activities can be clearly represented and arranged at an aggregate level by using patterns from the repository:

In the PRELIMINARY PHASES, the course platform is initialized with relevant information, followed by ELABORATION OF GOALS AND EXPECTATIONS in an INITIAL MEETING, where also the course concept is presented and project topics are elaborated and presented. Following online TEAM BUILDING, the main course phases mainly consist of PROJECT-BASED LEARNING, accompanied by interactive lectures that may include INTERACTIVE ELEMENTS such as TUTORIALS on tools and techniques as required, and by online project DIARIES, where participants/teams keep records of their process and activities. The PROJECT-BASED LEARNING thread is a mix of theory and content input, self-study, presentation of milestones and projects, discussions, as well as exchange with partner teams. The course finishes with ASSESSMENT PHASES including COLLECT FEEDBACK (in the form of REACTION SHEETS and QUESTIONNAIRE) and BLENDED EVALUATION of projects.

Activity Model: *Soft Skills*

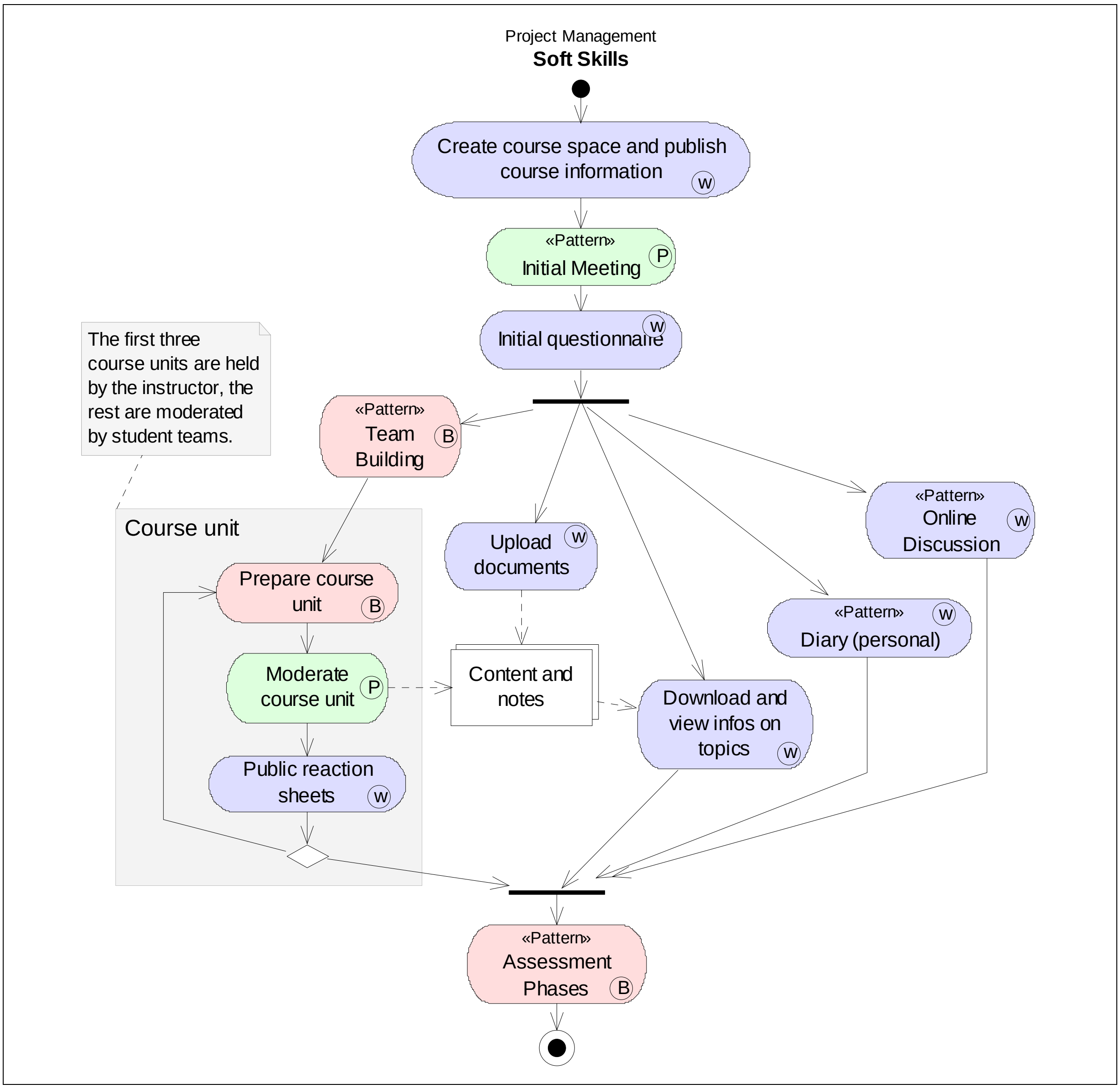


Figure 52: Soft Skills course in the Project Management module.

Note that the diagram additionally utilizes the "object flow" notation[442], which shows how objects are input to and output from activities. In the current case, the *Content and notes* object represents material and documents that are uploaded in the *Upload documents* activity and produced in the scope of the moderated course units. All documents can subsequently be viewed and downloaded online.

The course diagram in Figure 52 mixes the use of patterns with "normal" activities. The activity flow follows the typical arrangement of phases presented in COURSE, starting with PRELIMINARY PHASES that are concluded by an INITIAL MEETING, and followed by an initial QUESTIONNAIRE to be returned by participants. The main course phases consist of a series of

[442] OMG (2003, p. 3-163)

course units moderated initially by the instructor as well as subsequently by small teams of participants (prior TEAM BUILDING). The teams prepare the moderation of their chosen course unit and PUBLISH all relevant documents on a public space on the learning platform. After each moderated course unit the online material is complemented, and online surveys on issues that arose as well as public REACTION SHEETS are collected. Participants can view and download all online material prior to and after the course units. These moderated course units are accompanied by an online, private DIARY for participants, as well as by a general ONLINE DISCUSSION forum. The course is concluded as usual by an ASSESSMENT PHASES arrangement including BLENDED EVALUATION and COLLECTING FEEDBACK through a final QUESTIONNAIRE and REACTION SHEETS.

With respect to the pattern repository, one new pattern can be identified: The provision and downloading of documents in an online space was generalized in the **MARKET** pattern, as a specialized form of INFORMATION GATHERING.

3.5.4 PhD Seminars

General Information

The computer science PhD study at the Faculty of Computer Science includes three different kinds of seminars, each at 2 hours per week (usually blocked, and held every semester):

- ***Research Seminar***: Participants present the topics and progress of their PhD theses[443]. Typically, this includes an initial meeting where the topics are discussed and a final presentation meeting where each participant presents his or her topic.
- ***Literature Seminar***: Each participant elaborates some topic which may be out of the scope of his or her own PhD thesis. The elaboration is done in the form of a literature study and a written report, which is presented in the final presentation meeting. The thematic context of the seminar is set by the instructor and is usually located within the instructor's current field of research.
- ***Methods Seminar***: Typically, participants elaborate write reports regarding research methodology in general (e.g., writing scientific papers, using digital libraries for literature review, etc.) or methods specific to their particular PhD research. There is usually a final presentation meeting where each participant presents his or her report.

While the typical seminar at the University of Vienna is held in a conventional "*two-meeting*" style including one initial meeting with topic assignment and one final meeting with oral presentations of the term papers, our group employs a more interactive style based on Per-

[443] The official course description says: "*Current research questions in (Business) Informatics with respect to the PhD topics; methods of presenting scientific problems and solution approaches as well as defending them.*" (http://www.cs.univie.ac.at/institute/index.html?5_9621021_=9621021_)

son-Centered e-Learning in the seminars. The basic activity model underlying all our PhD seminars is given in Figure 53.

Activity Model

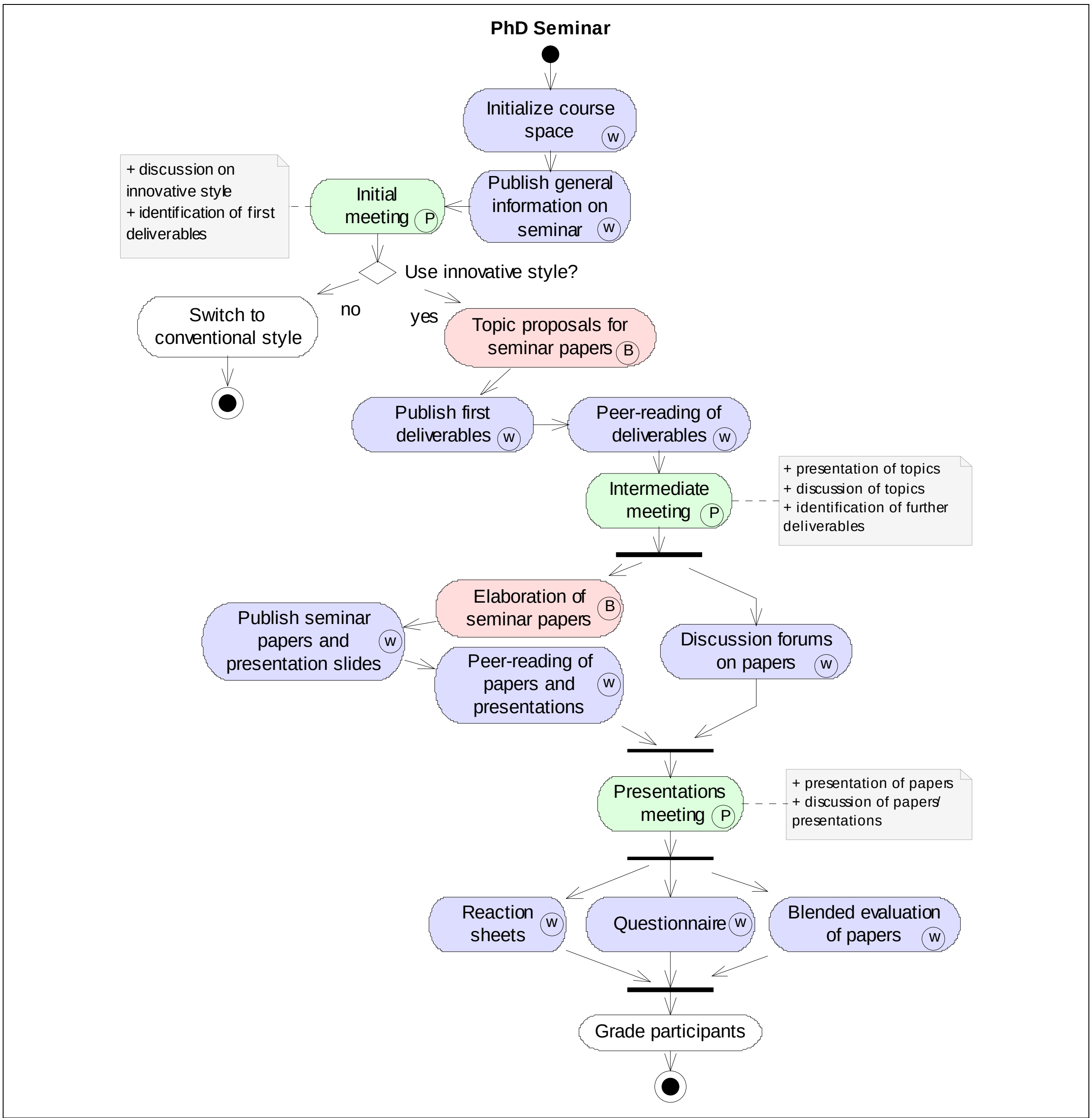


Figure 53: Activity diagram of our PhD seminars.

The flow of activities in the PhD seminars is described as follows:

- PRELIMINARY PHASES: The course space is initialized, and general information on the seminar is PUBLISHED online. The preliminary phases are concluded by an INITIAL MEETING, which includes a discussion on the innovative seminar style, collaborative ELABORATION OF GOALS AND EXPECTATIONS and setting of the thematic focus, as well as identification of first deliverables for the intermediate MEETING. Additionally, the instructor offers the participants to choose between the conventional and the innovative course style. So far, all decisions were unanimously made for the innovative style.

- The main course phases are mainly dedicated to fostering blended interaction among seminar participants:
  - After a PROPOSAL phase where participants propose their specific topics within the thematic context of the seminar, they are encouraged to PUBLISH abstracts of their PhD theses and prospective seminar report topics online. Peers are asked to view these contributions prior to the intermediate MEETING, where these issues are discussed. Usually (not in all seminars), each participant has to host one ONLINE DISCUSSION forum dedicated to his or her seminar/PhD topics. This thread of activities has been generalized in the **EXCHANGE OF CONTRIBUTIONS** pattern.
  - Contrary to conventional settings at the University of Vienna, our PhD seminars solicit short presentations of about 15 to 20 minutes followed by longer discussions. To achieve this, participants are instructed to PUBLISH their completed reports and presentation slides on the learning platform about one week prior to the presentation MEETING. This enables participants to prepare for their peers' presentations by downloading and reading their reports and presentations before the presentations are held. This thread of activities has been generalized in the **PRESENTATION PHASES** pattern.
- In the ASSESSMENT PHASES, which are conducted "as usual" including BLENDED EVALUATION and COLLECT FEEDBACK, the PhD seminar introduces a new specialization of COLLECT FEEDBACK, i.e. **FEEDBACK FORUM**, where the instructor initiates ONLINE DISCUSSION on feedback topics in dedicated discussion threads. Participants subsequently post their feedback in reply to these predefined threads. However, feedback may equally be collected through REACTION SHEETS.

The process pattern underlying all these interactive PhD seminars is defined in the **SEMINAR** pattern. It shows a slightly generalized model of the activities in Figure 53.

4 Related Approaches

This Chapter is structured as follows:

- First, ***related approaches*** in the field of (e-)learning patterns and learning design are presented (Section 4.1, p. 149).
- Subsequently, the PCeL pattern approach is ***differentiated*** from the related approaches by outlining its distinguishing key features (Section 4.2, p. 156).
- After that, the PCeL pattern approach is related to the theory of ***promotive activities*** in education by Tausch and Tausch[444] (Section 4.3, p. 158): Points of overlap / complementation in the underlying educational philosophy are elaborated.
- Finally, the support that the PCeL pattern approach can provide in the traditional ***instructional design*** theory is discussed (Section 4.4, p. 165).

4.1 Related (E-)Learning Design Approaches

Surprisingly, even though the pattern approach has found its way into many different disciplines, the field of e-learning clearly seemed to lag behind until very recently more and more projects and efforts have emerged in the field. These approaches are characterized in the following sub-Sections.

4.1.1 Pedagogical Patterns Project

The Pedagogical Patterns Project[445], which was presented in detail in Sections 2.2.1.1.3 and 2.2.2.5.5[446], provides a compilation of prose-style patterns for many educational scenarios. However, these patterns are neither tied to any pedagogical baseline, nor do they include or address explicitly the use of learning technology.

[444] Tausch and Tausch (1998)

[445] Pedagogical Patterns Project (2002)

[446] See p. 39 and p. 51, respectively

4.1.2 E-LEN Project

The E-LEN project[447] aims to create a network of e-learning centers and organizations in the learning technologies, as well as to develop and disseminate pedagogically informed technology for effective e-learning experiences in the form of design patterns. The project started in mid-2003 at a workshop of the Computer-Supported Collaborative Learning (CSCL) conference in Norway[448]. Its results are not yet completely available to the public.

At the 2004 ED-MEDIA[449] conference the E-LEN project coordinator hosted a symposium, where no complete results but initial concepts of various approaches and aspects regarding the use of design patterns in e-learning were presented:

- Integrating pedagogical approaches with hypermedia design patterns[450] and providing a taxonomy of design patterns of adaptive/adaptable hypermedia[451].
- Implementation of wizards that guide e-learning system administrators in configuring the instantiation of pedagogical patterns (e.g., discussion patterns, drill-and-practice patterns, etc.).[452]
- Focusing on the organizational view on e-learning by defining organizational patterns for developing and implementing e-learning centers within institutions.[453]
- Theoretical investigation of deductive and inductive pattern mining and identification in the domain of collaborative learning and CSCL.[454]

4.1.3 E^{2}ML – The Educational Environment Modeling Language

E^{2}ML[455] is a novel visual language for supporting the design of educational environments. It allows producing a comprehensive documentation of the instructional design process by defining learning goals and modeling action diagrams and overview diagrams of the whole design. The language does not use a standardized notation system, nor does it rely on any learning theory or didactic baseline. Its central aims are stated as follows:

---

447 E-LEN Project (2003), http://www.tisip.no/E-LEN; see also Section 2.2.2.5.6, p. 54

448 See the “Design Patterns for CSCL” Web page at http://www.intermedia.uib.no/cscl/workshop/workshop11.cscl

449 The 2004 ED-MEDIA (World Conference on Educational Multimedia, Hypermedia & Telecommunications) took place during June 21-26, in Lugano, Switzerland (see http://www.aace.org/conf/edmedia).

450 Garzotto et al. (2004)

451 Cristea and Garzotto (2004)

452 Kolas and Staupe (2004)

453 Steeples and Zenios (2004)

454 Baggetun, Rusman and Poggi (2004)

455 Botturi (2003)

- Facilitating communication among all involved in the educational design process.
- Supporting the derivation of requirements that have to be met by e-learning tools.
- Supporting quality assessment during the setup stage of an e-learning environment, thus providing decision support for management decisions.
- Providing a diagnostic toolkit for supporting redesign of existing educational environments.

The primary modeling elements in E^{2}ML are actions and dependencies. An action (e.g., a lecture or some online activity) is represented by a rectangle including written statements that are organized in the following spots (cf. Figure 54):

- *Action identification*: name, roles, identifier, and duration
- *Initial state*: pre-requirements, preconditions, and input
- *Final state*: outcome, side-effect, and output
- *Action performance*: procedures, locations, and tools.

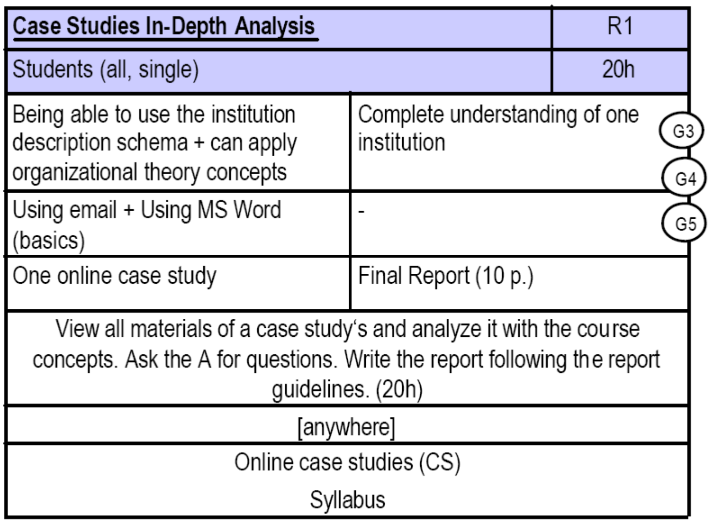


Figure 54: Example of an E²ML action "Case Studies In-Depth Analysis."[456]

An action diagram represents a static view that allows connecting actions through aggregation and inheritance relationships. Through the former, actions may be conceptually composed of sub-actions, while through the latter certain actions can be derived from more or less abstract action templates. Additionally, E^{2}ML provides a method of modeling the timeline of a course, which produces a Gantt-chart-like visualization of the "action flow."

4.1.4 CSCL Scripts

CSCL scripts formally describe collaborative learning scenarios that students and tutors have to play like actors play a movie script[457]. The scripts are transparent to tutors and learners.

[456] Source: Botturi (2003, p. 308)

[457] Cf. Dillenbourg (2002)

The approach is highly formalized regarding both syntax and semantics (inside as well as among scripts). Through this kind "programmed collaboration," there seems to be not much space for flexibility on the side of the learners and facilitators.

4.1.5 The Conversational Framework

Laurillard's book, "Rethinking University Teaching,"[458] is one of the most cited among current e-learning publications. It presents a framework that is centered on the dialog between instructor and learner (*conversational framework*) and that shall support the course designer in selecting and using new media (such as hypermedia, audio-visual media, interactive media, etc.) for teaching/learning processes. The main commonality to the PCeL pattern approach is the aim to make situated use of learning technology. One of the main differences is that it does not describe learning scenarios in a uniform, visualized manner, which particularly supports identifying targeted uses of learning technology with respect to the activity flow in the learning scenarios. Additionally, the conversational framework theory is based on the presumption that the instructor has to take the main responsibility in setting learning goals and for how the students learn[459]. This is definitely not a central presumption in Person-Centered Learning, which is based more on self-responsibility and self-pacing of learners as well as self-organization, participation, and collaboration.

Within the conversational framework (depicted conceptually in Figure 55), there are 12 processes defined between the instructor's conceptions of a theory, a corresponding learner's conception, the learning environment constructed by the instructor, and the learner's actions to adapt his/her conception:

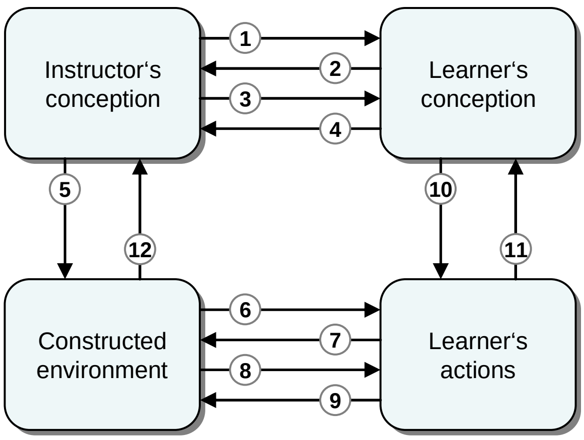


Figure 55: Conceptual model of the Conversational Framework.

1) Instructor describes conception

[458] Laurillard (2001) – The first edition of this book was released in 1993.

[459] Cf. Young, Foulkes and Thomas (2004)

2) Learner describes conception
3) Instructor clarifies conception in light of learner's conception
4) Learner re-describes conception
5) Instructor tailors learning tasks to address gaps in learner's conception
6) Instructor sets clear learning goals
7) Learner attempts to meet goals
8) Instructor provides feedback
9) Learner adapts to feedback
10) Learner adapts conception
11) Learner reflects on conception in light of his/her experience
12) Instructor evaluates/adapts descriptions of conception

One of the primary assumptions in the Conversational Framework is that there are always different options of supporting some process with media. Each of these processes can be supported by different types of media:

- *Narrative*: Web resources, video, TV, etc.
- *Interactive*: (Enhanced) hypermedia, Web resources
- *Communicative*: E.g., means computer-mediated communication such as conferencing, chat, discussions
- *Adaptive*: Simulations, tutorials
- *Productive*: E.g., modeling environments

An instructive, animated graphic of the conversational framework, including the process sup-
port covered by various types of media, can be found online at http://www2.smumn.edu/deptpages/~instructTech/lol/laurillard/.

4.1.6 Educational Modeling and Learning Design

The Educational Modeling Language[460] was one of the cornerstones in the specification of the IMS Learning Design (IMS/LD)[461], an XML-based language for specifying learning content and process in a widely pedagogy-independent way. The IMS/LD specification also includes a *learning design best practice and implementation* guide[462] that describes a number of learning design scenarios represented as UML use cases, e.g. "Adapting Units of Learning to Learner Profile." Each of these use cases is described by the following elements:

- Narrative
- Primary actors
- Stakeholders and interests

[460] Koper (2001)

[461] IMS Global Learning Consortium (2003d)

[462] IMS Global Learning Consortium (2003b)

- Preconditions
- Trigger
- Scenario steps
- Extensions

This model allows learning designers to describe learning content and resources in a process-oriented, formalized way by using use cases for analysis, activity diagrams for modeling of the use cases' narratives, and XML documents conforming to the IMS/LD specification that are used for content development and packaging. Thereby, the steps to be taken for developing an IMS/LD compliant unit of learning are[463]:

1) *Analysis*: A concrete educational problem is analyzed, resulting in a didactical scenario that is captured in a narrative.

2) *Design*: The narrative is represented as UML activity diagram, forming the basis for an IMD/LD compliant XML document instance.

3) *Development*: The XML document forms the basis for content and resources development.

4) *Evaluation*: The whole design (including content and resources) is evaluated.

Following these considerations, the IMS/LD is, like other highly formal approaches, a way of sequencing learning content according to an elaborated conceptual design model. The design model (called "Information Model" in IMS specifications) is given in Figure 56. It shows a UML static structure diagram showing the hierarchy and relationships of elements in the XML representation of a learning design. Such approaches are valuable for, but have limited scope of use in blended learning environments, as online learning content is just *one* important aspect of blended learning solutions. Additionally, in sharp contrast to the aims of IMS/LD, the approach presented in this thesis aims primarily at producing *semi-formal visual models* and *structured textual documentation* of blended learning designs.

[463] Adapted from IMS Global Learning Consortium (2003b, p. 20)

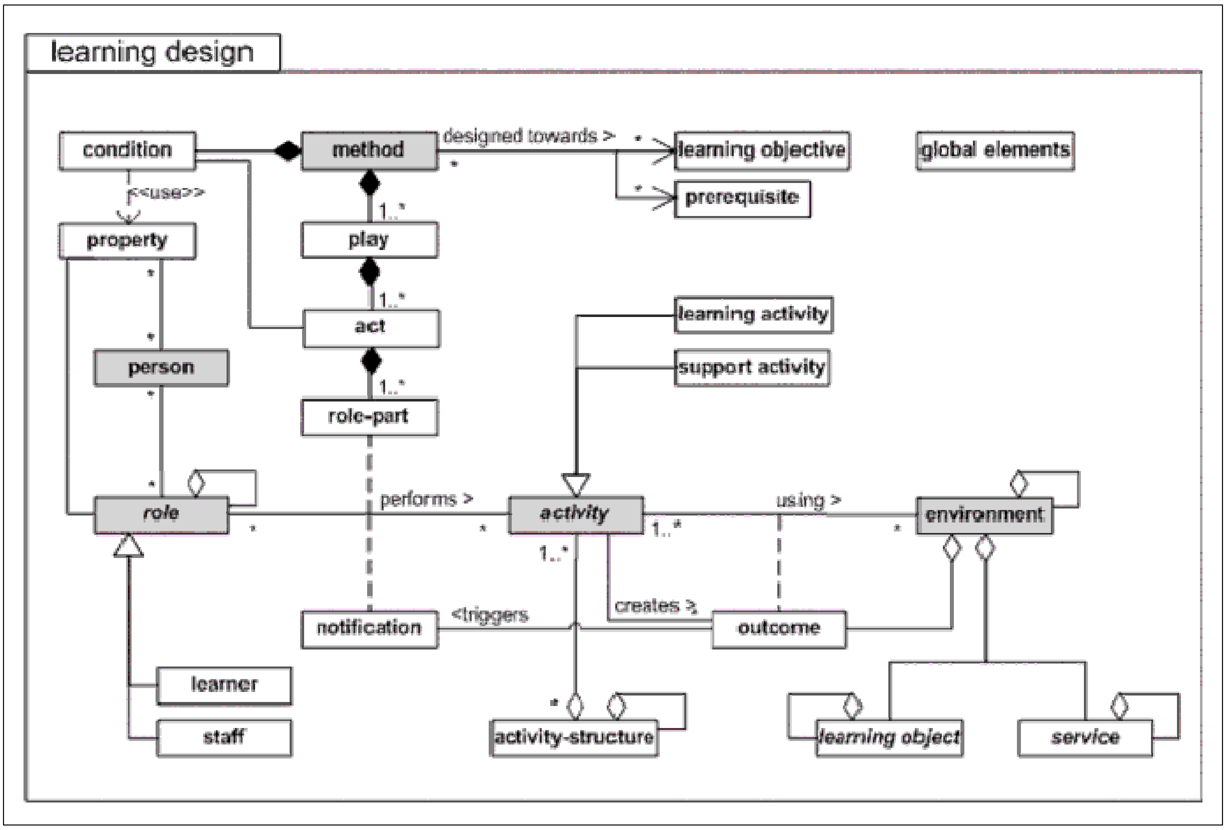


Figure 56: Conceptual structure of IMS Learning Design.[464]
Note that the most important concepts are highlighted in gray color.

4.1.7 Categorization of Virtual Learning Activities

Baumgartner and Bergner[465] have developed an approach that shows some similarities with the BLESS model. Their categorization scheme identifies three levels of abstraction: The first (top) level categorizes and describes a number of *educational scenarios*, however without any ties to a didactic baseline. The second level describes *educational interaction patterns*, which are descriptions of activities that as a whole define a learning scenario (e.g., a "guided discussion" scenario combines the interaction patterns "initiate a topic," "respond," "filter," etc.). The third (bottom) level describes learning and content management systems as well as other learning tools in terms of adequacy for a certain interaction pattern. Generally, the authors use the term "pattern" in its linguistic sense rather than in the Alexandrian sense that is relevant to this thesis. Baumgartner and Bergner admit that the description of the interaction patterns on the second level "*still lacks the greatest amount of analytical description.*"[466] Additionally, there seems to be no sound methodology to identify, describe and combine

464 Source: Taken from Figure 2.2 in IMS Global Learning Consortium (2003c). Copyright © 2003 IMS Global Learning Consortium, Inc.

465 Baumgartner (2003)

466 Baumgartner (2003, p. 7)

scenarios and patterns (in terms of the BLESS model, the link to the top layers comprising courses and learning theory is missing).

4.2 Differentiation from Related Approaches

The specifics of the approach presented in this thesis that differentiate it from the approaches presented in Section 4.1:

- A *didactic concept* or base to build upon: Person-Centered e-Learning is the didactic foundation on which both concrete courses (BLESS layer 1[467]) and the PCeL patterns (BLESS layer 3[468]) are built. That didactic concept is the most fundamental aspect of the *value system* of PCeL patterns, towards which central design decisions are oriented. Pattern collections without such a value system tend to aggravate the derivation of one homogenous *whole* from single patterns[469]. Additionally, this is line with the claim that pattern approaches to learning design should *not* be pedagogically neutral[470].
- A conceptual framework of decomposition of complexity into layers: The *BLESS model* shows stepwise transitions within the socio-technical layers of blended learning design, as well as an interface between platform independent scenario patterns and their user-centered platform support (i.e., the Web templates).
- The *methodological underpinning*: Action Research (AR) is the primary driver of cyclic mining, description, evaluation, and improvement processes, with each cycle of the process being structurally guided by the BLESS model. This kind of adaptation of AR aims to overcome its shortcomings as described in Section 3.1.2 on page 96.
- Usage of *standardized conceptual modeling* techniques inside as well as among patterns, using the UML. This supports all activities of each AR cycle through visualization and semi-formal description of person-centered learning scenarios involving the use of learning technology. The combination with the object-oriented paradigm (generalization hierarchies, dependency networks) inside the pattern repository additionally fosters analysis and design processes, which is relevant input to each AR cycle.
- The *pattern repository* is particularly designed for dissemination and subsequent reuse outside of its original context. This allows any instructor to reuse single patterns or pattern families and to subsequently provide evaluations, case study reports, and suggested improvements for further collaborative development of the pattern repository. In this re-

[467] See p. 92

[468] See p. 93

[469] Discussions on the importance of a value system underlying patterns can be found for example in Alexander (1979) or Goodyear et al. (2004); see also the discussion on the *Quality without a Name* (QWAN) in Section 2.2.3.2, p. 58.

[470] Goodyear et al. (2004)

spect, the repository as presented in this work does *not* claim to capture the "final wisdom," but rather an initial state based on a methodology that is accessible to and practicable by others.

- *Visual artifacts*: On one side, we have highly formal approaches like IMS/LD, which aims to produce machine-processible XML documents. On the other side, we have completely informal approaches, using unstructured textual descriptions of learning designs. The PCeL pattern approach chooses the "*golden mean*" by primarily producing ***semi-formal visual learning design models*** complemented by ***structured textual descriptions***.

Finally, the PCeL pattern approach presented here addresses all of the major issues that are perceived as problems in the field of e-learning patterns today. According to a recent study[471] on e-learning patterns the main problems currently are:

1) *Variability in focus and intention*: The focus of current proposals is distributed among software design, pedagogy, and content issues. There is no single focus.
2) *Diversity in description format*: Most of the approaches come by with their own templates and visions about how to encapsulate the obligatory pattern ingredients problem, context, and solution.
3) *High level of abstraction*: Most patterns are too abstract to be put into practice by non-expert users.
4) *Lack of organization*: There is no common organization scheme, and patterns are unrelated among different approaches.

The PCeL pattern approach is capable of resolving the problems stated above:

- The focus is clearly on learning design based on the pedagogical principles of the PCA. However, it is acknowledged that other aspects of e-learning may be equally important, e.g. content issues, which are currently very popular.
- It employs a uniform, simple description format that includes all essential pattern parts;
- There are *different* levels of abstraction in the PCeL patterns, ranging from abstract higher-level patterns to concrete, ready-to-use lower-level patterns. Different degrees of abstraction are a primary aspect of conceptual modeling and object-oriented thinking, aiming to support the understanding of complex concepts by humans.
- The patterns are organized structurally in a way that increases usability and enables reuse and extensibility[472]. Moreover, the structural model is based on a simple, standardized notation, which minimizes ambiguity in concepts and expressions.

[471] Caeiro, Llamas-Nistal and Anido (2004)

[472] Cf. Derntl (2004)

4.3 PCeL Patterns and Promotive Activities

In person-to-person interaction, promotive, non-directive activities are a direct consequence of living and holding the three personal dispositions of acceptance, realness, and empathic understanding[473] toward the partner. According to Tausch and Tausch[474], when holding all of these dispositions, the person will basically tend to act in a facilitative and non-directive way. Generally, promotive, non-directive activities are characterized as follows[475]:

- Promoting meaningful mental processes and constructive development of personality (e.g., self-respect, openness for experience) in the other person, and to some extent even in the person who communicates the dispositions.
- Alignment with the four psycho-social values of living: self-determination, respect for the person, social order, and mental as well as physical functioning.
- Being socially reversible, meaning that even young persons may hold these attitudes towards adults without being disrespectful.
- Furthering the quality of interpersonal relationships.
- Facilitating self-responsible, self-initiated learning processes and creativity in learners.
- Being equally promotive for the "holder," and not only for the "recipient" of the three person-centered dispositions.

Several studies[476] have confirmed the positive effects of promotive activities on learning quality, outcome, and student satisfaction. Consequently, it seems worthwhile to investigate existing and possible points of contact between promotive activities and PCeL patterns. But first, we give an outline of endeavors and settings that are particularly suited to promote non-directive and self-initiated learning processes[477]:

- *Facilitating temporary work in small teams*: This is a well-proven method of furthering self-directed learning by dividing the class into teams of 2–5 persons to work on well-defined tasks.
- *Endeavor to design transfer of knowledge in a comprehensible way*: Adhering to the four dimensions of comprehensibility (simplicity, organization, conciseness, and encouragement) helps the students in acquiring knowledge from oral or written information resources.

[473] Cf. Section 2.1.3.1, p. 24

[474] Tausch and Tausch (1998, p. 243-245)

[475] Cf. Tausch and Tausch (1998, p. 244-245)

[476] For example, Wittern and Tausch (1983), Cornelius-White (2003)

[477] Tausch and Tausch (1998)

- *Furthering of helpfully living together in a "good group"*: This may be characterized by furthering of exchange of personal feelings, personally important experiences, and allowing for autonomous interaction.
- *Facilitating learning through provision of resources*: Providing inspiring learning material as well as personal resources furthers autonomous, self-responsible learning.
- *Facilitating thinking processes in class*: Addresses facilitation of creative, longer-lasting, autonomous thinking processes by acting promotive and in a non-directive way, e.g., in the process of solving complex problems that are of personal interest to the students.
- *Furthering beneficial working progress in class*: This can be achieved by (1) getting to know the work personally, then by (2) elaborating and inspecting sub-tasks, and by (3) finalizing, exploiting and/or applying the work.

These endeavors and their connections to patterns of the PCeL pattern repository are scrutinized in the following sub-Sections. A key question that is subject of another PhD thesis[478] at the RLET is, which of these endeavors and settings can be transformed to online activities as well as which qualities are strengthened and which are weakened in computer-supported learning.

4.3.1 Teamwork

Work in small teams is a proven method for furthering self-directed learning and for a constructive personality development[479]. In the context of higher education it can be employed to solve complex problems/projects as well as for writing seminar and/or research reports[480]. Teamwork is explicitly addressed by the patterns in the *Project-Based Learning* (PBL) package, where participants elaborate projects iteratively and incrementally in several successive project milestones. They may work out individual projects, may be organized in teams, or may collaborate collectively on a single group/course project, whereas in most of the application cases work in small teams of 2–5 members is preferred. One specific form of PBL, which is also employing teamwork, is the LEARNING CONTRACT pattern, where teams propose topics they want to elaborate and sign contracts defining learning targets and expected contributions for each team. In teamwork scenarios, the instructor prepares and provides relevant content and working resources, coaches teams, and makes herself available to the students on demand[481]. For students, teamwork has many positive and facilitative effects:

1) they think and work individually as well as collaboratively,
2) they train their communication skills,
3) they learn to know how to cope with different opinions and conflicts,

[478] Bauer (2003)

[479] Tausch and Tausch (1998, p. 253)

[480] Tausch and Tausch (1998, p. 258)

[481] Tausch and Tausch (1998, p. 259)

4) they take responsibility for decisions made in teams,
5) they learn to organize themselves when working with peers,
6) they have more motivation and joy in working and learning.[482]

In the PCeL pattern repository, teamwork is technically supported by the TEAM WORKSPACES pattern, which aims to provide teams with private workspaces that they may use to create, store, work on, and share their contributions and other documents to allow for online collaboration within and among teams.

Facilitative aspects of teamwork may be additionally augmented through activities such as:

- Mutual exchange of information and collaborative construction and collection of information as well as resources[483]. These activities are also addressed by several patterns: EXCHANGE OF CONTRIBUTIONS (participants exchange and discuss their contributions and ideas online), INFORMATION GATHERING (participants and instructors interact with the primary target to collect information which shall be gathered collaboratively and shared among all participants; concrete examples include THEORY ELABORATION, BRAINSTORMING, and exchange MARKETS), and KNOWLEDGE BASE CONSTRUCTION (advancing the construction of a knowledge base in a specific subject area from single contributions and knowledge fragments).

- Mutual support and encouragement regarding work as well as thematic or personal questions[484]. Such activities are particularly addressed in the TUTORIAL pattern, which proposes that students' peers (tutors) should do introductory or collateral technical tutorials for complex technical or application-oriented scenarios involving new or sophisticated tools and methods.

- "Thinking aloud," giving and exchanging opinions on technical as well as personal issues helps students to direct their own learning, make individual progress, and learn from their mistakes[485]. These promotive activities are addressed by several interactive patterns, such as COMPUTER-MEDIATED COMMUNICATION for online exchange of ideas and opinions independently of time and local displacement, COLLECT FEEDBACK as a way to convey personal opinions on learning scenarios to the instructor (e.g., for subsequent improvement of the learning scenarios), or DIARY, which is a pattern that can be used collaterally with COURSES or complex learning scenarios to collect personal thoughts, insights, opinions, and reports from students/teams.

[482] Compiled from Tausch and Tausch (1998, p. 260-261)
[483] Tausch and Tausch (1998, p. 263)
[484] Tausch and Tausch (1998, p. 263)
[485] Tausch and Tausch (1998, p. 264)

4.3.2 The "Good Group"

The climate in a good group is characterized by the three Rogers' Variables and by a high degree of mutual support and communications. Such a climate facilitates self-directed and self-responsible learning in individuals.[486] In blended learning scenarios online phases can be utilized to continue face-to-face interaction and communications, as well as to prepare for subsequent face-to-face meetings and discussions[487]. However, COMPUTER-MEDIATED COMMUNICATION (CMC, also included in the PCeL pattern repository), which can be implemented synchronously (CHAT) or asynchronously (ONLINE DISCUSSION), is only considered more enjoyable, uninhibited, and capable of producing a greater diversity of perspectives when coached and facilitated accordingly[488].

4.3.3 Learning Resources

Provision of relevant information resources is essential for facilitating the learning process. In terms of PCeL patterns, one pattern is used for providing content and other resources online: PUBLISH, which generically describes disclosure of an information item (e.g., text, file) to a certain target person, role, or group of roles and/or persons.

- The economic learning methods as already described in the previous Section are addressed, e.g., by the SELF-EXAMINATION pattern, which aims to provide participants with the option of evaluating themselves in a uniform, structured way by providing questions and expected answers. Such examination may optionally be used by students to assess their current status of acquired knowledge (*voluntary learning checkup*[489]) while on the side of the instructor "only" one-time compilation of questions is required.
- Specifying learning goals and making them transparent is an inherent intention of the LEARNING CONTRACTS pattern, where teams propose topics they want to elaborate and sign contracts defining learning targets and expected contributions/outcomes for each team. This allows for monitoring of compliance with requirements and deliberately set targets.
- Providing personal learning resources is addressed in the TUTORIAL pattern, where more advanced students are available for concerns of their younger or less experienced peers.

---

[486] See Tausch and Tausch (1998, p. 277-281).

[487] Cf. Dietz-Uhler and Bishop-Clark (2001)

[488] See for example Ensher, Heun and Blanchard (2003), McNeil, Robin and Miller (2000)

[489] Tausch and Tausch (1998, p. 291)

- Furthering of the constructive development of the learner, as well as the reflective engagement with one's self[490] can be supported online through the DIARY and COLLECT FEEDBACK patterns.

Tausch and Tausch assert that a certain degree of freedom of choice and acting enables the students to collect valuable personal experiences[491]. The PROPOSAL pattern and its derivates address precisely this issue: Students are asked to more or less freely choose and propose and the instructor subsequently reviews and approves the proposals. This may be used in many circumstances, e.g. in TEAM BUILDING, LEARNING CONTRACTS, PROJECT-BASED LEARNING, seminar reports, etc.

4.3.4 Thinking Processes

Thinking itself and primary aspects of thinking processes as described by Tausch and Tausch[492] may also be supported by a number of patterns:

- Thinking processes evolve when problems have to be solved and solutions are not readily available and/or perceptible. Central problem solving patterns are PROJECT-BASED LEARNING and LEARNING CONTRACTS, where students tackle complex problems incrementally and iteratively in several PROJECT MILESTONES.
- Thinking processes are non-linear, unpredictable, and highly different among persons, even for similar problems. The Internet with its manifold possibilities of interlinking documents (hyperlinks) provides perfect options for learners to explore highly diverse and dependent information in their own ways and tracks.
- Thinking processes are particularly furthered when problems and solutions of personal interest are tackled. This is supported by the PROPOSAL pattern that encourages participants to propose problems of personal interest within a certain (instructor-supplied, curriculum-compliant) context.
- Combining and structuring is one main aspect of thinking processes. KNOWLEDGE BASE CONSTRUCTION is a pattern that resembles this in the large: knowledge fragments (e.g., single contributions of different type) are assembled to constitute a whole (knowledge base).
- Comparison, restructuring, and abstraction are thinking processes prevalent in evaluation and valuing of decisions and actions. The *Evaluation* pattern package defines a set of patterns that address such processes: Students evaluate their peers' contributions (PEER-EVALUATION), they evaluate themselves (SELF-EVALUATION), and they provide valuing FEEDBACK to the instructor.

[490] See Tausch and Tausch (1998, p. 295-296)

[491] Tausch and Tausch (1998, p. 294)

[492] Tausch and Tausch (1998, p. 298-307)

- Longer-lasting and autonomous thinking processes are furthered by tackling of complex situations and problems, which is addressed by PROJECT-BASED LEARNING and derived patterns.

4.3.5 Working Progress

Beneficial working progress is supported by adequate design and structuring of the teaching and learning process[493]. Thereby it is essential that the instructor (or the learning designer) takes into account perceptions and concerns of participating students, which is explicitly supported by the following *Feedback* patterns:

- REACTION SHEETS describes collecting reactions on specific aspects of learning scenarios and activities to collect feedback in an open, unstructured way.
- FEEDBACK FORUM describes collecting feedback in a semi-structured way by soliciting postings to instructor-initiated ONLINE DISCUSSION threads. This additionally allows for open discussion of feedback postings.
- QUESTIONNAIRE is a form of collecting feedback in a structured way by specifying and providing a set of questions along with scaled, possible responses.

Tausch and Tausch identify three consecutive phases of the problem-solving process[494]:

- Kick-off, getting to know the task: This phase is connected with activities such as confronting oneself with the problem, spontaneous expression of personal thoughts and feelings, intuitive attempts of problem solving, and collecting relevant material[495]. This is embodied in some interactive INFORMATION GATHERING patterns: BRAINSTORMING for collecting ideas gathered in brainstorming sessions (either online or present), THEORY ELABORATION for collecting and elaborating certain aspects of theories or problems, MARKET for sharing information and contributions in learning activities, or ELABORATING GOALS AND EXPECTATIONS for upcoming tasks and activities.
- Inspection and elaboration of (sub-)tasks: Relevant material is collected, worked through, and exchanged among participants (compare the THEORY ELABORATION, EXCHANGE OF CONTRIBUTIONS, and MARKET patterns). Occasionally, elaboration phases alternate with plenary sessions, where single participants or teams present their progress or solutions (this is addressed by the PRESENTATION PHASES or EXCHANGE OF CONTRIBUTIONS patterns). Additionally, in this phase communication and ONLINE DISCUSSION among participants or problem-oriented WORKSHOPS are fruitful activities. In elaboration phases it is often helpful for participants to be able to consult the instruc-

[493] Tausch and Tausch (1998, p. 307)

[494] Interestingly, we had already elaborated patterns that support individual aspects of these phases *before* consulting the work of Tausch and Tausch.

[495] See Tausch and Tausch (1998, p. 307-310)

tor, tutor, or expert for specific problems or questions. The CONSULTATION pattern describes means of doing this online.

- Finishing the work (which does not necessarily mean that a complete solution has been elaborated). What is more important is that the participants have found the right way and that they have learned from the process. In such a concluding phase, reflective processes may be used by participants to structure and rethink the process.[496] This can either be achieved present or online. As already mentioned above, the *Feedback* package includes patterns that provide means of supplying personal thoughts and reflections (REACTION SHEETS, FEEDBACK FORUM), and structured responses (QUESTIONNAIRE) online.

4.3.6 Integration

The following matrix summarizes the above textual integration efforts. The left-most column depicts concrete PCeL patterns and the header row depicts promotive activities as presented in the previous sub-Sections. An 'X' in a matrix cell means that the promotive activity in this cell's column is in some way supported by the respective pattern in this cell's row. The degree of support/overlap is indicated by the number of X's in that cell (up to three X's are used.)

Table 9: Integrating promotive activities with PCeL patterns.

	Teamwork	The "Good Group"	Learning Resources	Learning Progress	Working Progress
BRAINSTORMING	X X				X X X
CHAT	X	X X			
COLLECT FEEDBACK	X			X X X	
CONSULTATION					X X
DIARY	X X		X		
ELABORATE GOALS AND EXPECTATIONS					X X
EXCHANGE OF CONTRIBUTIONS	X X X				
FEEDBACK FORUM					X X
KNOWLEDGE BASE CONSTRUCTION	X X X			X	
LEARNING CONTRACTS	X X X		X X	X X X	
MARKET	X				X

496 See Tausch and Tausch (1998, p. 313-315)

	Teamwork	The "Good Group"	Learning Resources	Learning Progress	Working Progress
ONLINE DISCUSSION	X	X X			X
PEER-EVALUATION				X X X	
PRESENTATION PHASES					X X X
PROJECT-BASED LEARNING	X X X		X X	X X	
PROJECT MILESTONE			X	X X	
PROPOSAL			X X	X	
PUBLISH	X		X X X		
QUESTIONNAIRE					X
REACTION SHEETS					X
SELF-EVALUATION				X X	
SELF-EXAMINATION			X	X X X	
TEAM BUILDING			X		
TEAM WORKSPACES	X X X				
THEORY ELABORATION	X X		X X		X X X
TUTORIAL	X		X X X	X	

4.4 PCeL Patterns and Instructional Design

[In a] comprehensive sense, instruction must be planned if it is to be effective.[497]

Instructional design is concerned with systematically planning the events which are aimed at aiding individuals to learn[498]. The result of such planning processes is an *instructional system*, which constitutes the environment in which the instructional events take place based on an "organized way of accomplishing certain goals".[499] Note that while the general term "system" does not imply any boundaries of the instructional design process, the discussion here is clearly focused on designing an instructional system in the *scope of a course*. Gagné identifies 14 steps to be taken in the design of such a system, each located on the system, course, or lesson levels[500]. These stages are arranged schematically in chronological order in Figure 57.

[497] Gagnè and Briggs (1979, p. 3)

[498] Cf. Gagnè and Briggs (1979, p. 3-5)

[499] Gagnè and Briggs (1979, p. 19)

[500] Gagnè and Briggs (1979, p. 23)

As we focus specifically on course design, the boundaries of the system and course levels in Gagné's process model converge.

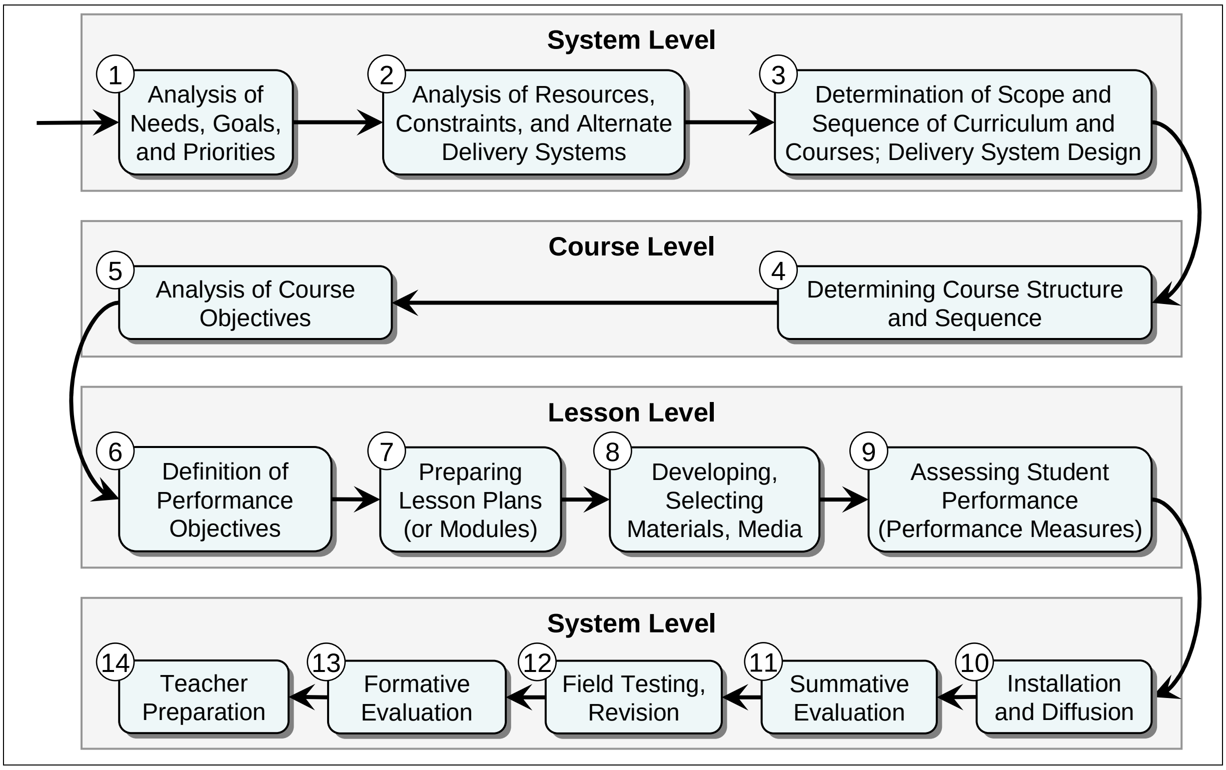


Figure 57: Gangé's stages in the design of instructional systems.[501]

Note that Gagné emphasizes that the concrete design process following the stage model is barely conducted in the same sequential manner in which the stages are arranged: "*In practice there is much working backwards and forward in a non-linear fashion, because work done at any one stage gives new insights into other stages.*" [502]

Conceptually, blended learning design is a special application case of the more general discipline of instructional design: The basic steps from the traditional instructional design process are still valid. Only the process of design becomes more complex in blended learning, as the instructional designer has to consider additional design options and requirements introduced by employing learning technology. As the PCeL pattern repository along with its learning scenario modeling and description method aims at supporting the task of blended learning design, it is worthwhile to analyze the support the PCeL pattern approach can provide in each stage of the instructional design process.

The first three stages at system level focus on goals and desired outcomes of an instructional system from an analytical point of view[503] and thus offer few points of contact with the PCeL pattern approach. However, in step 3, sequences and structural relationships of courses within curricula can be modeled using static and dynamic UML diagram types (static struc-

[501] Source: Reproduced from Table 2-1 in Gagnè and Briggs (1979, p. 23)

[502] Gagnè and Briggs (1979, p. 40)

[503] Cf. Gagnè and Briggs (1979, p. 23-28)

ture diagrams and activity diagrams, respectively). Course models can be ***organized in curriculum packages***, much like patterns are organized in pattern packages. This would allow for visualizing dependencies among courses, as well as their chronological order in separate activity diagrams.

Among the following two steps on the course level, step 4 is particularly suited for applying the pattern modeling approach. Course sequences can be ***modeled and visualized*** at different levels of aggregation using activity diagrams with the proposed PCeL extensions. Each activity (or course module, depending on the level of aggregation) can be complemented with results of the previous analysis steps, namely goals and objectives of the activity, as well as available and required resources. Course objectives and goals can be ***matched with pattern intents*** and motivations to support the instructional designer in choosing an appropriate pattern, if available. This way, course objectives can be ***decomposed*** collaterally with the refinement of course models from general objectives/patterns at course or module level to specific objectives/sequences at activity level. Particularly when designing PCeL courses, the ***Course Types package*** may provide useful, generic arrangements of activities in a course sequence appropriate for the desired goals/outcomes.

At lesson level, the PCeL patterns approach can provide valuable input and tools for steps 7 through 9: For the preparation of lesson/module plans (step 7), which include activities of both learners and instructors[504], activities can be arranged using the PCeL scenario modeling approach. These models represent ***detailed descriptions and visualizations of events/activities*** employed to reach particular course or module objectives. Again, utility- or collateral patterns can be used to complement the primary instructional method in a course phase (e.g. DIARY or ACHIEVEMENT AWARD attached to LEARNING CONTRACTS). For the selection and development of materials and media at stage 8, the ***course documentation*** including models and descriptions can be used to identify required materials (books, links, Web resources) and delivery channels (online/distant, blended, face-to-face). At stage 9, assessing student performance, the instructional designer is supported by the ASSESSMENT PHASES pattern, which provides a model of ***incorporating person-centered assessment practices*** through the use of various patterns in the *Evaluation* pattern package. Through multiple views on student performance provided for example by BLENDED EVALUATION the instructor receives a comprehensive set of evaluations to assess students' performance with respect to desired performance objectives. Gagné considers performance measures to be also considered for the whole instructional design, which is supported by various *Feedback* patterns in the PCeL pattern repository.

Back at system level, the final three stages (11-14) also provide multiple points of overlap with intentions of the PCeL approach: At stage 11 and 13, formative and summative evaluation, which is used to revise and improve materials, course/lesson plans, and performance measures[505], the ***Feedback patterns can provide valuable input for improvement***, e.g.

[504] Gagnè and Briggs (1979, p. 32)

[505] Gagnè and Briggs (1979, p. 37)

by Collecting Feedback and obtaining qualitative and quantitative data through Reaction Sheets, Questionnaires, and Feedback Forums. For initial testing and for installation and diffusion of the instructional design (at stages 12 and 14), the overall course design including models and documentation can be used to ***distribute the design for application and adoption*** in different environments and contexts. Thereby, the models provide clear visualizations of the whole design at different levels of detail, and can therefore be employed and adopted more easily by instructors (and designers) than purely text-based design documentations.

The discussion shows that the PCeL pattern approach and the repository can substantially support instructional designers in many important stages of the instructional design process, even at different levels of the instructional system. Even though the approach is capable of providing a versatile toolbox for instructional design, it is still left to the designers and particularly to instructors to convert the models into situated and effective educational experiences.

5 The Pattern Repository

This Chapter is organized according to the pattern packages that the PCeL pattern repository defines. Each package starts a new section at heading level 2.

5.1 Assessment

Assessment Phases

Package: *Assessment*

Intent

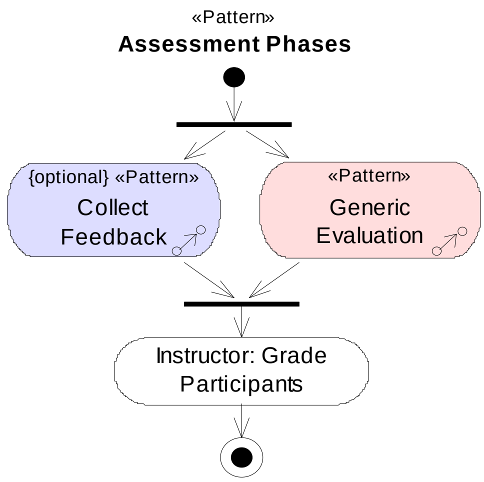
Use Generic Evaluation to assess participants, and Collect Feedback on the learning activity from participants.

Motivation

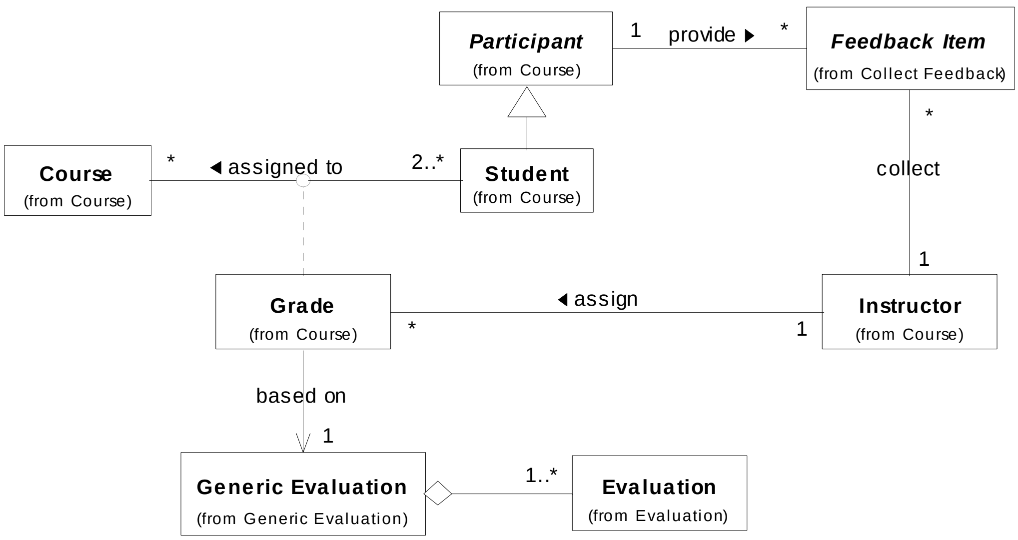
Assessment of participants' achievements is one central activity in educational course settings. However, in conventional courses the assessment phases consist of just one single activity, namely *grading* of participants by the instructor. Such an assessment scenario has no deeper meaning and produces few insights for participants, it is just a requirement. In Person-Centered settings, the participants are actively involved in the assessment phases:

- They provide valuable feedback to the instructor, if the instructor chooses to Collect Feedback.
- They engage in the evaluation of the own and/or other participants' contributions (Generic Evaluation).

Sequence

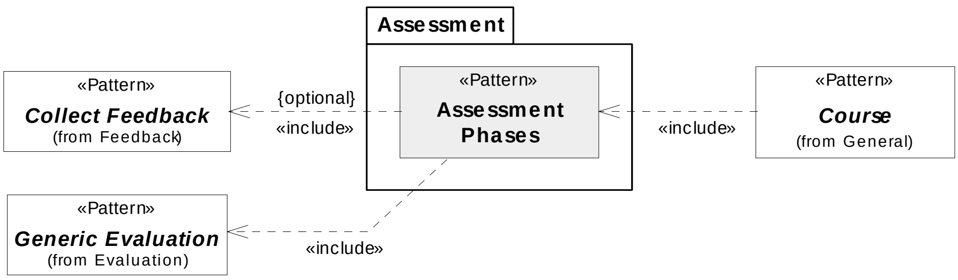


Structure



The structure of this pattern is completely composed by reusing and integrating structural elements of the hosting pattern COURSE and included *Feedback/Evaluation* patterns.

Taxonomy/Dependencies



Parameters

- Primary pattern author: Michael Derntl
- Primary pattern source: Research Lab for Educational Technologies (University of Vienna)
- Pattern categories: Generic, Composite, Utility
- Level of abstraction: High
- Scope: Phase
- Flexibility: High
- Level of confidence: 5
- Input: Definition of feedback to be collected, evaluation scenarios to be employed
- Output: Feedback, evaluation, grades

Web Template

The administration view links the user to the administration of GENERIC EVALUATION and optionally to COLLECT FEEDBACK. This pattern does not define a participant view or report view as these are provided by the aforementioned patterns.

Examples

Assessment phases are employed in different forms in any course underlying this repository, for example:

- In ***Person-Centered Communication***, the assessment phases were conducted by COLLECTING FEEDBACK (REACTION SHEETS) for the course, as well as by involving the participants in the evaluation process by asking them to SELF-EVALUATE and to PEER-EVALUATE their contributions.
- In ***Web Engineering***, additionally a final QUESTIONNAIRE (as a form of COLLECTING FEEDBACK) was distributed to allow for comparison of the initial questionnaire results with that of the end of the course.

Evaluation

Not available: Refer to included patterns from the *Taxonomy/Dependencies* section.

5.2 Course Types

Interactive Lecture

Package: *Course Types*

Intent

In courses or scenarios where transmission of information is the main goal use Interactive Elements to minimize pure lecturing.

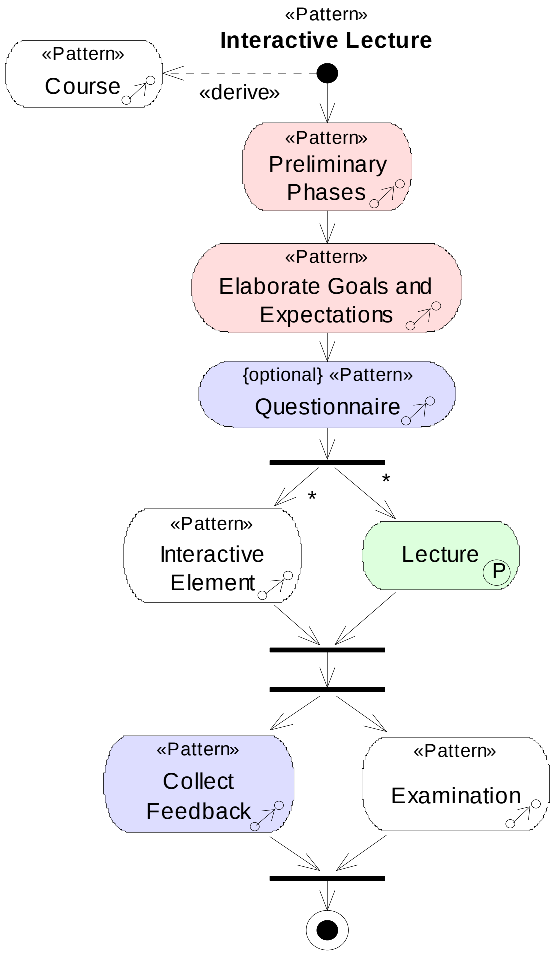
Motivation

> *Why the lecture is regarded as a major means of instruction is a mystery. It made sense before books were published, but its current rationale is almost never explained.*[506]

Interactive Lecture is intended to enhance pure lecturing sessions by concurrently or alternatively employing Interactive Element scenarios. Even though transmission of relevant content (theories, methods, etc.) is undoubtedly important, this pattern aims to actively involve participants in this process, e.g., by incorporating Theory Elaboration or Brainstorming sessions. As this pattern repository includes a number of different scenarios of Interactive Elements, the instructor is assisted in gradually increasing active participation in otherwise traditional lecture settings.

[506] Rogers and Freiberg (1994, p. 210)

Sequence

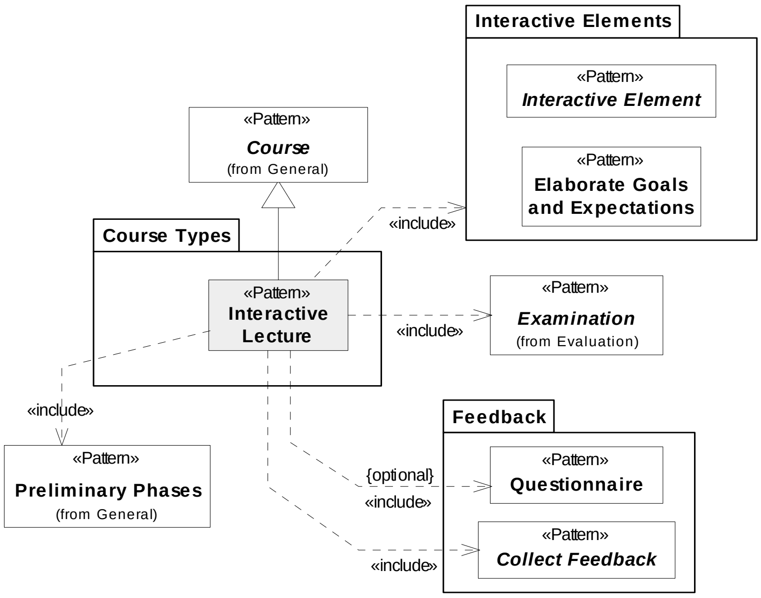


Activity	Description
PRELIMINARY PHASES	
ELABORATE GOALS AND EXPECTATIONS	
QUESTIONNAIRE	The initial questionnaire aims to survey participants' a-priori attitudes and motivations. If complemented with a concluding questionnaire, this can be perfectly used to compare responses at the beginning and the end of the course (see the sequence of COURSE).
INTERACTIVE ELEMENT, Lecture	Any number of lecture sessions and INTERACTIVE ELEMENTS may be used concurrently, or in an alternating way. Note that this thread of the course can also be used stand-alone in other course scenarios.
COLLECT FEEDBACK	
EXAMINATION	

Structure

Not available. Inherited from parent pattern COURSE and dependent on included patterns.

Taxonomy/Dependencies



Parameters

- Primary pattern author: Michael Derntl
- Primary pattern source: Research Lab for Educational Technologies (University of Vienna)
- Pattern categories: Course type, Composite
- Level of abstraction: Medium
- Scope: Course
  Note that the lecture and INTERACTIVE ELEMENTS thread can also be employed at phase scope.
- Primary presence type: Blended
- Flexibility: High
- Level of confidence: 3
- Number of participants: Unrestricted
  Restrictions may arise when using interactive elements which are restrictive on number of participants.
- Application effort: High
- Level of expertise required: High
- Suggested assistance: Administrator, Tutor
- Target skills: Interpersonal Skills, Communication, Collaboration, Problem solving

Web Template

Inherited: See parent pattern COURSE.

Examples

Interactive lecturing was used in ***Web Engineering*** to present and elaborate relevant content. The instructor had prepared relevant content for the lectures, but offered the participants to suggest additional topics of interest in the initial lecture session. This was done in a face-to-face BRAINSTORMING session, where participants were asked to raise additional topics and subjects. Subsequently, these topics were published on the learning platform, giving participants the opportunity to add further comments and suggestions. Three of the collected topics have been incorporated in the lecture schedule while the remaining topics turned out to be mainly congruent with the gross agenda prepared by the instructor. In the beginning of the course the participants were also asked to complete QUESTIONNAIRES regarding their expectations, goals, and motivations to participate. At the end of the course they were asked to complete a second QUESTIONNAIRE that was used to compare their responses with those made in the beginning of the course. Additionally, REACTION SHEETS were solicited to collect open feedback on any course aspect. Regarding the assessment phases, participants had freedom of choice between two options of receiving a grade: they could either pass a conventional written EXAMINATION, or they could commit themselves to achieve certain self-defined learning targets in a LEARNING CONTRACT scenario.

More detailed usage examples can be found in included patterns.

Evaluation

Refer to parent pattern COURSE as well as to any other included pattern.

References

Rogers, C. R., & Freiberg, H. J. (1994). *Freedom to Learn* (3rd ed.). Columbus, Ohio: Charles E. Merrill Publishing Co.

LAB COURSE

Package: *Course Types*

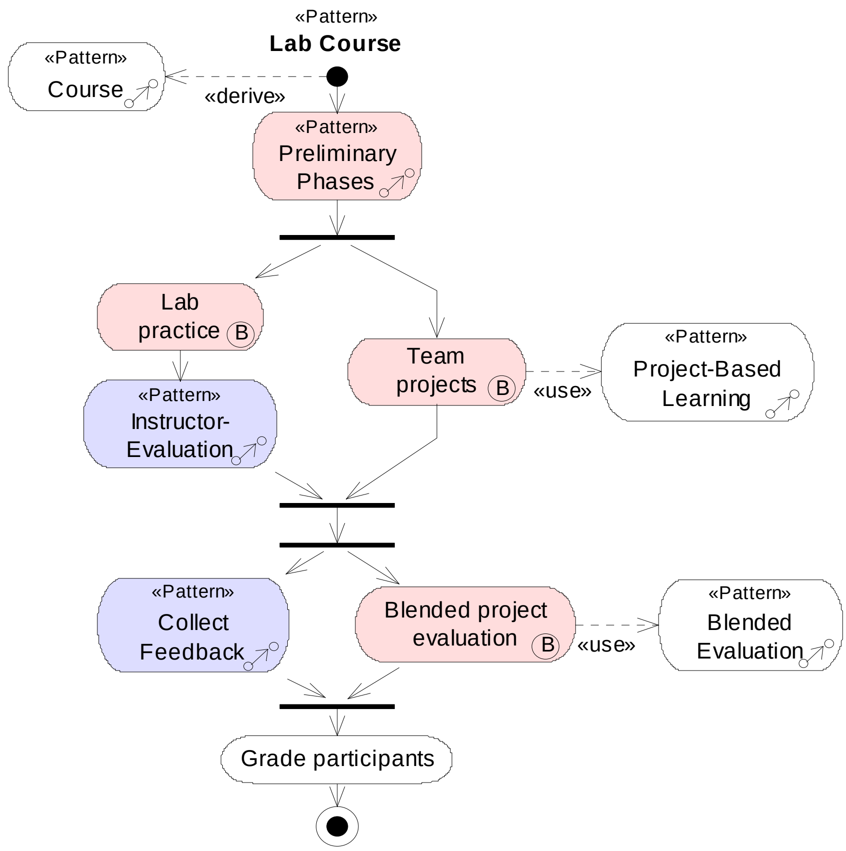
Intent

Describes a course type where application-oriented lab practice, with concurrent PROJECT-BASED LEARNING is used throughout the course.

Motivation

This pattern does not describe a radically new approach to lab courses. Rather it proposes that individual lab work is complemented with teamwork on more complex, self-chosen projects, which are self- and peer-evaluated by participants.

Sequence

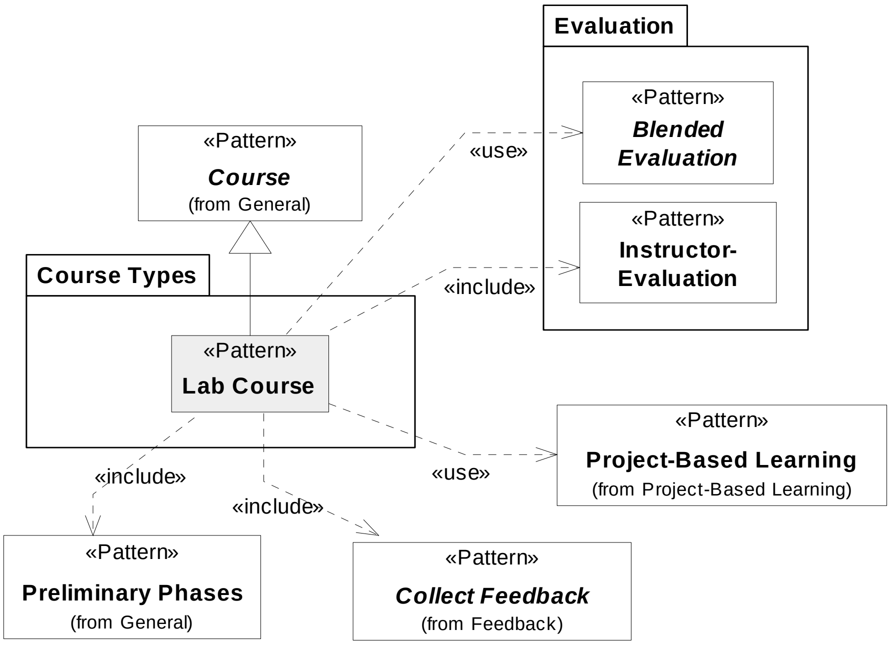


Activity	**Description**
Preliminary Phases	
Lab practice, Instructor Evaluation	In this stream, assignments, tasks, and/or examples are elaborated individually by participants. Subsequently, the instructor evaluates the participants' achievements.
Team projects, Blended project evaluation	In this parallel stream, team projects are elaborated as a form of Project-Based Learning. The focus here is on applying theories and methods to develop and acquire practical as well as interpersonal skills. This is concluded by Blended Evaluation of the team projects, where Self-Evaluation as well as Peer-Evaluation are used and considered in subsequent grading of participants by the instructor.
Collect Feedback	

Structure

Not available.

Taxonomy/Dependencies



Parameters

- Primary pattern author: Michael Derntl
- Primary pattern source: Research Lab for Educational Technologies (University of Vienna)
- Pattern categories: Course type, Composite
- Level of abstraction: Medium
- Scope: Course
- Primary presence type: Blended
- Flexibility: High
- Level of confidence: 3
- Number of participants: up to 30
  Lab courses are usually restricted by infrastructure (e.g., number of available computers, tools, etc.). Additionally, a large number of participants may impose negative effects as individual coaching in such an application-oriented learning scenario seems impracticable with more than 30 participants.
- Application effort: High
- Level of expertise required: High
- Suggested assistance: Administrator, Tutor
- Target skills: Technical skills, Interpersonal Skills, Communication, Collaboration, Problem solving, Practical skills

Web Template

Inherited: See parent pattern COURSE.

Examples

A lab course setting similar to that described in this pattern was used in the ***Web Engineering*** courses (summer terms 2003/4). In the lab practice stream, participants had to individually elaborate some practical examples and tasks with the *Extensible Markup Language* (XML). In the concurrent PROJECT-BASED LEARNING scenario, participants were organized in teams to realize self-chosen Web application projects. The projects had to be completed by following a standard process, namely the *Rational Unified Process* (RUP), which includes 4 major milestones: inception, construction, elaboration, and transition. During the whole process, the instructor and their tutors were available as providers of resources and technical assistance/tutorials, respectively.

More detailed usage examples can be found in included patterns.

Evaluation

Not available.

PROJECT-BASED LEARNING COURSE

Package: *Course Types*

Intent

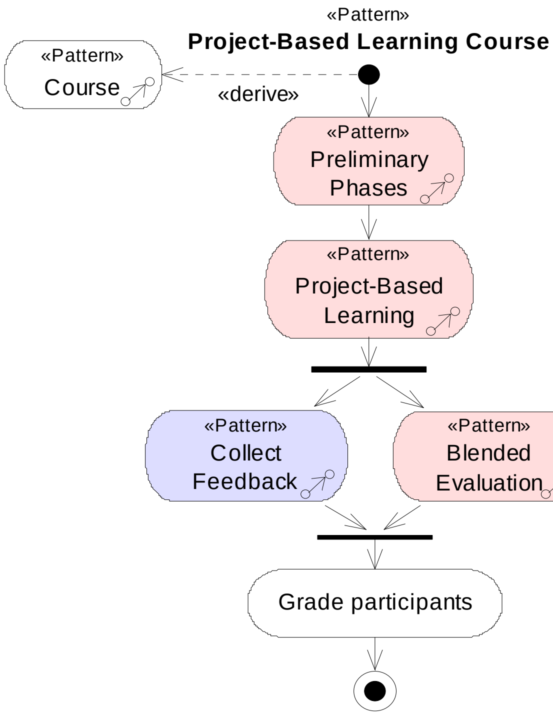
Use PROJECT-BASED LEARNING as the primary method of the learning process, and BLENDED EVALUATION of projects for evaluation of participants.

Motivation

This course type employs a PROJECT-BASED LEARNING (PBL) scenario as the primary teaching and learning approach. One central feature of PBL is that is perfectly suited for assessment practices that involve participants in different ways and roles[507], e.g., as self- and peer-evaluators in BLENDED EVALUATION. See the PROJECT-BASED LEARNING pattern for more detailed discussion of motivational aspects involved in PBL.

[507] Cf. San Mateo County Office of Education (2001), or Kraft (2003)

Sequence

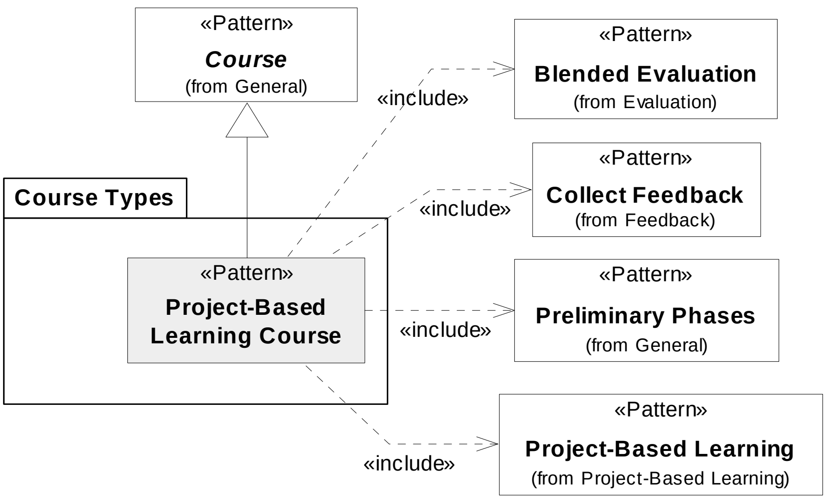


The sequence is self-explanatory: the abstract "*Main course phases*" activity of parent pattern COURSE has been specialized by including PROJECT-BASED LEARNING as the primary learning activity.

Structure

Not available.

Taxonomy/Dependencies



Parameters

- Primary pattern author: Michael Derntl
- Primary pattern source: Research Lab for Educational Technologies (University of Vienna)
- Pattern categories: Course type, Composite
- Level of abstraction: Medium
- Scope: Course
- Primary presence type: Blended
- Flexibility: Low
- Level of confidence: 4
- Number of participants: up to 30
- Application effort: High
- Level of expertise required: High
- Suggested assistance: Administrator, Tutor
- Target skills: Technical skills, Interpersonal Skills, Communication, Collaboration, Problem solving, Practical skills, Reflective thinking

Web Template

Inherited: See parent pattern COURSE.

Examples

At the pattern's source department, PROJECT-BASED LEARNING COURSE has not yet been applied as a *standalone* course scenario as outlined in this pattern, but PBL as a didactic approach is applied frequently (see the PROJECT-BASED LEARNING and LAB COURSE patterns).

Evaluation

Not available: refer to pattern COURSE and PROJECT-BASED LEARNING.

Remarks

As this is a pure composite pattern it does not define an own *Structure* section.

References

Kraft, N. (2003). *Criteria for Authentic Project-Based Learning.* Retrieved Dec 22, 2003, from http://www.rmcdenver.com/useguide/pbl.htm

San Mateo County Office of Education. (2001). *Why Do Project-Based Learning?* Retrieved Dec 22, 2003, from http://pblmm.k12.ca.us/PBLGuide/WhyPBL.html

Seminar

Package: *Course Types*

Intent

Increase active participation in an otherwise presentation-centric seminar by Exchange of Contributions, short presentations with longer discussions in the Presentation Phases, and Blended Evaluation of contributions in the Assessment Phases.

Motivation

The typical seminar scenario (at least at the University of Vienna) includes an initial meeting, where the thematic context as well as topics to be elaborated and/or presented, are discussed. Subsequently, each participant (or team) chooses a topic to elaborate from a list of predefined topics. Afterwards there is usually a period of "silence" until the final meeting, which is used for extensive presentations as well as for handing over written reports. Such a scenario has advantages but also many drawbacks: Speaking from personal experiences, observations, and exchange with colleagues, the degree of interaction and cooperation among participants is kept at a minimum, especially when participants have to work out reports individually. Verbal interactions only take place during presentation meetings when presentations are discussed. Additionally, as the focus is mainly on comprehensive presentations, not much time and energy is left for inspiring discussions. The primary target does not seem to be learning from contributing, from peers, and from the elaboration process, but rather to elaborate and present material selected solely the instructor.

A person-centered seminar as described by this pattern addresses the shortcomings of such typical conventional seminar scenarios:

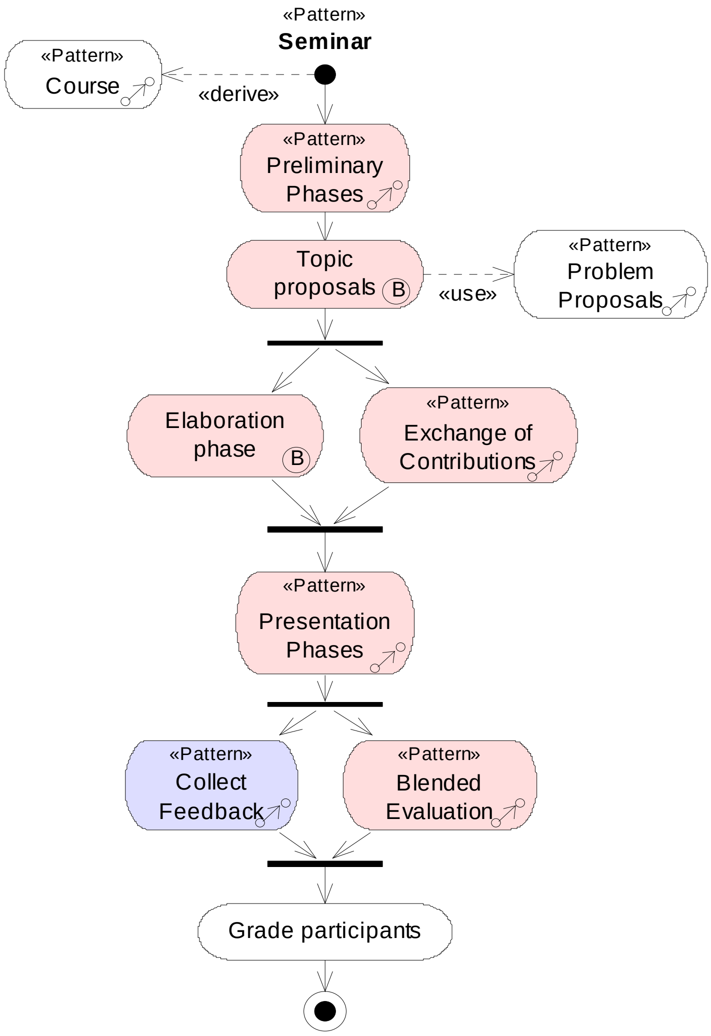
- Within a certain thematic context, which is either set by the curriculum or deliberately set by the instructor, participants are provided with a certain degree of freedom in selecting or proposing topics within that context, stemming from their personal interest or curiosity. These topics need *not* be defined in the initial meeting. To allow for such a proposal process, relevant material, resources, and possibly some predefined topics related to the thematic context of the seminar are provided by the instructor.

- Interaction and cooperation is furthered by a subsequent Exchange of Contributions scenario with an optional additional Meeting, and optionally by including any suitable *Interactive Elements* such as Online Discussion or Theory Elaboration on certain topics.

- Participants are asked to prepare themselves for the presentation meetings by reading/viewing presentations, resources, and reports of their peers *prior* to the meetings (see Presentation Phases). This way, participants come along with enough background information on their peers' topics to allow for shorter presentations, followed by engaging in longer, more insightful and fruitful discussions.

- Participants are actively involved in the ASSESSMENT PHASES by providing SELF- and PEER-EVALUATIONS, as well as feedback regarding the seminar to the instructor.

Sequence



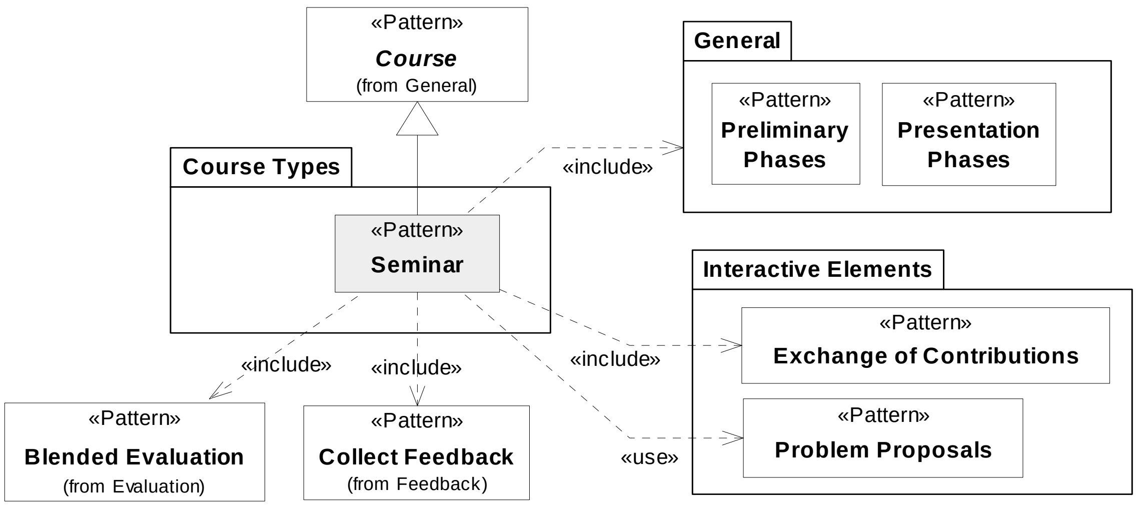
Activity	Description
PRELIMINARY PHASES	In addition to the scenario described in the INITIAL MEETING pattern (which is included in PRELIMINARY PHASES), some predefined topics are offered to participants for elaboration, and the subsequent PROPOSAL phase is explained.
Topic proposals	This is a specialized form of PROBLEM PROPOSALS: Participants are free to propose topics for elaboration within the thematic context defined in the prior INITIAL MEETING. The proposals are subsequently APPROVED by the instructor.
EXCHANGE OF CONTRIBUTIONS	Participants are asked to share, exchange, and discuss relevant information and points of contact of their topics on the learning platform. In a

Activity	Description
	subsequent meeting, proposed topics are presented and approved, and evolved issues/problems, as well as further steps are discussed.
Elaboration phase	Participants elaborate reports and other contributions. During this phase, additional INTERACTIVE ELEMENT scenarios may be used to further (online or face-to-face) interaction among participants, e.g., ONLINE DISCUSSIONS anchored to specific topics.
PRESENTATION PHASES	
BLENDED EVALUATION	Presentations, contributions during online phases, and reports of participants are evaluated using a mix of SELF-, PEER-, and INSTRUCTOR-EVALUATION.
COLLECT FEEDBACK	
Grade participants	The instructor assigns a grade for each participant, taking into account relevant evaluation reports of the prior BLENDED EVALUATION scenario.

Structure

Not available.

Taxonomy/Dependencies



Parameters

- Primary pattern author: Michael Derntl
- Primary pattern source: Research Lab for Educational Technologies (University of Vienna)
- Pattern categories: Course type, Composite
- Level of abstraction: Medium
- Scope: Course

- Primary presence type: Blended
- Flexibility: High
  The described scenario allows for a lot of variations, especially by including INTERACTIVE ELEMENTS.
- Level of confidence: 4
- Number of participants: up to 20
  Designing meaningful EXCHANGE OF CONTRIBUTIONS as well as PRESENTATION PHASES may become futile with more than 20 participants.
- Application effort: High
  Of course, fewer participants mean less effort required.
- Level of expertise required: High
- Suggested assistance: Administrator, Tutor
- Target skills: Interpersonal skills, Communication, Technical skills, Problem solving

Web Template

Inherited: See parent pattern COURSE.

Examples

This SEMINAR scenario is (and was) used in many of the PhD seminars at of the pattern's source institution. A quite representative example is the ***PhD Literature Seminar*** that was held in the winter term 2002[508]:

The thematic focus of the seminar was collaboratively determined and assigned by the facilitator and the participants in the INITIAL MEETING. The broad field of *e-learning* was preset as the thematic context by the facilitator, as most of participants' PhD thesis topics showed relations to this field. Regarding CONSIDER CONVENTIONAL STYLE, the participants unanimously agreed that the innovative seminar style be employed. Each participant was asked to publish three documents as first deliverables on the learning platform: His or her GOALS AND EXPECTATIONS in the seminar, the topic and an abstract of his/her PhD thesis, and the topic PROPOSAL of his/her seminar report. Before the next meeting they had to read each other's documents to be prepared for discussion (EXCHANGE OF CONTRIBUTIONS). In the meeting, each participant orally presented his or her thematic focus, and embedded that focus in the seminar's thematic context. During the discussions, many questions and issues arose. Finally, the topic PROPOSALS were approved, however, sometimes in a slightly adapted form, mirroring the instructor's and participants' perspective in the meeting. Each participant was requested to PUBLISH relevant information regarding his/her topic on the platform.

Subsequently, while the instructor set the deadlines for the documents to be published on the platform before the begin of the PRESENTATION PHASES, the participants, in addition to writing their seminar report, had to host ONLINE DISCUSSION forums, each addressing a

[508] See also Derntl and Motschnig-Pitrik (2003a)

major issue of the seminar that was related to the specific topic of the respective participant's PhD thesis. For example, one participant's PhD thesis was about action research; the forum he had to host was titled, "*How can action research techniques be used in e-learning research?*"

During the *elaboration phase*, participants elaborated reports and discussion contributions. The facilitator offered the participants to host an additional informal MEETING, if they felt the need and desire to do so. In fact, no additional meetings took place.

In the PRESENTATION PHASES, participants uploaded their contributions one week prior to the presentation meeting, so that participants could prepare themselves for their peers' presentations and subsequent discussions. Contrary to traditional settings, there were short presentations of about 15 – 20 minutes followed by intensive discussions. Each participant's PhD mentor has been invited to join the presentation meeting. The vast majority of participants' feedback on this meeting mode was positive.

Finally, to COLLECT FEEDBACK, participants were offered facilities to provide their feedback on the seminar as well as on the learning platform online in a separate FEEDBACK FORUM.

Evaluation

The reactions collected in the FEEDBACK FORUM of the PhD Literature Seminar (winter term 2002; see *Example* section) were predominantly positive, and very encouraging. For example, regarding the PRESENTATION PHASES and the use of the learning platform, one student wrote:

> "I liked using the learning platform very much, because I had the opportunity to gain deeper insight into the topics elaborated by the other participants. The idea to have short presentations and long discussions turned out to be very effective: Because of the exchange of viewpoints in the discussions following the presentations everyone could get more into the other topics, contrary to the one-way communication predominant at long presentation sessions..."

Remarks

This pattern does not define an own *Structure* section as it is mainly composed of other patterns.

References

Derntl, M., & Motschnig-Pitrik, R. (2003). *Employing Patterns for Web-Based, Person-Centered Learning: Concept and First Experiences.* Proceedings of ED-MEDIA 2003 - World Conference on Educational Multimedia, Hypermedia & Telecommunications, Honolulu, HI, USA.

5.3 Evaluation

BLENDED EVALUATION

Package: *Evaluation*

Intent

Use a mix of SELF-, PEER- and INSTRUCTOR-EVALUATION to actively involve participants in the ASSESSMENT PHASES and to take into account as many views on participants' contributions as possible.

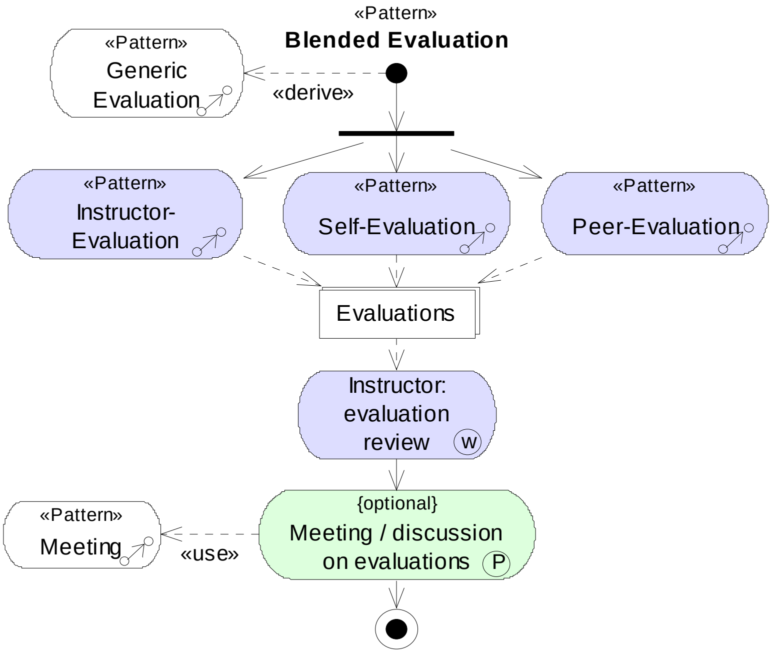
Motivation

> *The learner is the primary evaluator of the extent and significance of student learning, although this may be influenced and enriched by caring feedback from other members of the group and from the facilitator.*[509]

This pattern describes a special form of GENERIC EVALUATION, including SELF-EVALUATION as well as PEER-EVALUATION, as this combination is a frequently used form to evaluate participants' contributions in a student-centered mode. It enables the collection of multiple views on contributions and achievements, and it can be applied in any learning activity through which participants produce contributions that are open for review by their peers.

[509] Rogers and Freiberg (1994, p. 213)

Sequence

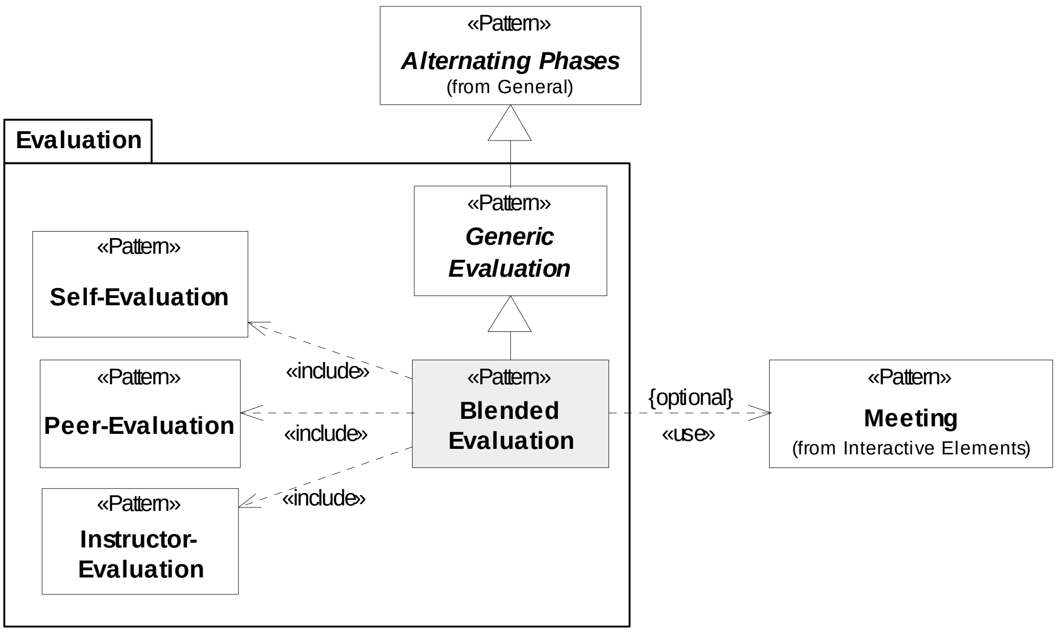


For a description of the activities, refer to parent pattern GENERIC EVALUATION.

Structure

Inherited.

Taxonomy/Dependencies



Parameters

- Primary pattern author: Michael Derntl

- Primary pattern source: Research Lab for Educational Technologies (University of Vienna)
- Pattern categories: Composite, Utility, Motivational
- Level of abstraction: Low
- Scope: Phase
- Primary presence type: Blended
- Flexibility: Low
- Level of confidence: 5
- Application effort: High
- Level of expertise required: Medium
- Person-Centered variables addressed: Acceptance, Transparency, Understanding
- Suggested assistance: Administrator, Tutor
- Target skills: Reflective thinking, Communication, Collaboration
  Many different ways of aligning different evaluation methods may be identified.
- Input: Evaluation targets, evaluation guidelines
- Output: Evaluation reports

Web Template

Similar to parent pattern GENERIC EVALUATION, the administration view simply links the user to the administration of the included patterns SELF-EVALUATION, PEER-EVALUATION, and INSTRUCTOR-EVALUATION. The participant view and the report view (see the *evaluation review* activity) are provided by these included patterns.

Examples

In ***Web Engineering*** a mix of oral INSTRUCTOR-EVALUATION combined with online SELF-EVALUATION and PEER-EVALUATION of LEARNING CONTRACTS was used in the ASSESSMENT PHASES of the course. For additional examples, refer to parent pattern GENERIC EVALUATION.

Evaluation

Not available.

References

Rogers, C. R., & Freiberg, H. J. (1994). *Freedom to Learn* (3rd ed.). Columbus, Ohio: Charles E. Merrill Publishing Co.

EVALUATION

Package: *Evaluation*

Intent

Evaluation is used to produce valuing assessment of a participant's learning performance. It generically characterizes scenarios that may be used collateral to learning activities as well as in the ASSESSMENT PHASES of a course.

Motivation

Evaluation of participants based on their performance and contributions (i.e., the *evaluation targets*) is one central aspect of any educational course setting. Chronologically, it is most often located in the ASSESSMENT PHASES – thus, at the end – of a course, but it may also be used anywhere in the sequence of a course or learning activity where evaluation is reasonable and situated.

Basically, two different methods of *producing* evaluation targets are possible:

- ***Oral***: evaluation targets are produced face-to-face, e.g., in oral examinations or colloquia, or in presentation meetings.
- ***Written***: written evaluation targets are – as opposed to orally produced targets – *tangibly* available to the evaluator. Two sub-methods exist:
  - Present or distant (*offline*): tangible evaluation targets are produced offline and thus in a presence scenario, e.g., written EXAMINATIONS or reports.
  - *Online*: Evaluation targets are produced online, e.g., ONLINE DISCUSSION contributions or online EXAMINATIONS such as multiple-choice tests.

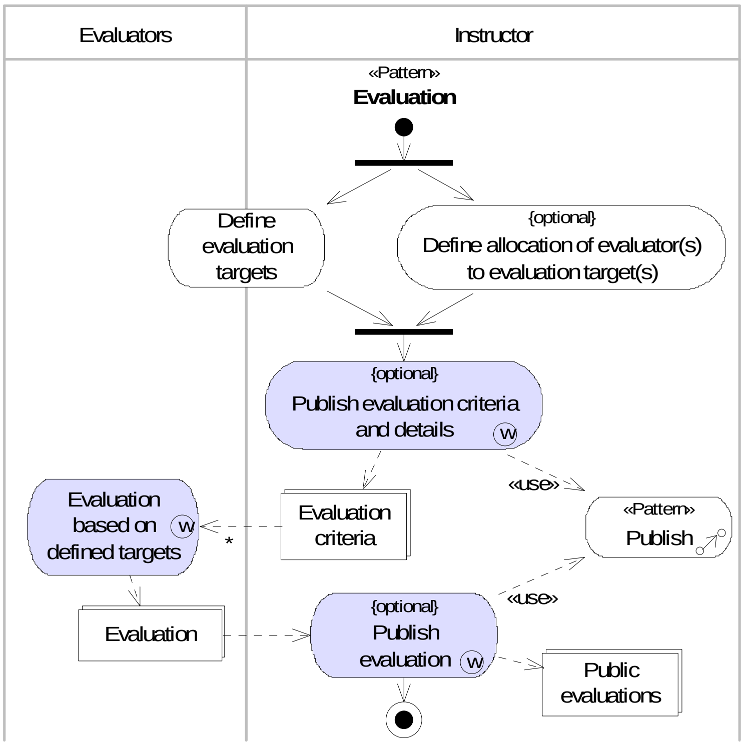
The "production method" of the evaluation targets as described above is not to be specified within the scope of EVALUATION. Instead, it is given by the learning activity through which evaluation targets are produced. This is different for the actual method of evaluation, which has to be explicitly defined by the instructor. For each distinct evaluation scenario, it can be one of the following:

- ***Oral***: oral evaluation is a frequently used form of evaluation. It is feasible in face-to-face settings, e.g., oral evaluation of EXAMINATIONS or oral evaluation of other contributions. Oral evaluation is also possible in pure distant settings, yet only when appropriate video and/or audio conferencing tools are available (this particular case will not be considered here).
- ***Written***: evaluation is provided in a written, tangible way, which can be:
  - Present (*offline*): for example, a certificate.

- *Online*: this kind of evaluation is provided online, e.g., by completing an evaluation form which is published to the producer of the evaluation target. This can also be an automatically generated evaluation, e.g., immediately after completing an online test.

EVALUATION describes an abstract scenario intended to be refined by more specialized patterns. However, as Person-Centered *e-Learning* is the target domain, the EVALUATION sub-patterns in this repository concentrate on Web-supported evaluation processes.

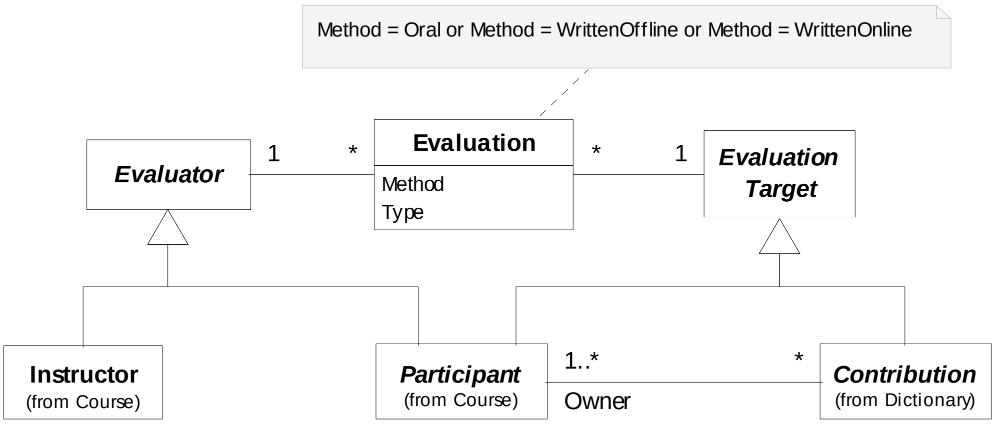
Sequence



Activity	Description
Define evaluation targets	The instructor, possibly collaboratively with the evaluators (students), defines what is target to evaluation. Targets are mostly specific contributions of participants.
Define allocation of evaluator(s) to evaluation target(s)	The instructor always defines who the evaluator is. The actual allocation has to be refined by sub-patterns, whereas there are three possible evaluators: self, peers, or the instructor.
Publish evaluation criteria and details	In order to transparently provide the participants with information, evaluation criteria, and guidelines, the instructor should publish the evaluation details. This is always advisable, and is explicitly required when participants evaluate themselves or their peers.

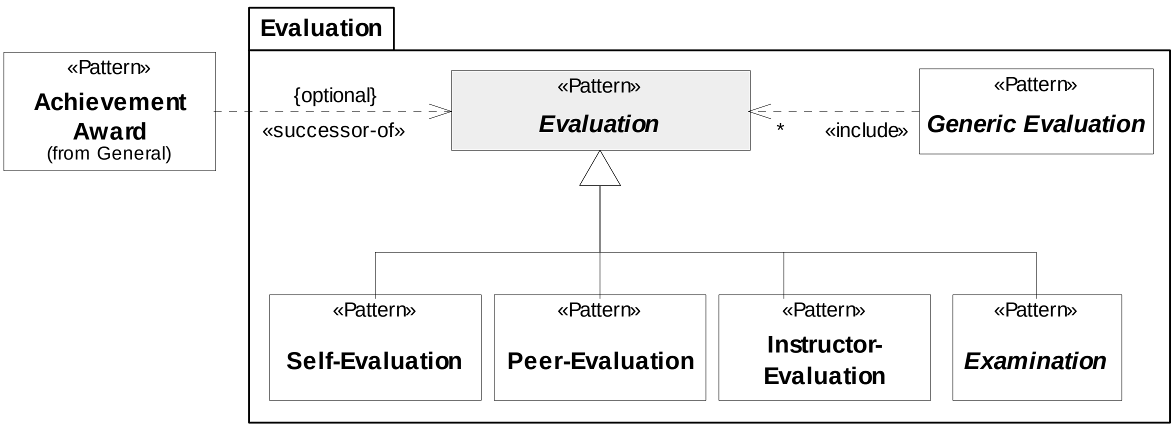
Activity	Description
Evaluation based on defined targets	According to the procedure defined in the activities above, the evaluators execute the evaluations. This is a generic Web-supported activity which may be refined by sub-patterns.
Publish evaluations	Depending on the concrete form of EVALUATION as well as on an eventually negotiated procedure, evaluations may be PUBLISHED to evaluated participants.

Structure



EVALUATION defines three structural elements. The *evaluator* is an abstract entity capable of supplying *evaluations*. An evaluation is linked to exactly one *evaluation target*, whereas the *method* of evaluation is either written (online/offline) or oral as described in the *Motivation* section above. This pattern does **not** define who the evaluators are, what method is used, and what the evaluation targets are.

Taxonomy/Dependencies



EVALUATION is the central pattern in the *Evaluation* package and it is related to a number of other patterns:

- As it is an abstract, generic pattern, it is refined by sub-patterns which define more concretely the roles of evaluators and evaluation targets: SELF-EVALUATION, PEER-EVALUATION, EXAMINATION, and INSTRUCTOR-EVALUATION.
- GENERIC EVALUATION provides a mix of possible evaluation scenarios.
- In scenarios where outstanding participants or contributions are awarded with an ACHIEVEMENT AWARD, results from EVALUATION may be used to determine the winner(s).

Parameters

- Primary pattern author: Michael Derntl
- Primary pattern source: Research Lab for Educational Technologies (University of Vienna)
- Pattern categories: Generic, Collateral, Utility
- Level of abstraction: High
- Scope: Activity, Phase
- Flexibility: Low
- Level of confidence: 4
- Input: Evaluation method, criteria, and guidelines
- Output: Evaluations

Web Template

This pattern supplies Web templates for written, online evaluations, which may be reused and/or redefined by sub-patterns.

Generally note that evaluation scenarios are almost always linked to learning activities which produce documents that are subject to evaluation. See for example the connection to the TEAM WORKSPACES Web template, where the administrator can select some evaluation form to be hyperlinked from the dedicated evaluation folder. Standalone evaluation scenarios are obviously useless, unless for examinations, or when the evaluation is linked to a whole course.

The activities performed by various actors in EVALUATION scenarios are depicted in the use case diagram in Figure 58. If the evaluation is linked to contributions produced by some learning activity, participants may browse these contributions. Evaluators can complete the evaluation form. The results of these evaluations can be viewed by creators/owners (the *targets*) of the respective contributions. Only if the results are configured to be public, all participants can view these results. The administrator configures the evaluation, which includes construction of the evaluation form to be completed by evaluators. Finally, instructors can view aggregate and detailed reports of the evaluations.

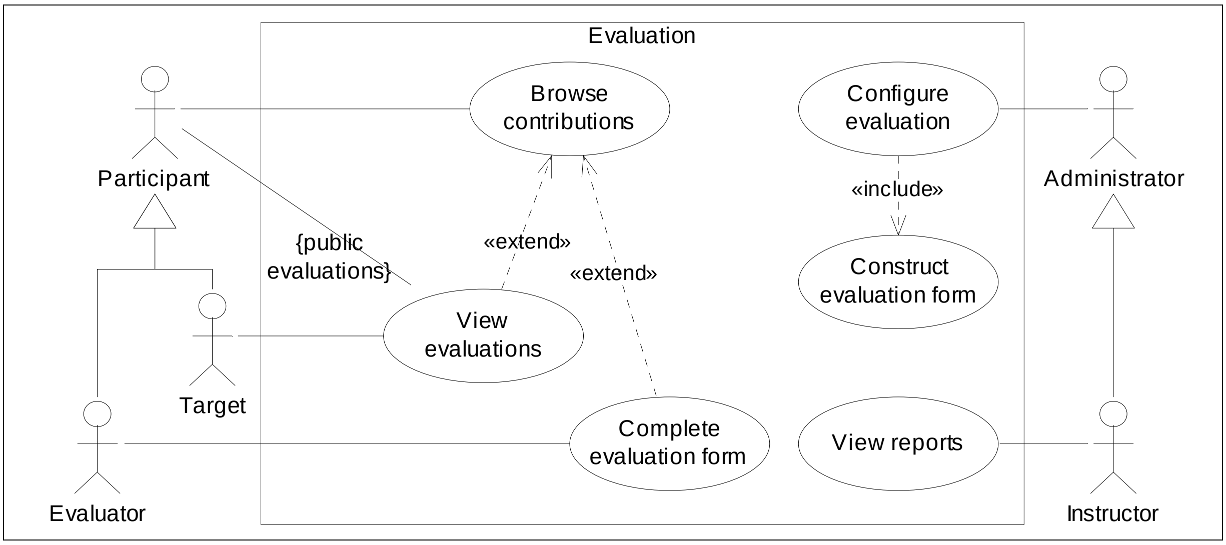


Figure 58: Use case diagram showing supported activities in online EVALUATION scenarios.

Administration View

This view supports the use cases "*Configure evaluation*" and "*Construct evaluation form*". The general configuration step (Figure 59) includes the following settings:

- *Evaluation mode*: Used to determine whether *individual* participants or *teams* of participants have to act as evaluators.
- *Public evaluations*: If the evaluations are configured to be public, every participant may view the results, even those of their peers.
- *Anonymous*: Specifies whether viewing the evaluations includes the name of the evaluator to be visible to the viewer.

The second configuration step is used for construction of the actual evaluation form page. This step uses the QUESTIONNAIRE form editor to allow the administrator to construct the evaluation form or to select an already existing form for reuse. Note that the introductory block of the evaluation form should be used for providing evaluation criteria and guidelines. Refer to the QUESTIONNAIRE Web template for details on the form editor[510]. It might also be useful to provide some predefined, simple evaluation forms, from which the administrator can choose, instead of constructing a new form. The following simple forms have been used frequently in previous courses (see the *Examples* sections of SELF-EVALUATION and PEER-EVALUATION for screenshots/links):

1) *Text box*: Just one large text box for written evaluation
2) *Text box and points*: One large text box for written evaluation and a single-choice block for choosing bonus points on a numeric scale.

[510] See page 233

3) *Multiple text boxes*: A number of text boxes, each used for providing comments on specific aspects of the contributions.

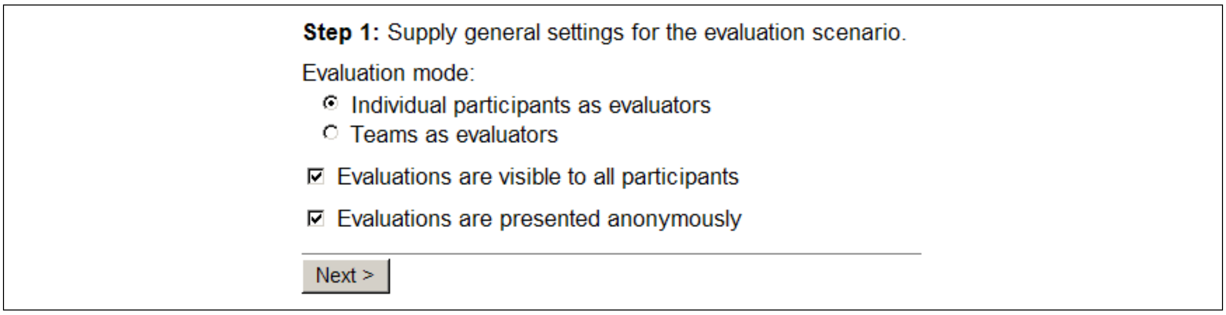


Figure 59: General configuration step for EVALUATIONS.

Participant View

This view supports the use cases "*View evaluations*" and "*Complete evaluation form*", while the "*Browse contributions*" use case has to be supported by the learning activity that produces the contributions to be evaluated. Typically, the respective page carries hyperlinks to the evaluation form and (public) evaluation results.

- *Complete evaluation form*: This page is shows the of the evaluation form, which was previously constructed by the administrator, as a Web form. It consists of a heading, an introductory text including evaluation criteria and guidelines, and the evaluation form itself. All these sections are configured in the administration view through the QUESTIONNAIRE form editor. Refer to the participant view of the QUESTIONNAIRE Web template to see how the form is presented to evaluators.

- *View evaluations*: This page is a simple report, whose content is determined by the settings made in the general configuration in the administration view. It shows the evaluations grouped by evaluation targets and evaluators. Depending on the configuration settings, the evaluations may remain anonymous. In case of a SELF-EVALUATION, an INSTRUCTOR-EVALUATION, or a non-public PEER-EVALUATION scenario, only the evaluations concerning the currently logged in participants are presented. In case of a public PEER-EVALUATION, all evaluations are presented. See Figure 60 for a general layout template for this page, including filtering options for evaluation targets and evaluators.

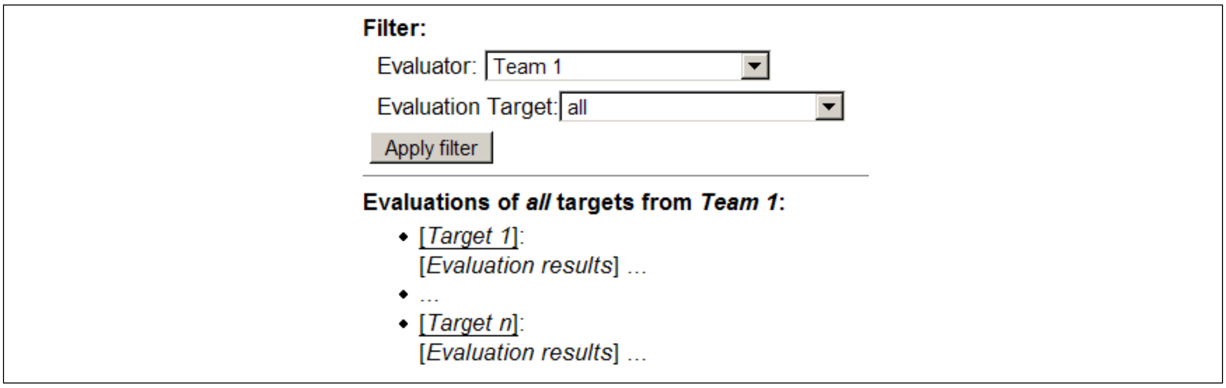


Figure 60: Page layout template for viewing Evaluation results.

Report View

The "*View reports*" use case allows the instructor to access the following reports:

- *Evaluation targets*: Shows submitted evaluations grouped by evaluation targets; includes filter options for restricting the report to specific targets and/or evaluators.
- *Evaluators*: Shows the evaluations grouped by evaluators; includes filtering options as above.
- *Binary*: Allows to download the evaluation results in some machine-processible format, e.g., in comma-separated value (CSV) format.
- *Missing evaluations*: Lists all evaluators who have not yet submitted their evaluations; includes options of notifying evaluators with missing evaluations (e.g., per e-mail or other messaging facilities provided by the learning platform).

Examples

Not available: refer to concrete sub-patterns.

Evaluation

Inapplicable.

Examination

Package: *Evaluation*

Intent

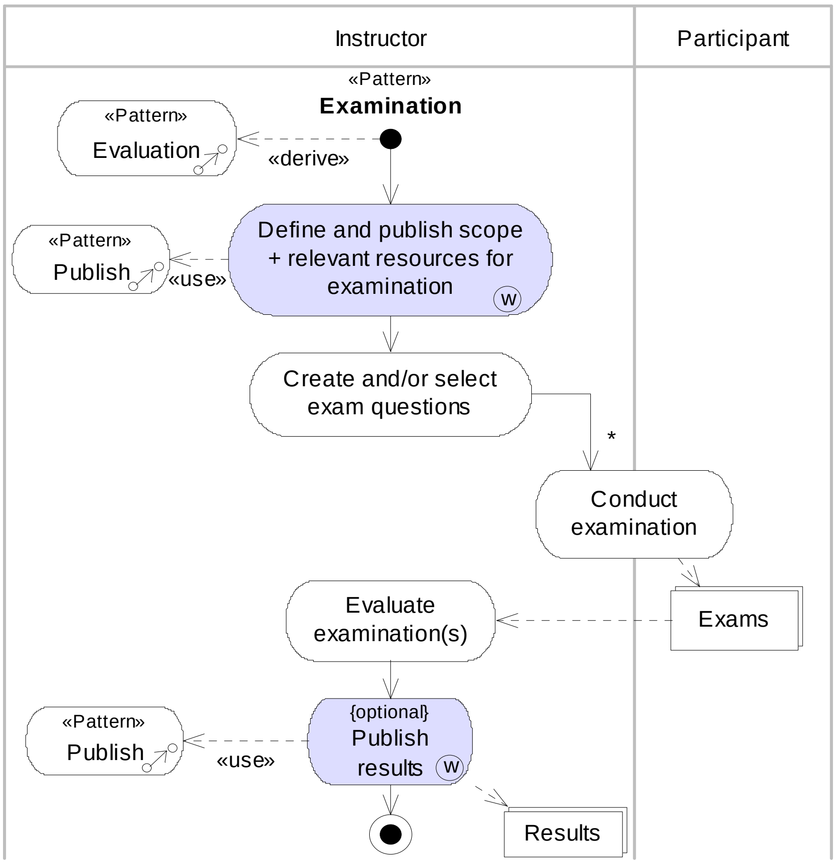
Evaluate participants' learning progress in a structured way by doing oral or written examinations using predefined questions.

Motivation

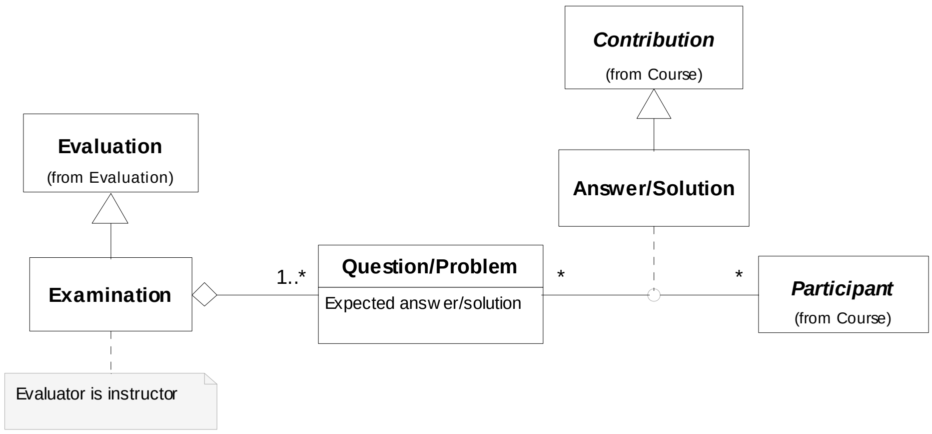
This pattern provides a generic scenario for doing structured examination of participants, which is common practice in many courses independent of the actual educational environment. To aid participants in preparing for the examination, it may be useful to publish relevant resources as well as clues on what the instructor expects them to learn/know. This can be achieved by publishing recent examination questions or by pointing to relevant chapters in the respective content pool.

Generally, EXAMINATION is a form of EVALUATION. The main specialization is that examinations are executed by providing *structured* questions and that the person who is evaluated is not passive during the process as it is the case in EVALUATION. Rather, the participant has to actively contribute oral or written answers/solutions to questions/problems in EXAMINATION scenarios. Additionally, in EXAMINATION the instructor *always* takes over the role of the evaluator, either himself/herself (INSTRUCTOR-EXAMINATION) or impersonated by an automated examination processing procedure (SELF-EXAMINATION).

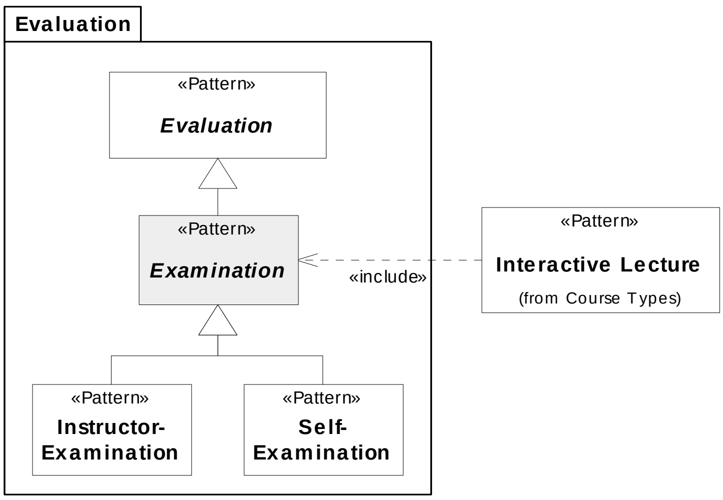
Sequence



Structure



Taxonomy/Dependencies



Parameters

- Primary pattern author: Michael Derntl
- Primary pattern source: Research Lab for Educational Technologies (University of Vienna)
- Pattern categories: Generic, Traditional, Utility
- Scope: Activity
- Primary presence type: Online or Present
  Depending on the method of examination, which may be oral (present) or written (online or present)
- Application effort: Medium
  Depending on the degree of reuse of previous exam questions the effort may be lower
- Level of abstraction: Medium
- Level of expertise required: Low
- Target skills: Problem solving
- Variants: Few
- Input: Relevant resources, Questions

- Output: Examination results

Web Template

To be done. As no form of EXAMINATION was yet used online in our courses, no detailed Web template is provided here. The following information just gives hints on what the Web template views for SELF-EXAMINATION and INSTRUCTOR-EXAMINATION would address, given the examinations proceed online:

- *Administration View*: Needs to provide an online test/quiz creation tool, similar to the QUESTIONNAIRE form editor[511]. It has to provide means for specifying correct results to the test questions to allow for automatic result generation in the participant view. Obviously, this is only possible for test items with closed responses (i.e., single and/or multiple choice), as open questions can not be analyzed automatically.
- *Participant View*: Provides Web view of the online test/quiz. Participants complete the test and, if configured accordingly, get the automatically generated results. If the instructor corrects the tests manually, he/she needs some Web form to provide the results to participants.
- *Report View*: Various types of report could be useful here, e.g., for providing aggregate test results for participants/items and for providing detailed results for participants/items.

In case of an "offline" examination, the instructor/tutor would just PUBLISH relevant material and organizational details for the examination.

Remarks

This pattern does not define *Example* or *Evaluation* sections as it is abstract.

GENERIC EVALUATION

Package: *Evaluation*

Intent

Use INSTRUCTOR-EVALUATION plus any mix of other EVALUATION scenarios in ASSESSMENT PHASES. This allows instructors to involve participants in the assessment process and to collect multiple views on contributions.

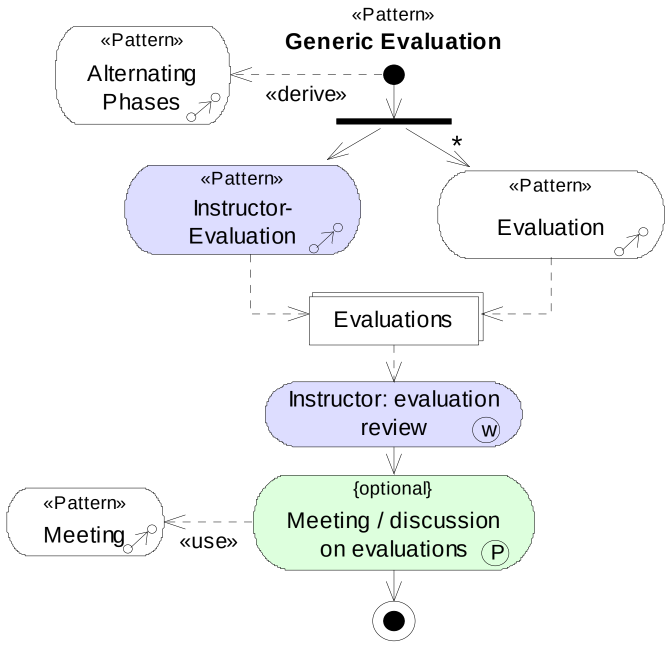
511 See the *Web Template* section of QUESTIONNAIRE on p. 233

Motivation

> *The increasing stress on the examination is [...] mysterious. It has come to be regarded as the most important aspect in education, the goal toward which all else is directed.*[512]

Generic evaluation aims to provide a generic scenario for employing different kinds of EVALUATIONS for participants' contributions. In traditional scenarios, some form of oral or written EXAMINATION is usually the one and only activity in participant evaluation. By employing this pattern, it is possible to produce a greater diversity of evaluations by including INSTRUCTOR-EVALUATION, SELF-EVALUATION and/or PEER-EVALUATION in addition (or as an alternative) to EXAMINATION.

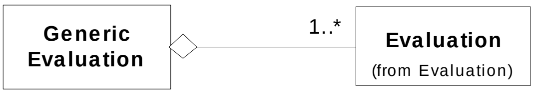
Sequence



Activity	Description
INSTRUCTOR-EVALUATION	
EVALUATION	The instructor has to specify what additional types of EVALUATION are to be employed for evaluating contributions.
Evaluation review	After the evaluation phase has concluded, the instructor can review the evaluation reports for each contribution.
Meeting / discussion on evaluations	Optionally, a concluding MEETING for discussions on the evaluations may be useful to provide oral feedback to participants regarding their contributions.

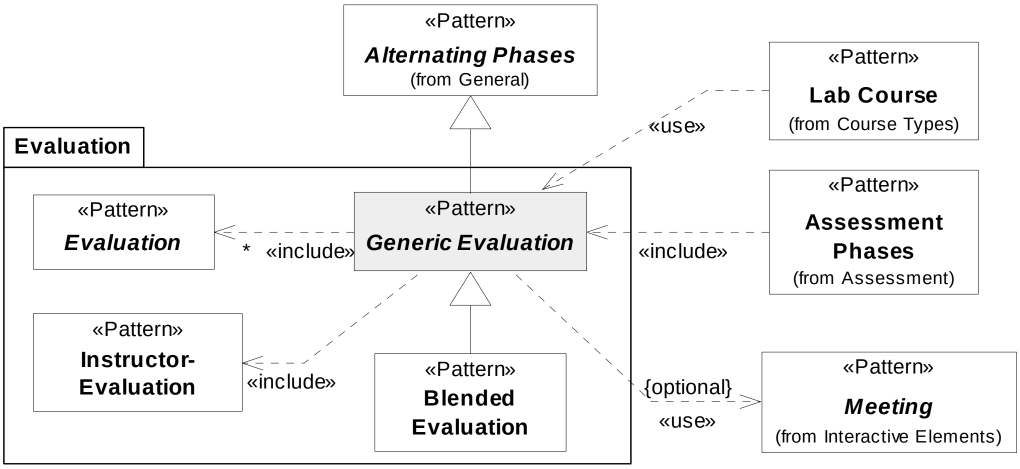
512 Rogers and Freiberg (1994, p. 210)

Structure



In structural terms, a *generic evaluation* is just a collection (or combination) of *evaluations* (instructor-, self-, and/or peer-evaluations).

Taxonomy/Dependencies



Parameters

- Primary pattern author: Michael Derntl
- Primary pattern source: Research Lab for Educational Technologies (University of Vienna)
- Pattern categories: Generic, Composite, Motivational, Utility
- Level of abstraction: Medium
- Flexibility: High
- Level of confidence: 5
- Scope: Phase
- Level of expertise required: Medium
- Target skills: Communication, Reflective thinking
- Person-Centered variables addressed: Acceptance, Transparency, Understanding
- Input: Evaluation methods to be used
- Output: Evaluation reports

Web Template

As this pattern is a generic composite pattern, there are no participant and report views. The administration view just forwards the administrator to the administration views of the

EVALUATIONS patterns he or she chooses to include in the GENERIC EVALUATION in addition to INSTRUCTOR-EVALUATION.

Examples

This pattern is typically included in the ASSESSMENT PHASES of a COURSE. Therefore it is applied in some form in any course instance which is deriving from COURSE. Concretely, generic evaluation was used in the following courses:

- In the ***Project Management*** courses (Basics and Techniques, Soft Skills) to evaluate team contributions, homework, and reports.
- In ***Person-Centered Communication*** to evaluate reports/papers. The optional concluding MEETING for discussion on the different evaluations was also held: the instructors orally presented the report writers their views (positive and negative aspects) on each report. Additionally, peer-evaluators had the chance to "justify" their evaluations.
- In ***Web Engineering***, BLENDED EVALUATION as a specialized form of GENERIC EVALUATION was used to evaluate the teams' LEARNING CONTRACT contributions and lab projects.
- In the ***PhD seminars***, to evaluate seminar papers.

Evaluation

Not available.

References

Rogers, C. R., & Freiberg, H. J. (1994). *Freedom to Learn* (3rd ed.). Columbus, Ohio: Charles E. Merrill Publishing Co.

INSTRUCTOR-EVALUATION

Package: *Evaluation*

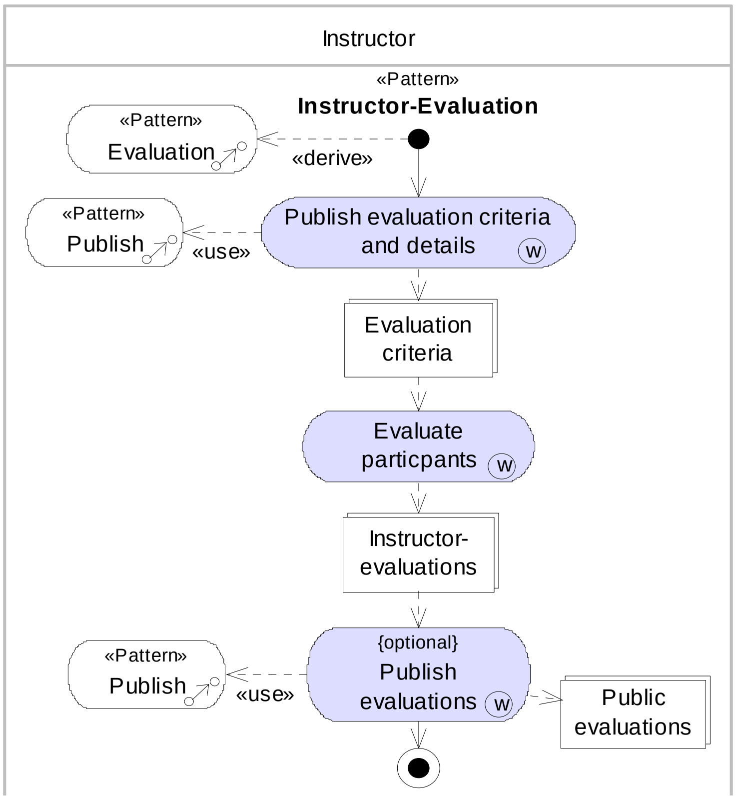
Intent

The instructor evaluates participants' achievements, contributions, and/or performances in COURSES and learning activities. Instructor-evaluation is a necessity in almost any educational scenario.

Motivation

Instructor-evaluation is the prevalent form of assessing participants' achievements in traditional teaching scenarios. The fact that most of the final evaluations in courses never reach the participant they address, is interesting. All that participants usually see is the final grade, which does not convey much information or comments to draw from, or to gain insight. This might be due to instructors who seldom provide evaluation in written form. This pattern proposes a form of instructor-evaluation that makes evaluation criteria as well as the evaluation itself transparent to the participant even in non-face-to-face settings.

Sequence

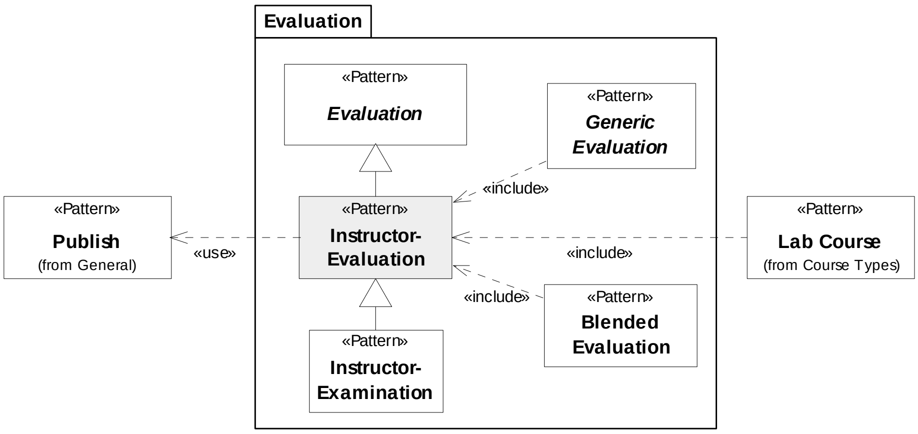


Acticity	Description
Publish evaluation criteria and details	In order to transparently provide the participants with relevant information as well as evaluation criteria and guidelines, the instructor should publish the evaluation details.
Evaluate participants based on targets	The instructor evaluates participants.
Publish evaluations	If an evaluation is produced in written form, it should at least be provided to the respective participant.

Structure

Inherited. The structure is inherited from EVALUATION with the constraint that the *instructor* is the *evaluator*.

Taxonomy/Dependencies



Parameters

- Primary pattern author: Michael Derntl
- Primary pattern source: Research Lab for Educational Technologies (University of Vienna)
- Pattern categories: Utility, Traditional
- Level of abstraction: Low
- Scope: Activity, Phase
  May be at activity scope, e.g., after oral colloquiums, but also at phase scope, e.g., when evaluating written examinations or reports.
- Primary presence type: Online or Present
- Flexibility: Low
- Level of confidence: 5
- Number of participants: Unrestricted
  As instructor-evaluation is a necessity, it cannot be restricted by number of participants.
- Person-Centered variables addressed: Transparency
- Application effort: High
  The application effort may be low when few participants or few evaluation targets are involved.
- Level of expertise required: High
  Technical know-how and fairness is required when evaluating participants.
- Input: Evaluation targets, Evaluation criteria
- Output: Evaluation reports

Web Template

This pattern reuses the Web template of parent pattern EVALUATION. Note that all the settings in the general administration step are preset for this pattern:

- The evaluation mode is *individual* as the instructor is the sole evaluator;
- The evaluations are *non-public*, which means that each evaluation is only visible to the participant or team whom it concerns;
- The evaluations are *non-anonymous*, as the participants know they are evaluated by the instructor anyway.

This means that the instructor only has to construct the desired evaluation form and to submit the evaluations.

Examples

The basic scenario is used in every educational course setting. However, providing written evaluation comments by the instructor has not yet been frequently applied in our teaching activities.

Evaluation

Some points of interest that could be subject to evaluation can be identified. Responses may be collected as any form of feedback, whereas in unstructured scenarios (e.g., REACTION SHEETS) the following items can be formulated as questions and subsequently provided as feedback guidelines. When collected as a QUESTIONNAIRE, an interval scale (e.g., 1 = *not at all* to 5 = *very much*) seems appropriate:

- Explicitly knowing the evaluation criteria helped me a lot to arrange and to distribute my working and learning efforts in a targeted way.
- Receiving written evaluation comments on my achievements from the instructor is important to me.
- I benefited from the instructors written evaluation comments.
- Written evaluations contribute more to my improvement than just receiving a grade.

INSTRUCTOR-EXAMINATION

Package: *Evaluation*

Intent

Evaluate participants using a structured set of questions.

Motivation

This pattern describes a sub-form of EXAMINATION that is frequently used in the ASSESSMENT PHASES of a course or learning scenario. Basically, INSTRUCTOR-EXAMINATION is the default form of EXAMINATION as it is initiated by the instructor and its results are mostly relevant to actual grading of participants. It is provided as a sibling to SELF-EXAMINATION, where the participant themselves initiate the examination process and the results typically do not influence the grades. As attributes of EVALUATION are inherited, INSTRUCTOR-EXAMINATION may take place orally (present) or written (online or present). In traditional scenarios, the variation which is most often used is written, present examination.

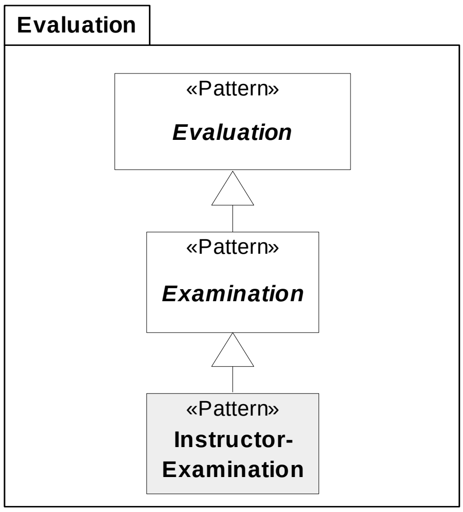
Sequence

Inherited.

Structure

Inherited.

Taxonomy/Dependencies



Parameters

- Primary pattern author: Michael Derntl
- Primary pattern source: Research Lab for Educational Technologies (University of Vienna)
- Pattern categories: Utility, Traditional
- Level of abstraction: Low
- Scope: Activity
- Primary presence type: Online or Present
- Flexibility: Low
- Level of confidence: 3
- Number of participants: Unrestricted
- Application effort: Medium
- Level of expertise required: Medium
- Target skills: Problem solving, Communication
  Communication may be addressed in oral examinations only.
- Input: Examination questions
- Output: Corrections (if written examination), Examination results

Web Template

Not available: see parent pattern EXAMINATION.

Examples

In almost any traditional course setting (apart from seminars), written or oral examinations are the sole activity in the ASSESSMENT PHASES. In this repository, examinations are in most cases used as an optional alternative to more interactive, collaborative scenarios. In such a way it was used in ***Web Engineering***, where participants had freedom of choice among engaging in LEARNING CONTRACTS *or* written EXAMINATIONS.

Evaluation

Not available.

PEER-EVALUATION

Package: *Evaluation*

Intent

Peer-evaluation fosters active participation and engagement of participants, as they take on the roles of peers as evaluators in EVALUATION scenarios.

Motivation

Typically, in traditional course settings, students may not be interested in or come to know work contributed by other students and teams, as long as they get their own work right. Often they do not even know what other students and teams are working on. Only students who come equipped with prior personal interest in the course's subject are likely to be motivated to engage with their own and other students' contributions. Additionally, it is often hard to accomplish for instructors to assess fair grades to the students, taking into account the diverse topics and interests.

In peer-evaluation scenarios participants evaluate certain contributions of each other. The infrastructure for students to communicate the evaluation to the instructor is provided, e.g., via a web form. There are several variants of implementing this pattern in a course. They can be derived from the questions that specify the peer-evaluation process as shown in Table 10. The questions in the left-hand side column aim at certain independent variables of a peer-evaluation solution. The right-hand side column shows possible answers to the questions, which may be adapted by the pattern user and combined arbitrarily to assemble one feasible solution.

Table 10: Possible solutions of implementing PEER-EVALUATION.

Variable	Possible solutions
Who are the evaluators?	Single participants or teams
What is the evaluation method?	Written (offline/online), Oral

Variable	Possible solutions
What kind of evaluation is expected?	Grades, bonus points, structured or unstructured comments, or any combination of these
Are the evaluations visible to peers?	No, anonymously, or with name

What is potentially subject to evaluation depends largely on the type of learning activity that produces the evaluation targets, e.g. designing a website will impose different criteria on evaluation than writing an essay. Additionally, the instructor may define certain aspects and criteria to be evaluated, e.g. focusing on content, ignoring writing style. Imposing criteria even seems recommendable, as some studies[513] stress the rather moderate validity of peer-evaluation results, which means that participants often evaluate their peers using different criteria than the instructor.

As depicted in Table 10, there exist a host of possible peer-evaluation solutions, each feasible, but with different consequences and meaning. For example, consider the following two peer-evaluation solutions to a course with 20 students, organized in teams of four. Say, there were three assignments each team had to work out.

- *Let each team write evaluations of at least two other dedicated teams*: This scenario will result in (five teams) x (two dedicated teams) = 10 evaluations which the instructor has to consider in the evaluation process.
- *Let each participant grade and briefly comment the main contribution of every other student*: Here the instructor will have to take into account (20 students) x (19 peer students) x (one final assignment) = 381 peer-evaluations, which is organizationally nearly impracticable, unless the instructor receives the evaluations in some structured, formatted form.

The revision of a participant's evaluation by the instructor may be postponed until the INSTRUCTOR-EVALUATION of the respective participant, which may reduce the overhead of considering large numbers of peer-evaluations. The latter of the above scenarios shows an extreme, but it reveals the impact of peer-evaluation design decisions on organization and the instructor's schedule in the ASSESSMENT PHASES.

On the students' side, peer-evaluation can have many positive effects:

- Motivation is likely to rise because the students know their work will be reviewed and evaluated by peers. This situation can produce an atmosphere of positive reciprocal stimulation and competition.

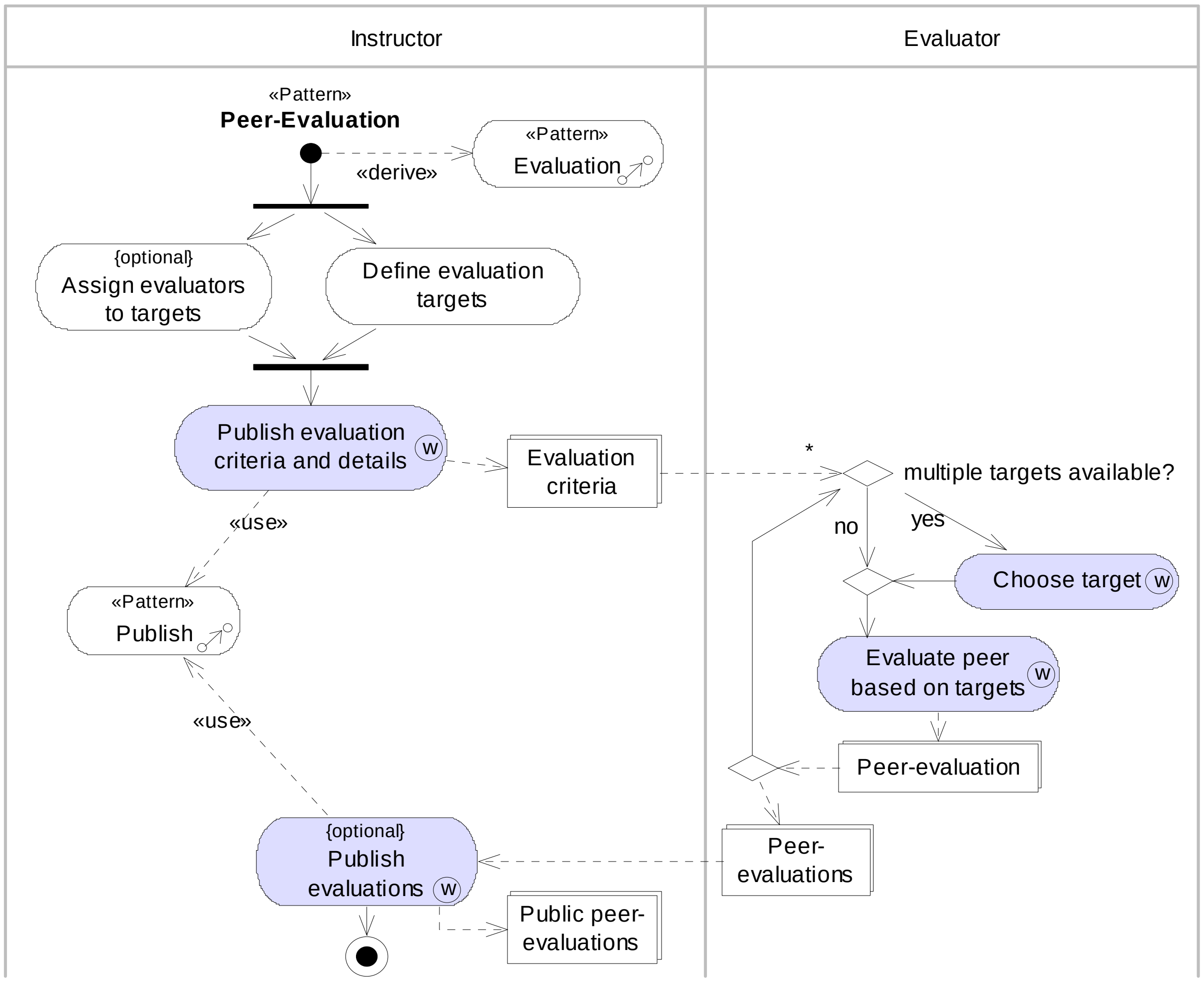
513 For example, Cho and Schunn (2003), Mockford (1994)

- Students are given the opportunity to compare their contributions and evaluations to those of others. This can lead to exchange of ideas on the subjects, what itself may arouse deeper interest in the whole subject area.
- Contributions are considered from different points of view.
- From the perspective of learning, peer-evaluation using comments helps students improve their performances.

On the instructor's side:

- Peer-evaluation provides the instructor with insights and perspectives in what the students think about and expect from each other.
- If introduced accordingly with students' cooperation, it has the potential of supporting fair assessment of a grade as it gives broader, more diverse perspectives.
- Less enjoyable is the organizational overhead that peer-evaluation tends to introduce when not being properly dealt with. Especially in cases with many evaluations among the students, the instructor may require a tutor or assistant to electronically solicit, collect and prepare grades and comments. The first class solution to this problem is structured delivery of grades and comments, e.g. through a suitable Web form that produces and stores data the instructor can process as needed. Given lack of technical learning platform support or only few participating evaluators, simpler means of collecting and processing evaluations may be employed, such as e-mail.
- Implementations of this pattern have shown that soliciting unstructured comments (maybe in connection with a grade or bonus points) yield more perspectives and more meaningful opinions on the evaluation targets than a structured evaluation template approach (see *Examples* section).

Sequence



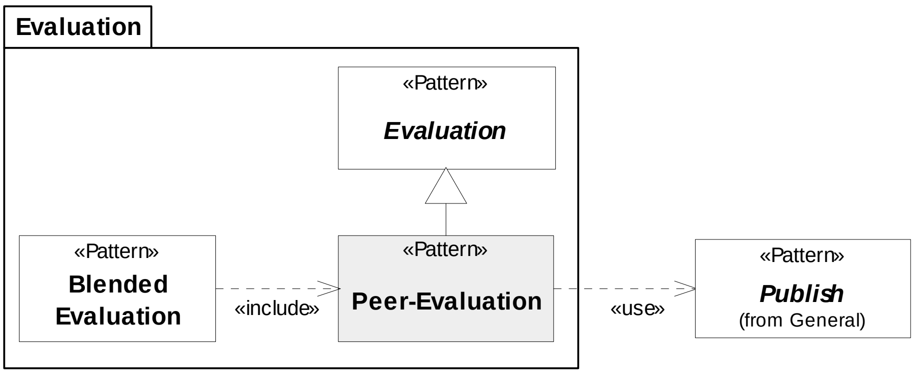
Item	Description
Define evaluation targets	The instructor has to decide which contributions of the participants he or she wants to be evaluated as evaluation targets. For example, in a LEARNING CONTRACT scenario, the elaborations in the scope of a participant's contract may be defined as evaluation targets. There is a web template for online evaluation.
Assign evaluators to targets	The instructor may optionally assign peers (evaluators) to evaluation targets for evaluation. Usually, it is more useful to let the participants choose their evaluation targets, as the meanings und insights of their evaluations certainly increase when they are free to evaluate targets based on their own personal interests. Alternatively, the instructor may assign certain evaluation targets to each evaluator.
Publish peer-evaluation criteria and details	All details the instructor has defined so far have to be made clear to the participants along with guidelines and criteria for evalua-

Item	Description
	tions. Apart from the whole evaluation procedure, it is particularly important to define what kind of evaluations and comments the instructor expects (see Table 10).
Choose target	If the instructor did not assign certain targets to be evaluated by the peers, the participants may now freely choose evaluation targets for evaluation. The instructor should define the expected number of targets each participant has to evaluate.
Evaluate peers based on targets	This activity is inherited from EVALUATION. It is supported by a web template.
Publish evaluations	Depending on what has been negotiated with the participants, the instructor may PUBLISH the evaluations to the participants, either anonymously or with name.

Structure

Inherited from EVALUATION. One constraint is added, namely that the evaluator has to be a different person than the evaluation target or owner of the evaluation target (if it is a contribution).

Taxonomy/Dependencies



Parameters

- Primary pattern author: Michael Derntl
- Primary pattern source: Research Lab for Educational Technologies (University of Vienna)
- Pattern categories: Motivational, Generic, Collateral
- Level of abstraction: Medium
- Scope: Activity, Phase
- Primary presence type: Online, Present
- Flexibility: High
- Level of confidence: 4

- Application effort: High
- Level of expertise required: Medium
- Suggested assistance: Administrator, Tutor
  Assistance is only necessary in online scenarios.
- Target skills: Interpersonal skills, Communication
- Input: Evaluation criteria and guidelines, Evaluation targets
- Output: Peer-evaluation reports

Web Template

This pattern reuses the Web template of parent pattern EVALUATION, which provides all necessary features for peer-evaluation.

Examples

In any of the examples below, the evaluations were anonymous for the recipients, but transparent to the instructors.

Web Engineering: Peer-evaluation was used in the LEARNING CONTRACT scenario in Web Engineering, in the so called *Web Engineering Learning License* (WELL) project. Each WELL participant had to peer-evaluate at least three contracts elaborated by peer teams. They had to provide a written, unstructured evaluation, and could optionally supply 0 – 5 bonus with each evaluation. Evaluation guidelines were published prior to the start of the evaluation phase. The evaluations were used by the instructors to prepare themselves for the final oral EXAMINATION with each participating WELL team. Additionally, the ranking of contracts calculated from bonus points served as input for a subsequently handing over ACHIEVEMENT AWARDS to contributors of outstanding (i.e., best-evaluated) contracts.

The evaluation form can be viewed at:

http://elearn.pri.univie.ac.at/patterns/examples/peer-evaluation/we.html

Person-Centered Communication: Each student had to evaluate seminar reports of at least two teams. Thereby the students had to supply a written evaluation, whereas the following criteria were defined as evaluation guidelines:

- Is the report interesting? Is it readable and well structured?
- What is appealing, what is not so interesting?
- Did the authors include own comments and experiences? Do I find these comments interesting?
- Are there any thought-provoking impulses?

Additionally, the evaluators had to supply 1 – 10 points, whereby 1 = *unsatisfying* and 10 = *very satisfying*.

The evaluation form can be viewed at:

http://elearn.pri.univie.ac.at/patterns/examples/peer-evaluation/pcc.html

Project Management/Soft Skills: Students were organized in five small teams. Each team had to evaluate the work (the sum of all contributions) of all other teams. Each team had elaborated a number of unique, different contributions throughout the course. The following aspects were formulated and published as evaluation criteria:

- Provided resources: Readability, structure, degree of interests/motivation, applicability as soft skill, ...
- Design concept (thread to follow)
- Implementation of the concept
- Overall impression that evolved through the team, the subject, and its moderation in the course module.

The evaluation form can be viewed at:

http://elearn.pri.univie.ac.at/patterns/examples/peer-evaluation/pmss.html

Evaluation

As given in Figure 61, empirical investigation in the Web Engineering case study (in 2003) substantiated that peer-evaluation is perceived as very or rather useful by more than 45.6% of the students, while it is considered rather useless or useless by only 21.4% of the participants. About one third of the participants found the procedure just acceptable. However, more supportive is the fact that more than half of the participants valued reading the evaluations of their own contributions as meaningful or very meaningful (52.9% combined as given in Figure 62), while only 19.1% consider this as meaningless.

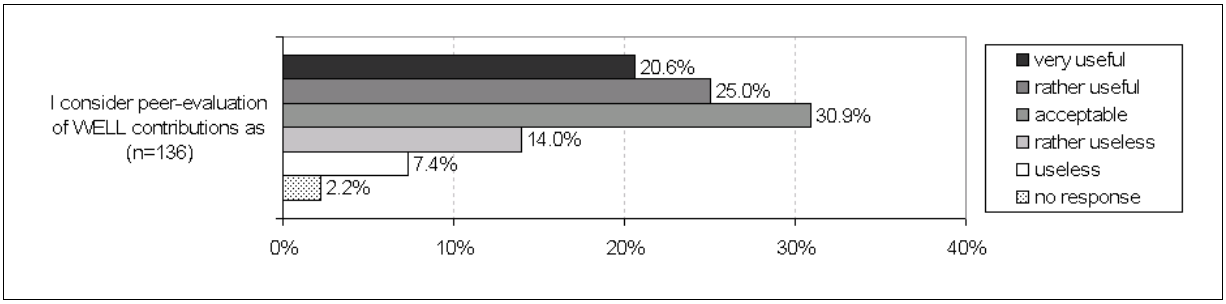


Figure 61: Students' perception of the usefulness of peer-evaluation in Web Engineering 2003.

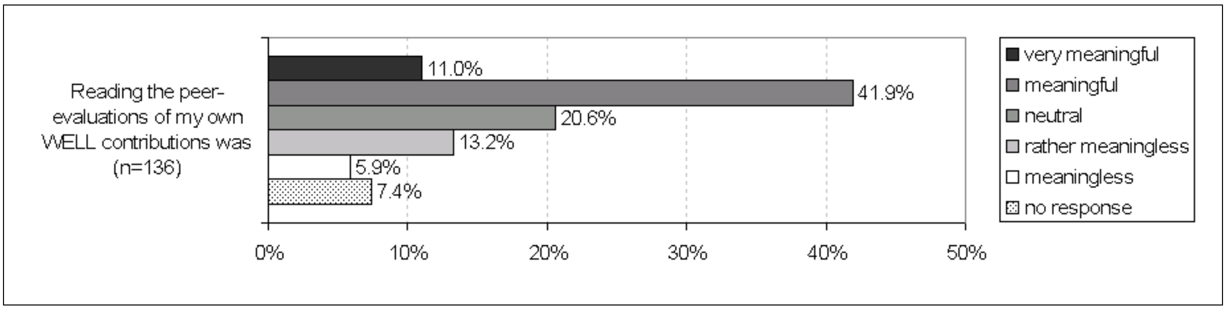


Figure 62: Students' valuing of reading evaluations of their own contributions (Web Engineering 2003.)

Regarding the anonymity of peer-evaluations, the study revealed that participants rather want to their evaluations to appear anonymous to their peers (47.1% as opposed to 14% opting for non-anonymity; see Figure 63).

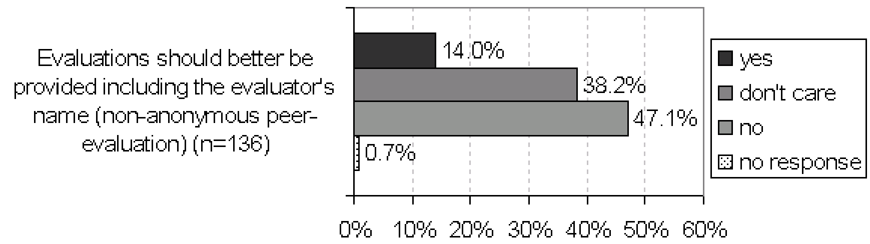


Figure 63: Students' opinion on the anonymity of peer-evaluations.

Other studies

Wen and Tsai[514] have investigated students' perceptions of and attitudes toward online peer-evaluation using 4 subscales with responses ranging from 1 (*strongly disagree*) to 5 (*strongly agree*)[515]:

- Positive Attitude Subscale (PAS): Peer-evaluation...
  - is helpful to my learning
  - makes me understand more about teacher's requirement
  - activities can improve my skills in verbal interaction
  - activities motivate me to learn
  - activities increase the interaction between my teacher and me
  - helps me develop a sense of participation
  - activities increase the interaction between my classmates and me
  - is fair to assess students' performance
- Online Attitude Subscale (OAS): Online peer-evaluation activities...
  - can be timesaving
  - can increase the interaction among classmates
  - can be economical
  - can increase the interaction between the teacher and the students
  - are fair when assessing students' performance
- Understanding-and-Action Subscale (UAS):
  - Peer-evaluation activities help me to understand what other classmates think
  - The teacher should develop criteria of peer-evaluation activities for students

[514] Wen and Tsai (2003)

[515] Note that this reproduction uses the term *evaluation* instead of *assessment* and *mark* (or *grade*).

    - Students should participate in the development of criteria for peer-evaluation activities

- Negative Attitude Subscale (NAS):
    - I think students should not be responsible for making evaluations
    - Peer-evaluation is time-consuming
    - My evaluation giving to classmates are affected by the evaluations given to me
    - If I receive evaluations worse than expected, then I will give worse evaluations to classmates

The results of this study showed that this questionnaire is a highly reliable instrument to assess students' attitudes toward peer-evaluation regardless of level of education. University students generally perceive peer-evaluation in a positive way. Interesting is the fact that students who have prior experience with peer-evaluation tend to have more positive opinions on this technique when compared to students who have never acted as evaluators. Thus, it seems recommendable to establish this evaluation technique as an integral part of blended learning courses.

A study on online peer-evaluation by ***Akahori and Kim***[516], which was conducted in a problem-based learning (PBL) course, revealed some interesting results:

- Novices (i.e., students) tend to show lower appreciation to problem-solving methods and problem proposals than experts (i.e., instructors). Instead they focus more on comprehensive literature references in a report as well as on the validity of the solution.

- Peer-evaluation activities highly promote students' motivations to learn by themselves as well as by modeling from their peers.

- Students show high level of interest and concern towards peer-evaluation, which is especially due to the unusual experience of coming to review peers' reports.

- Contributions that are subject to peer-evaluation are of higher quality (higher score) than reports evaluated traditionally.

- Feedback included many comments about learning how to improve report writing and about how to distinguish different qualities of reports.

Two studies in project-based learning (PBL) by ***Sluijsmans and colleagues***[517] show rather diverse results. Subject to peer-evaluation was the PBL process itself, not its products:

- The question of reliability of peer-evaluation in PBL could not be answered consistently and affirmatively (p. 168).

- Students are positive about peer-evaluation, yet they agree that this technique needs further improvement, perhaps focusing on products rather than on process (p. 169).

---

516 Akahori and Kim (2003)

517 Sluijsmans et al. (2001)

- Students show highly different views and weights on different criteria (p.170).

References

Akahori, K., & Kim, S. M. (2003). *Peer Evaluation Using the Web and Comparison of Meta-cognition between Experts and Novices.* Proceedings of ED-MEDIA 2003 - World Conference on Educational Multimedia, Hypermedia & Telecommunications, Honolulu, HI, USA.

Cho, K., & Schunn, C. D. (2003). *Validity and Reliability of Peer Assessments with a Missing Data Estimation Technique.* Proceedings of ED-MEDIA 2003 - World Conference on Educational Multimedia, Hypermedia & Telecommunications, Honolulu, HI, USA.

Mockford, C. D. (1994). The use of peer group review in the assessment of project work in higher education. *Mentoring and Tutoring, 2* (2), 45-52.

Sluijsmans, D. M. A., Moerkerke, G., van Merrienboer, J. J. G., & Dochy, F. J. R. C. (2001). Peer Assessment in Problem Based Learning. *Studies in Educational Evaluation, 27*, 153-173.

Wen, M. L., & Tsai, C.-C. (2003). *College Students' Perceptions of and Attitudes Toward Online Peer Assessment.* Proceedings of ED-MEDIA 2003 - World Conference on Educational Multimedia, Hypermedia & Telecommunications, Honolulu, HI, USA.

SELF-EVALUATION

Package: *Evaluation*

Intent

Self-evaluation fosters critical reflection on a participant's own contributions and learning progress, as the participant is both evaluator and evaluation target at the same time.

Motivation

> *[Self-evaluation] is one of the major means by which self-initiated learning becomes also responsible learning.*[518]

Self-evaluation is a form of EVALUATION where participants evaluate their own contributions, performances, and achievements throughout a course or learning activity. Reflecting on one's own performance critically and *becoming aware* of the own criteria, strengths and weaknesses requires taking responsibility and is one of the keys to personal improvement and self-initiated learning.[519]

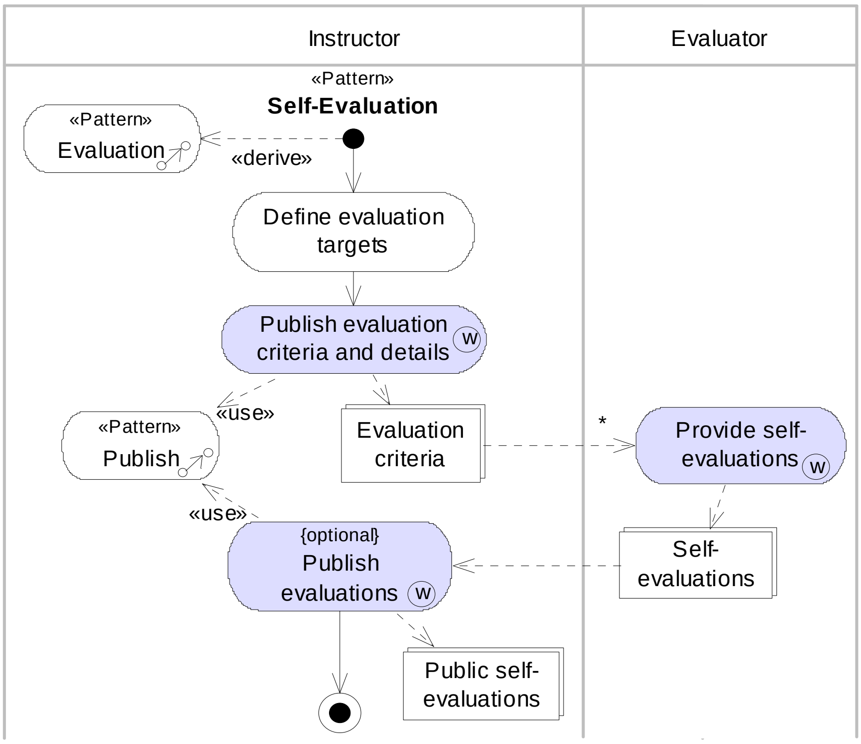
Participants are encouraged to evaluate their own performances and/or specific contributions, whereas there are different ways of implementing self-evaluation. It has to be decided

[518] Rogers (1983, p. 158)

[519] Cf. Rogers (1983, p. 158-159), Rogers and Freiberg (1994, p. 206)

whether the evaluation should be provided orally or written. In online settings, mostly written comments and optionally bonus points or suggested grades will be solicited. The criteria for evaluation may be set by the instructor, but it may also be left to the participants to define their own criteria and measures. Anyway, it is important to PUBLISH how, when, and why self-evaluation is going to be employed, as early and as clearly as possible.

Sequence

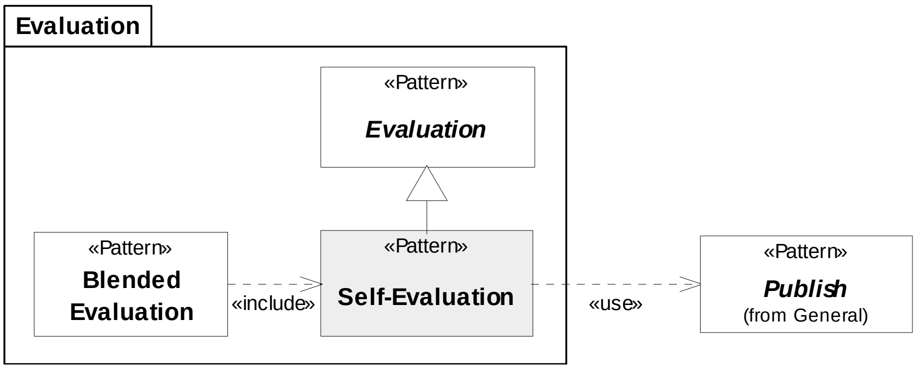


Item	Description
Define evaluation targets	The instructor has to decide what should be target to self-evaluation. For example, in a LEARNING CONTRACT scenario, the elaborations in the scope of a participant's contract may be defined as evaluation targets.
Publish self-evaluation details	In order to transparently provide the participants with relevant information as well as evaluation criteria and guidelines, the instructor should publish the evaluation details.
Evaluate self based on assigned targets	The participants supply their self-evaluations.
Publish evaluations	When written self-evaluations are solicited, the participants have to make their evaluations available to the instructor, and possibly to other participants.

Structure

The structure is inherited from EVALUATION, only one constraint is added: The *evaluator* is the same person as the *evaluation target* or the *owner* of the evaluation target (if it is a contribution).

Taxonomy/Dependencies



Parameters

- Primary pattern author: Michael Derntl
- Primary pattern source: Research Lab for Educational Technologies (University of Vienna)
- Pattern categories: Utility, Motivational
- Level of abstraction: Low
- Scope: Activity, Phase
  In present settings self-evaluation is an activity, while in online settings it typically covers phase scope.
- Primary presence type: Online or Present
- Flexibility: Low
- Level of confidence: 5
- Number of participants: Unrestricted
  In presence scenarios fewer participants are surely preferable, whereas in online scenarios the number of participants just affects the application effort on the side of the instructor
- Person-Centered variables addressed: Acceptance, Understanding
  Acceptance and understanding are required to let the participants actively take part in their own evaluations.
- Application effort: Low
  Putting the pattern into practice requires not much effort, except in online settings with many participants involved.
- Level of expertise required: High
  For self-evaluation to be reflective and beneficial for participants facilitative attitudes are required by the instructor.
- Target skills: Communication, Reflective thinking
- Input: Evaluation guidelines and criteria
- Output: Self-evaluation reports

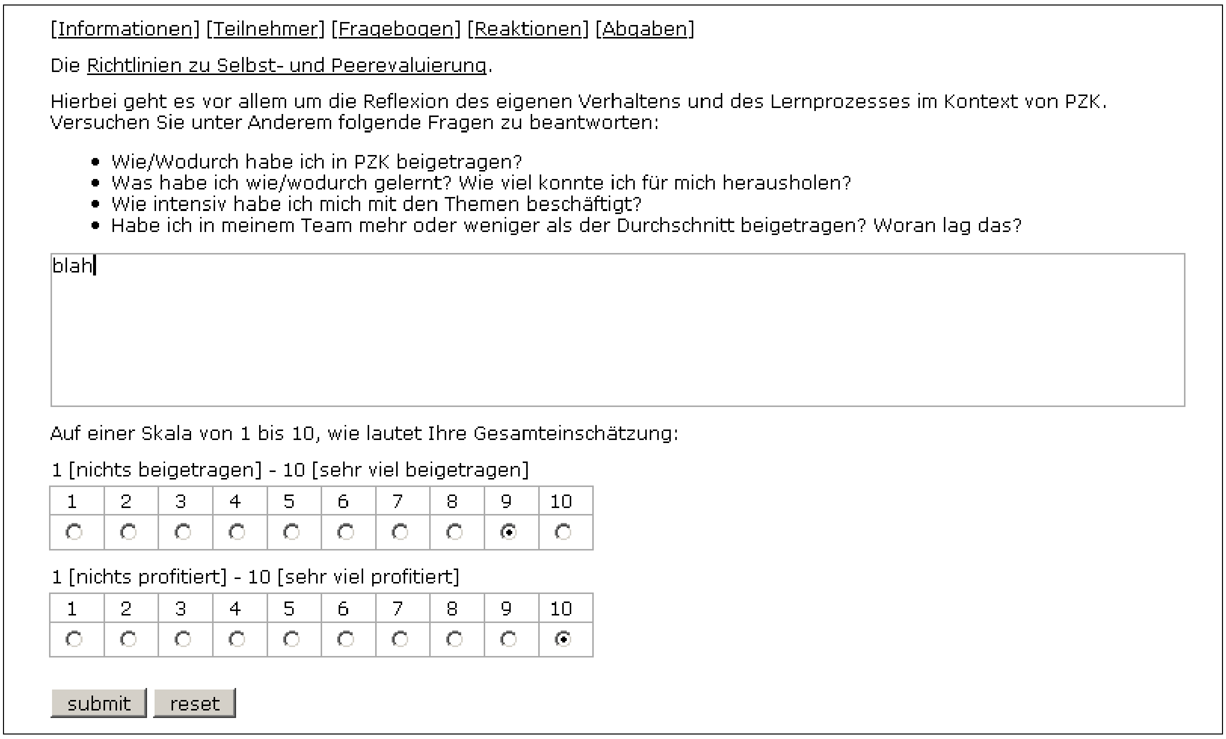
Web Template

This pattern reuses the Web template of parent pattern EVALUATION. Not that, as only one evaluation target is available (=*self*), no contribution overview page is really needed. Still, it may be there to provide a link to the evaluation form. Additionally, the evaluation view should only visible to the instructor and to the evaluator, and not to other participants. This can be configured by making the evaluations non-public in the general administration step.

Examples

Self-evaluation was used to reflect on the learning experience in the LEARNING CONTRACT scenario in ***Web Engineering***, as well as on general aspects of the course on ***Person-Centered Communication*** (see Figure 64 below). An online evaluation mode was employed where teams had to supply written evaluation comments with respect to the following questions/criteria:

- What have I contributed in the course?
- What and how did I learn in the course? To what extent was I capable of drawing value from the learning activities?
- How intensely did I engage with topics and subjects?
- Did I contribute equally, or above/below average when compared with team mates/group members?
- Overall appraisal of contribution and profit on a scale from 1 (low) to 10 (high).


[Informationen] [Teilnehmer] [Fragebogen] [Reaktionen] [Abgaben]

Die Richtlinien zu Selbst- und Peerevaluierung.

Hierbei geht es vor allem um die Reflexion des eigenen Verhaltens und des Lernprozesses im Kontext von PZK.
Versuchen Sie unter Anderem folgende Fragen zu beantworten:

- Wie/Wodurch habe ich in PZK beigetragen?
- Was habe ich wie/wodurch gelernt? Wie viel konnte ich für mich herausholen?
- Wie intensiv habe ich mich mit den Themen beschäftigt?
- Habe ich in meinem Team mehr oder weniger als der Durchschnitt beigetragen? Woran lag das?

blah

Auf einer Skala von 1 bis 10, wie lautet Ihre Gesamteinschätzung:

1 [nichts beigetragen] - 10 [sehr viel beigetragen]

1	2	3	4	5	6	7	8	9	10
○	○	○	○	○	○	○	○	◉	○

1 [nichts profitiert] - 10 [sehr viel profitiert]

1	2	3	4	5	6	7	8	9	10
○	○	○	○	○	○	○	○	○	◉

submit reset

Figure 64: Screenshot of the self-evaluation form in Person-Centered Communication. Participants had to supply written comments, as well as ratings (1-10) of their contributions in the course and of the benefits they have drawn from participating. The form can also be viewed at http://elearn.pri.univie.ac.at/patterns/examples/self-evaluation/pcc.html

In ***Project Management/Basics and Techniques***, participants had to do oral self-evaluations in front of their peers and the instructor. One week prior to this self-evaluation session, participants were asked to prepare themselves for it, taking into account their learning, achievements, and oral as well as written contributions.

Evaluation

Not available.

References

Rogers, C. R. (1983). *Freedom to Learn for the 80's*. Columbus, OH: Charles E. Merrill Publishing Company.

Rogers, C. R., & Freiberg, H. J. (1994). *Freedom to Learn* (3rd ed.). Columbus, Ohio: Charles E. Merrill Publishing Co.

SELF-EXAMINATION

Package: *Evaluation*

Intent

Provide participants with the option of evaluating themselves in a uniform, structured way by providing questions and expected answers.

Motivation

This pattern describes a sub-form of EXAMINATION which may be used by participants to examine *themselves*. Such scenarios are frequently used in online courses by participants to assess their current knowledge, e.g., in self-initiated online tests. Thereby the expected, correct answers to exam questions are specified by the instructor, while the evaluation results presented to the participant are computed automatically.

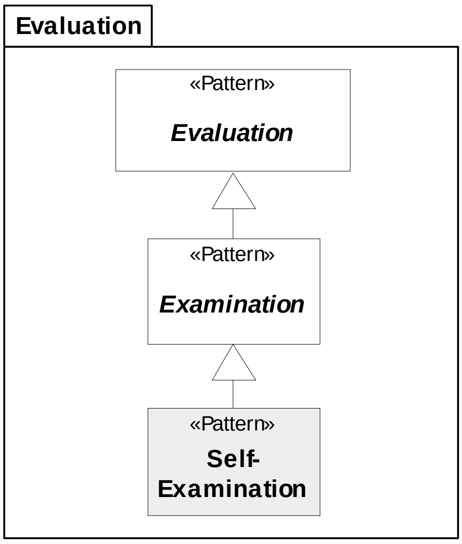
Sequence

Inherited.

Structure

Inherited.

Taxonomy/Dependencies



Parameters

- Primary pattern author: Michael Derntl
- Pattern categories: Utility, Collateral
- Level of abstraction: Low
- Flexibility: Low
- Level of confidence: 4
- Scope: Activity
- Primary presence type: Online
- Number of participants: Unrestricted
- Application effort: Low
- Level of expertise required: Low
- Target skills: Problem solving
- Input: Examination questions
- Output: Examination results

Web Template

Not available: see parent pattern EXAMINATION.

Examples

Not available.

Evaluation

Not available.

Remarks

SELF-EXAMINATION inherits all its properties from EXAMINATION, except for the fact that self-examination is initiated by the participant himself, not by the instructor.

5.4 Feedback

COLLECT FEEDBACK

Package: *Feedback*

Intent

Solicit feedback from participants to enable qualitative analysis and subsequent improvement of the employed learning scenarios.

Motivation

The collection of feedback from participants is a central activity in person-centered e-learning scenarios. It fosters a climate of openness that entails benefits for both the participants and the instructor:

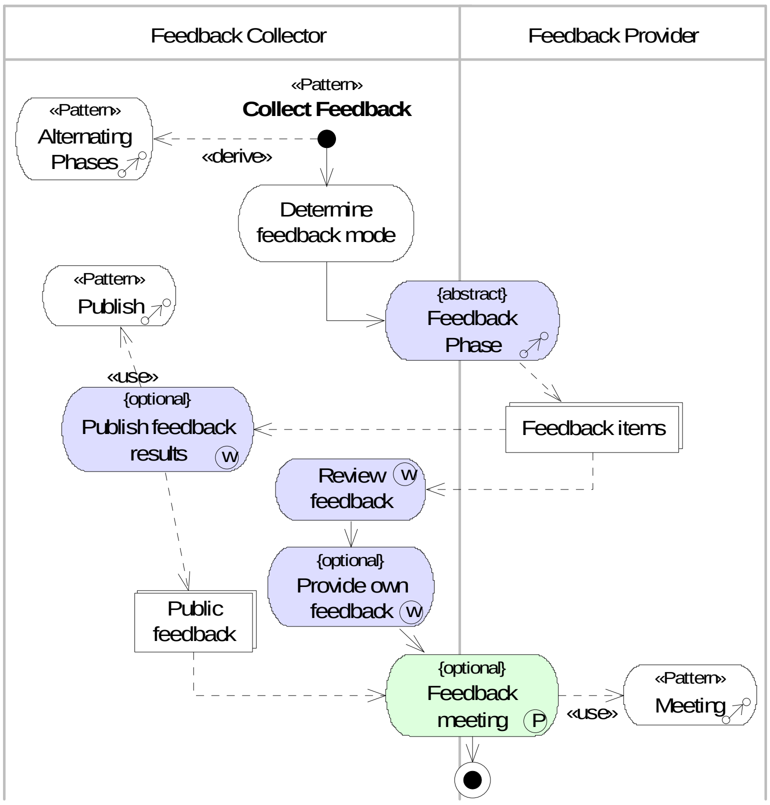
- For participants, providing feedback on (certain aspects of) a course or learning activity activates reflective thinking processes that allow for recapitulation of personal experiences and insights, as well as for transparently explicating personal feelings and opinions.
- For the instructor, collected feedback can be a valuable resource for consequent improvement of the course/scenario: Feedback collected during the course can be reflected upon in following meetings and thus allows for immediate consideration of issues of interest and helps to achieve a common understanding or better feeling of the current process. As such it provides an undistorted "mirror" for the instructor, based on different viewpoints. Additionally, feedback is a valuable asset in the qualitative and quantitative evaluation of the course.

Feedback may be collected online or in face-to-face meetings. This pattern provides an abstract sequence that describes a generic process of collecting feedback online. The actual means of collecting feedback is left to be defined in other patterns. Generally, there are two basic feedback types:

1) Unstructured or semi-structured, open feedback (e.g., REACTION SHEETS, FEEDBACK FORUM)

2) Structured feedback, aiming to survey specific aspects of courses or learning activities (QUESTIONNAIRE).

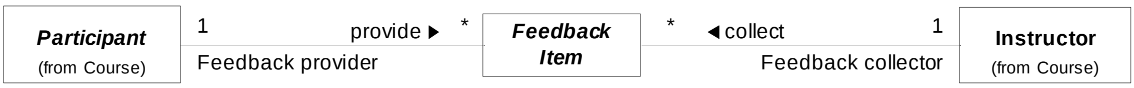
Sequence



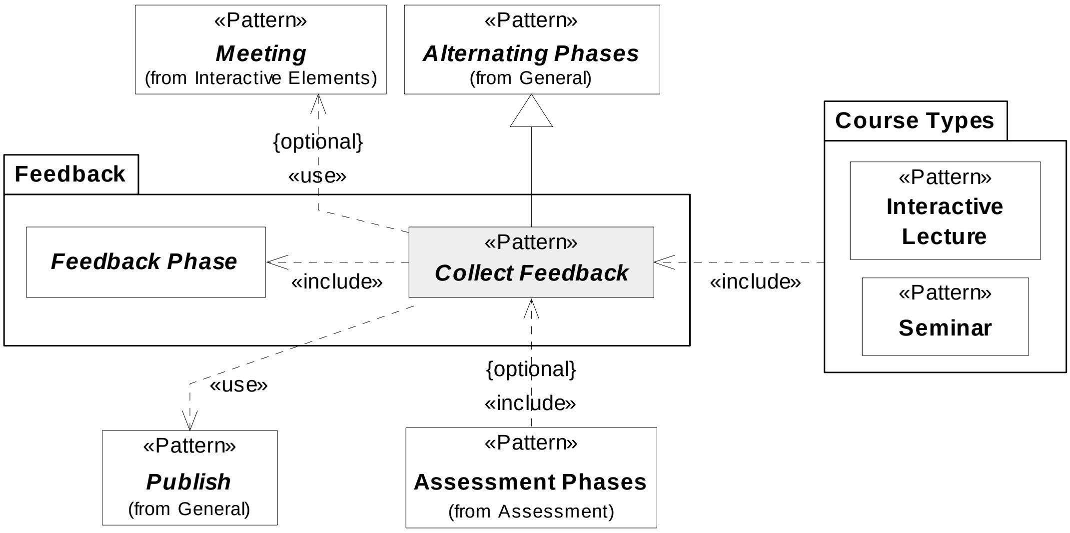
Activity	Description
Determine feedback mode	Participants and the instructor agree upon the mode of collecting feedback, which concretizes the following *feedback phase* activity.
Feedback Phase	This is an abstract placeholder that has to be specialized by patterns describing concrete means of providing feedback, e.g., REACTION SHEETS, FEEDBACK FORUM, or QUESTIONNAIRE.
Review published feedback	Concurrently with the feedback phase, the instructor may review arrived feedback by retrieving the respective feedback report that is described in the *Web Template* section.
Publish feedback results	Optionally, feedback published to the instructor may be published to be accessible to all participants.
Provide own feedback	If appropriate and desired, the instructor may provide own feedback to the participants on the learning platform.
Feedback meeting	To reflect on the feedback process it might be useful to offer a feedback MEETING in which the issues that have been posted as feedback are

Activity	Description
	discussed. Positive effects have been achieved when feedback that had been collected during courses was discussed in a subsequent meeting. This way, opinions and issues can be immediately discussed.

Structure



Taxonomy/Dependencies



Note that the included *feedback phase* activity has to be specified by other patterns. For concrete examples, see the REACTION SHEETS, FEEDBACK FORUM, and QUESTIONNAIRE patterns.

Parameters

- Primary pattern author: Michael Derntl
- Primary pattern source: Research Lab for Educational Technologies (University of Vienna)
- Pattern categories: Utility
- Level of abstraction: Medium
- Scope: Activity, Phase
  If feedback is collected face-to-face, it is an activity; if it is collected online it covers a whole phase.
- Flexibility: Low
- Level of confidence: 5
- Primary presence type: Blended, Online, or Present

Collecting feedback can either be done *online* with an optional, subsequent MEETING for feedback discussion (→ *blended*), or *present* (written or oral).

- Number of participants: Unrestricted
  Collecting feedback is only restricted in oral mode, because only a limited number of participants can be considered.
- Application effort: Low
- Level of expertise required: Low
- Person-Centered variables addressed: Acceptance, Understanding
- Input: Feedback request, Guidance on giving feedback
- Output: Feedback reports

Web Template

This is a pass-through pattern to any pattern that specifies and implements the abstract *feedback phase*, i.e., REACTION SHEETS, QUESTIONNAIRE, and FEEDBACK FORUM. Therefore, the administration view of this pattern just redirects the administrator to the administration view of one of the aforementioned patterns.

There is no separate participant or report view, as these are provided by the concrete feedback patterns.

Examples

Collecting feedback was (and is being) used in every course underlying this pattern repository. Examples of the various forms of feedback provision are described by patterns that concretely specify the *feedback phase* activity: REACTION SHEETS, QUESTIONNAIRE, and FEEDBACK FORUM.

Evaluation

Not available.

FEEDBACK FORUM

Package: *Feedback*

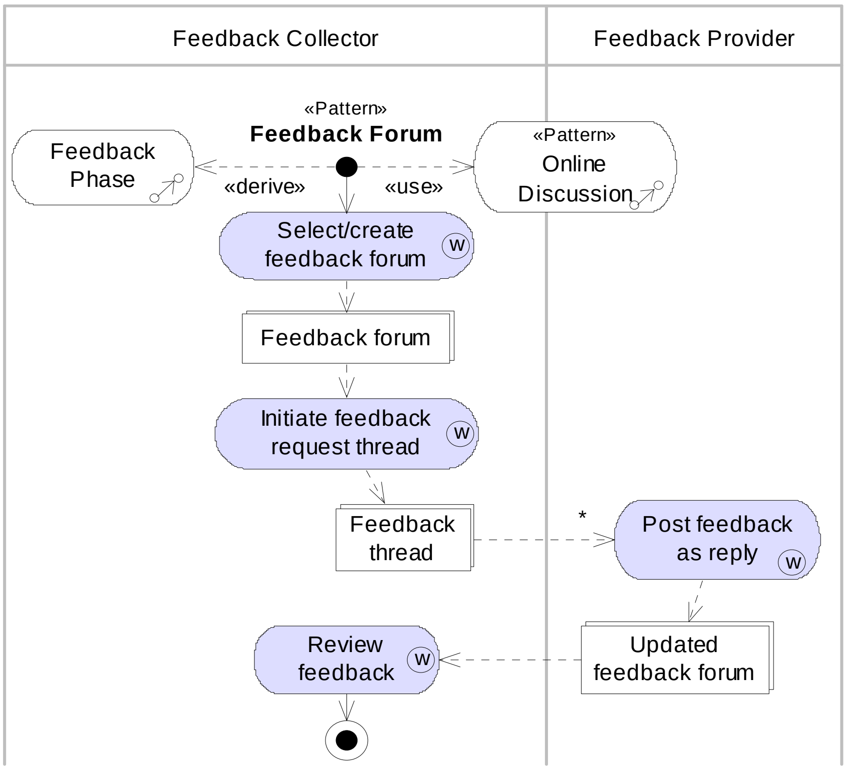
Intent

COLLECT FEEDBACK in a semi-structured way by soliciting postings to instructor-initiated ONLINE DISCUSSION threads. This additionally allows for open discussion of feedback postings.

Motivation

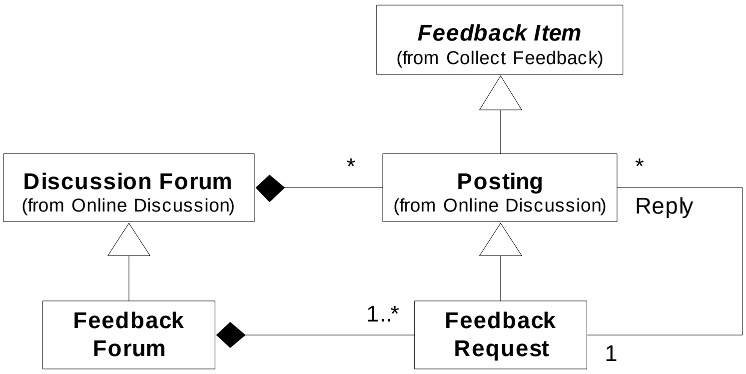
FEEDBACK FORUM describes a concrete form of COLLECTING FEEDBACK. Participants provide semi-structured, written feedback in the form of postings to feedback request threads in an ONLINE DISCUSSION forum. The advantage of using a discussion forum for collecting feedback is that each posted feedback is immediately visible to all users (participants and instructor). Subsequently, the instructor may even specify that participants are allowed to reply to their peers' feedback postings.

Sequence



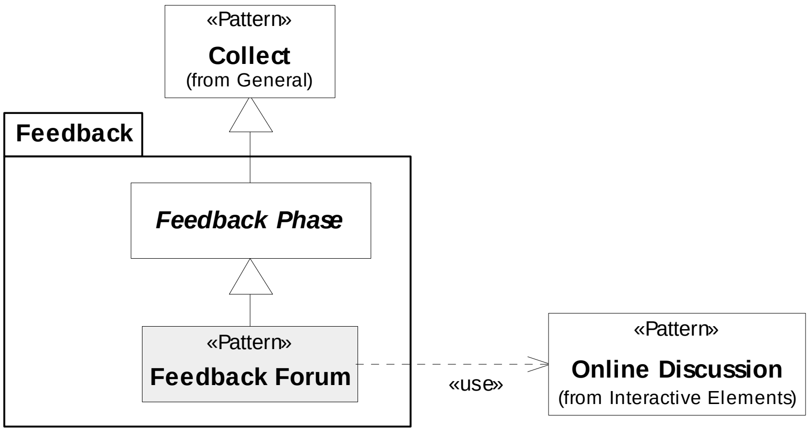
Activity	Description
Select/create feedback forum	For collecting feedback in a discussion forum, either an existing forum may be selected or a new one has to be created.
Initiate feedback request thread	The feedback collector initiates one or more threads that comprise the feedback request(s) in the designated discussion forum.
Post feedback as reply	Feedback providers just post their feedback as replies to the feedback request thread(s).
Review feedback	Feedback can be reviewed by all as it is posted in a discussion forum.

Structure



The diagram shows the FEEDBACK FORUM classes embedded into the structural model of ONLINE DISCUSSION.

Taxonomy/Dependencies



Parameters

- Primary pattern author: Michael Derntl
- Primary pattern source: Research Lab for Educational Technologies (University of Vienna)
- Pattern categories: Utility, Collateral
- Level of abstraction: Low
- Scope: Phase
- Primary presence type: Online
- Flexibility: Low
- Level of confidence: 4
- Number of participants: Unrestricted
- Application effort: Low
  Reviewing feedback requires higher effort when many participants are involved.
- Level of expertise required: Low
- Target skills: Reflective thinking
- Input: Feedback thread(s), Feedback guidelines

- Output: Feedback postings

Web Template

Basically, the FEEDBACK FORUM Web template allows for COLLECTING FEEDBACK in qualitatively much the same way as the REACTION SHEETS Web template, except that the participant view in this pattern resembles the appearance of an ONLINE DISCUSSION forum. So technically, this pattern's realization is kind of a simple merging of these two patterns.

Administration View

One difference to the standard ONLINE DISCUSSION forum Web template is that additional configuration options are required in the administration view:

- *Multiple replies*: Controls whether participants are allowed to post multiple feedback messages to one single feedback thread. In ONLINE DISCUSSION forums this is always allowed, whereas in FEEDBACK FORUMS it might be useful to restrict that participants may only post one single feedback message.
- *Reply to feedback*: Used to allow/deny participants to reply to feedback messages posted by their peers. If allowed, regular discussions can arise on any feedback message. If denied, participants are only allowed to post feedback messages to the instructor-initiated, predefined threads.

Additionally, the following setting in the standard administration view of ONLINE DISCUSSION is predefined and need not be configured manually:

- *Thread creation*: In FEEDBACK FORUMS, participants are not allowed to create their own threads. They can only reply to instructor-initiated, predefined threads.

Participant View and Report View

These views are the same as in ONLINE DISCUSSION. Refer to that pattern's Web template for details.

Examples

Feedback forums were used in the ***PhD Literature Seminar***. The instructor initiated threads in discussion forums where feedback was solicited regarding the learning platform that was used as well as regarding the style and process of the seminar. For example, in the seminar feedback forum the instructor posted the following simple feedback request: "*I would like to know from everyone how the seminar appealed to you (eventually in comparison to other courses) and whether there was something to get from it (except for the certificate). I am curiously awaiting your reactions, which always help me to improve on it next time*." Each participant was asked to post one feedback message to this thread (see Figure 65).

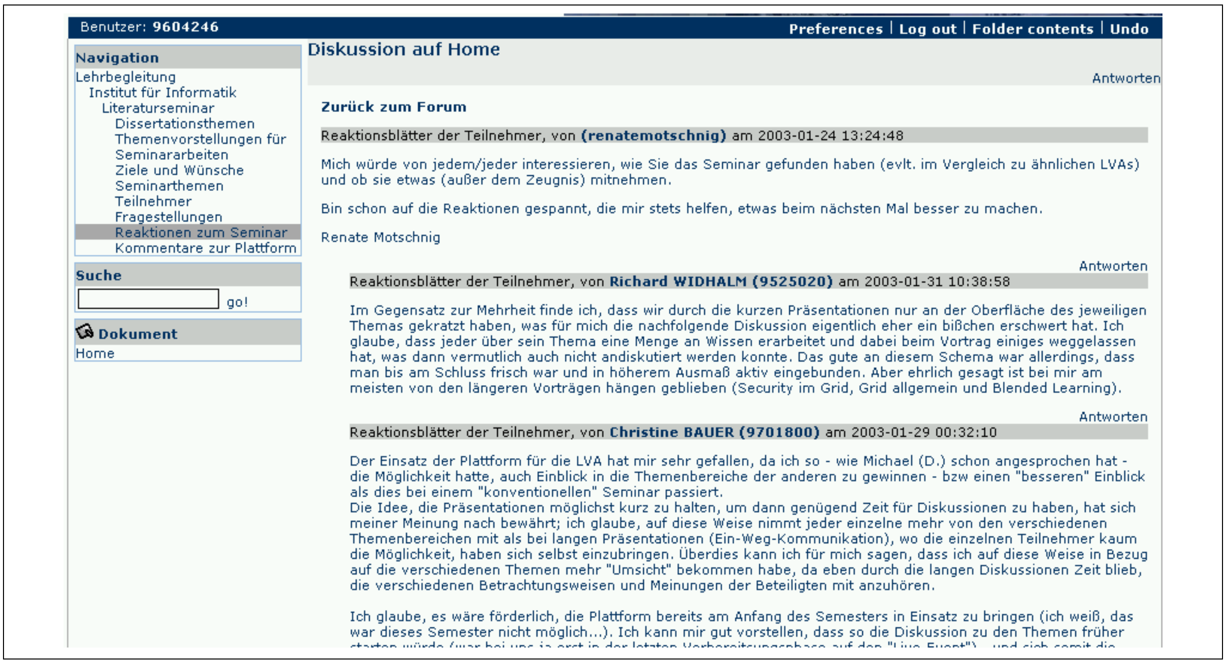


Figure 65: Example of a feedback forum in a PhD seminar.
Only one portion of the forum is visible. The rest was cut to save space.

In another course (***Project Management/Basics and Techniques***) a feedback forum was used to collect structured error reports for the employed learning platform (see Figure 66).

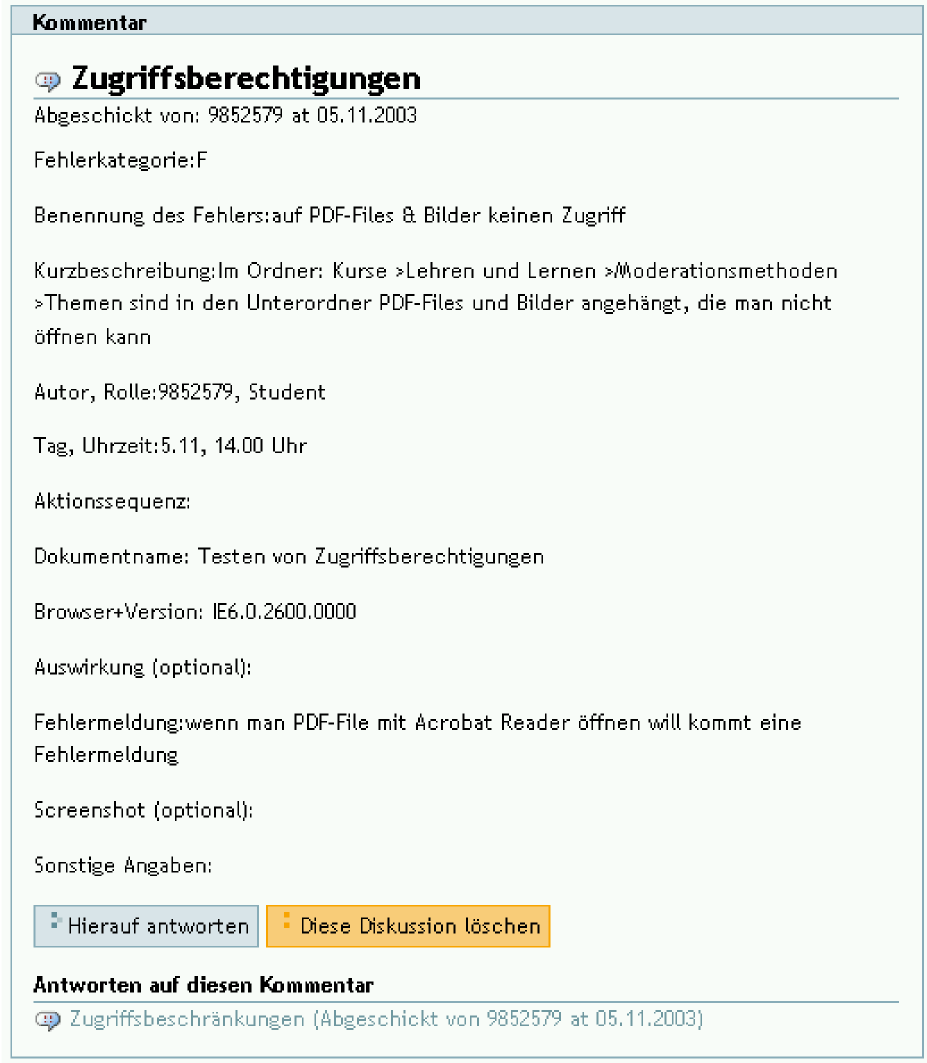


Figure 66: Feedback forum to collect structured error reports.

Evaluation

Not available.

QUESTIONNAIRE

Package: *Feedback*

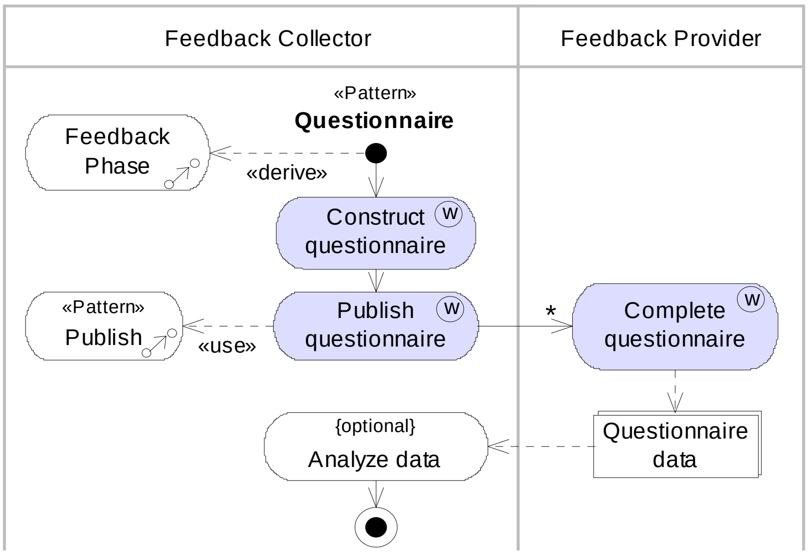
Intent

A questionnaire is a form of COLLECTING FEEDBACK in a structured way by specifying and providing a set of items/questions along with scaled, possible responses.

Motivation

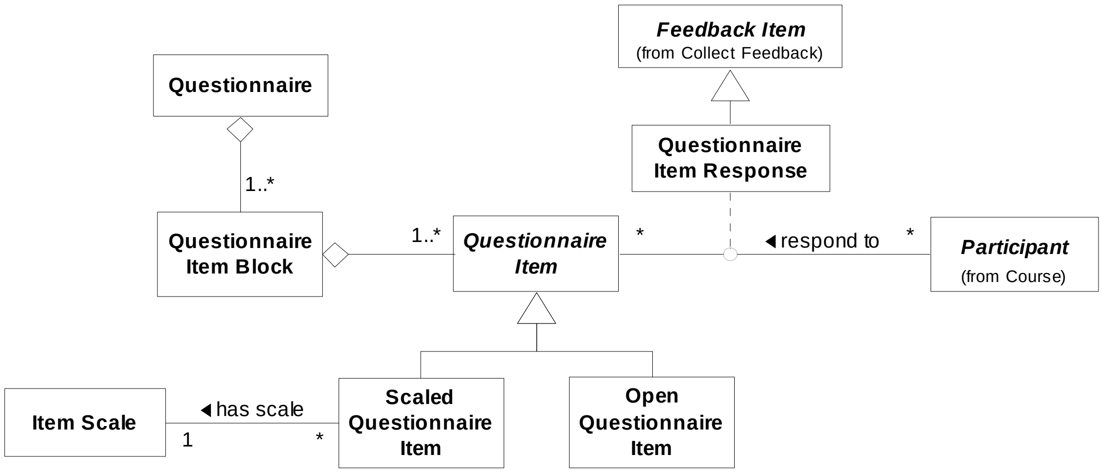
This pattern describes a concrete form of COLLECTING FEEDBACK, where participants are asked to provide structured feedback by completing a predefined questionnaire. Questionnaire results are essential input to quantitative evaluation of employed scenarios. Therefore, questionnaires should be collected in any course, either at the beginning, at the end, or both at the beginning and at the end to allow for comparative analyses.

Sequence



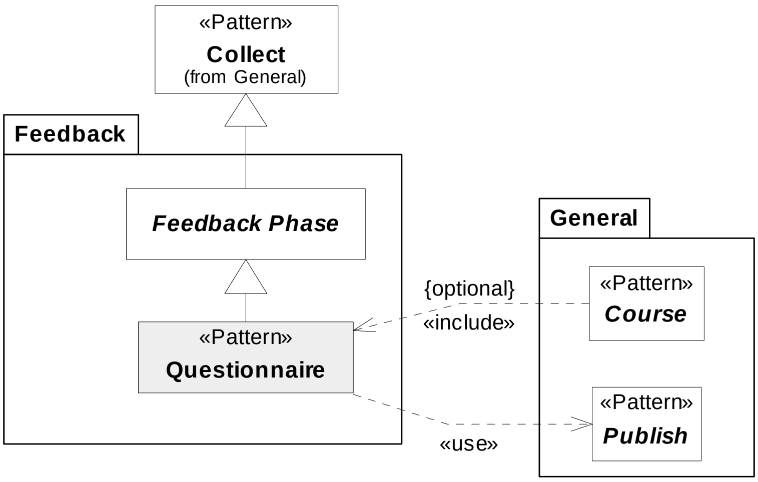
Activity	Description
Construct questionnaire	First, a questionnaire has to be constructed by compiling a number of questionnaire items to blocks and by subsequently attaching the blocks to the questionnaire (see the *Structure* section). If previously used questionnaires are available, their items may be selected for reuse.
Publish questionnaire	The compiled questionnaire is published on the platform (or distributed on paper) to be completed by participants.
Complete questionnaire	Each participant (*feedback provider*) completes the questionnaire by filling out the respective web form on the learning platform or by completing the questionnaire on paper.
Analyze questionnaire	The completed questionnaires have to be prepared in a machine-processible form to be ready for subsequent analysis.

Structure



Each *questionnaire* consists of a number of *questionnaire item blocks*. Each of these blocks comprises a number of *questionnaire items* that participants respond to. Each of these responses is considered as a *feedback item* (see COLLECT FEEDBACK). Questionnaire items may be either *open* (free text response) or *scaled* by an appropriate *item scale* (e.g., *yes* = 1, *no* = 2).

Taxonomy/Dependencies



Parameters

- Primary pattern author: Michael Derntl
- Primary pattern source: Research Lab for Educational Technologies (University of Vienna)
- Pattern categories: Utility
- Level of abstraction: Low
- Scope: Activity, Phase
- Primary presence type: Online, Present
- Flexibility: High

- Level of confidence: 5
- Number of participants: Unrestricted
- Application effort: High
  Assembling and analysis of questionnaires is typically very time- and resource-intensive.
- Level of expertise required: High
  Constructing and analyzing sound questionnaires is a science itself; statistical as well as socio-psychological research methods have to be employed.
- Suggested assistance: Expert, Administrator
- Input: Questionnaire items, Guidelines for completing the questionnaire
- Output: Questionnaire results, Analyses

Web Template

Even though questionnaires often appear quite complex, they are usually made up from a small number of distinct item/response types that may be presented differently. The Web template for this pattern presents the template views, which were necessary and sufficient in the questionnaires used in the last two years at the department.

Administration View

Each questionnaire carries a heading and an introductory text, as well as a number of item and text blocks. The former two options are configured in the initial administration wizard step (Figure 67), while the latter two are configured in a more complex second step.

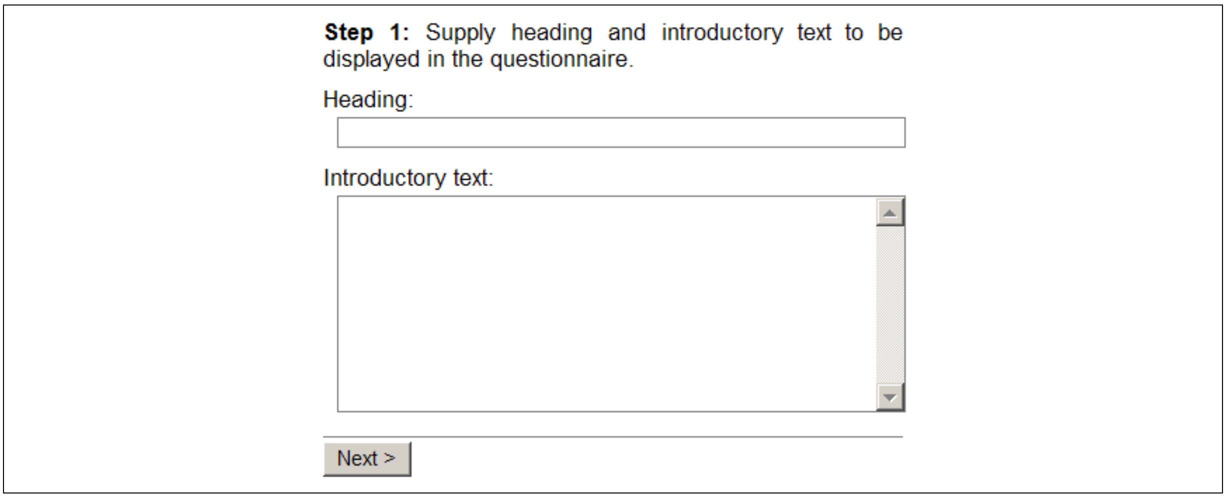


Figure 67: General QUESTIONNAIRE administration page.

While the general configuration step is quite simple, the second step has to host options for configuring the different types of item/text blocks, which make up the body of the questionnaire. We distinguish four different types of blocks:

- *Text block*: This may either be a heading, a subheading, or some normal text paragraphs.

- *Choice block*: Used to pose one or a group of items with a list of common responses. The responses may be mutually exclusive (single-choice), or non-exclusive (multiple-choice).
- *Open response block*: Used to pose an item with an open text response.

Each questionnaire consists of an arbitrary number of instances of the above block types. The configuration step for block arrangement (or the ***questionnaire form editor***) is shown in Figure 68. There is a table that includes the currently configured blocks, along with an option of rearrangement (*move up* hyperlink), *editing*, and *removal* for each block, as well as for *adding* a new block by choosing from one of the different block types.

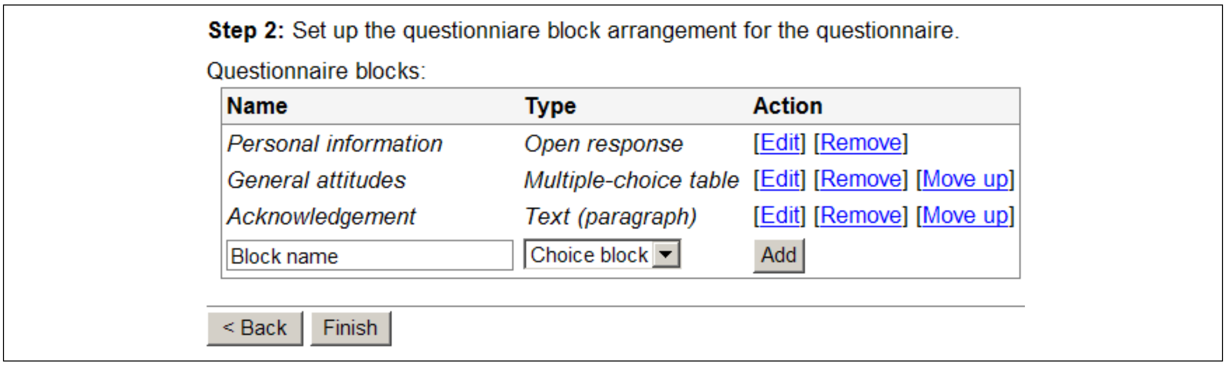


Figure 68: Questionnaire form editor for QUESTIONNAIRE.

Note that in each of the following block administration screens there is one common field that allows the administrator to specify some *name* (or a handle) for later referral.

By choosing to add a new block to the questionnaire, the user is redirected to a page for configuring that block. Figure 69 depicts the text block configuration screen, which only consists of two fields:

- An option for selecting the type of text to be displayed in the participant view, which is one of the following: *heading*, *subheading*, or *simple text*.
- An option for entering the actual *text* to be displayed.

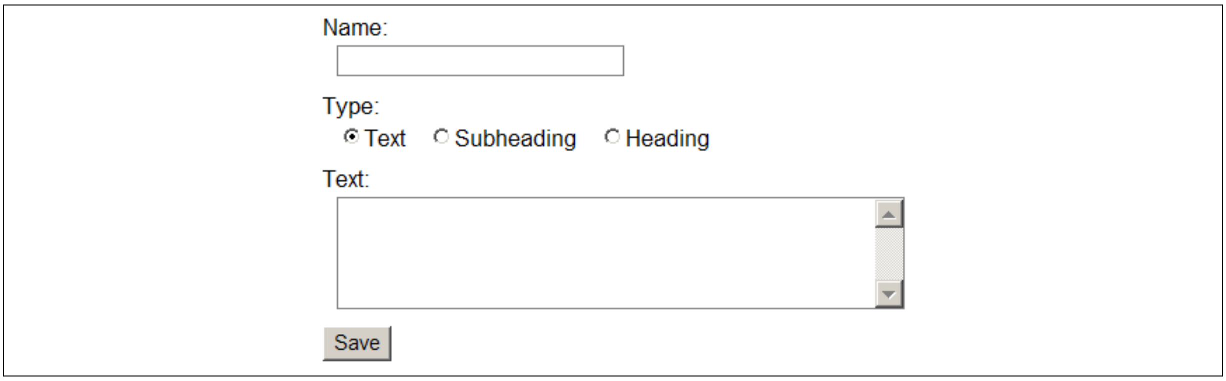


Figure 69: Text block configuration in QUESTIONNAIRE.

The choice block configuration screen shown in Figure 70 is more complex. It offers the following options:

- *Block style*: This option configures the appearance of the block in the participant view. There are two choices (see Figure 73 and Figure 74 in the participant view):
  - *Table*: Displayed as a table where the leftmost column holds the items and the remaining columns hold the radio controls for selecting the response, which is displayed in the heading of the respective column.
  - *List*: The responses are displayed one below the other in list-style manner for each item. The item is displayed on top of the response list. This is suitable for longer responses, which would not fit in the table style display.
- *Response type*: Each item may have exactly one response or an arbitrary number of responses from the response domain. The former is called *single-choice* and the latter is called *multiple-choice*. In Web forms, radio buttons are used for single-choice responses and checkboxes are used for multiple-choice responses.
- *Introduction*: Some text that may optionally be entered to set the context for the following question.
- *Common beginning*: Often when a number of items are grouped together, it can be useful to specify a common beginning for all items, such that this part of the item need not be displayed repeatedly. For example, a group of items about learning behavior has the common beginning "What is your preferred learning style...", with two concrete items stating "...at home?" and "...at work?".
- *Items*: A list of items that represent the questions or statements to be posed for response. The table lists the items already available, as well as options for *editing*, *deleting*, *rearranging*, and *adding* items. Adding a new item requires two fields to be set:
  - *Text*: The actual item text displayed to the user.
  - *Type*: The type of item, which may either be *closed* (displayed as normal text in the participant view) or *open*, which means that the user may fill in an additional desired item (displayed as a text box in the participant view), which may be missing from his/her point of view. Clearly, most questionnaire items are closed ones, and setting this type of item is mostly useful in table-style arrangement.
- *Responses*: The set of possible responses (e.g., "yes" and "no") is configured much the same way as the item configuration above. The table lists the responses already available, as well as options for *editing*, *deleting*, *rearranging*, and *adding* responses. Additionally, the administrator may specify whether selecting one of the responses for that block is *required* (this should be the default setting). Like for items, adding a new response requires two fields to be set:
  - *Text*: The actual response text displayed to the user.
  - *Type*: The type of response, which may either be *closed* (displayed as normal text in the participant view) or *open*, which means that the user may fill in any desired re-

sponse (displayed as a text box in the participant view). Setting this kind of response, if at all, is certainly only useful in list-style arrangement.

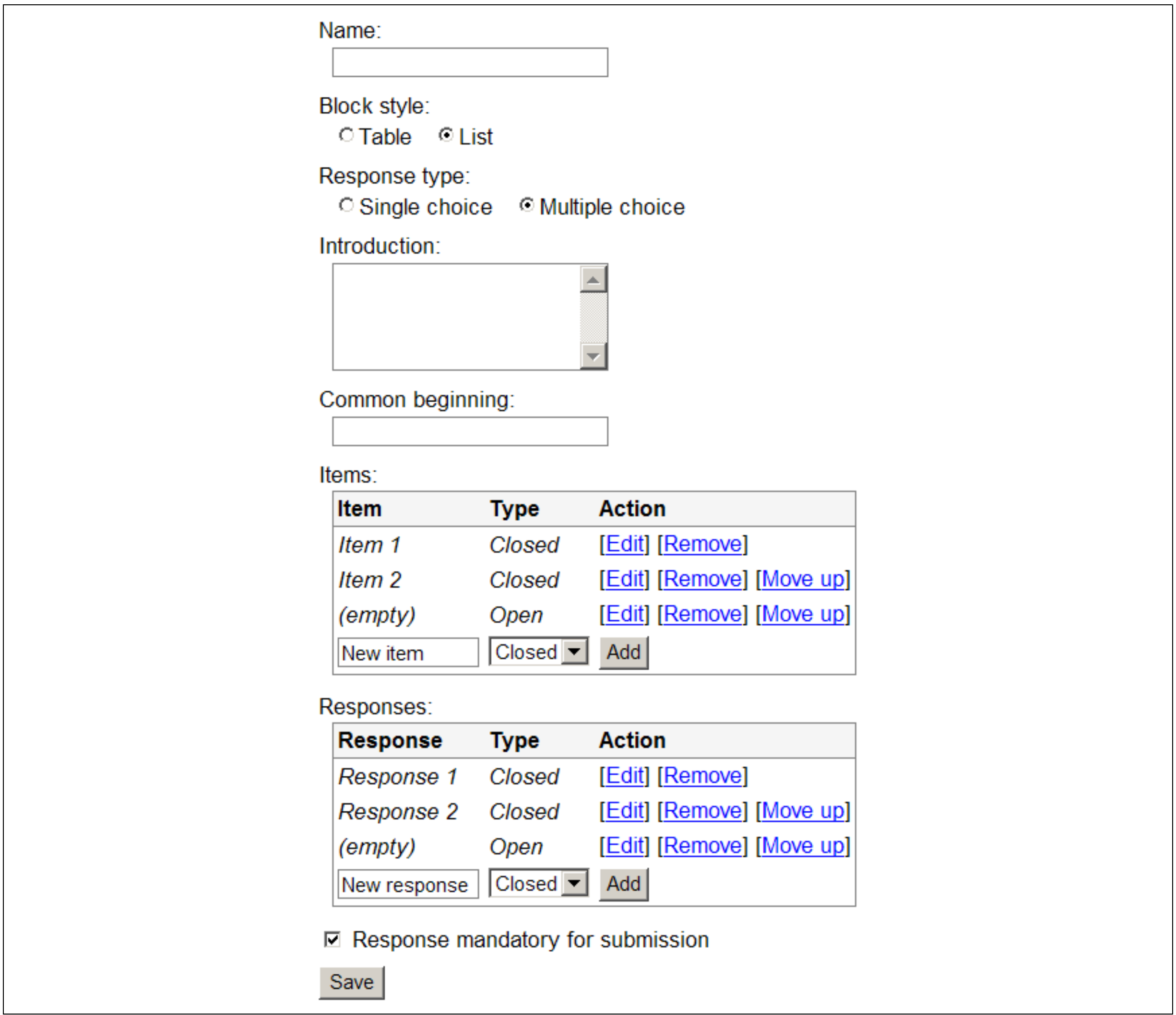


Figure 70: Choice block configuration in QUESTIONNAIRE.

Finally, the open response block (Figure 71) requires fewer configurations than the choice blocks: Besides the *introduction* and *item* (question/statement) fields, there is one option for setting the *text box size* for the control in which the user will enter his or her open response.

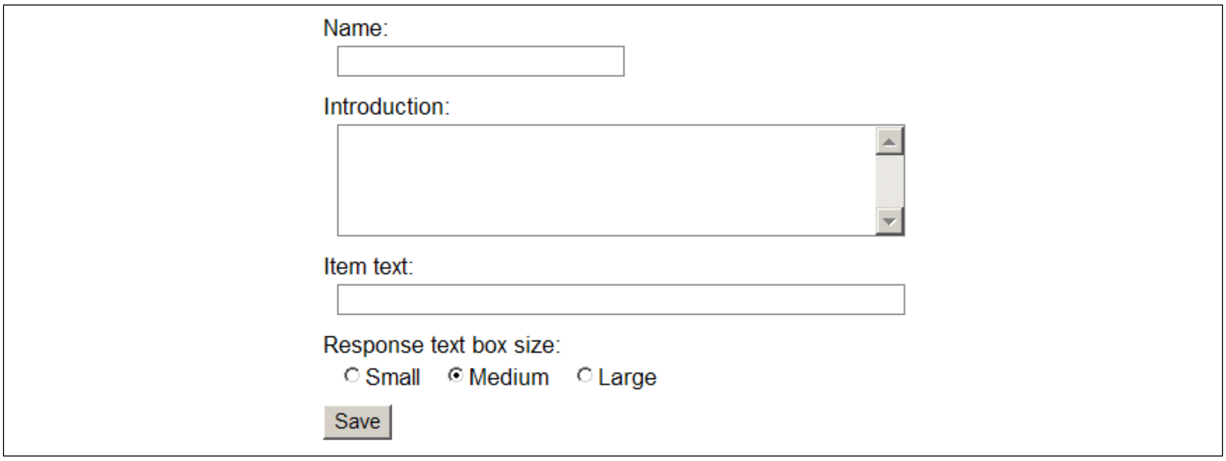


Figure 71: Open response block configuration in QUESTIONNAIRE.

Participant View

The participant view results in one single page that is displayed to the user for completing the configured questionnaire. Sequentially, it starts with the *heading*, the *introduction* text, the list of configured questionnaire *blocks*, and finally a button initiating *submission*. For clarity, the different visual layouts of the different block types are depicted in separate figures (Figure 72 to Figure 75), even though they appear on one single page.

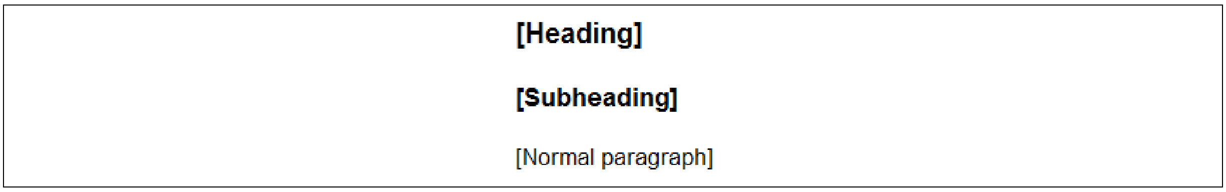


Figure 72: The different text blocks in the QUESTIONNAIRE participant view.

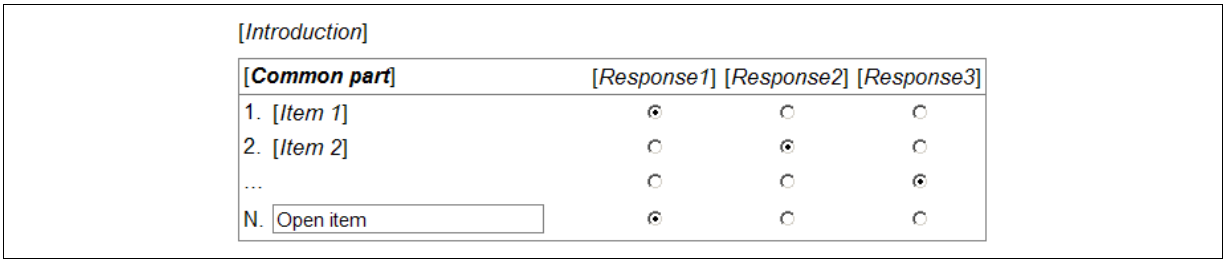


Figure 73: A table-style single-choice block in the QUESTIONNAIRE participant view. Note that a table-style multiple-choice block looks much the same, only the radio buttons are replaced by checkboxes.

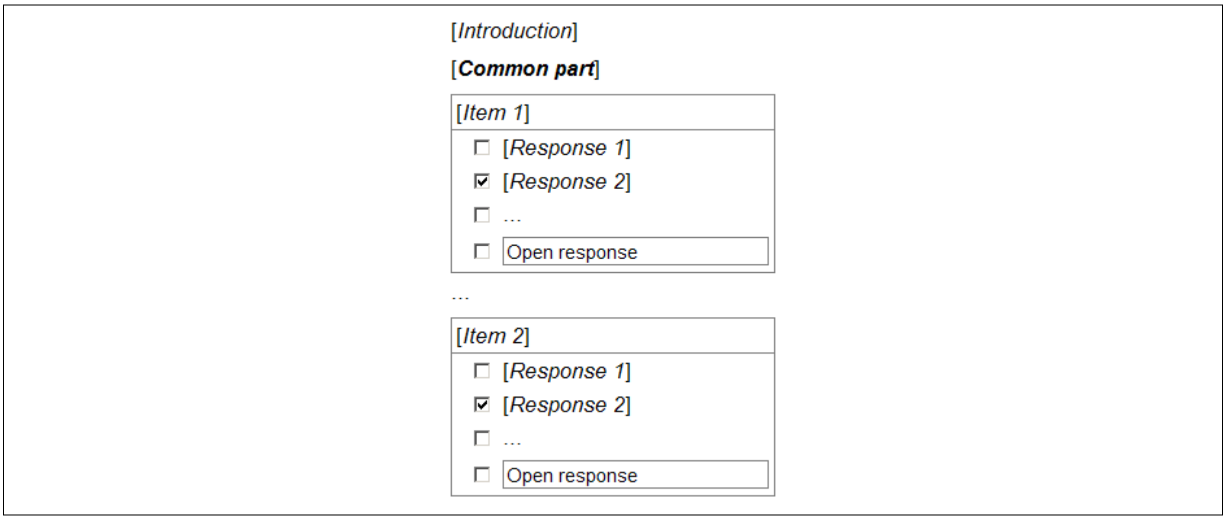


Figure 74: A list-style multiple-choice block in the QUESTIONNAIRE participant view. Note that a list-style single-choice block looks much the same, only the checkboxes are replaced by radio buttons.

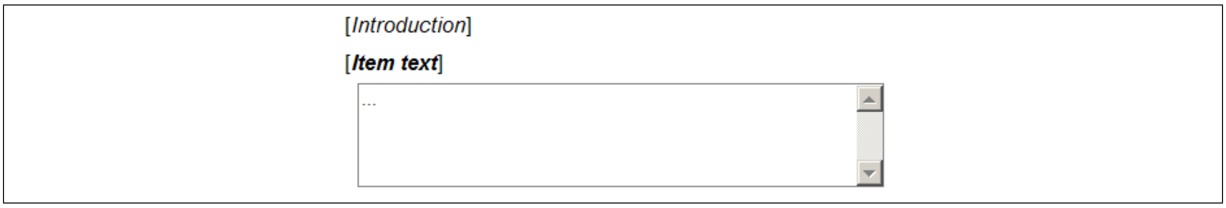


Figure 75: Open response block type in the QUESTIONNAIRE participant view.

Report View

Generally, questionnaire data must be stored using some response coding scheme that is transferable to other coding schemes for both human- and machine-processing. The simplest scheme is just coding the responses to each item numerically in a sequence: No response (if allowed) is coded as 0, and the preconfigured responses are coded beginning with 1, each following response incremented by 1. This way, the coding scheme is non-ambiguous and can be easily transformed to virtually any other coding scheme.

For appropriate handling of questionnaire data, the following types of report are required:

- ***Overview***: Lists the participants who have already *submitted* and those whose data is yet *missing*, along with an option to contact one single or all participants by e-mail or platform messaging services.
- ***Item-based***: Prepares submitted data from all participants suitable for printout, grouped by questionnaire blocks/items. Each participant's response, as well as the average value for each choice block item should be visible, along with each question and response text. There should also be an option to restrict reporting to single participants.
- ***Participant-based***: Generally identical to the item-based report, except that this report shows the responses grouped by participants.

- ***Binary***: For analytical/machine data processing (e.g., MS Excel®, SPSS, etc.) the data should be exportable to some format that can be imported by most spreadsheet applications, such as CSV (*comma separated values*).

Examples

The following URLs point to examples of questionnaires and reports:

- http://elearn.pri.univie.ac.at/patterns/examples/questionnaire/webeng-ss04-begin.html
  The initial questionnaire from Web Engineering 2004
- http://elearn.pri.univie.ac.at/patterns/examples/questionnaire/pcc-ss04-begin.html
  The initial questionnaire from Person-Centered Communication, summer term 2004
- http://elearn.pri.univie.ac.at/patterns/examples/questionnaire/webeng-ss04-begin-report-complete.html
  Complete report of the Web Engineering 2004 initial questionnaire (made anonymous)
- http://elearn.pri.univie.ac.at/patterns/examples/questionnaire/webeng-ss04-begin-report-binary.csv
  Binary report of the Web Engineering 2004 initial questionnaire (made anonymous)

See the *Evaluation* sections of the patterns in this repository for examples of uses of questionnaires.

Evaluation

Not available.

Reaction Sheets

Package: *Feedback*

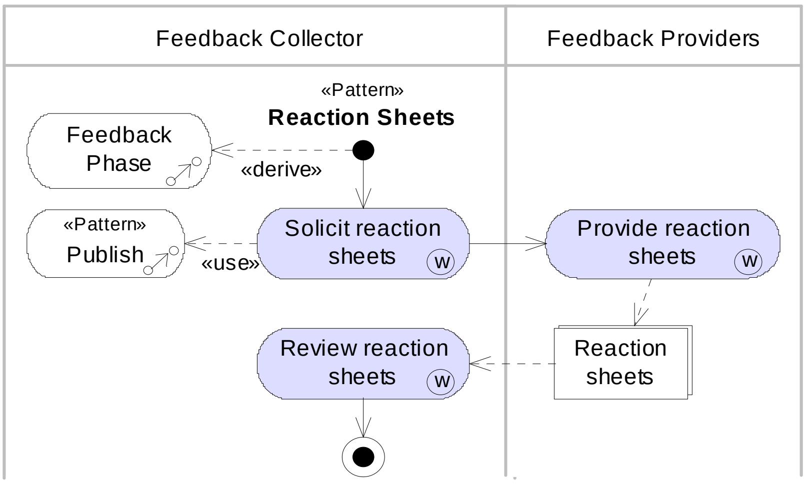
Intent

Solicit reactions sheets on specific aspects of learning scenarios and activities to Collect Feedback in an open, unstructured way.

Motivation

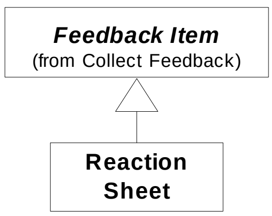
This pattern describes a concrete form of Collecting Feedback. Participants (*feedback providers*) provide unstructured, written feedback. According to the mode set by the instructor (*feedback collector*) the collected feedback may be publicly visible to all participants, or otherwise only to the instructor.

Sequence



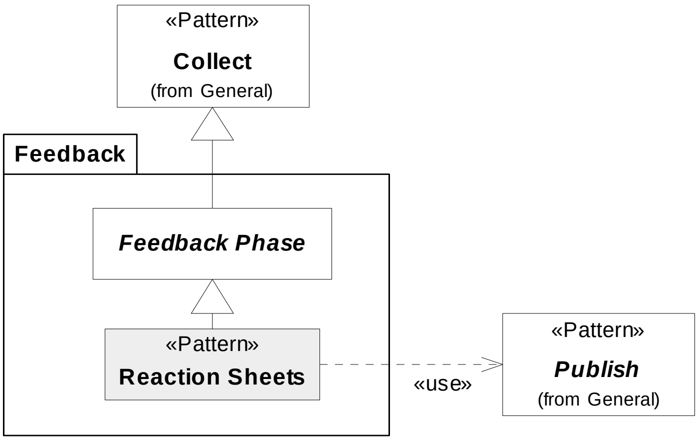
Activity	Description
Solicit reaction sheets	The feedback collector publishes a feedback request including information on what the collector expects the feedback providers to write in their reaction sheets. Typically, the collector provides some clues (e.g., guidelines, questions) on what reaction sheets should comprise.
Provide reaction sheets	Feedback providers respond to the reaction feedback by publishing their reaction sheets via the respective web form.
Review reaction sheets	The feedback collector reviews published feedback in the respective reaction sheet report. Optionally, it may be defined that feedback providers also have the right to review the reaction sheets.

Structure



A reaction sheet is just a specialized form of a feedback item. It only comprises the feedback text supplied by the feedback provider (see also the COLLECT FEEDBACK structure).

Taxonomy/Dependencies



Parameters

- Primary pattern author: Michael Derntl
- Primary pattern source: Research Lab for Educational Technologies (University of Vienna)
- Pattern categories: Utility, Collateral
- Level of abstraction: Low
- Scope: Activity, Phase
- Primary presence type: Online
- Flexibility: Low
- Level of confidence: 5
- Number of participants: Unrestricted
- Person-Centered variables addressed: Transparency, Acceptance
- Application effort: Low
  Reviewing a lot of reaction sheets may increase the application effort.
- Level of expertise required: Low
- Target skills: Reflective thinking
- Input: Reaction sheet request, Facility for providing reaction sheets
- Output: Reaction sheets

Web Template

Participant View

The REACTIONS SHEETS pattern is implemented by providing at least three pages:

(1) Start Page: Shows general information on reaction sheets to be collected. This information is provided by the feedback collector. The start page may be used to display multiple instances of REACTION SHEETS for a course. Thereby, some instances may share the same information text. Consequently, the start page layout follows the following pattern (see Figure 76): A paragraph holding information followed by a list of REACTION SHEETS instances with links to the submission and view pages, whereby this pattern may recur on the same page.

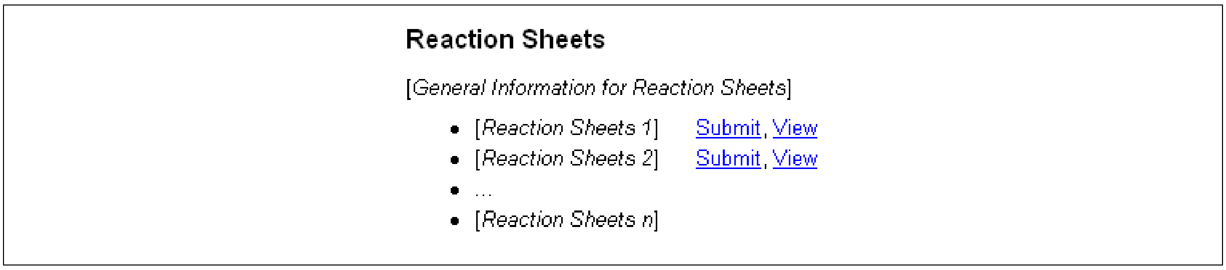


Figure 76: Start page for Reaction Sheets.
Note that the links to the *View* pages may be hidden to participants and thus only accessible to the instructor (i.e., private reaction sheets). This option is discussed in the *Administration View*.

Provision of a start page is superfluous in one special case: When only one Reaction Sheets instance is needed for a course, and the participants are not allowed to view the posted reactions of their peers, it is useless to provide a start page. In this case, the general information may as well be provided at the submission page, as the start page would only comprise one *Submit* hyperlink to the submission page.

Finally, the hyperlink to the start page on the learning platform may either be provided as a separate platform section in the course space, or anchored to a specific learning activity. In the latter case, the hyperlink may be placed at an appropriate spot on the respective learning activity's page.

(2) Submission page: The submission page can be reached via a *Submit* hyperlink from the start page. It provides a Web form for reaction sheet submission (see Figure 77). A reaction sheet Web form typically consists of a large text area, where participants can write their feedback. Additionally, some useful hints on submitting valuable feedback may be output above the form. Note that the submission page requires authorization.

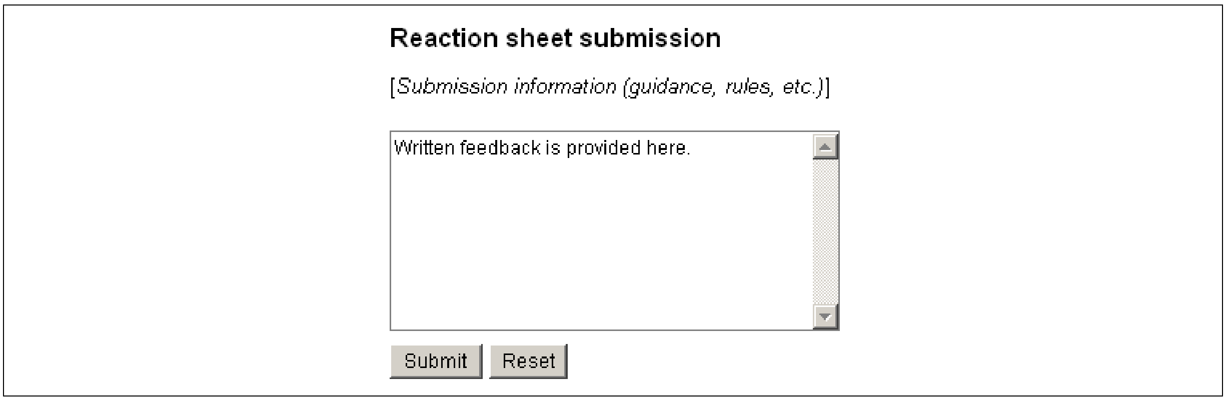


Figure 77: Submission page for Reaction Sheets.
Note that additional form fields may be provided depending on the feedback collector's desire (see *Administration View*).

(3) View page: Shows a list of reactions posted for a particular Reaction Sheets instance. This page may be reached via the respective *View* hyperlink on the start page. Depending on administrative settings, this page may require user authentication (e.g., reaction sheets are only visible to course participants; see *Administration View*). Basically the view page is identical to that in the report view for submitted reactions (see Figure 79).

Administration View

Creating and maintaining an instance of a REACTION SHEETS pattern is relatively simple for administrators, as they only have to provide some simple input parameters (see Figure 78, which shows the administration page):

- *General information* on the start page (an existing paragraph may be chosen for reuse)
- A *caption* for the links to the submission and view pages for the reaction sheets, also on the start page
- Specific information for reaction sheet *submission* on the submission page
- Whether the submitted reaction sheets will be *private* to a certain instructor

Note that for providing more complex reaction sheet forms (e.g., multiple text boxes and/or choices for feedback regarding specific topics or issues) it would be useful to provide some kind of *reaction sheet editor*, analogous to the *questionnaire form editor*[520] in the administration view of the QUESTIONNAIRE pattern.

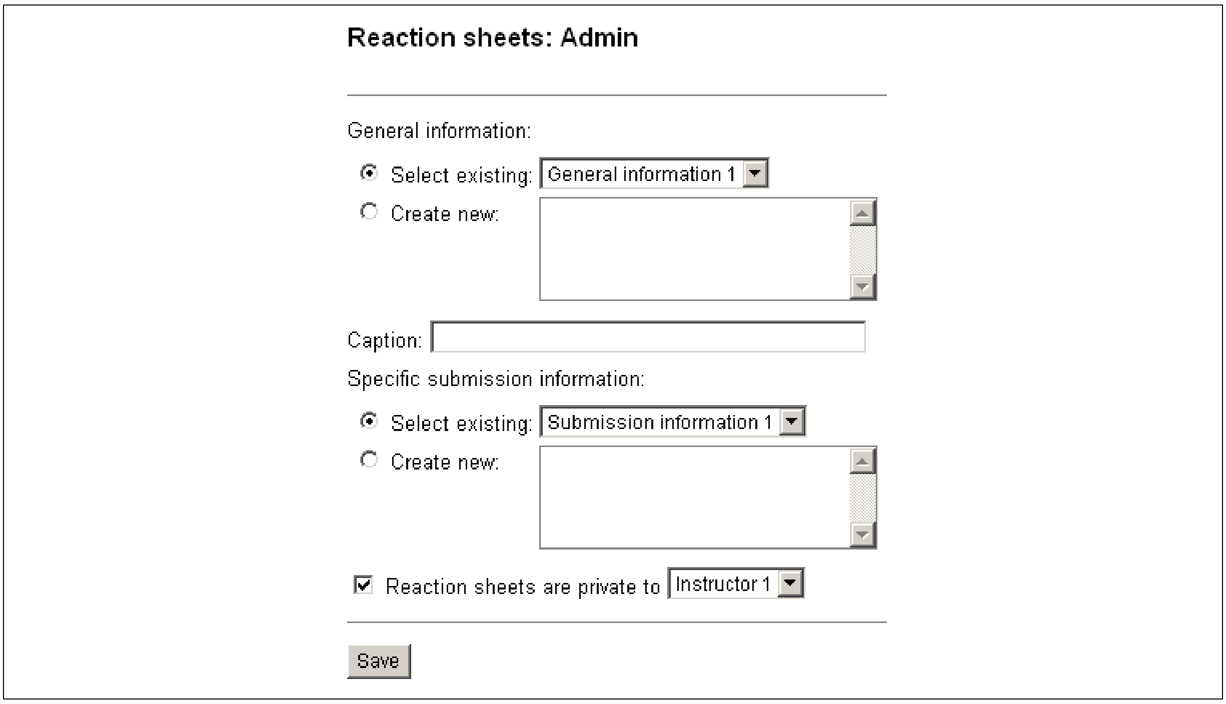


Figure 78: Administration page for REACTION SHEETS.

[520] See p. 233

Report View

There are two different report pages for reaction sheets:

1) ***Submitted reaction sheets***: Shows reaction sheets that have already been submitted for a given REACTION SHEETS instance (see Figure 79).

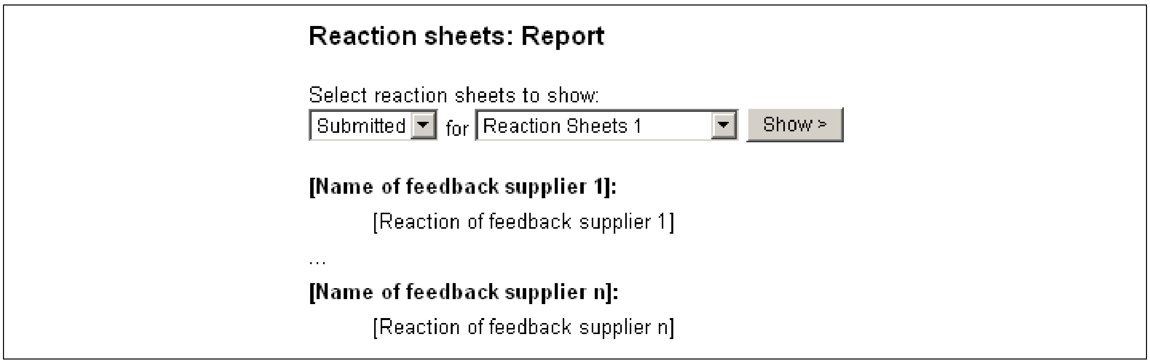


Figure 79: Report view for submitted REACTION SHEETS.

2) ***Missing reaction sheets***: Show a list of participants that have not yet submitted their reactions (see Figure 80). Additionally, there is a hyperlink *Notify* that should point to a page where the administrator can send a notification message to these participants.

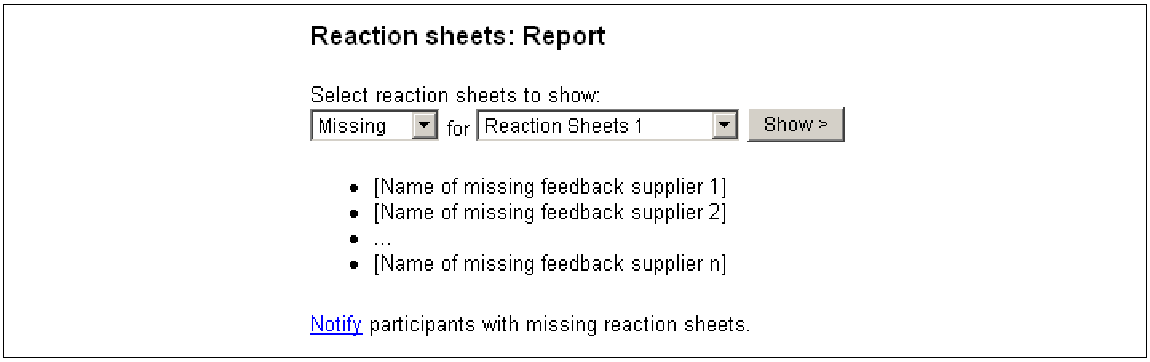


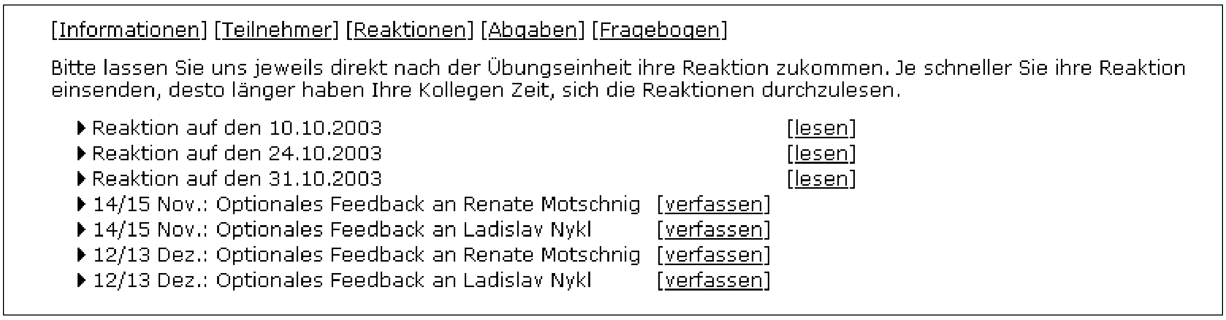
Figure 80: Report view for missing REACTION SHEETS.

Examples

Figure 82 shows an example of a reaction sheet form that was offered to participants in the ***Person-Centered Communication*** course to submit their reactions to the instructor after each course unit:

- There was a section for "reactions" on the platform, where participants had access to previously written reaction sheets, as well as to the web form for reaction provision (see Figure 81)
- After the initial WORKSHOP units, provision of reaction sheets was mandatory. Additionally, each reaction sheet was publicly accessible to participants and instructor (for an example see http://elearn.pri.univie.ac.at/patterns/examples/reactionsheets/pcc-publicreactions.html)

- After the following encounter group sessions, provision of reaction sheets was optional. These reaction sheets were visible only to the instructor (private reaction sheets).


[Informationen] [Teilnehmer] [Reaktionen] [Abgaben] [Fragebogen]

Bitte lassen Sie uns jeweils direkt nach der Übungseinheit ihre Reaktion zukommen. Je schneller Sie ihre Reaktion einsenden, desto länger haben Ihre Kollegen Zeit, sich die Reaktionen durchzulesen.

- Reaktion auf den 10.10.2003 [lesen]
- Reaktion auf den 24.10.2003 [lesen]
- Reaktion auf den 31.10.2003 [lesen]
- 14/15 Nov.: Optionales Feedback an Renate Motschnig [verfassen]
- 14/15 Nov.: Optionales Feedback an Ladislav Nykl [verfassen]
- 12/13 Dez.: Optionales Feedback an Renate Motschnig [verfassen]
- 12/13 Dez.: Optionales Feedback an Ladislav Nykl [verfassen]

Figure 81: Reaction sheet section on the Person-Centered Communication platform.
(See http://elearn.pri.univie.ac.at/patterns/examples/reactionsheets/pcc-selection.html)


[Informationen] [Teilnehmer] [Reaktionen] [Abgaben] [Fragebogen]

12/13 Dez.: Optionales Feedback an Renate Motschnig - verfassen

Hier kann persönliches Feedback an mich weitergeleitet werden. Der Text kann jederzeit ergänzt werden.

blah

submit reset

Figure 82: Reaction sheet submission form in Person-Centered Communication.
(See http://elearn.pri.univie.ac.at/patterns/examples/reactionsheets/pcc-form.html)

Evaluation

Not available.

5.5 General

ACHIEVEMENT AWARD

Package: *General*

Intent

Reward originators of outstanding contributions as determined in the ASSESSMENT PHASES with achievement award certificates.

Motivation

Offering positive reward in the form of an award certificate for originators of high-quality contributions is definitely one way of being acceptant and transparent toward students. In behavioral psychology this type of positive reward is called reinforcement[521] and intended as a motivation to learn. Thereby it is a precondition that the participants are provided with the opportunity to bring themselves into the EVALUATION process as deciders. For example, if the award winners are determined by PEER-EVALUATION results, acceptance becomes manifest in the fact that the instructor puts basic trust in the ability of peers to supply meaningful evaluation results Additionally, realness is augmented by transparently disclosing the mode and the criteria for determining distinguished contributions.

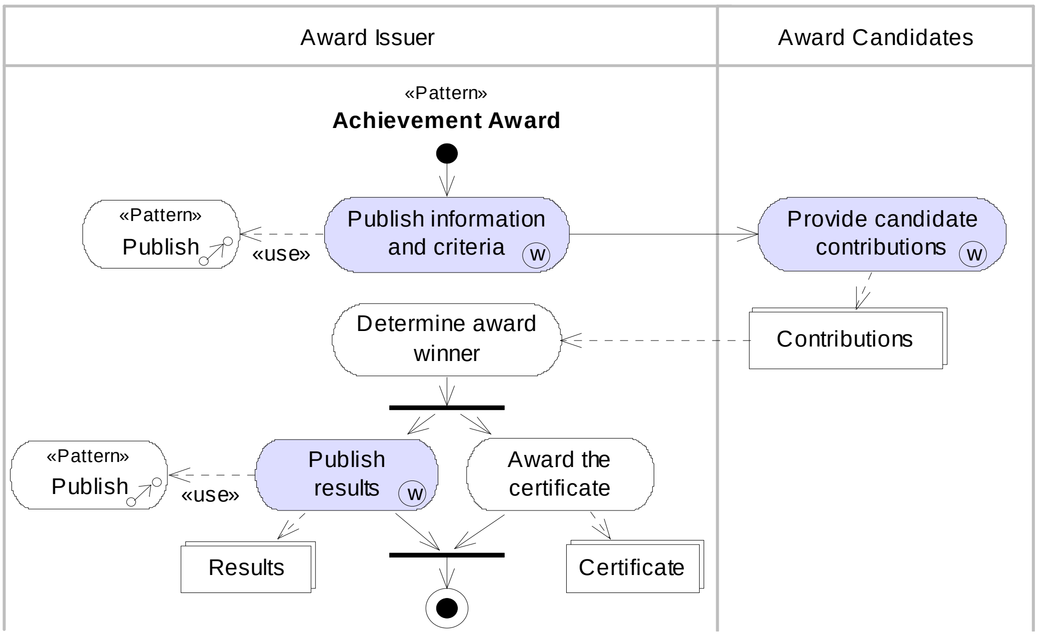
Nevertheless, care has to be taken when implementing this scenario: If only *winners* are determined, chances are that non-winners may think of themselves as being considered *losers*. So the instructor has to assure that winning an award is not the only option of receiving gratification for contributing and participating.

From the organizational point of view, the application effort of this incentive scenario is relatively low and is certainly justified by the chance of an increase in the participants' learning motivations and activities.

ACHIEVEMENT AWARD scenarios are usually used in conjunction with learning scenarios where different, comparable contributions are elaborated by participants. To make the ascertainment of the award winner traceable, the underlying EVALUATION process has to produce comparable results, and the criteria for evaluation as well as the metrics for determining the winners have to be well-chosen and made transparent.

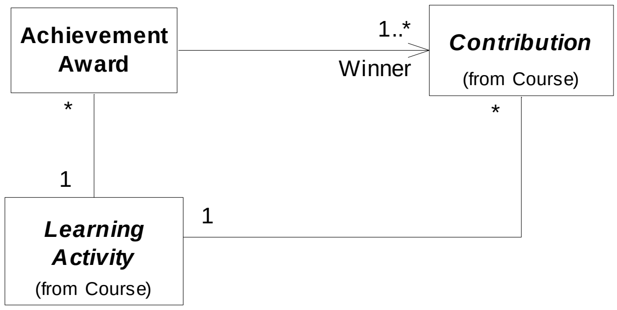
[521] Cotton (1995, p. 54)

Sequence



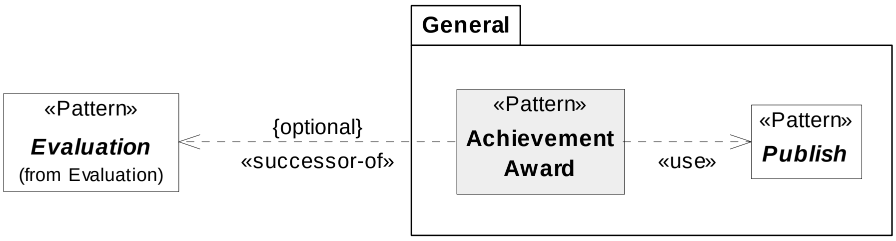
Activity	Description
Publish criteria	Initially, general information as well as criteria on which the ascertainment of the award winner is based have to be published by the award issuer (i.e., the instructor in the majority of the cases).
Provide candidate contributions	Award candidates provide candidate contributions to the award issuer; this is mostly done implicitly, e.g., by uploading a contribution which belongs to a certain learning activity.
Determine award winner	Based on a chosen metric, the winner is determined. For example, when this is based on PEER-EVALUATION results, suggested grades or received bonus points may be taken into account, either in average (determines quality) or absolute (determines popularity) measures.
Publish results	To sustain transparency as well as to attain public effect, the award winner should be published to participants.
Award the certificate	To give the process an official touch, the award certificate is issued with stamp and signature of the educational or scientific institution, and finally handed over personally to the winner.

Structure



Each *achievement award* is linked to a specific *learning activity*, and one or more of the *contributions* elaborated in the scope of this learning activity are determined as award *winners*.

Taxonomy/Dependencies



This pattern is located in the *General* package. It uses PUBLISH and optionally depends on number of EVALUATION scenarios as the base for determining the award winner.

Parameters

- Primary pattern author: Michael Derntl
- Primary pattern source: Research Lab for Educational Technologies (University of Vienna)
- Pattern categories: Motivational, Collateral
- Level of abstraction: Low
- Scope: Phase
- Flexibility: High
- Level of confidence: 2
- Number of participants: Unrestricted
- Application effort: Medium
- Level of expertise required: Low
- Person-Centered variables addressed: Acceptance, Transparency
- Input: Determination criteria, Evaluation results, Calculation metrics
- Output: Award winners, Certificates

Web Template

Web support for this pattern just consists of two usages of the PUBLISH pattern. First the award criteria are published and finally the award winners are published in text form. Regarding the participant view, the achievement award criteria and results will typically be published in the platform section of the respective learning activity. No visual Web template is provided here.

Examples

This pattern was applied in the summer term 2003, in an INTERACTIVE LECTURE course on ***Web Engineering*** at the University of Vienna. The award winning teams were determined by PEER-EVALUATION of LEARNING CONTRACTS. One certificate (which is given in Figure 83, translated from German) was awarded to the team which has achieved the highest average in bonus points per evaluation. This metric was chosen as a means to determine the learning contract contribution with the highest quality. Additionally, the originators of the most popular contribution were awarded a certificate for receiving the highest total number of bonus points.

University of Vienna
Department of Computer Science and Business Informatics

Certificate

The team of

Doris [redacted]
Christof [redacted]
Wolfgang [redacted]

in the course of the compulsory Business Informatics study module

Web Engineering

at the Department of Computer Science and Business Informatics of the University of Vienna, for their contributions on the topic

" Personalisierung im Internet "

received the

highest peer-evaluation ranking

among 85 participating teams.

Vienna, June 18, 2003

Deparmtent Stamp

Figure 83: Web Engineering Achievement Award certificate example.

Evaluation

The following proposed block of questions ask for general attitudes towards being awarded for contributions, as well as for the impact of the chance to win an award on participants' learning efforts and outcome. The items can be scaled based on a numerical interval, e.g., 1 = *not at all* to 5 = *very much.*

- Winning an award for an outstanding contribution is important to me.
- The chance to win an award was an incentive to work more intense, longer, or more accurate on the elaboration of my/our contribution.
- The chance to win an award contributed to my overall learning success.

Remarks

Usage of ACHIEVEMENT AWARD as an incentive seems justified; nevertheless this is clearly not a Person-Centered concept. Its inclusion is justified only if it is made clear that the award is a positive side-effect of learning and does not say much about personal learning.

References

Cotton, J. (1995). *The Theory of Learning.* London: Kogan Page.

ALTERNATING PHASES

Package: *General*

Intent

Presence phases alternate with online phases. This embodies the essence of blended learning scenarios.

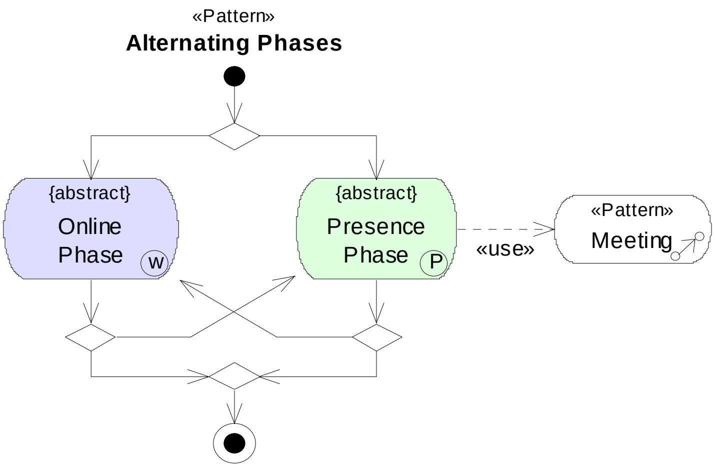
Motivation

It is the nature of blended learning that online phases alternate with presence phases. Presence phases comprise any kind and/or number of face-to-face MEETINGS, like WORKSHOPS, STAFF MEETINGS, etc. For this reason, the ALTERNATING PHASES pattern is one of the central patterns in the repository, lending an abstract scenario to many child patterns.

In a Person-Centered, blended learning scenario the Web acts as a kind of a backbone to supply the participants with learning material and to offer them facilities for interaction and collaboration. In this respect, online phases in which these inter-actions take place may be compared with background processes: always active, sometimes more intensive, sometimes idle. This process is only 'interrupted' when face-to-face MEETINGS take place. The meetings may be used to:

- Continue, deepen, diversify, and reflect on online processes, such as Online Discussion, or Feedback.
- Motivate, prepare, or initiate subsequent online phases. This does not necessarily mean that meetings always terminate online phases. A meeting may just suspend online phases. As for this, alternating phases may also be considered parallel or concurrent phases.
- Do interactions which can hardly or not at all take place online in a useful, supporting way.

Sequence



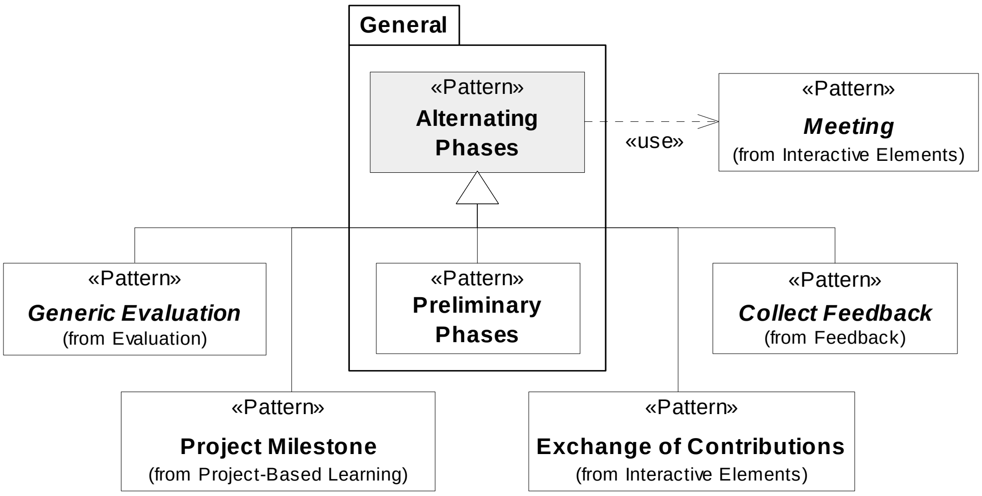
Even though the activity flow of Alternating Phases looks complex, its essence is simple. The arrangement of decision and merge elements serves two combined purposes:

- Online phases and presence phases *alternate*: It can never occur that an online phase follows an online phase, and it can never occur that a presence phase follows a presence phase.
- Which phase initiates the flow is *irrelevant*: When looking at an arbitrary point in a blended learning scenario, it is not always clear whether online phases follow presence phases or vice versa. For this reason, this pattern includes a decision at the start, thus commencing with an online phase or a presence phase. The constraint is that at least one of the activities has to take place.

Structure

Not available. Involved entities and their relations have to be identified by more concrete patterns.

Taxonomy/Dependencies



Parameters

- Primary pattern author: Michael Derntl
- Primary pattern source: Research Lab for Educational Technologies (University of Vienna)
- Pattern categories: Generic, Utility
- Level of abstraction: High
- Primary presence type: Blended
- Scope: Phase
- Flexibility: High

Web Template

Inapplicable: Due to the abstract sequence and missing structure, no Web template is provided.

Examples

Not available. Refer to more concrete scenarios.

Evaluation

Not available.

COLLECT

Package: *General*

Intent

Collect an information item by issuing a collect request so as to make the holder/owner of the item PUBLISH it to the collector.

Motivation

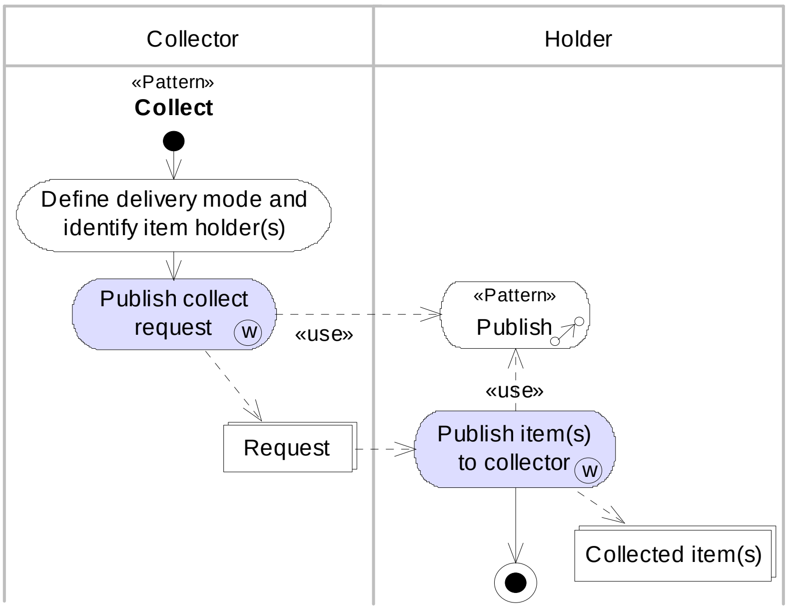
At certain points it is necessary to explicitly initiate PUBLISH processes, e.g., when soliciting project PROPOSALS from participants. In such a COLLECT scenario the *collector* is at the same time the *publication target* of the corresponding PUBLISH scenarios. Any *addressee* of the collect request is called upon to act as a *publisher* of the desired *information item*. COLLECT is a utility pattern that is used for concrete means of publishing / requesting by other patterns.

The aim of COLLECT is to make information items (i.e., text, files, or Web form content) accessible to the collector. So the desired information items, as well as the person(s) possessing or creating the items are known to the collector. The collector has to address the *collect request* to these holders of the desired items. In turn, they publish the desired items to the collector.

Basically, two roles are involved in this scenario: the *collector* and the *holder*[522]. The collector knows the item she seeks from the holder of the item. After defining restrictions on the mode of delivery of the item (which may be restricted and/or predetermined by the employed scenario) and after identifying the holder of the item, the collector has to PUBLISH to the holder that she seeks the item. If the item is available to the holder, the holder in turn initiates a PUBLISH process to the collector, making the item available to the latter.

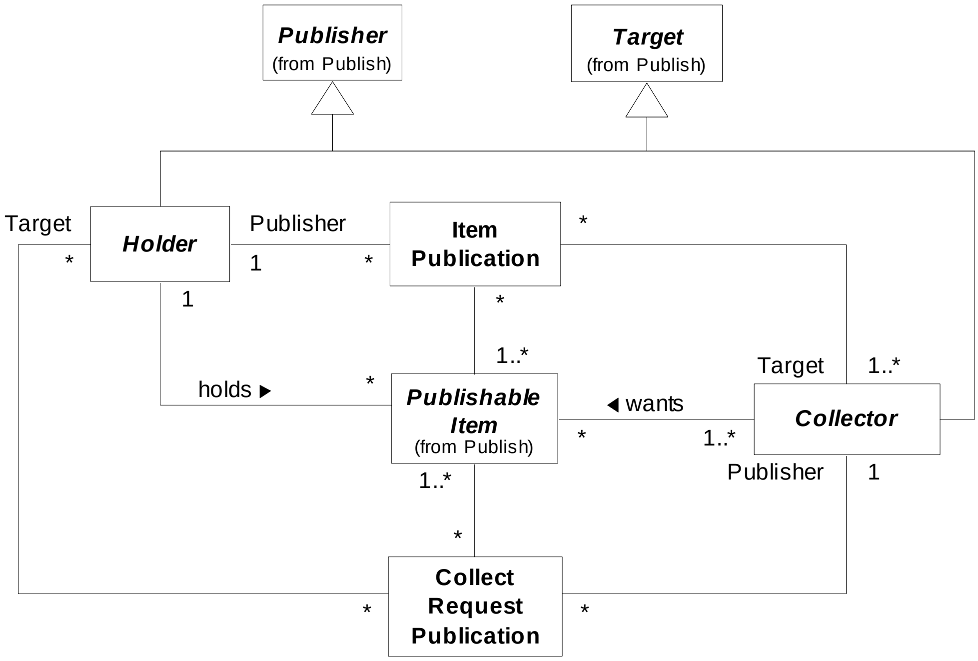
[522] Within the scope of COLLECT, the *holder* may also be called *supplier* (supplies the desired item), *creator* (creates the desired item before supplying it to the collector), or *addressee* (target of the collect request).

Sequence



Activity	Description
Define delivery mode and identify item holder(s)	The collector may define restrictions on how the desired item has to be published by the item holder. Usually, also a restriction on the time of delivery of the desired item is defined by the collector. The collector also has to identify the holder(s) of the desired item(s).
Publish collect request	The collector publishes a collect request to the holder of the item.
Publish item(s) to collector	Usually, the desired item is available to the holder. In some cases, however, the item has to be created first (e.g., a contribution). Anyway, after having the item available, the holder publishes the item to the collector.

Structure



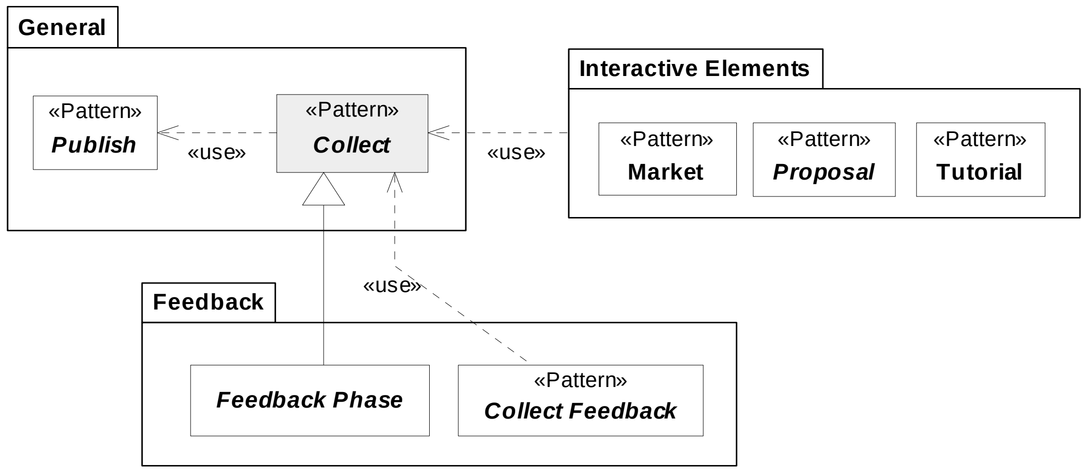
The structural entities of PUBLISH have already been integrated into this diagram, as two instances of PUBLISH are involved in the sequence of COLLECT, represented by the following publication classes:

- *Collect request publication*: This class represents the publication of the collect request by the *collector*. In this context, the collector acts as a *publisher*, while the *holder* of the item is the *target* of the collect request.
- *Item publication*: This class models the publication of the desired item by the *holder* (*publisher*) to the *collector* (*target*).

There are three abstract entities involved in relationships in the structure of COLLECT, which map to the abstract entities defined in PUBLISH according to the following:

- *Publishable item* remains untouched by the integration of PUBLISH and COLLECT, even though it is involved in two publications.
- In the context of *collect request publication*, *holder* maps to *target*, and *collector* maps to *publisher*.
- In the context of *item publication*, *holder* maps to *publisher*, and *collector* maps to *target*.

Taxonomy/Dependencies



Parameters

- Primary pattern author: Michael Derntl
- Primary pattern source: Research Lab for Educational Technologies (University of Vienna)
- Pattern categories: Utility, Generic
- Level of abstraction: Medium
- Scope: Activity, Phase
- Primary presence type: Online
- Flexibility: High
- Level of confidence: 4
- Application effort: Medium
- Input: Items to be collected, Holders of these items, Mode of delivery expected by collector, Deadline for holders
- Output: Items collected, Report

Web Template

COLLECT is a generic utility pattern, which needs to be implemented by other concrete patterns employing this scenario. See for example COLLECT FEEDBACK.

Examples

- The instructor collects PROPOSALS from participants, e.g., in the scope of a LEARNING CONTRACT scenario
- The instructor collects reports and other contributions from participants, e.g., seminar reports.

Evaluation

Not available.

COURSE

Package: *General*

Intent

Courses are arranged primarily in three consecutive phases: PRELIMINARY PHASES in the beginning, followed by the main course phases, and concluded by ASSESSMENT PHASES.

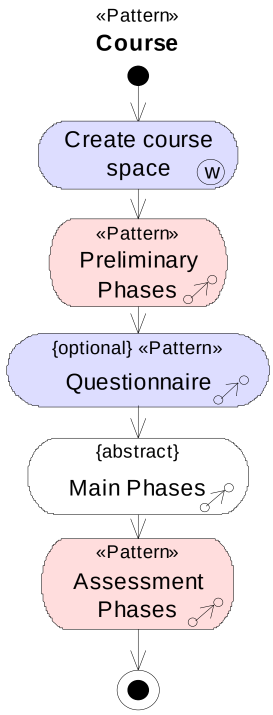
Motivation

This pattern generically describes the arrangement of central phases in any blended course scenario, and serves as a base to be refined by concrete course- or course type patterns, such as PROJECT-BASED LEARNING COURSE.

Blended courses start off with PRELIMINARY PHASES, where the course space on the learning platform is initialized and initial material as well as general information on the course is published to the participants. Subsequently, the INITIAL MEETING is the starting point for learning activities. The following "*main phases*" within the flow of activities in this pattern act as a generic placeholder for concrete blended learning scenarios to be specified by sub-patterns. Finally, any educational course scenario concludes with some form of learner assessment in the scope of ASSESSMENT PHASES.

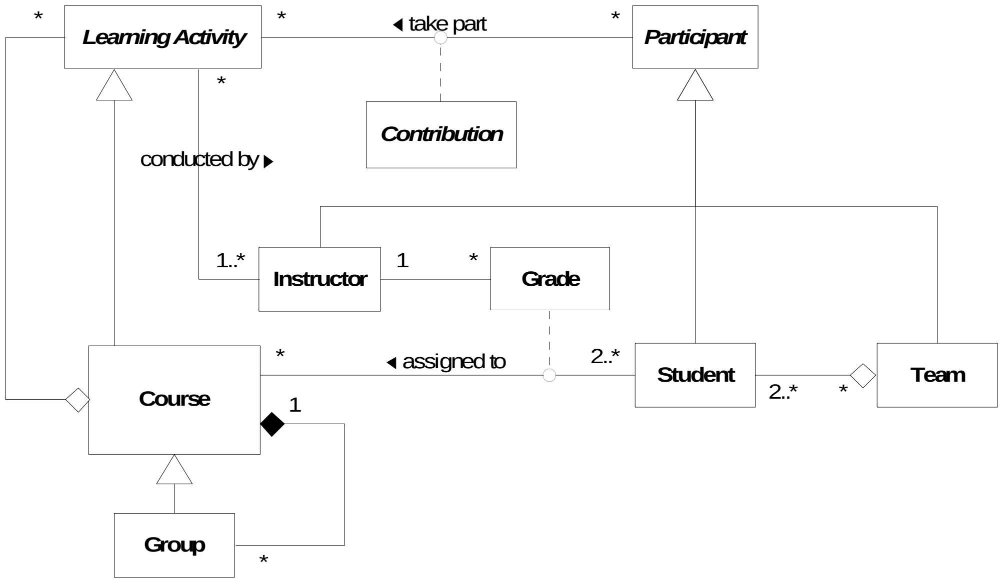
Additionally, the *Structure* section of this pattern provides a generic structural model which is essential for understanding basic relationships among concepts involved and referred to throughout the pattern repository.

Sequence



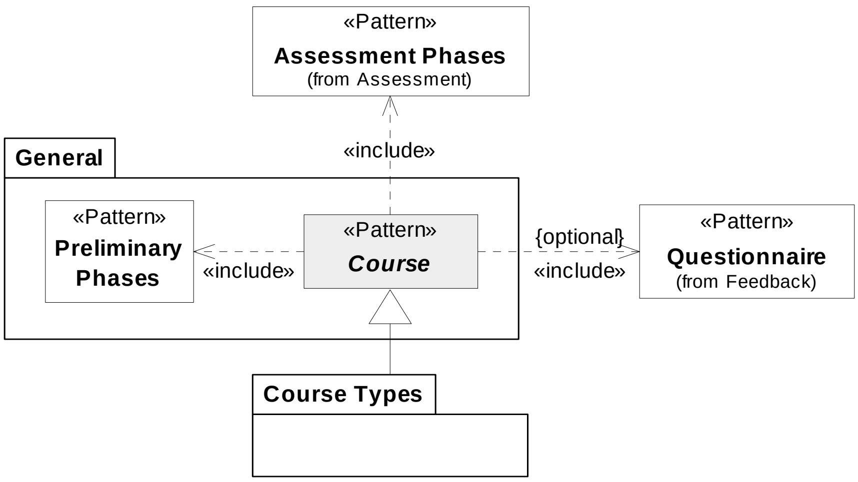
Activity	Description
Create course space	A course space has to be created on the learning platform.
Preliminary Phases	
Questionnaire	The initial questionnaire aims to survey participants' a-priori attitudes and motivations. If complemented with a concluding questionnaire, this can be perfectly used to compare responses at the beginning and the end of the course.
Main Phases	This is an abstract placeholder to be refined by sub-patterns that specify concrete blended learning scenarios as main phases of the course, e.g. Project-Based Learning in a Project-Based Learning Course.
Assessment Phases	

Structure



The structural model shows basic relationships among general entities involved in *courses*. In many educational environments, courses are divided into a number of *groups*. A course consists of a number of *learning activities*, while it itself is a certain form of a (compound) learning activity. At least one *instructor* is associated with a learning activity. Additionally, at least two *students* participate in a course. Each instance of this relationship has an associated *grade*, which is assigned by the instructor. At least two students may be joined to form one *team*. Usually, teams consist of 2 to 5 members, depending on the respective learning activity. Within the scope of this work, participating students, tutors, as well as teams are considered as *participants* of the course; thus, a participant is an abstract concept which is often used here to describe someone who takes part in learning activities and courses, regardless of his or her actual role. Participants elaborate, deliver, or perform a number of *contributions* during taking part in a learning activity.

Taxonomy/Dependencies



COURSE is a generic top-level pattern that has no parent. It is located in the *General* package, and is refined by a number of derived sub-patterns, each representing a course type with different learning goals and scenarios in its main phases.

Parameters

- Primary pattern author: Michael Derntl
- Primary pattern source: Research Lab for Educational Technologies (University of Vienna)
- Pattern categories: Generic, Composite
- Level of Abstraction: High
- Scope: Course
- Primary presence type: Blended
- Flexibility: Low
- Level of confidence: 5
- Number of participants: Unrestricted
- Input: Course parameters, Groups, Participants, Teams, Instructors (all for initialization)

Web Template

This pattern contains one Web-supported, top-level activity: *Create course spaces*. This activity is the starting point for any blended learning course that is conducted using this pattern repository.

Participant View

The participant view contains only the course homepage. Its layout depends on the visual styles of the employed learning platform solution and it is initially empty, so no screenshot is depicted here. Anyway, in the subsequent instance of the PRELIMINARY PHASES pattern, the homepage is populated with relevant information.

Administration View

The administrator has the major part in the course initialization activities. While course initialization is definitely a platform-specific activity, the administration view of this composite, generic pattern supplies links to the administration views of the following patterns, each of which is included in the activity sequence of COURSE:

- PRELIMINARY PHASES, for populating the initially empty course space on the learning platform with relevant information
- QUESTIONNAIRE, which may optionally be used at the beginning of the course to quantitatively assess participants expectations, estimations, or any other matter of interest (e.g., for comparison with results of a final questionnaire).
- Any pattern that is appropriate for being used in the abstract *main course phases* of the course sequence, which may include patterns at course scope such as PROJECT-BASED

LEARNING or LEARNING CONTRACTS, but also accompanying “helper” patterns such as DIARY, TUTORIAL, ONLINE DISCUSSION, etc.

- ASSESSMENT PHASES, which is a composite pattern itself and will guide the administrator to concrete assessment/evaluation patterns.

Report View

Not applicable.

Examples

Not available. Refer to concrete course patterns.

Evaluation

This section relies on a questionnaire that was developed, constructed and used at the Department of Computer Science and Business Informatics (University of Vienna) [523]. The questionnaire items can be used in various courses, and examples are given for one PCeL course on Web Engineering held in the summer term 2003.

General questions

The following item block shows a list of possible general questions to ask in a course. The scale reaches from 1 (=*does not apply*) to 5 (=*applies totally*).

1) The course was conducted differently compared to other courses of my study
2) I have shown high engagement in the course
3) I tried to keep my working in the course at a minimum
4) I worked through presented subject matter thoroughly
5) I tried to complete the course with minimal efforts
6) I have completed my tasks with pleasure

Figure 84 below shows results for these items in Web Engineering (summer term 2003).

[523] The questionnaire was constructed by Renate Motschnig and Katharina Mallich (see also Mallich (2003))

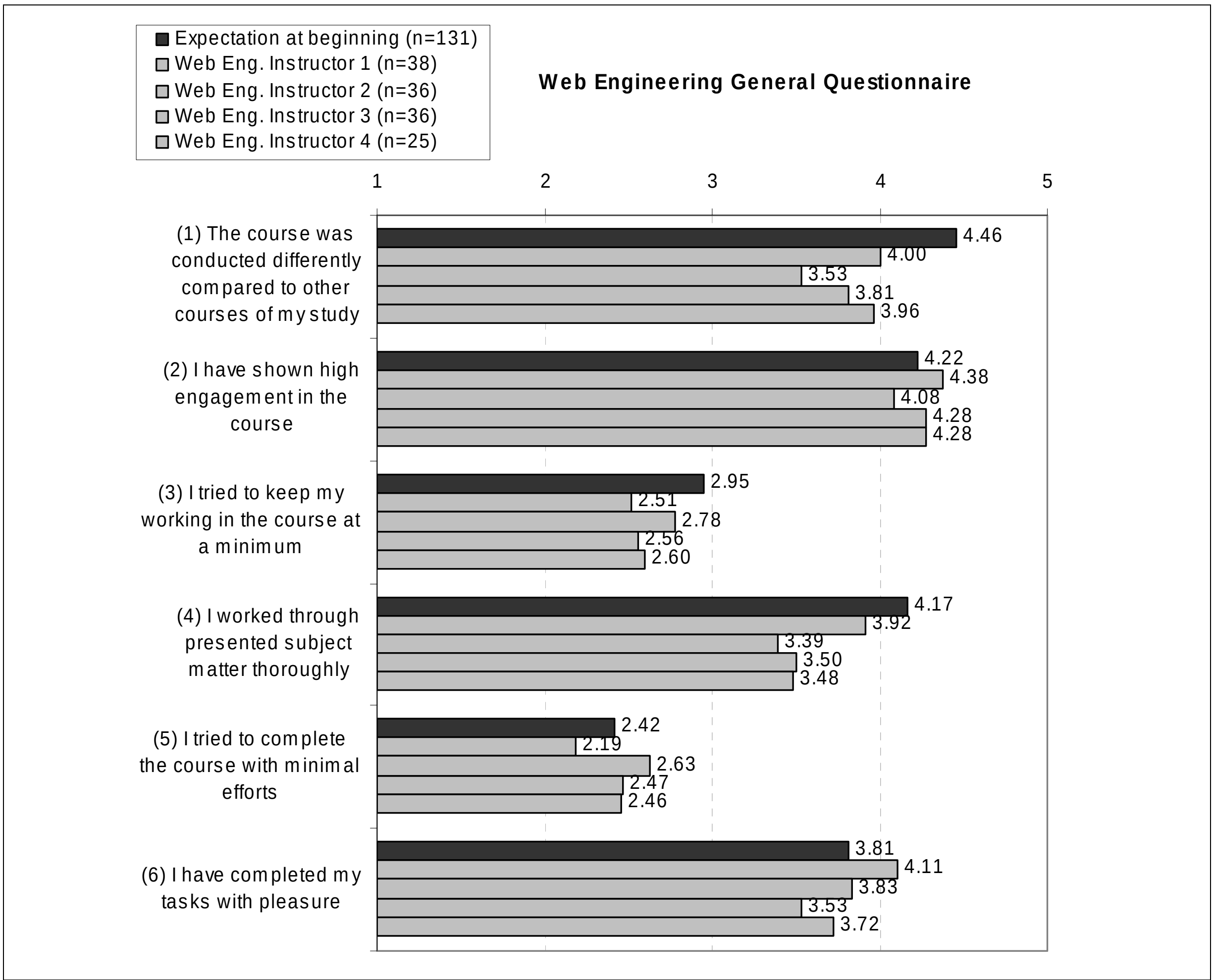


Figure 84: Results of general questionnaire in Web Engineering.
The top black bar for each item shows the mean expected value that was surveyed in an initial questionnaire at the beginning of the course. The bars below depict mean values surveyed at the end of the course grouped by instructors 1–4.

Motivational orientation

The following item block aims at determining different dimensions of the *motivation* of students to participate in a specific course. These include:

- ***Success*** (items 1, 2, and 21): The primary aim of participating is completing the course successfully and getting a good grade.
- ***Obligation toward the instructor*** (items 4, 10, 15, and 18): The primary motive is a perceived obligation toward the instructor, so to say, "learning for the sake of the instructor".
- ***Competition*** (items 5, 8, and 9): The primary learning motivation is grounded on competition with peers/colleagues.
- ***Competence*** (items 11, 12, and 19): Acquiring competence and technical skills is the primary motivation.

- ***Interest*** (items 3, 6, and 14): Influence of personal interests on motivation to participate.
- ***Course style*** (items 7, 13, 16, 17, 20, and 22): The course style/mode itself is a motivating factor.

The items are equally interval scaled from 1 to 5 (whereas 1 means *low motivation* and 5 means *high motivation*).

I participated in the course, because...

1) I wanted to get a good grade.
2) passing a course successfully makes me feel well
3) I had great interest in the content provided
4) the instructor expected this from me
5) I wanted to get a better exam grade than my colleagues
6) I wanted to get further education in the course's subject matter
7) the way the course was conducted appealed to me
8) it was important to me to show that I study more diligently than others
9) I wanted to be among the best in the course
10) I wanted to avoid getting in trouble with the instructor
11) I wanted to acquire comprehensive knowledge in subject area
12) subject matter was elaborated with practical application in mind
13) I liked the atmosphere / the working climate of the course
14) I liked to occupy myself with the course's content
15) I didn't want to disappoint the instructor
16) active participation of all students was possible
17) there was sufficient room for discussions
18) I wanted to be considered a good student by the instructor
19) I wanted to improve my professional skills
20) the cooperation with peers was very collegial
21) I wanted to prove to myself that I can pass the course very successfully
22) I found it important to improve my soft skills

Figure 85 through Figure 90 below show results of the motivational survey in the Web Engineering course conducted in the summer term 2003. As the results vary significantly for different instructors, the histograms show the underlying data grouped by instructors. One general pattern that can be observed from these histograms is that instructor 1 typically receives the highest results. This is presumably due to the fact that motivational orientations correlate with participants' perception of person-centered attitudes of the instructor[524],

[524] See, for example, Derntl and Motschnig-Pitrik (2004c), or Motschnig-Pitrik, Derntl and Mangler (2003)

whereas instructor 1 was perceived as being by far the most person-centered of the four instructors (cf. Figure 94).

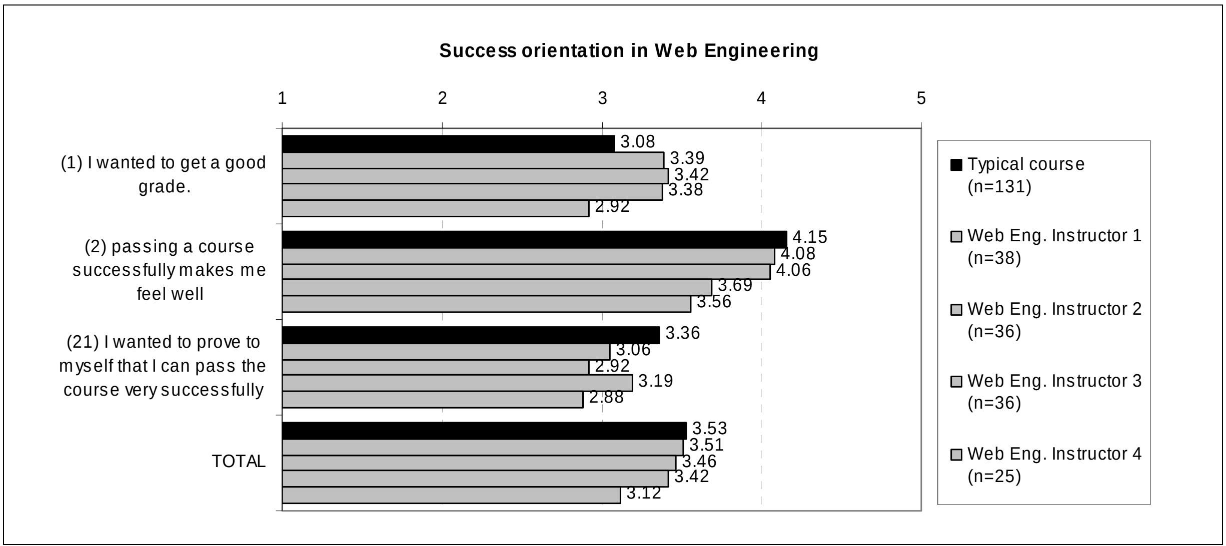


Figure 85: Motivation due to success orientation in Web Engineering.
The overall success orientation is slightly lower when compared with a typical course.

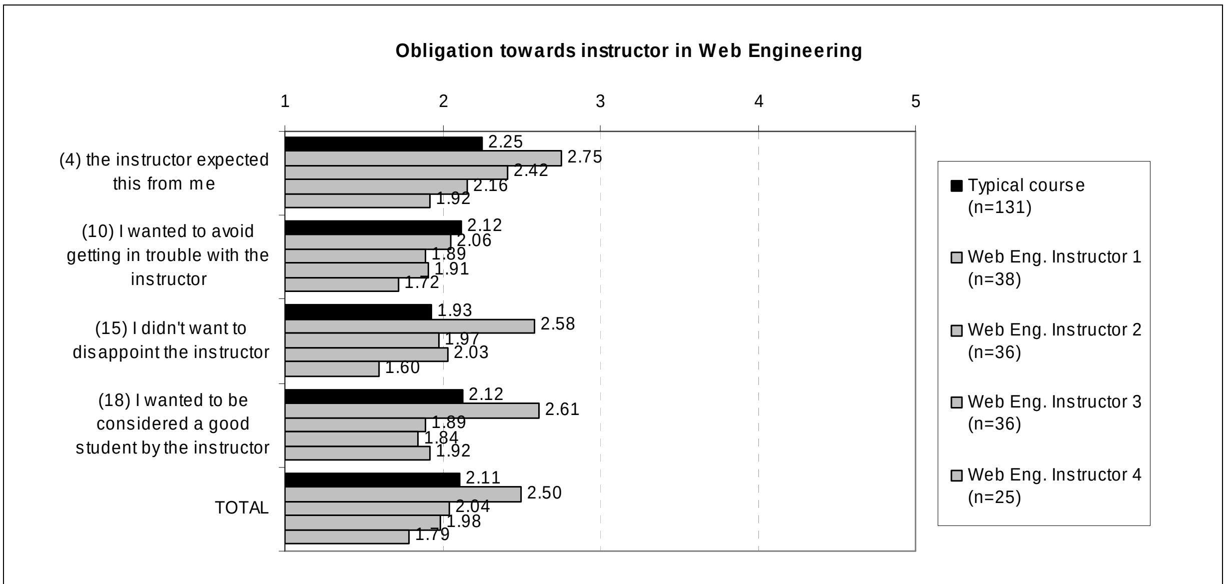


Figure 86: Motivation due to obligation towards instructor in Web Engineering.
Obligation towards instructor 1 is significantly higher in Web Engineering when compared with a typical course.

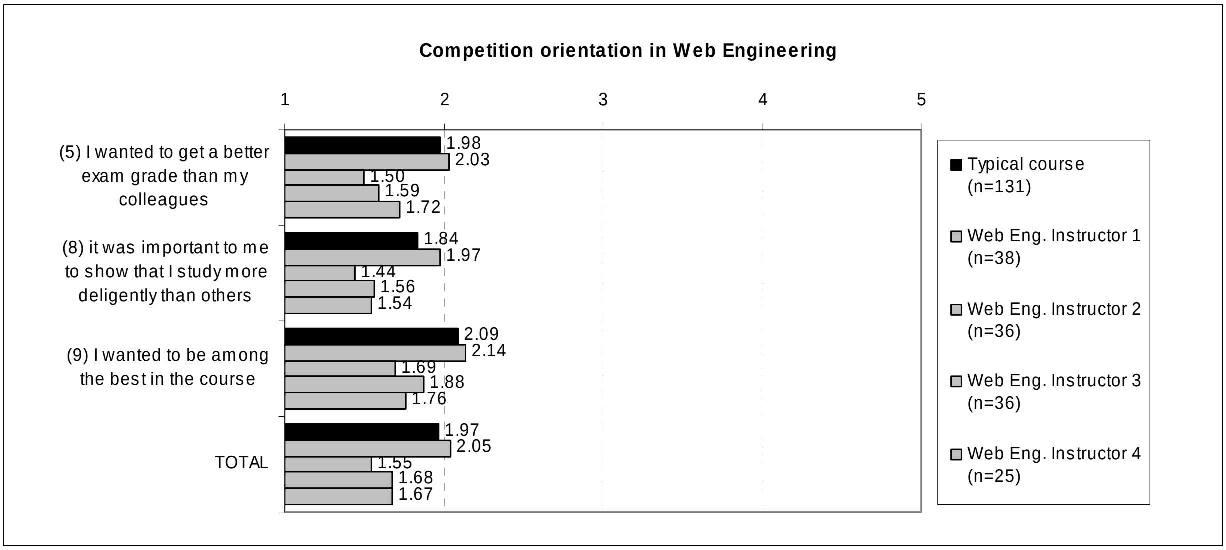


Figure 87: Motivation due to competition orientation in Web Engineering.
Competition is obviously not a primary influencing factor of motivation in Web Engineering. Participants had been encouraged to cooperate more than to compete.

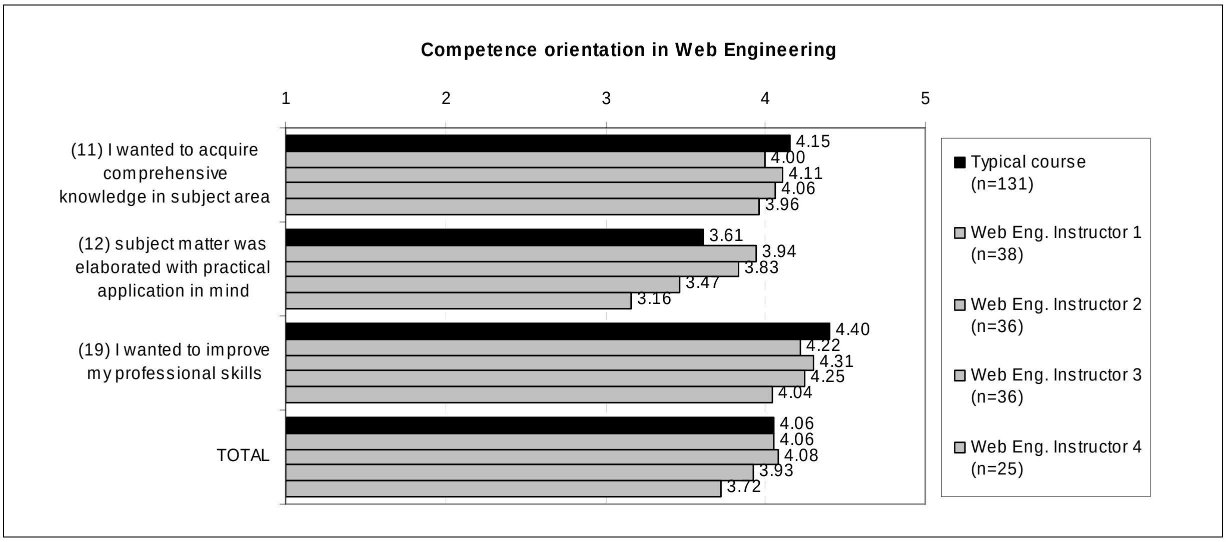


Figure 88: Motivation due to competence orientation in Web Engineering.
As this histogram shows, acquiring skills and competence is one of the central motivational factors for participants, in conventional as well as in person-centered courses.

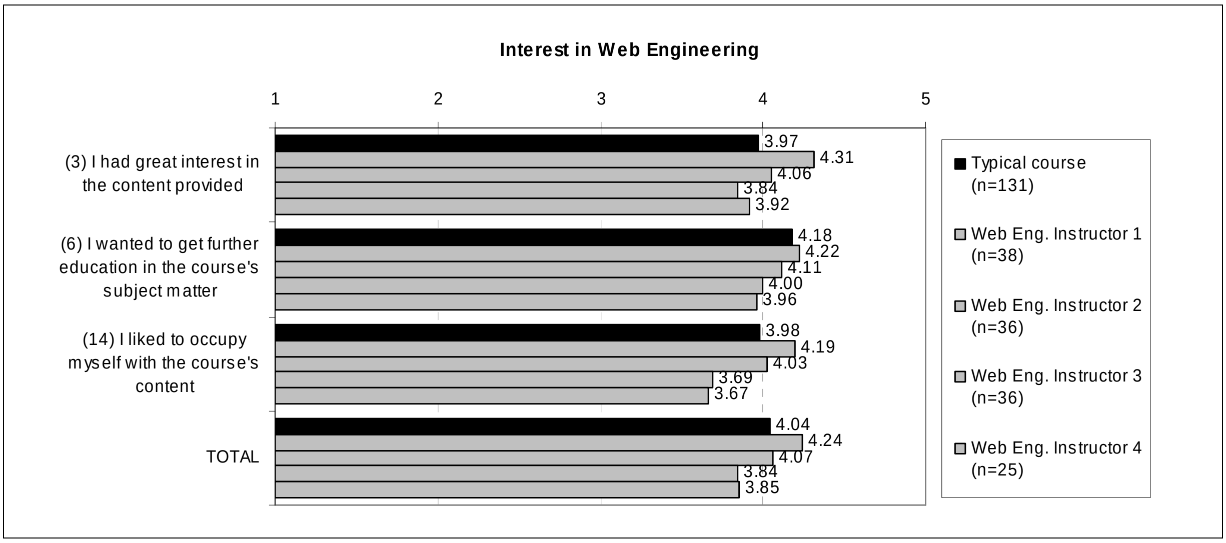


Figure 89: Motivation due to interest in Web Engineering.

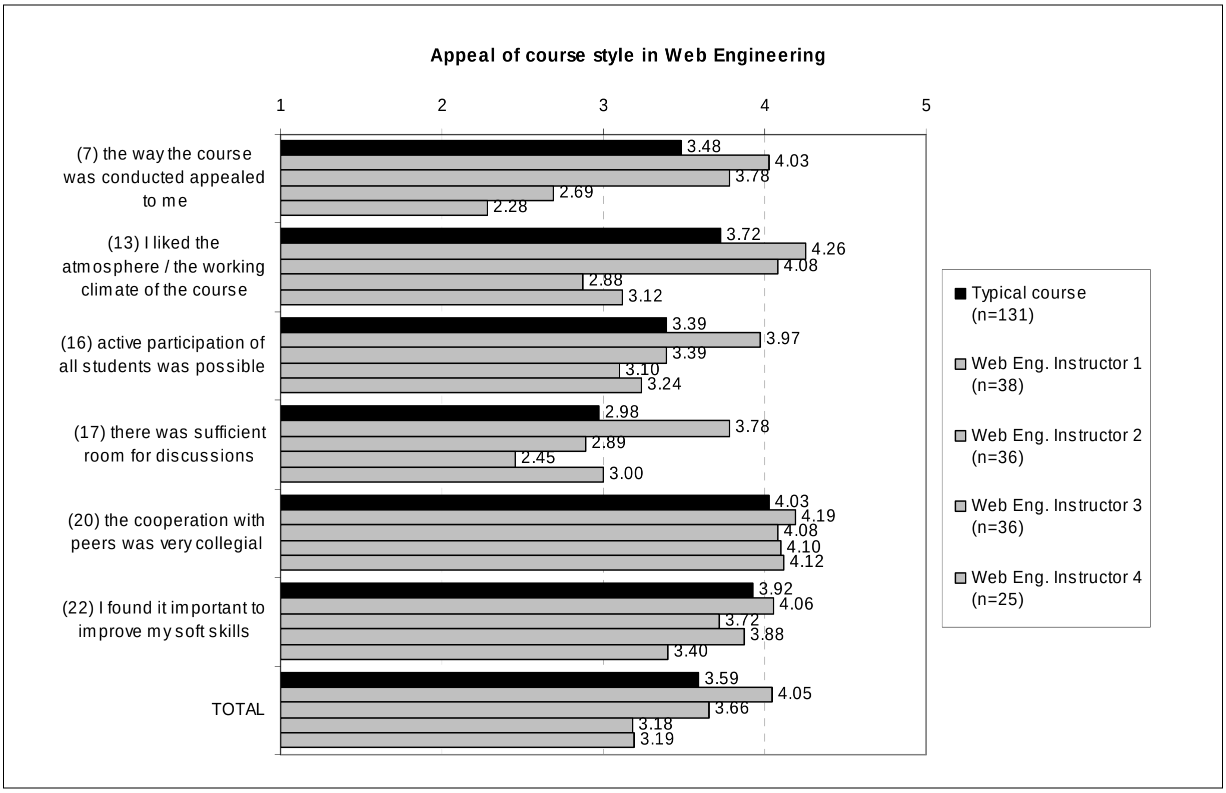


Figure 90: Motivation due to course style in Web Engineering.
The innovative course style of Web Engineering is perceived as superior to typical conventional scenarios only when accompanied by high interpersonal values of instructors (there is high correlation with person-centered attitudes).

Learning Aspects

The next item block surveys participants' perception of specific learning aspects in a course, whereas the scale reaches from 1 (= *not at all*) to 5 (= *very much*)

I benefited from...

1) the materials and literature references provided in the course
2) the materials I collected myself (library, Internet, etc.)
3) the practical exercises during the lab hours
4) the practical work at home
5) the web-based communication on a learning platform
6) the active participation in the course
7) cooperation with peers in teams
8) exchange and discussion with colleagues
9) exchange and discussion with the instructor

Figure 91 shows these items' results for Web Engineering. For most of the items, a typical course receives higher values than the Web Engineering course. Thereby it is important to mention, that the values for the typical course have been surveyed at the beginning of the term where participants' motivations and prospects are usually higher than those at the end of a long, hard term[525] (the Web Engineering data were collected at the end of the term).

[525] See also Rogers and Freiberg (1994)

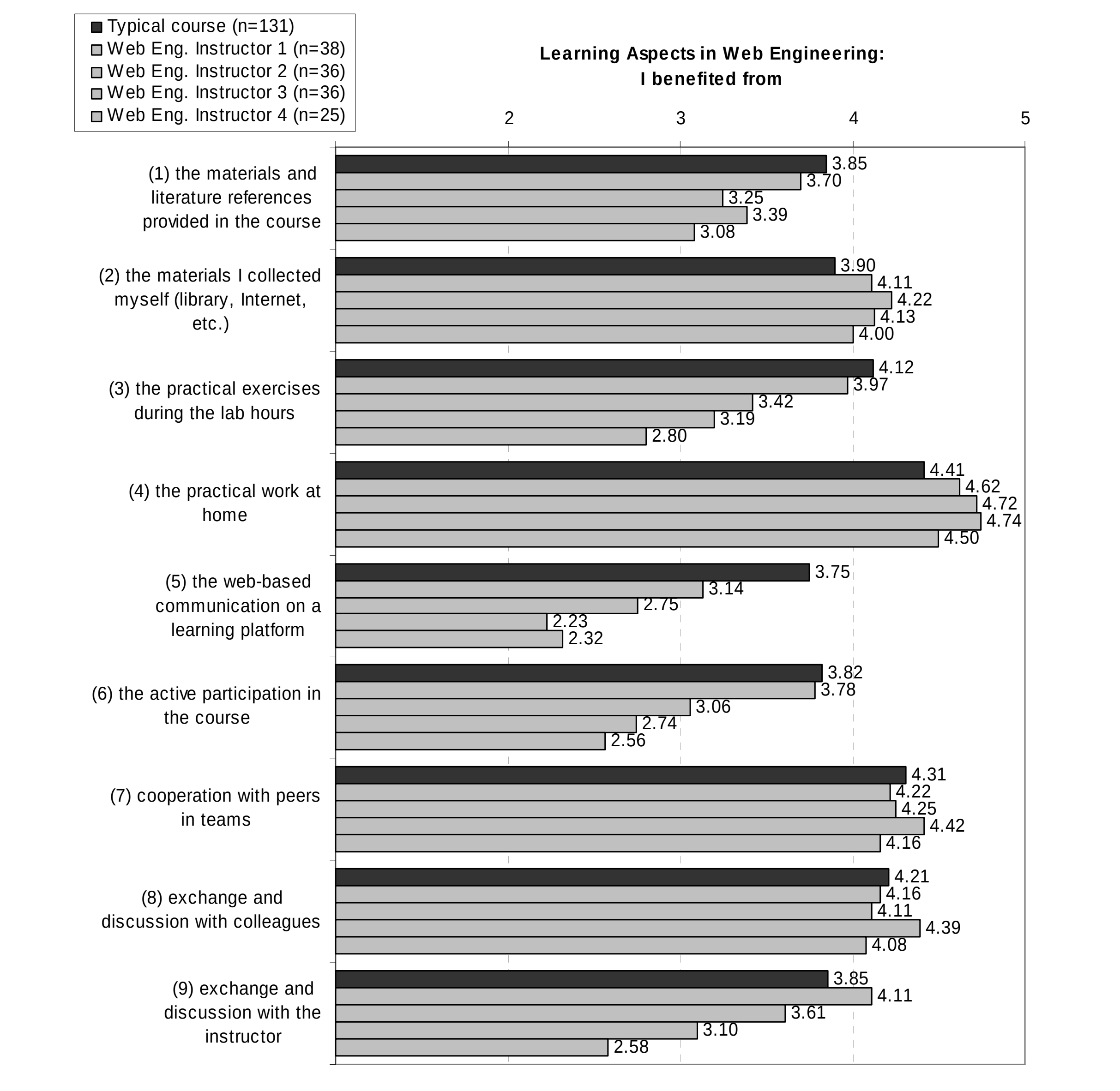


Figure 91: Learning aspects in Web Engineering.

Skills

The next block of items surveys participants' perception of skills the course has imparted and transported. The item scale is the same as for the learning aspects above (1 = *not at all* to 5 = *very much*).

The course has imparted:

1) Factual knowledge in the subject area
2) Practical knowledge
3) Orientation within the subject area
4) Producing work reports
5) Presenting results

6)	Personal time management
7)	Collaboration with colleagues in teams
8)	Communication in the team
9)	Importance of interpersonal relationships within the team

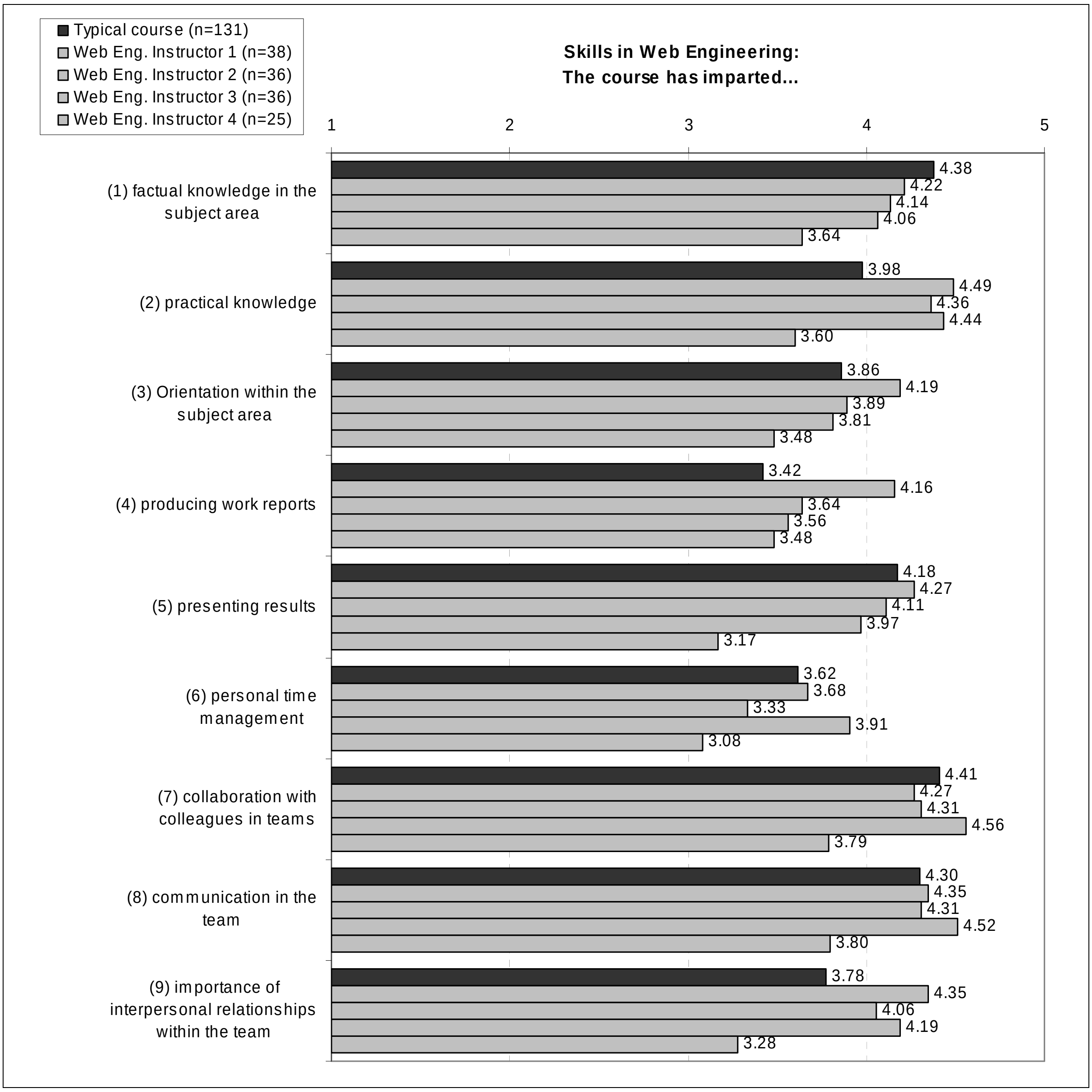


Figure 92: Skills in Web Engineering.
Particularly for practical knowledge (item 2), producing work reports (4), and interpersonal relationships (9), Web Engineering is perceived superior when compared to a typical course.

Learning platform support

It seems worthwhile to survey the degree of support the employed learning platform solution offered to participants. However, as such a survey is highly platform-specific and course-specific no default questions can be offered here. It is advisable to anchor questions to scenarios that are employed in a course (e.g., "*degree of learning platform support in peer-evaluation?*"). The items that were used in Web Engineering (along with evaluation results) are depicted in the histogram in Figure 93 below.

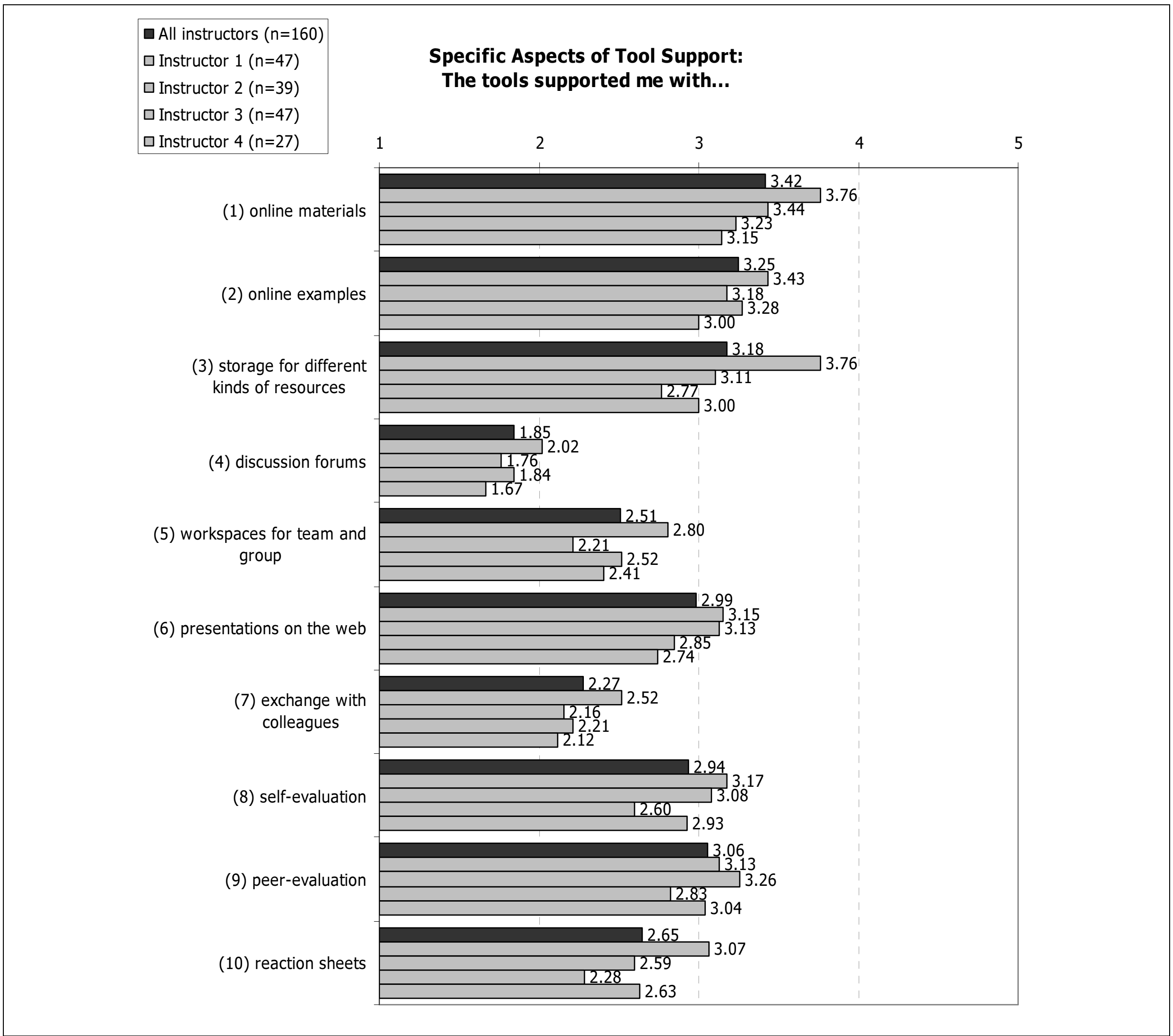


Figure 93: Specific aspects of tool support in Web Engineering.[526]

Person-Centered Attitudes

As the significant effects of instructor's Person-Centered attitudes have been reported in past[527] and recent[528] studies, it is worthwhile to query students' perception of these attitudes

---

[526] Remarks regarding the comparatively low value of items 4 and 5: online discussion and workspaces have been supported on a learning platform that was still at an early development stage and thus showed many flaws. Naturally, this had caused participants to feel inconvenience and reluctance during usage. (See also the *Evaluation* section of ONLINE DISCUSSION.)

[527] E.g., Rogers (1983)

[528] E.g., Chase and Geldenhuys (2001), Derntl and Motschnig-Pitrik (2004c)

in any course. For example, the following questionnaire was used in the Software Engineering, Project Management, and Web Engineering courses[529]:

Response Behavior: The instructor responds...	
1 =	in a destructive, de-motivating way
2 =	ineffectively, presumptuously
3 =	minimally effectively
4 =	in a way that notably supports making progress
5 =	in an encouraging, supportive way that significantly contributes to making progress
Transparency: The instructor...	
1 =	avoids questions, in not accessible and refuses open conversation
2 =	hides between his/her position and it is difficult to transparently communicate with him/her
3 =	gives clear answers to some minor degree
4 =	tends to communicate openly and transparently
5 =	communicates totally transparently, gives frank responses and is perceived as genuine and real
Acceptance: The instructor...	
1 =	meets students without any respect, does not consider their requests
2 =	hardly respects the students' requests and demands
3 =	respects the students' needs and requests to some minor degree
4 =	is generally respectful towards students and encourages them
5 =	is friendly, full of trust in students, encourages them, and lets them perceive his/her respect
Empathic Understanding: The instructor...	
1 =	completely ignores the students' needs
2 =	hardly responds to the students' needs and interests
3 =	to a minor degree reacts to what the students communicate
4 =	often reacts to what the students say such that they feel understood
5 =	completely understands students' needs and interests, reacts to students in a supportive way
Professional Competence: The instructor...	
1 =	does not at all technically find his/her way
2 =	seems technically rather incompetent
3 =	has passable factual knowledge and is sufficiently able to convey subject matter
4 =	leaves a competent impression and skillfully conveys subject matter
5 =	shows technical expertise and is capable of inspiring students

529 An initial version of this questionnaire is reported in Motschnig-Pitrik (2002b)

Figure 94 below depicts results obtained in Web Engineering. It shows significant differences for different instructors. These differences correlate significantly[530] with differences among instructors for many of the items presented above for motivation, learning aspects, and skills.

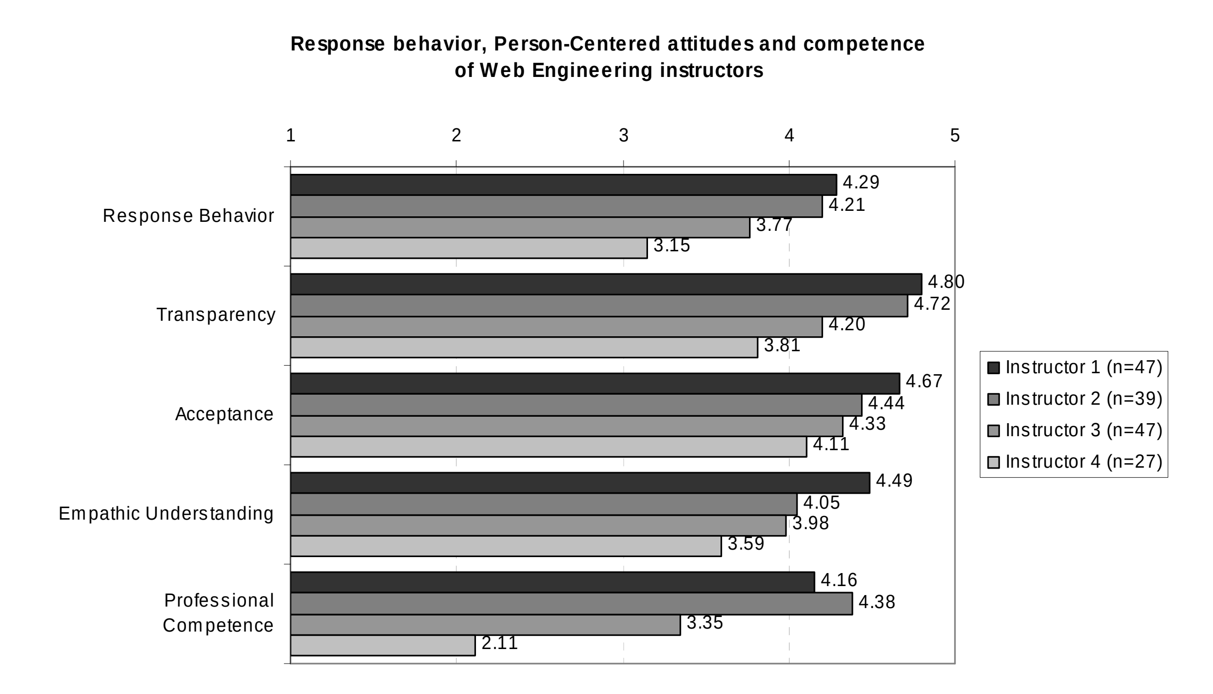


Figure 94: Response behavior, Person-Centered attitudes, and competence of four Web Engineering instructors.

References

Chase, C. C., & Geldenhuys, K. M. (2001). Student-centred teaching in a large heterogeneous class. *Medical Education, 35* (11), 1071.

Derntl, M., & Motschnig-Pitrik, R. (2004). *Patterns for Blended, Person-Centered Learning: Strategy, Concepts, Experiences, and Evaluation.* Proceedings of 2004 ACM Symposium on Applied Computing (SAC), Nicosia, Cyprus, pp. 916-923.

Mallich, K. (2003). *Auswertungen zur Lehrveranstaltung Web Engineering - SS 2003* (Technical Report): Department of Computer Science and Business Informatics, University of Vienna.

Motschnig-Pitrik, R. (2002). *Supporting Student-Centred Teaching with New Media: Case Study and Experience Report.* Proceedings of 3rd International Conference on Networked Learning (NLC) 2002, Sheffield, U.K, pp. 456-464.

Motschnig-Pitrik, R., Derntl, M., & Mangler, J. (2003). *Web-Support for Learning Contracts: Concept and Experiences.* Proceedings of Second International Conference on Multimedia and Information & Communication Technologies in Education (m-ICTE'03), Badajoz, Spain.

Rogers, C. R. (1983). *Freedom to Learn for the 80's.* Columbus, Ohio: Charles E. Merrill Publishing Company.

[530] Cf. Mallich (2003)

Rogers, C. R., & Freiberg, H. J. (1994). *Freedom to Learn* (3rd ed.). Columbus, Ohio: Charles E. Merrill Publishing Co.

DIARY

Package: *General*

Intent

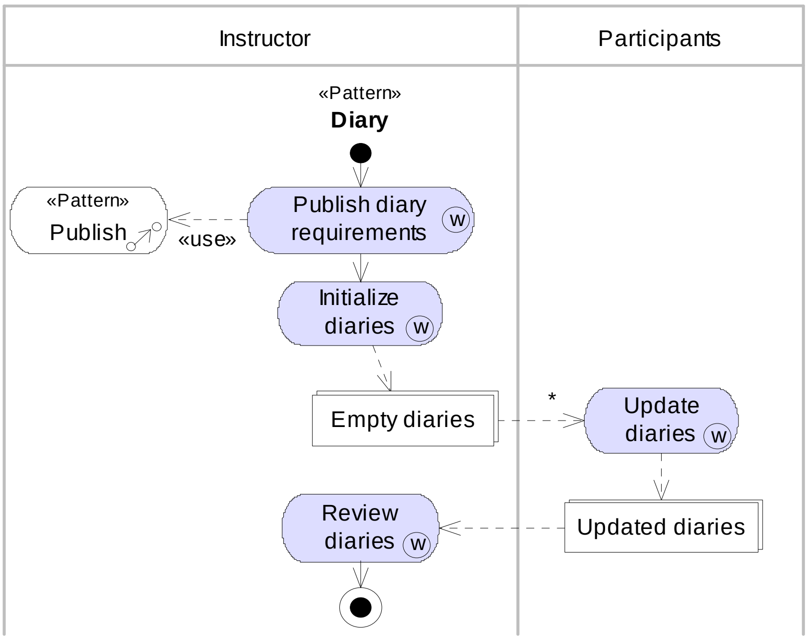
Make participants' efforts transparent by making them keep track of their work in diaries, especially in collaborative and/or iterative learning processes.

Motivation

Keeping a diary during learning activities has some positive aspects: First, diaries help memorizing and reflecting on experiences, problems, and achievements encountered during problem-solving processes. So diaries are valuable assets to initiate self-reflective thinking. Second, diaries provide the instructor with valuable insight on working progress, activity, and thoughts of participants. Finally, in teamwork scenarios diaries allow for monitoring the (equal) distribution of activities among team members.

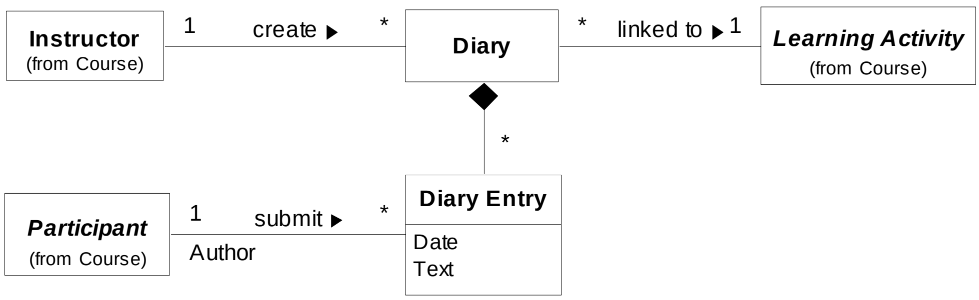
All of the above points only hold when participants do not treat diaries as annoying appendages but take a stake in keeping their diaries. To facilitate this, it may be worthwhile to provide writing guidelines for participants (e.g., providing a set of questions and points to be considered) and to transparently disclose the intentions of making them keep diaries.

Sequence

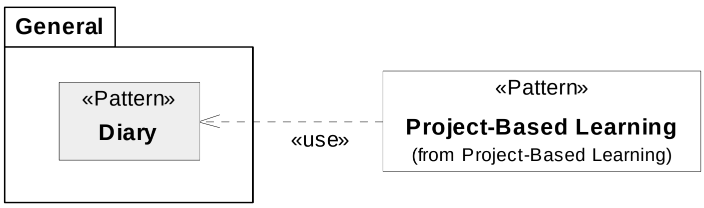


Activity	Description
Publish diary requirements	The instructor publishes the diary requirements including guidelines on what kind diary entries the instructor expects.
Initialize diaries	Diaries are initialized on the platform for each team or student, producing empty diaries to be populated by participants.
Update diaries	Participants periodically update their diaries according to the requirements and guidelines specified by the instructor.
Review diaries	Diaries are accessible for review by the instructor. When desired, diaries can also be made public to other participants.

Structure



Taxonomy/Dependencies



Even though DIARY is explicitly used only by PROJECT-BASED LEARNING in the PCeL pattern repository, it may be used in many other circumstances as a collateral activity, for example for whole courses, or just for single learning activities.

Web Template

Administration View

The administration view for diaries bears only few options and can be presented on one single page (see Figure 95):

- *Diary title*: Is displayed as the heading on the diary page in the participant view. Additionally, this title may be used as the link text leading to the diary page.

- *Introduction/requirements*: Here the instructor supplies information and requirements regarding usage of the diary. This introductory text is displayed below the heading in the participant view.
- *User mode*: Specifies whether diaries are used *individually* (i.e., each participant has his/her own diary) or in *team mode* (each team shares one diary). This setting has no visual impact on the participant view.

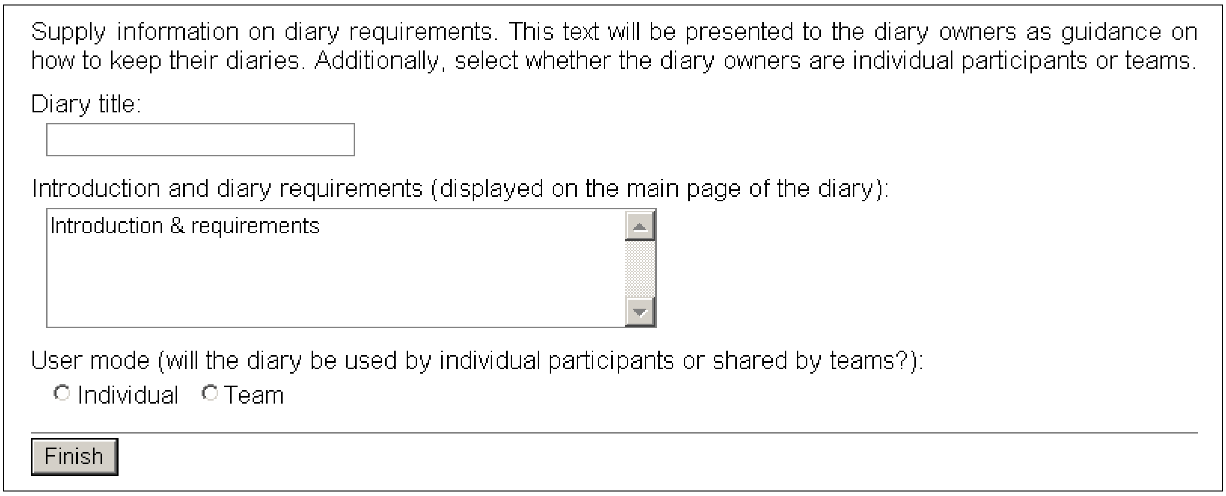


Figure 95: One-step DIARY administration template.

Participant View

The diary overview page (Figure 96), which is the entry page to the Web-based diary, shows the current diary log of the currently logged in user (or team member). From here, the user can click on the *new entry* hyperlink to create a new diary entry (see Figure 97), *reload* the list, or *delete* existing entries. Additionally, by clicking on one of the entries subjects in the log table, the details (including the body text) of the entry are displayed to the user.

Figure 96: DIARY overview page.

The page for submitting diaries includes three controls: The *date/time* which is associated with that diary entry, as well as the *subject* and body *text* of the diary entry. Note that the date/time and subject fields are presented in the diary overview page.

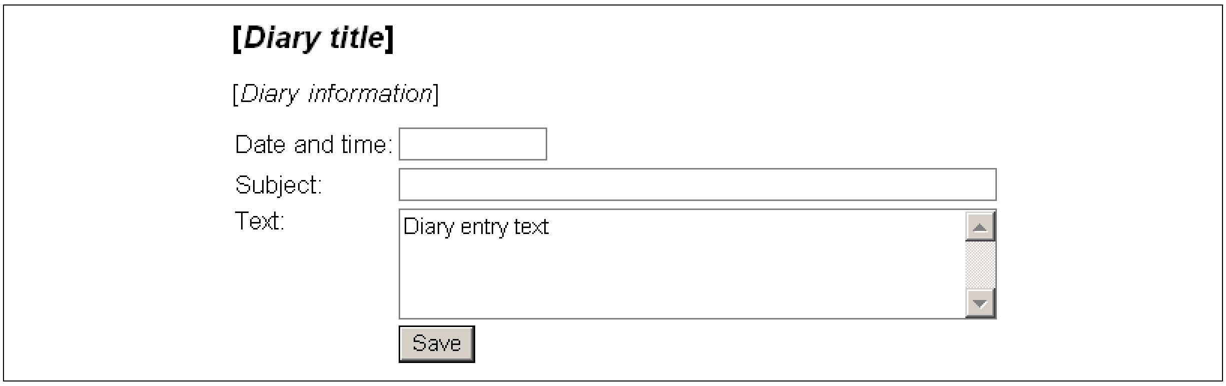


Figure 97: DIARY entry page template.

Report View

The report view for DIARY includes simply a control for selecting the diary owner whose diary contents shall be displayed in full detail (see Figure 98). Additionally, instructors and reviewers may browse through the diaries even in participant view.

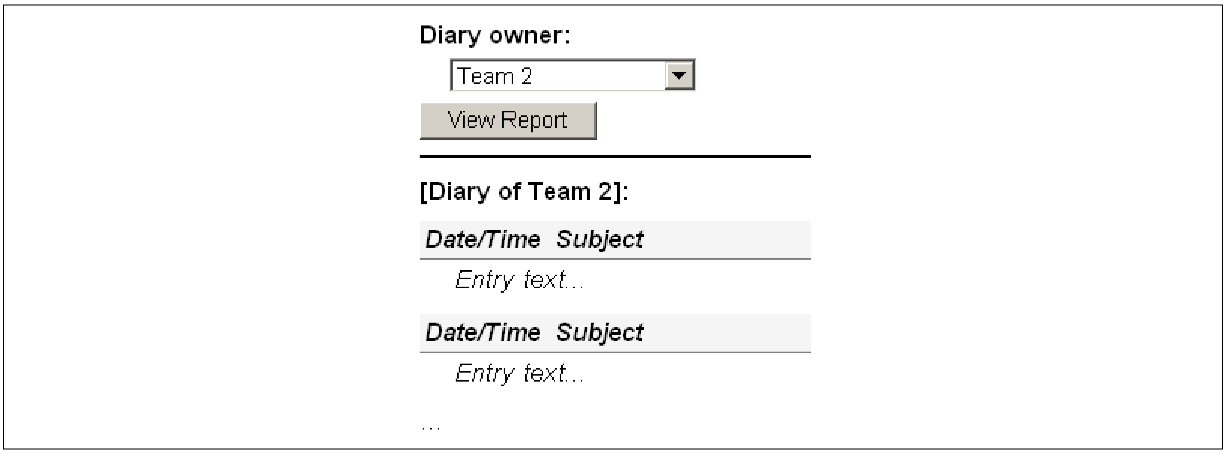


Figure 98: DIARY report page.

Parameters

- Primary pattern author: Michael Derntl
- Primary pattern source: Research Lab for Educational Technologies (University of Vienna)
- Pattern categories: Collateral, Utility
- Level of abstraction: Low
- Scope: Phase, Course
- Primary presence type: Online
- Flexibility: High
- Level of confidence: 3
- Number of participants: Unrestricted
- Application effort: Medium
  Depending on number of participants
- Level of expertise required: Low

- Target skills: Reflective thinking
- Input: Diary space, Guidelines
- Output: Diaries

Examples

Diaries were used in ***Web Engineering*** to document work on the LEARNING CONTRACTS. However, in 2003 and 2004, no online diaries were used. Rather, participants were asked to document their project work in a diary that was subject to inspection in the final oral colloquium of the learning contracts. In 2005, online diaries were provided for participants. A screenshot from the diary log of a random participant is given in Figure 99.

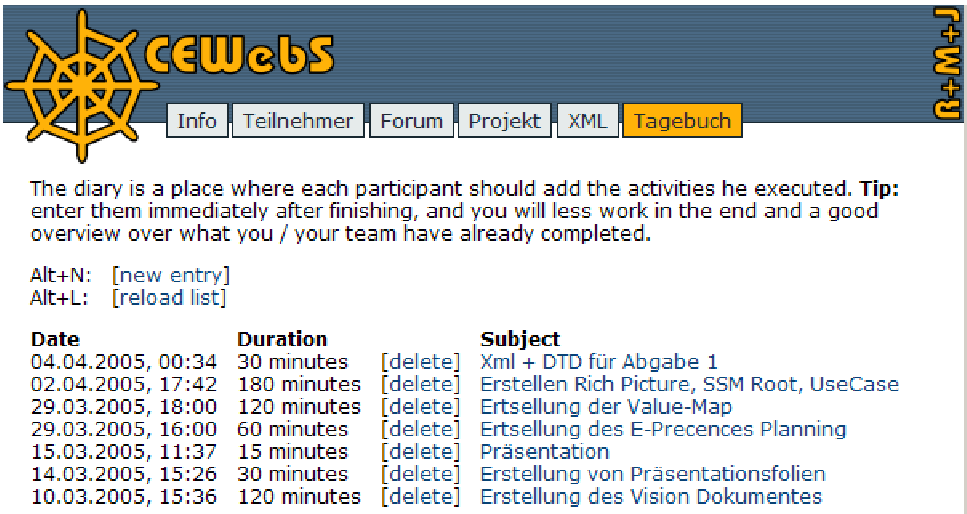


Figure 99: Some participant's lab DIARY in Web Engineering 2005.

Evaluation

Not available.

INITIAL MEETING

Package: *General*

Intent

Outline course style and objectives in an initial MEETING and CONSIDER CONVENTIONAL STYLE.

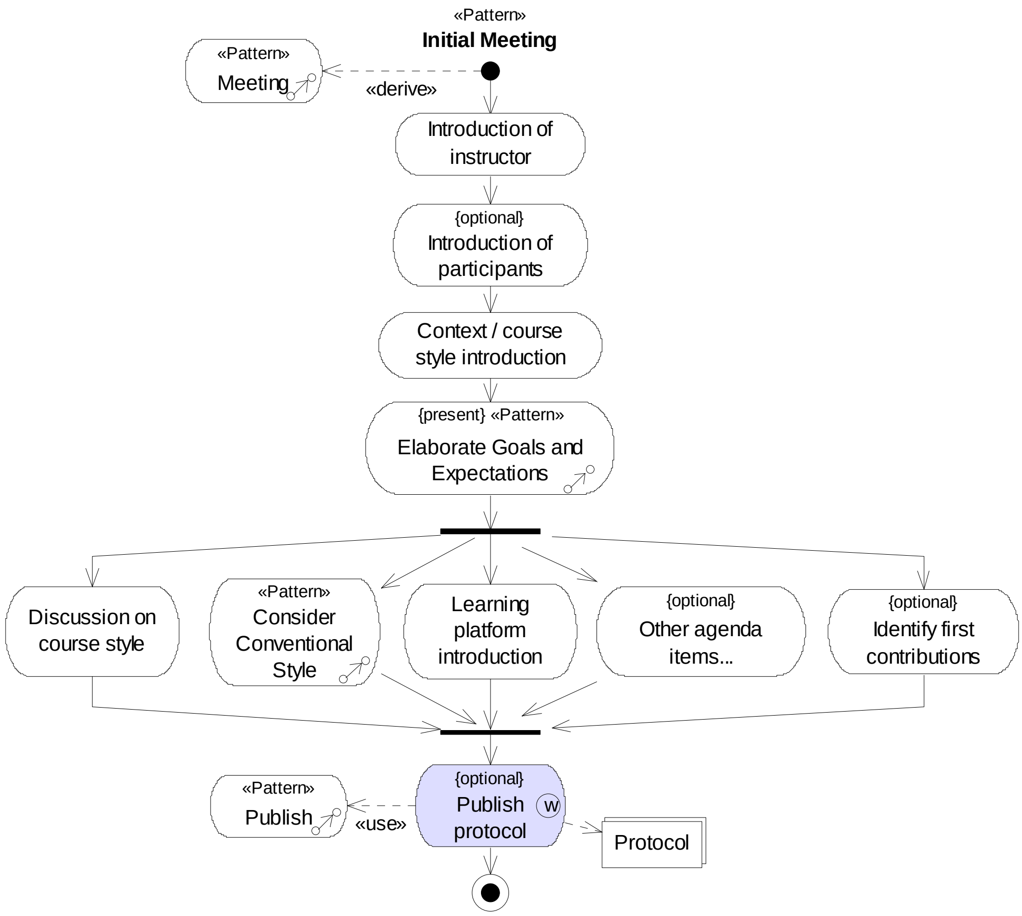
Motivation

Initial meetings act as face-to-face kickoffs for courses. It is the first time that participants and instructor come together to confer on general organizational and administrative issues, to present first content, to ELABORATE GOALS AND EXPECTATIONS and to prepare subsequent phases.

It is crucial to be careful and transparent in an INITIAL MEETING to avoid later reneging on promises, or demanding something which was not announced initially. If a novel, unusual course scenario is employed, some of the participants may feel reluctance or uneasiness. In such a case, the instructor may acceptingly CONSIDER CONVENTIONAL STYLE, offering them a higher degree of direction.

This pattern is a specialized form of MEETING and is typically included in PRELIMINARY PHASES, so the meeting relies on a prior preparatory online phase where general information of interest to participants was published by the instructor.

Sequence



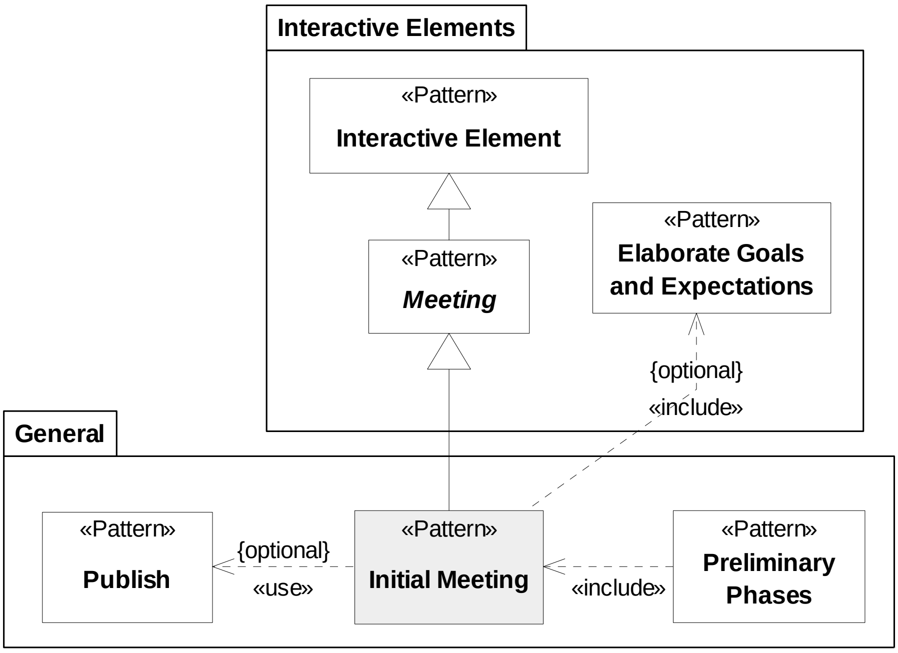
Activity	Description
Introduction of instructor	The instructor personally introduces himself to the participants. Typically, this includes personal interests and background, affiliation, availability to participants (office hours, e-mail address, etc.), and others as appropriate.
Introduction of partici-	When participants introduce themselves, their interests and back-

Activity	Description
pants	grounds, the instructor gets a picture of existing skills, expertise, and knowledge of the learners.
Context/course style introduction	Thematic context as well as focus of the course is of central interest to participants. Additionally, course style and objectives should are introduced.
Elaborate Goals and Expectations	Even if the thematic course context is set by the instructor (and mostly influenced by curricular requirements), providing space for shifting or adding specific topics of interest is an eminent motivational factor capable of raising learner motivation and dedication. To achieve this, Elaborate Goals and Expectations (constrained to presence activities) may be used in a.
Discussion on course style	Questions may arise on the presented course style, so participants' opinions should be considered in a discussion on the course style.
Consider Conventional Style	Students are asked to choose between a more conventional style and a person-centered one. If students choose conventional style the instructor conducts the course in a mainly directive way. Else students are provided with much more freedom and participation. Those feeling reluctant or uneasy to participate in an unusual scenario may be offered to switch to a more directed, conventional style. In the case studies underlying this pattern no participant ever really refused to join the innovative style, although in some cases a considerable degree of uneasiness in some participants was perceived by the instructor. Offering a directed style on demand to these participants has proven to strongly relieve uneasiness.
Identify first contributions	First contributions to be elaborated by participants may be identified or assigned here.
Instructor: publish protocol	A protocol should be published to make decisions, to-dos, and assignments made in the meeting available for lookup.

Structure

Inherited.

Taxonomy/Dependencies



Parameters

- Primary pattern author: Michael Derntl
- Primary pattern source: Research Lab for Educational Technologies (University of Vienna)
- Pattern categories: Utility, Motivational
- Level of abstraction: Low
- Scope: Activity
- Primary presence type: Present
- Flexibility: High
- Level of confidence: 5
- Number of participants: Up to 30
- Person-Centered variables addressed: Acceptance, Transparency, Understanding
- Application effort: High
- Level of expertise required: High
- Input: Preparation of meeting agenda, Online information, Course scenario
- Output: Additional focal topics, List of participants, Protocol (optional)

Web Template

Not available. See parent pattern MEETING.

Examples

This pattern was used in every course underlying this repository. Especially in more complex scenarios, e.g. in the ***Web Engineering*** LEARNING CONTRACTS scenario, the initial meeting was prepared very carefully to prepare the participants for upcoming learning activities. Still, some aspects of the courses remained open and confusing for some participants, which was unveiled by REACTION SHEETS at the end of the course. Also in Web Engineering, collection of participants' goals and expectations as well as their topics of interest for the INTERACTIVE LECTURE yielded some additional topics raised by participants to be presented in the lectures.

Evaluation

Not available.

PRELIMINARY PHASES

Package: *General*

Intent

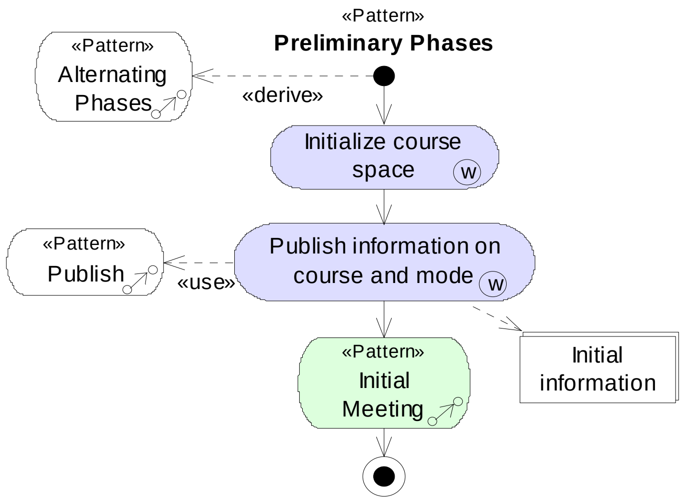
PUBLISH relevant content and resources as well as information on course style, activities, and objectives prior to an INITIAL MEETING where these issues are discussed.

Motivation

PRELIMINARY PHASES are the initial phases of any course. As such, this pattern provides a generic arrangement of activities for the initialization of COURSES: Potential participants are transparently provided with relevant content, resources, and information online as preparation for a subsequent INITIAL MEETING, where course style, objectives, and first tasks are presented and discussed.

Even though the pattern appears to be quite simple, the preparation of a course is definitely not. The initial phases and information in a course seem crucial for the further progress. Well structured information and consideration of students' goals and expectations (see for example ELABORATE GOALS AND EXPECTATIONS in INITIAL MEETING) are definitely central elements in PCeL courses. The only way of assisting the instructor in this effort is by providing Web support to publish relevant course resources and information and by providing guidance for INITIAL MEETINGS.

Sequence

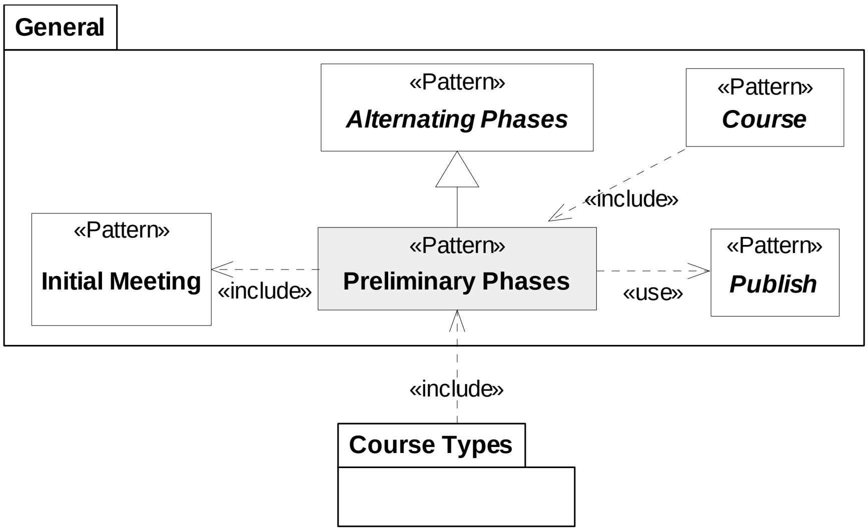


Activity	Description
Initialize course space	On the learning platform, the prior created course space is populated with information on groups, participants, instructors, and the course. Additionally, the course space is populated with relevant information (organizational issues, schedule, etc.), content, and resources.
Publish information on course and mode	General information regarding the course is PUBLISHED on the platform (if not already done in the previous activity).
INITIAL MEETING	

Structure

Not available: this pattern's structural elements are already embedded into the structure of COURSE.

Taxonomy/Dependencies



Parameters

- Primary pattern author: Michael Derntl
- Primary pattern source: Research Lab for Educational Technologies (University of Vienna)
- Pattern categories: Utility, Generic, Composite, Administrative
- Level of abstraction: Medium
- Scope: Phase
- Primary presence type: Blended
- Flexibility: High
- Level of confidence: 4
- Person-Centered variables addressed: Transparency
- Application effort: High
- Level of expertise required: High
- Suggested assistance: Tutor
- Input: Relevant course resources
- Output: Common understanding of course concept

Web Template

The primary Web-based activities in PRELIMINARY PHASES are:

- Initialization of the course platform, which may include import of participants and other general course data from some local course management system. Likely, most of the initialization steps will be custom to the institution where the solution is used.
- PUBLISHING of general information on the course, which is more appropriate for defining Web templates. Even though the concrete template used may vary significantly depending on the type of course (e.g., lecture vs. lab course), some common sections are identifiable from instances of PRELIMINARY PHASES in our recent teaching activities:

- *General information* and links to documents describing the course mode, schedule, and other useful information for participants.
- *Resources*, in the form of links to other Web pages or presentation slides or other content and links regarding specific topics addressed in the course (may also lead to further or additional information).
- *A link section* with hyperlinks to any additional information that may be relevant to the course participants.

Examples

This pattern is used with very few variations by any COURSE. The screenshot in Figure 100 gives an example of the course information that was posted in the ***Project Management/Basics and Techniques*** course in winter term 2004.

Figure 100: Project Management/Basics and Techniques course information homepage.

Evaluation

Not available.

Presentation Phases

Package: *General*

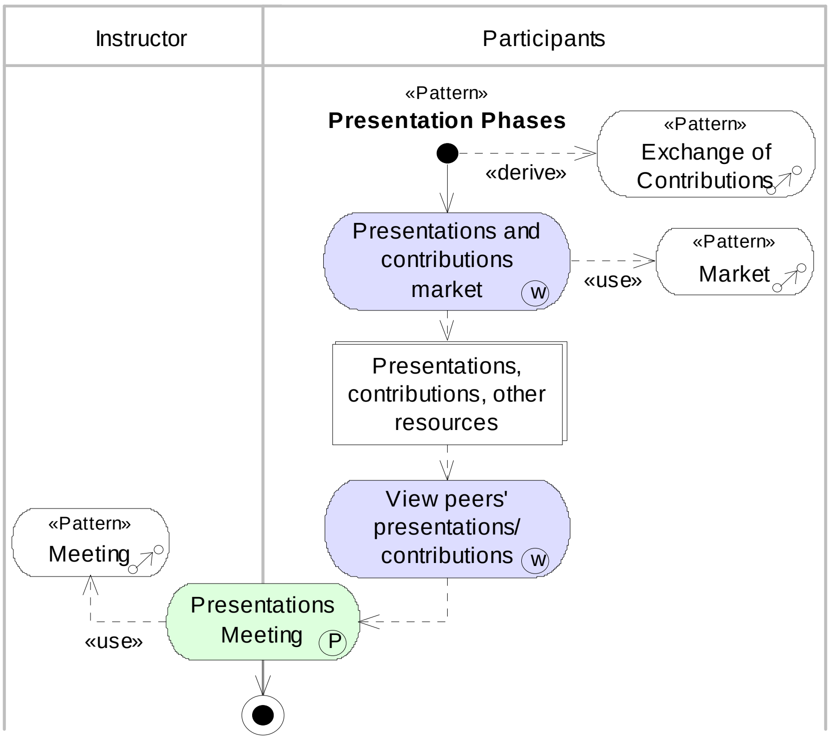
Intent

Let participants prepare themselves for presentation Meetings by Exchange of Contributions prior to the presentations. Prepared this way, the traditional long-presentation-and-short-discussion-scenario can be replaced by active discussions following short, concise presentations.

Motivation

The intent above conveys most of this pattern's essentials. Long presentations or lectures tend to be avoided in PCeL scenarios. The best way to avoid long presentations is to get the audience prepared before the presentations take place. This can be achieved by employing an Exchange of Contributions scenario specifically tailored to exchange of presentation slides, contributions, and resources in an online Market. Each participant is asked to upload his or her presentation slides until about one week before the presentation meeting(s). Additionally, participants are to prepare themselves for their peers' presentations by viewing/reading others' presentation resources *prior* to the presentation meeting(s). Such preparation has shown to produce an additional positive side effect: the combination of better preparation and shorter presentations leads to longer, more meaningful discussions, as everyone comes equipped with basic background knowledge in the subject, and short discussions do not allow going too deeply into the subject, which would quench evolving questions and discussion contributions.

Sequence

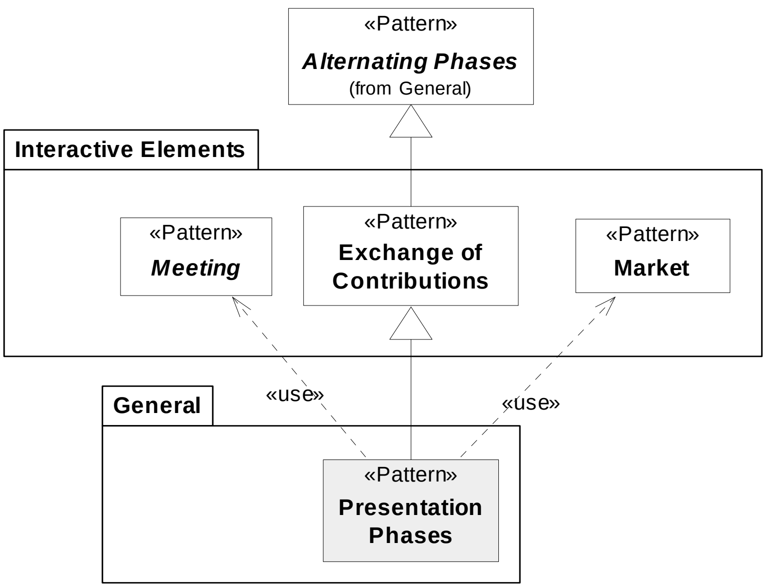


The flow of activities is almost identical to that of parent pattern EXCHANGE OF CONTRIBUTIONS, so no specific description is provided here.

Structure

Inherited.

Taxonomy/Dependencies



Parameters

- Primary pattern author: Michael Derntl
- Primary pattern source: Research Lab for Educational Technologies (University of Vienna)
- Pattern categories: Collateral, Motivational
- Level of abstraction: Low
- Scope: Phase
- Primary presence type: Blended
- Flexibility: High
  The exchange among peers can take on many different forms and is not restricted to MARKETS.
- Level of confidence: 3
- Number of participants: up to 20
  Participant restrictions are based on the number of presentations in the presentation meetings.
- Application effort: Medium
- Level of expertise: High
- Suggested assistance: Tutor
- Target skills: Interpersonal Skills, Communication, Collaboration
- Input: Guidelines for presentations, Meeting dates, Market space for presentations
- Output: Online presentation resources

Web Template

Inherited. Refer to parent pattern EXCHANGE OF CONTRIBUTIONS.

Examples

Presentations in most of the PhD Seminars of the primary author's department are commonly conducted according to this pattern. For example, in the ***Literature Seminar*** presentation phases, participants were asked to upload their contributions and presentation slides one week prior to the presentation meeting, so that participants could prepare themselves for their peers' presentations and subsequent discussions. Contrary to traditional seminar settings, there were short presentations of about 15 – 20 minutes followed by intensive discussions. Refer to the *Examples* section of the SEMINAR pattern for a more complete example context.

Evaluation

See the *Evaluation* section of the SEMINAR pattern, which shows written feedback regarding the PRESENTATION PHASES.

PUBLISH

Package: *General*

Intent

Disclose an information item (i.e., text, file, or completed form) to a certain target location, person, role, or group of roles and/or persons.

Motivation

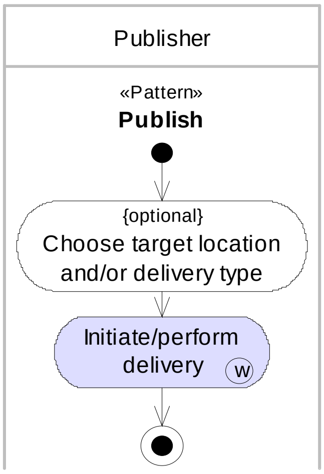
Publishing of information item (e.g., an electronic document, or just Web content) is one of the most basic tasks in deploying and using ICT in education. In the narrower sense, publishing may only be seen as disclosing a document or other information to anyone who has means of accessing the document. However, the PUBLISH pattern includes, but is not restricted to such a scenario. Within the scope of this pattern collection, publishing refers to any activity creating an information item or modifying the visibility of an information resource such that after the modification one of the following is achieved:

1) The same groups of roles or persons (i.e., the users) get an extended set of access rights to the resource;
2) The access rights remain at the same level of openness (or restriction) while more users get the right to access the resource.

Examples of PUBLISH include the disclosure of contributions in the scope of a LEARNING CONTRACT to make them accessible for PEER-EVALUATION purposes. Or, a participant sending some contribution for review to the instructor may also be seen as publishing. Finally, completing and submitting a Web form (e.g., in a QUESTIONNAIRE scenario) also lies in the scope of PUBLISH.

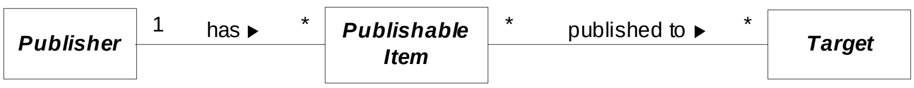
Publishing always serves the intent of disclosing information items, so the item to be published is given or created during the publishing process. Also, the *target* of publishing, i.e., the group of roles or person receiving extended access rights, is usually known to the *publisher* before publication. Subsequently, the way of publishing (*delivery*) is either given by restrictions imposed due to the nature of the item (e.g., submitting a Web form), or has to be explicitly selected by the publisher (e.g. sending a document via e-mail or changing the access rights of a folder). Thereby, means of publication are restricted by the nature of the item to be published. A piece of paper cannot be published by e-mail, but by pinning it to a blackboard, for instance.

Sequence



All activities in the sequence are executed by the *publisher*. The publisher holds the item to be published, defines the target and type of delivery (if required; this may include a decision about the publishing location), and finally commits the publishing process by initiating/performing the delivery.

Structure



Three abstract entities are involved in the publication process: The *publisher* has a number of *publishable items* to be published to a number of *targets*. However, PUBLISH does not answer the following questions:

- *Who is the publisher?* Many different roles may act as publishers, e.g., instructors, participants, teams, groups, administrators, or external guests.
- *What the publishable item?* For example, it may be any type of contribution supplied by participants or any kind of information provided electronically by the instructor. Technically, the location of the publishable item need not change during the publication process, as publication may also be initiated by changing access rights for an electronic document so that it becomes visible to others (i.e., to the *targets*).
- *Who are the targets?* Any resource with appropriate rights to view, access, or receive the published item may act as a target. In concrete scenarios, targets are mostly either a number of participants, teams, group, or the instructor.

Taxonomy/Dependencies

As PUBLISH is the most often used pattern, and due to its generic character, it is located in the *General* package. As almost every pattern uses it, the dependency diagram would have no informational value. Therefore, dependencies are not depicted here.

Parameters

- Primary pattern author: Michael Derntl
- Primary pattern source: Research Lab for Educational Technologies (University of Vienna)
- Pattern categories: Utility, Generic
- Level of abstraction: High
- Scope: Activity
- Flexibility: Low
- Level of confidence: 5
- Primary presence type: Online
- Application effort: Low
- Input: Item(s) to be published, Publication targets, Type of delivery/publication
- Output: Item(s) published

Web Template

Not available. As PUBLISH is a purely generic utility pattern, it does not define any Web template. Any other pattern using some form of the PUBLISH scenario must define this part of its Web template for itself.

Examples

There are numerous examples for concrete instances of the PUBLISH pattern:

- PEER-EVALUATIONS are published on the learning platform, SELF-EVALUATIONS are published to the instructor, and INSTRUCTOR-EVALUATIONS are published to participants.
- Elaborated documents are published in TEAM WORKSPACES.
- Teams of students make presentation slides visible to peers by placing them on the negotiated location on the learning platform, or just by giving them rights to access the slides.
- Instructor e-mails final grades to the participants. In this case the grade is the item, the instructor is the publisher, and the participants are the targets. Sending via e-mail is the type of delivery.
- ...

Evaluation

Not available.

STAFF MEETING

Package: *General*

Intent

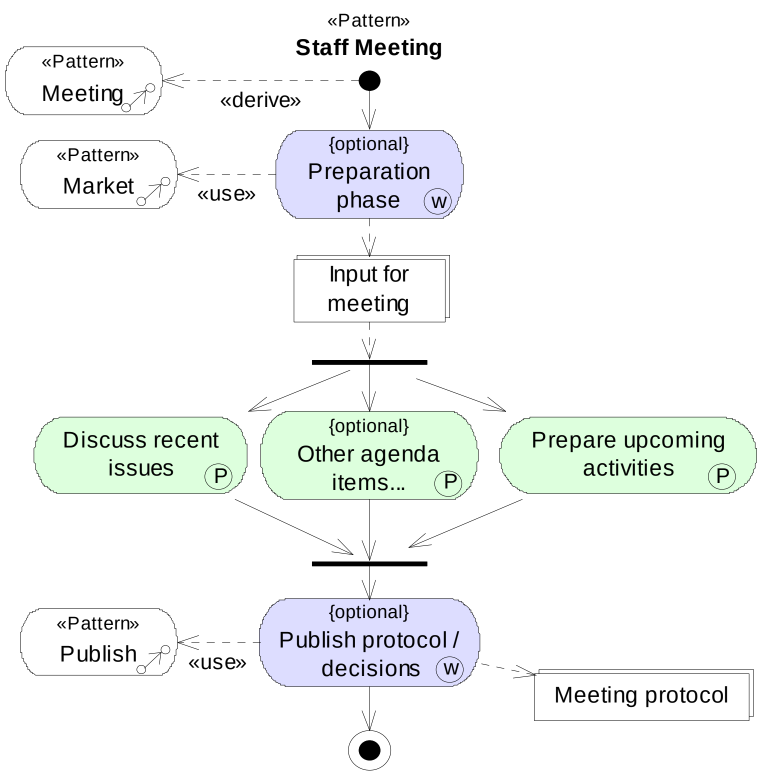
If more than one staff member is involved in the organization and/or execution of a COURSE or learning activity, staff members meet periodically to synchronize processes and discuss evolved issues and problems.

Motivation

It may be worthwhile to meet periodically with involved staff such as tutors, (external) instructors, guests, and administrators. Such meetings are especially useful when novel scenarios are employed to discuss recent issues and to prepare/synchronize for subsequent activities.

Note that, in preparation for the staff meeting, it may be useful to collaboratively collect agenda items and exchange ideas in an online MARKET.

Sequence



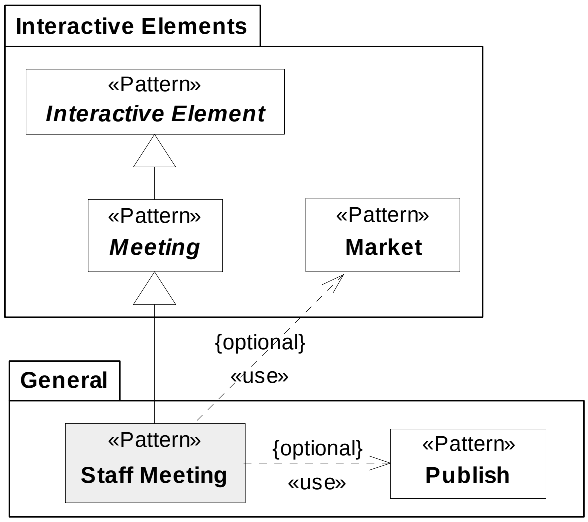
Activity	Description
Preparation phase	For preparation of the actual meeting, it may be useful to do prior exchange of information, issues of interest, and ideas online, e.g., in an

Activity	Description
	online MARKET.
Discuss recent issues, prepare upcoming activities, and optionally consider other agenda items in the actual MEETING.	
Publish protocol / decisions	Optionally, as after any meeting, it is certainly useful to PUBLISH a protocol containing any decisions that were made or important issues that were discussed.

Structure

Inherited.

Taxonomy/Dependencies



Parameters

- Primary pattern author: Michael Derntl
- Primary pattern source: Research Lab for Educational Technologies (University of Vienna)
- Pattern categories: Utility, Collateral, Administrative
- Level of abstraction: Low
- Scope: Activity
- Primary presence type: Present, Blended
- Flexibility: High
- Level of confidence: 5

Web Template

Optionally pass through to the MARKET Web template in the *preparation phase*.

Examples

In ***Web Engineering***, staff meetings were used intensively and have shown to be essential in the preparation of the course as well as throughout the course, because:

1) Many internal as well as external staff members (8 tutors, web master, assistant, and 4 instructors) were involved.
2) Some complex scenarios were used for the first time. Especially in the employed LEARNING CONTRACT scenario the participants frequently raised organizational and administrative questions which had to be handled carefully and consistently.

Evaluation

Not available.

TEAM WORKSPACES

Package: *General*

Intent

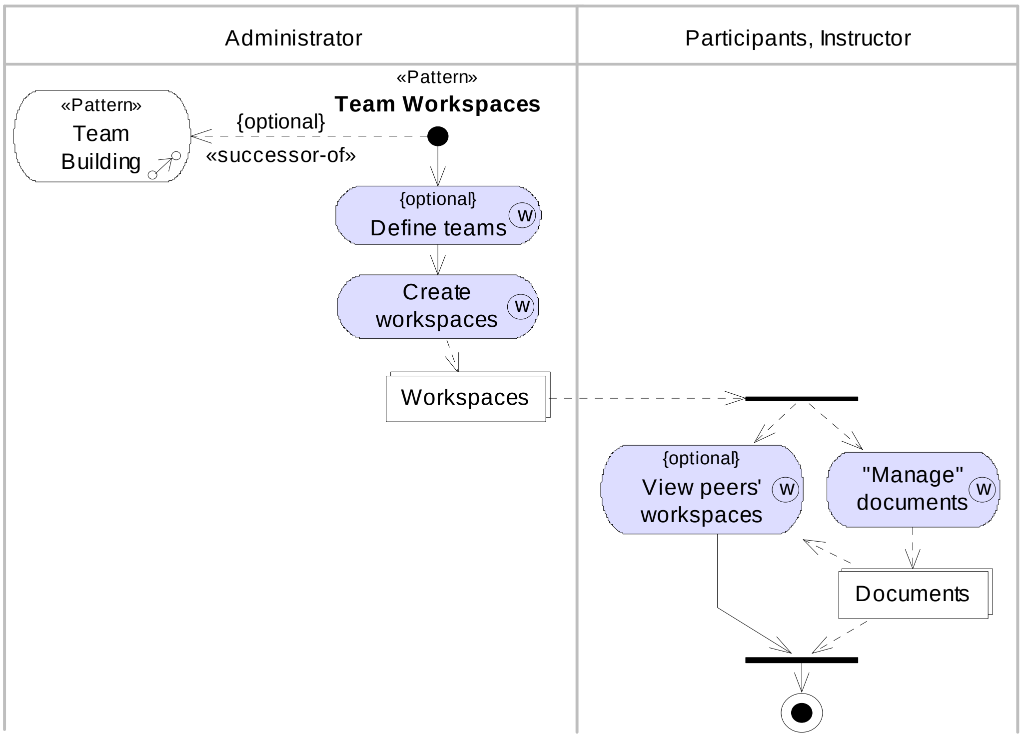
Provide teams with private workspaces, which they can use to create, store, work on, and share their contributions and other documents.

Motivation

Collaboration is one central element in blended, Person-Centered learning. To allow for online collaboration within and among teams, each team should get a dedicated, separate space on the learning platform which they can use to manage their documents. Document management includes creation, deletion, modification, and PUBLISHING of documents and folders, as well as upload and download.

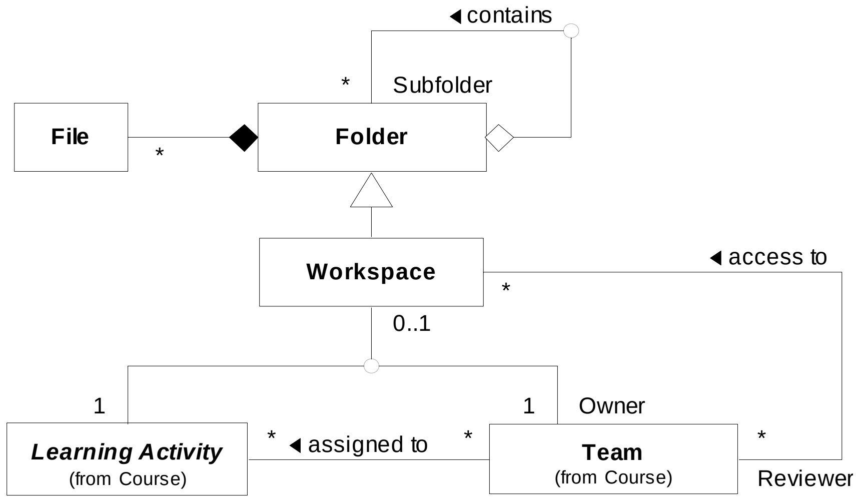
The scenario of this pattern depends on TEAM BUILDING, which has to be executed prior to creating workspaces: To create team workspaces, teams have to be defined for the respective learning activity. Finally, team workspaces may be created for the desired learning activity. This pattern is substantially supported by Web templates, as all interaction takes place online.

Sequence



Activity	Description
Define teams	First of all, teams have to be specified by grouping the participants in teams of the desired size. The "optional" constraint says that if teams have already been defined in the scope of another scenario (TEAM BUILDING), they do not have to be defined at this point.
Create workspaces	Team workspaces are created by selecting the respective learning activity. As teams have already been assigned to that learning activity, the required information is available to create workspaces.
"Manage" documents	After the workspaces were created, teams (and the instructor) can start using them for "managing" their documents, whereby document management includes uploading, downloading, deleting, and moving documents as well as organizing documents in a folder structure.
View peers' workspaces	If configured accordingly, the instructor and participants may view their other team workspaces, and download documents.

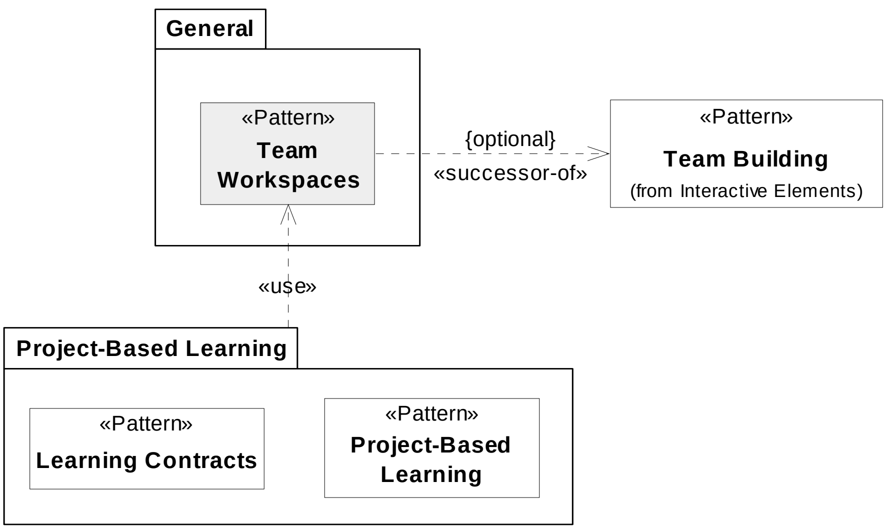
Structure



Each team *workspace* belongs to one *team*, which is the *owner* of the workspace. Additional teams may be assigned to workspaces as *reviewers* with read-only access, e.g., to enable PEER-EVALUATION of teams' contributions. Team workspaces are usually linked to *learning activities*, as workspaces without a certain document context are mostly useless, except: Workspaces assigned to courses (which are also learning activities[531]) are a special kind of workspaces that may be used by teams to manage documents not explicitly produced within the scope of a specific learning activity.

Conceptually, a workspace is nothing more than a document *folder* (or better the *root* folder) that is capable of hosting additional folders (*subfolders*) and/or *files*.

Taxonomy/Dependencies



[531] See the *Structure* section of COURSE

Parameters

- Primary pattern author: Michael Derntl
- Primary pattern source: Research Lab for Educational Technologies (University of Vienna)
- Level of abstraction: Low
- Pattern categories: Utility, Administrative
- Scope: Activity
- Primary presence type: Online
- Number of participants: Unrestricted
- Application effort: Low
- Level of expertise required: Low
- Input: Teams, Learning activity
- Output: Team workspaces

Web Template

Administration View

The administration view of TEAM WORKSPACES offers the following options:

1) *General information* (Figure 101): Includes the heading that is displayed in the workspace page (participant view; Figure 106) as well as an optional restriction on the usage period, during which workspace owners are allowed to change the contents of their assigned workspaces (i.e., uploading and deleting documents, creating and deleting folders).

2) *Initial folder structure* (Figure 102): It might be useful for instructors to define an initial folder structure that offers participants some kind of guidance for uploading their documents onto the desired folders. This would be useful for example when defining separate folders for each PROJECT MILESTONE in a PROJECT-BASED LEARNING setting, or for defining one dedicated folder for final reports in a SEMINAR, which could subsequently be subject to PEER-EVALUATION or SELF-EVALUATION (see wizard step 4). Defining the initial folder structure includes specifying a name and description for the *root* or home folder of each team. The following table allows for creating (and subsequent deleting) additional predefined folders, along with their *parent* folder, folder *name*, and folder *description*. The example given in the screen shows that the root folder will contain two subfolders (*Folder 1* and *Folder 2*), and *Folder 2* will additionally contain subfolder *Folder 3*. Note that the folder description will be visible to participants in the *Folder description* section of the workspace page (participant view; Figure 106).

3) *Additional options* (Figure 103): On this page, administrators can specify (a) whether participants are allowed to view into the workspaces of their peers, (b) whether workspace owners are allowed to create folders within the predefined initial folder structure, (c) the maximum allowed upload file size, and (d) the maximum allowed total work-

space size. Any file upload that exceeded any of the restrictions set in (c) or (d) would be rejected.

4) *Evaluation settings* (Figure 104): As already mentioned above, supplying direct hyperlinks to evaluation forms within dedicated folders (e.g., a folder for final contributions) supports central use cases in the evaluation phase of a learning activity: For example, participants who have to supply peer-evaluations for particular documents navigate into the workspace of a selected peer team (to the dedicated document folder), download the documents for review from that folder, and supply the peer-evaluation by clicking the hyperlink on that same workspace page. The settings offered in this final wizard step allow for configuring links to Self- and Peer-Evaluation forms. Thereby, the administrator has to select the dedicated evaluation folder, a link text that is displayed in that folder, an evaluation form that must have been previously designed in the course of an EVALUATION scenario, as well as an optional restriction of the time period in which the links to the evaluation forms are displayed.

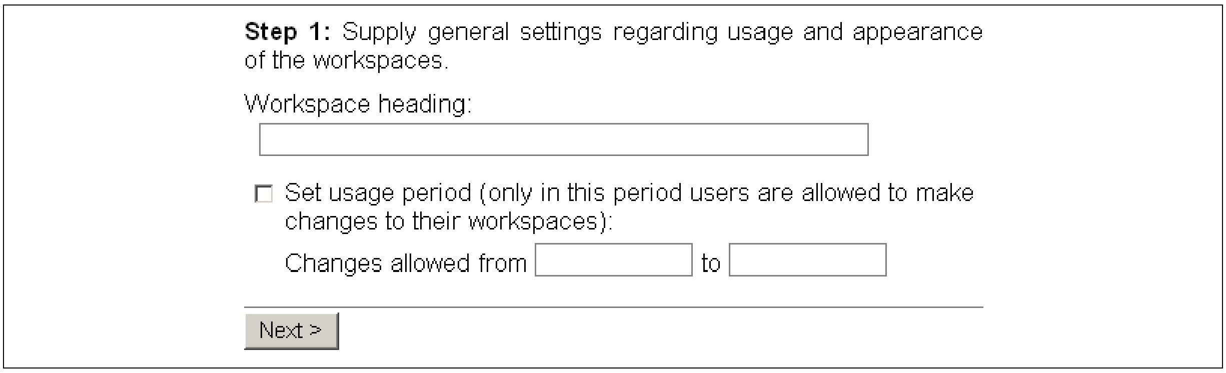


Figure 101: Wizard step 1 in Team Workspaces administration view.

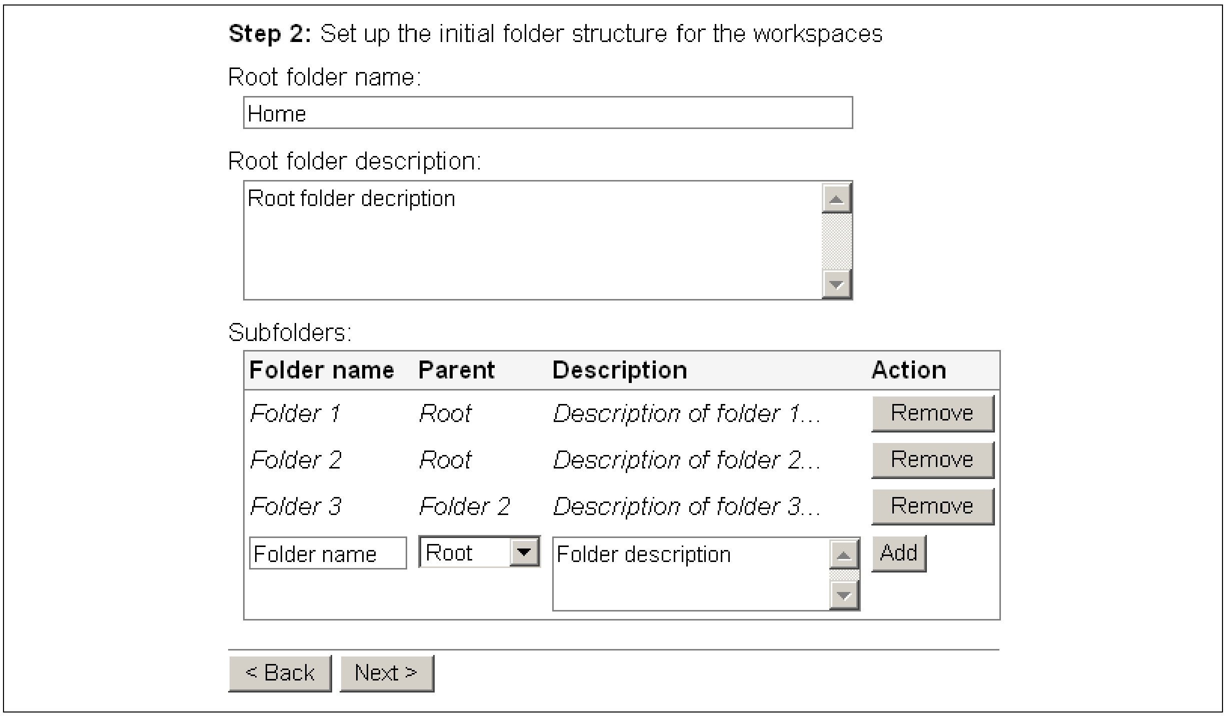


Figure 102: Wizard step 2 in Team Workspaces administration view.

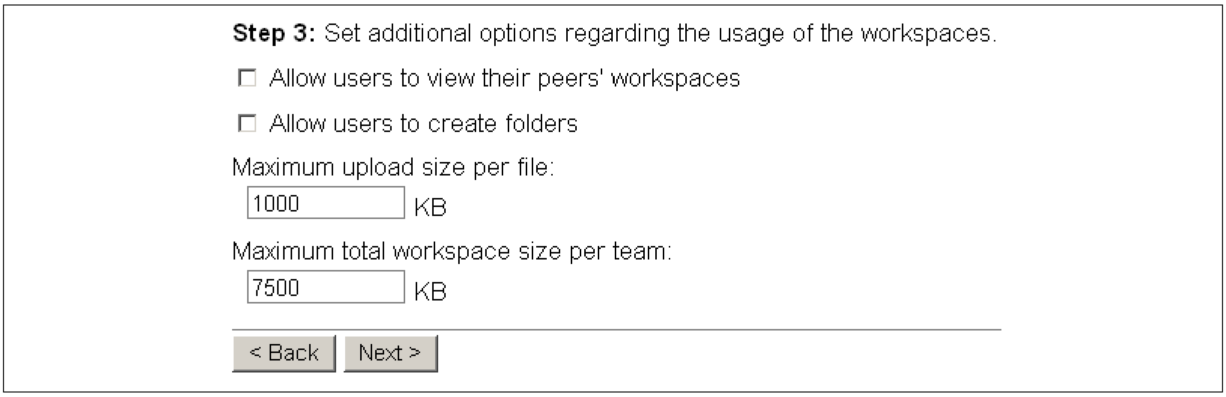


Figure 103: Wizard step 3 in Team Workspaces administration view.

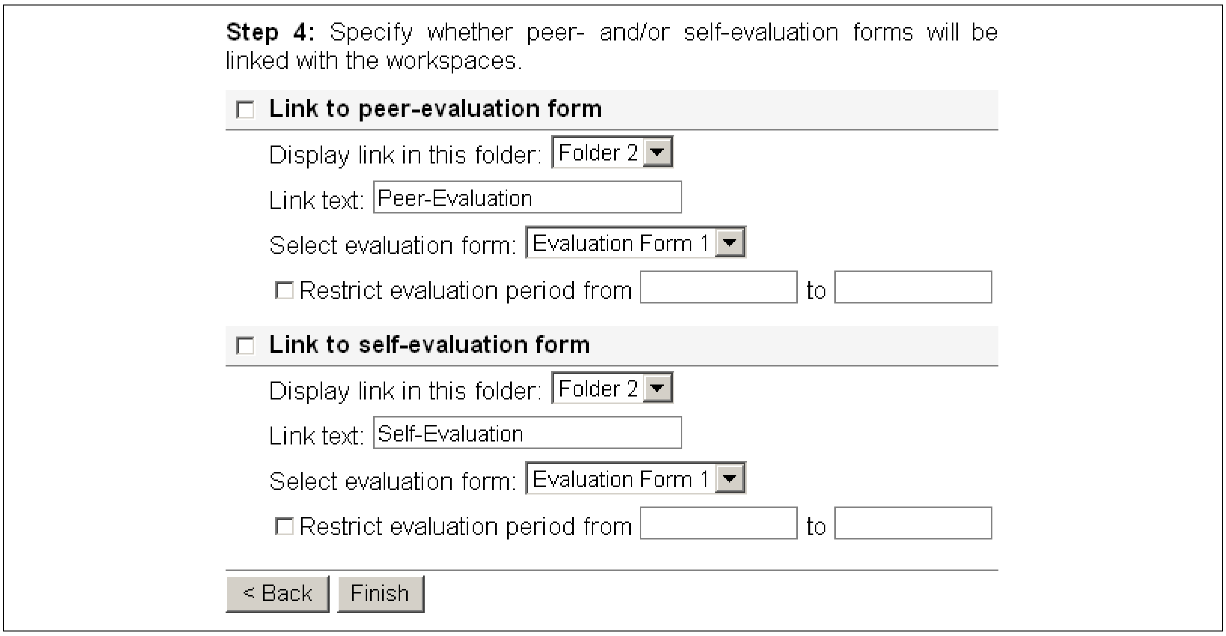


Figure 104: Wizard step 4 in Team Workspaces administration view.

Participant View

The participant view consists of just two pages, of which one is not necessarily accessible to participants:

The overview page (Figure 105) is only visible to participants, when the option for allowing participants to enter their peers' workspaces was activated in the administration view (see Figure 102), or if the team workspaces scenario is currently in a peer-evaluation period (see administration view, Figure 104). Instructors, administrators and reviewers are necessarily always allowed to access multiple workspaces, so the overview page is always the entry point to the workspaces to them (as long as there exist multiple workspaces). Note that the *Own workspace* hyperlink is only visible to workspace owners (i.e., team members). Depending on the structure of the course, links to the workspaces may be grouped by course groups and/or the teams' instructors. Additionally, after a possible PEER-EVALUATION phase is over, the team workspace hyperlinks in the overview page can each be amended with a hyperlink to the peer-evaluation results (if these are configured to be visible to owners/peers).

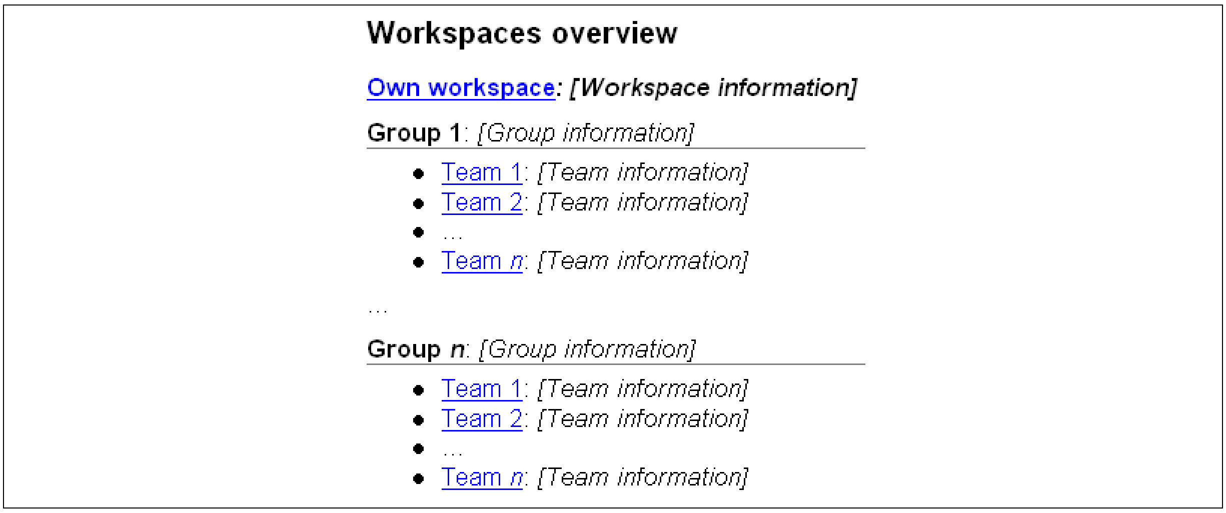


Figure 105: Workspaces overview page in Team Workspaces participant view.

Following one of the hyperlinks to the actual workspace, participants are presented with workspace the page depicted in Figure 106. It shows the configured *heading* (see administration view, Figure 101), the current folder description (administration, Figure 102), a structural navigation bar showing the parent folder hierarchy as hyperlinked folder names, as well as the *current folder name*. Below these informational elements, the interactive part of the workspace page includes controls for *creating a folder* within the current folder (if allowed) and for *uploading files* onto the current folder (optionally by *browsing* the local file system). Below these controls, *hyperlinks to peer- or self-evaluation* forms are displayed as configured in the administration view (Figure 104). The actual folder/file table allows for clicking on a *subfolder* hyperlink, resulting in the workspace page to be reloaded to display that subfolder's contents, and clicking on a *file* hyperlink, resulting in a download process of that file via HTTP. The title bar of the table contains hyperlinks for navigating *up* in the folder tree, as well as for *refreshing* the current folder view (other participants might have uploaded documents onto the current folder during the current participant's idle time). Below the file/folder table, there are controls for *deleting* checked files/folders (note the checkboxes in the first table column), as well as clipboard controls allowing for *cutting* or *copying* checked items, and for *pasting* items from the clipboard into the current folder.

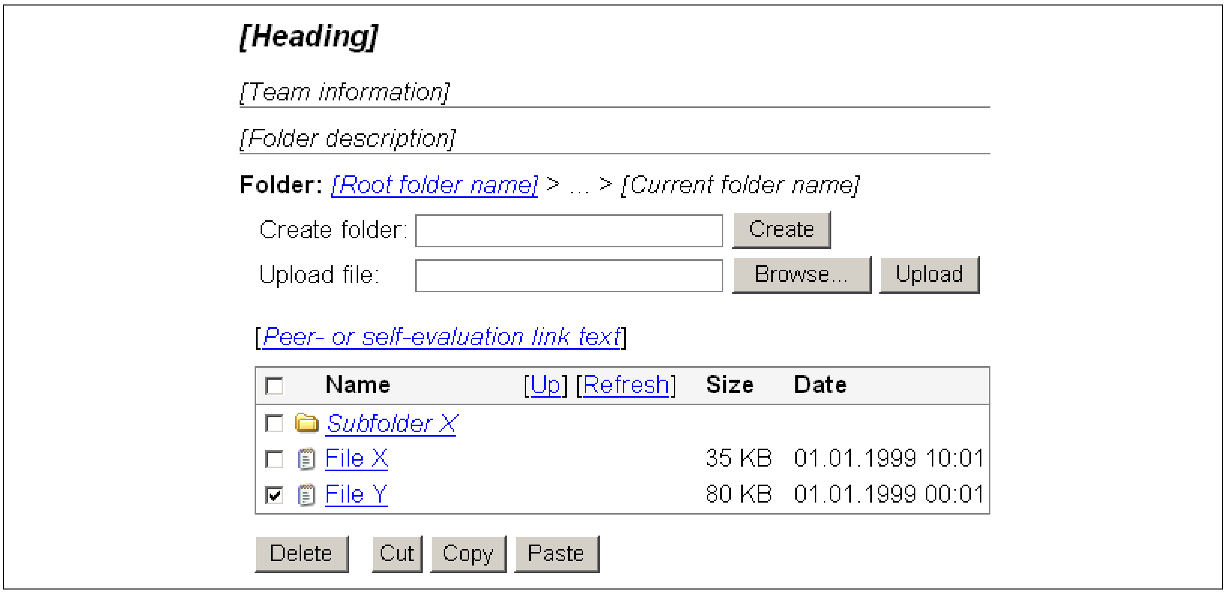


Figure 106: Workspace page in Team Workspaces participant view.

Report View

This view is not available because instructors, reviewers, and administrators can navigate through workspaces in which they are interested.

Examples

As workspaces for teams can be used in many learning activities where documents are produced, there are numerous application examples. A typical scenario is using TEAM WORKSPACES for LEARNING CONTRACTS: Any contribution relevant to a contract is uploaded to the respective team's workspace, and finally PUBLISHED to the instructor and peers for review and EVALUATION. Such an application scenario was employed in ***Web Engineering*** (2004): a screenshot from a sample team workspace is given in Figure 107.

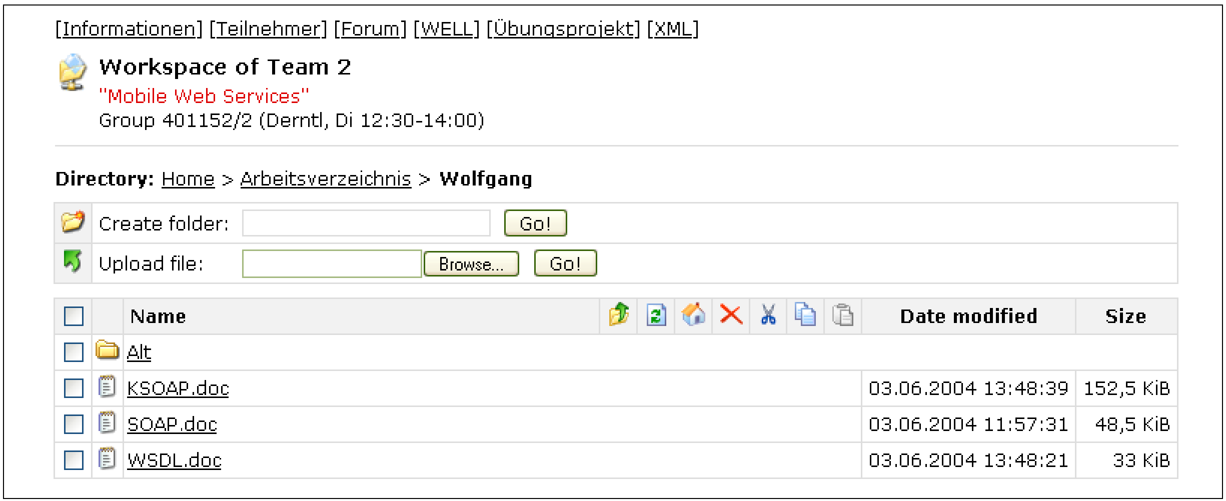


Figure 107: Example Learning Contract team workspace from Web Engineering (2004).
The figure shows the workspace of a team elaborating the topic "Mobile Web Services". There are facilities to create folders and to upload files. The lower part of the figure shows the contents of the current folder, which is one subfolder and three Word documents. The contents table also shows icons to be used for navigation and modification of files/folders as well as for the well-known clipboard functionality (i.e., *cut*, *copy*, and *paste*).

Evaluation

As Figure 108 shows, tools support for team workspaces and resources in general was not perceived as high as one would expect in ***Web Engineering 2003***. The assumed reason for that is that one of the two platforms that were employed was ranked very low in general tool support because it was still in a phase of development and handling was rather cumbersome and unintuitive.

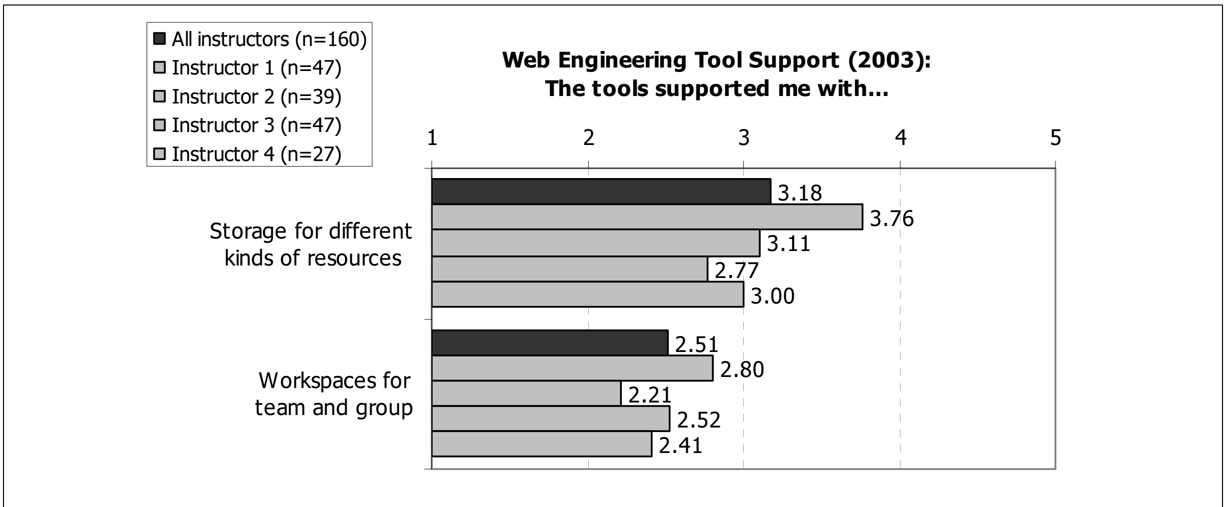


Figure 108: Tool support for resource storage and workspaces in Web Engineering (2003).
The results were obtained from the questionnaire that was distributed at the end of the course and shows mean values supplied by 160 participants for *all instructors* and instructors 1–4 separately. Scale: 1 = *not at all* ... 5 = *very much*

In the following year (***Web Engineering 2004***), the team workspace facilities were improved significantly in terms of usability and their use was an integral part of the LEARNING CONTRACT scenario. As expected, their perceived usefulness was much higher than the year before (see Figure 109).

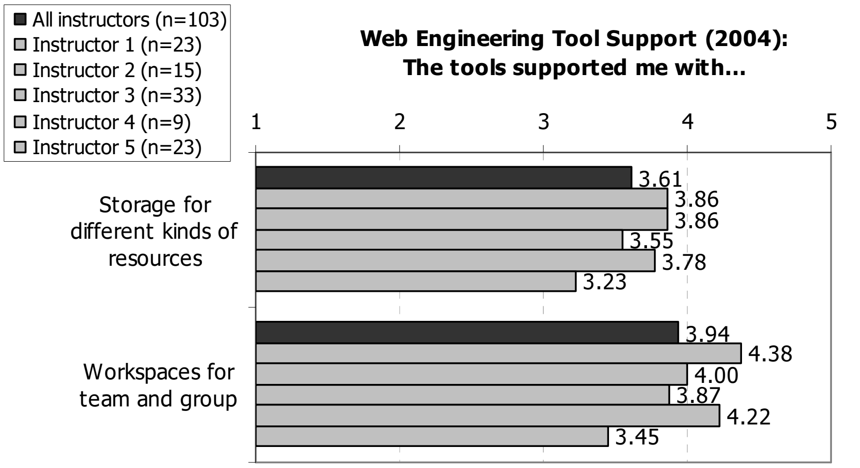


Figure 109: Tool support for resource storage and workspaces in Web Engineering (2004).

Procedure and scale are the same as in Figure 108 above. The mean value for "storage of different kinds of resources" was raised considerably by .43, while the mean value for "workspaces for team and group" was raised massively by 1.43 due to the improved version of the online team workspace facility.

5.6 Interactive Elements

APPROVAL

Package: *Interactive Elements*

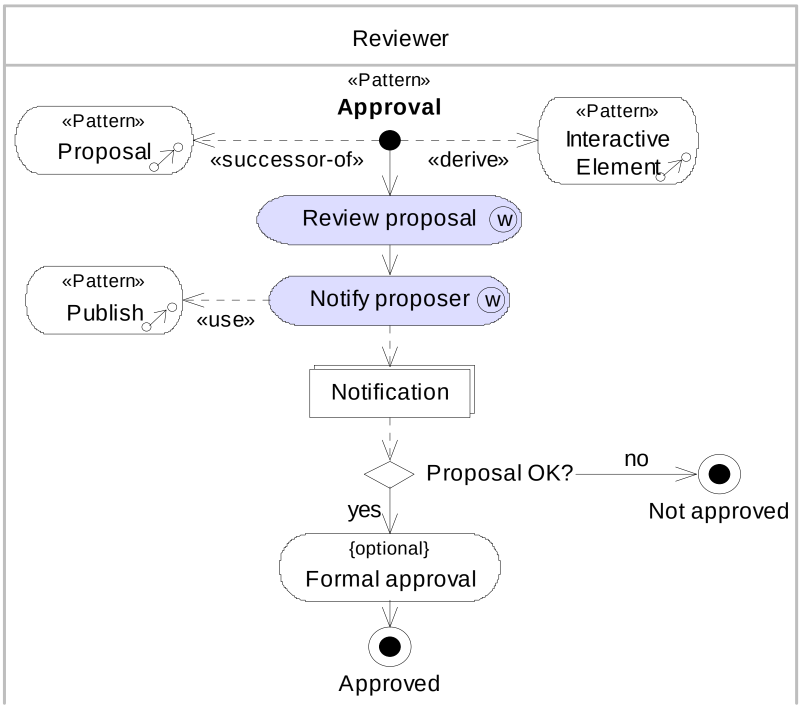
Intent

The instructor reviews and approves PROPOSALS according to guidelines PUBLISHED in a proposal request.

Motivation

APPROVAL is required within *any* PROPOSAL scenario, as proposals without a subsequent approval stage are senseless. The reason why this scenario is extracted from the PROPOSAL pattern is simply to allow for detaching the approval process from the employed concrete form of the proposal process. Additionally, PROPOSAL is certainly subject to more variations than APPROVAL, so separating these two should contribute to an increase of extensibility and adaptability of the pattern repository.

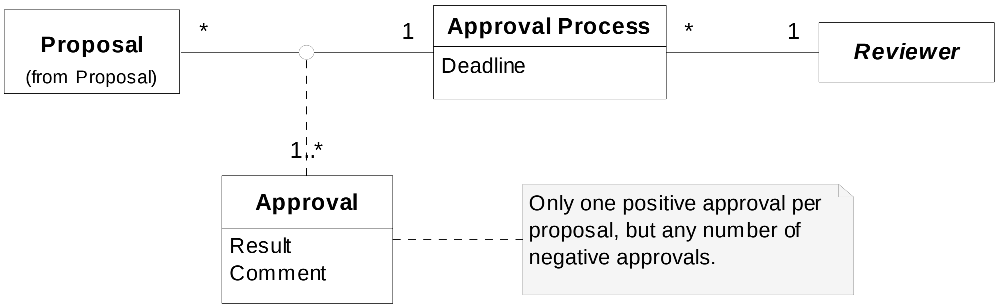
Sequence



Activity	Description
Review proposal	Initially, the published proposal is reviewed by the reviewer. This is usually the instructor using the report view of the pattern that is producing the proposals, except for approval scenarios between participants. The reviewer comes to a decision regarding the approval of the proposal. Note that it is important that the reviewer defines an acceptable time span in which the proposals will be processed, so that the projected deadline of the notification can be PUBLISHED to the proposer.
Notify proposer	The decision is published to the proposer, possibly along with further instructions regarding the revision of the proposal (see PROPOSAL for details)
End: Not approved	If the proposal is inappropriate, it is rejected. The proposal is *not approved* and the sequence ends.
Formal approval (*End*: Approved)	If the proposal is *approved*, the reviewer may issue a formal approval which may act as an agreement between the reviewer and the proposer, such as in LEARNING CONTRACTS, where an approved proposal is

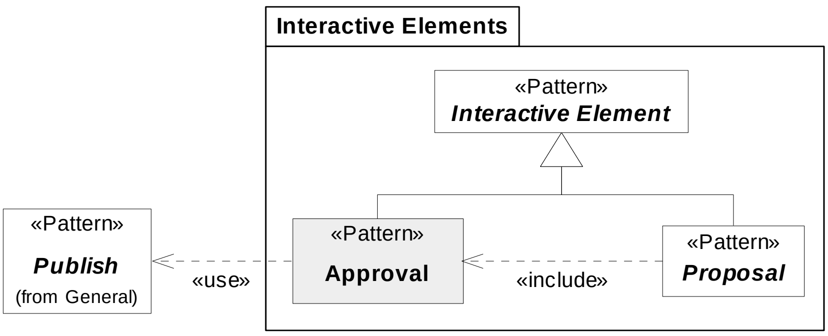
Activity	Description
	formally recorded as a contract, which is signed by all involved.

Structure



The structure of this pattern is connected to the Proposal pattern through class *proposal*. For each proposal there is an *approval process* which should be executed within a time span delimited by the approval *deadline*. In turn, in a single approval process multiple proposals may be approved by a *reviewer*. For each pair of proposal and approval process there exist a number of *approvals*, each with a *result* (positive or negative) and optionally a *comment* to the proposer. However, only one positive approval makes sense for one concrete proposal.

Taxonomy/Dependencies



As there is asynchronous interaction among proposer and reviewer, Approval is located in the *Interactive Elements* package, as a sub-pattern of Interactive Element. A bidirectional dependency exists with Proposal, as Approval is always included in the sequence of Proposal, and Approval is in turn useless without the existence of a proposal. Some form of Publish is used to notify the proposer of the outcome.

Parameters

- Primary pattern author: Michael Derntl
- Primary pattern source: Research Lab for Educational Technologies (University of Vienna)
- Pattern categories: Utility, Generic, Administrative
- Level of abstraction: Medium
- Scope: Activity, Phase

- Primary presence type: Online, Present
- Flexibility: Low
- Level of confidence: 5
- Application effort: High
- Level of expertise required: High
- Input: Approval criteria, Proposals
- Output: Positive or negative approval

Web Template

As APPROVAL is always part of a PROPOSAL scenario, its Web templates are defined in concrete sub-patterns of PROPOSAL.

Examples

Not available. Due to the tight coupling with the respective PROPOSAL pattern, any application example given there also applies to this pattern.

Evaluation

Inapplicable.

BRAINSTORMING

Package: *Interactive Elements*

Intent

COLLECT and subsequently PUBLISH ideas gathered in brainstorming sessions, either online or present.

Motivation

Brainstorming is a well-proven method of collecting ideas and opinions from different points of view in a group[532]. There are several available, good definitions for brainstorming. Some of them are listed below, with key characteristics being italicized:

- Brainstorming is a *group process* for generating ideas using four *divergent thinking*[533] guidelines: *Deferring judgment* (raised ideas are not judged during the process), striving

[532] Accoding to Steege (1999), the word 'brainstorming' was coined by Alex Osborn, who developed this technique in the 1930's when he was president of an advertising firm.

[533] Cotton (1995, p. 152) describes divergent thinking as "*the ability to spread your thoughts about a central theme rather than follow a straight line of thoughts.*"

for *quantity*, freewheeling (*encouraging ideas*, even if they seem improper), and seeking *combinations*.[534]

- Brainstorming is a process for developing *creative solutions* to problems by focusing on a problem, and then deliberately coming up with as many solutions as possible and by *pushing* the ideas as far as possible.[535]
- Brainstorming is the *unstructured generation* of ideas by a group of people. Solutions are sought to particular problems and every idea is acceptable.[536]
- Brainstorming is where a group of people put *social inhibitions and rules aside* with the aim of generating *new* ideas and solutions, with *free association* of different ideas to form new ideas and concepts.[537]

Brainstorming sessions are usually rather short, lasting about 10-15 minutes. After a short break, they may be continued. It is crucial that someone writes down the gathered ideas without valuing or interpreting them. This is deferred to the evaluation phase of the session, which need not necessarily take place immediately after the gathering session.

In addition to the option of having one person write down all the ideas, there are some variations that are frequently employed: *Post-it* and *brainwriting*[538]. In the post-it variation, every participant writes down his or her own ideas legibly on post-it cards. The cards may either be immediately collected and pinned up by the host, or this may be deferred until the participants have written down all their ideas. After this initial gathering phase, the clusters may be built with the ideas on the cards. In the brainwriting variation there is no verbal interaction initially. Ideas are written down on paper and handed over to other participants, who may refine, modify, or add to these ideas.

Obviously, there are numerous aspects in these scenarios which can be perfectly supported online. For example:

- Evaluation and clustering of ideas may be done in an ONLINE DISCUSSION forum that is attached to the initial brainstorming results.
- Modifying others' ideas as in the brainwriting variation can also be supported online by allowing the participants to post/attach modifications to other ideas.
- Subsequent collection of additional ideas can take place online by providing an upload space attached to the initial brainstorming results where notes can be uploaded.
- The brainstorming session may be carried out in the form of a moderated CHAT, where the moderator acts as the host by formulating the initial problems and collecting ideas

---

[534] Steege (1999)

[535] Clark (2000)

[536] Wideman (2002)

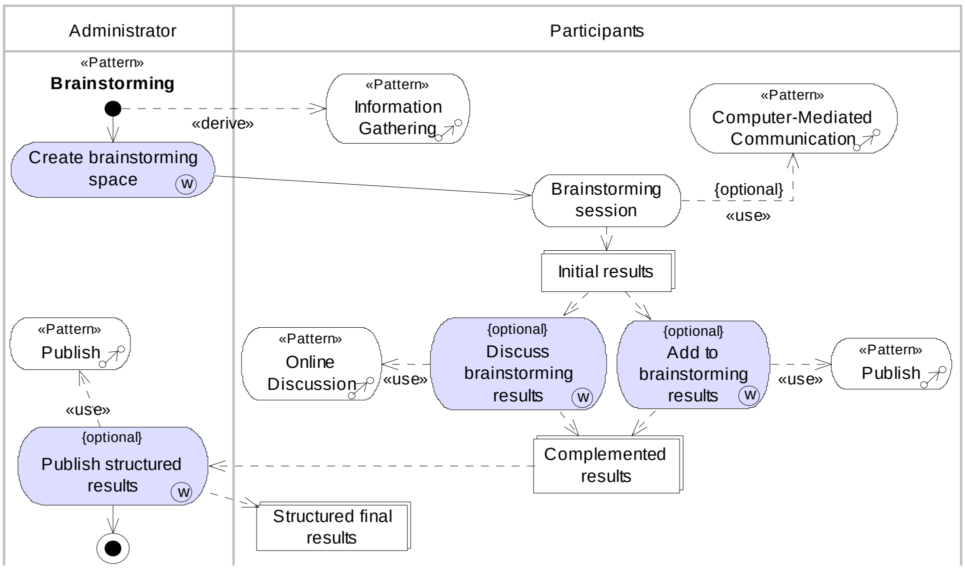
[537] Infinite Innovations Ltd. (2001)

[538] Steege (1999)

from participants as chat messages. Note that in this case the chat facility has to provide an option to record the chat log for later look-up and extracting of ideas.

Regarding usage in educational scenarios, brainstorming may be used, e.g., to collect ideas on a theory or problem before the theory or solution is presented by the instructor, which may arouse interest in the topic *before* factual information is transmitted. Additionally, brainstorming can even be used to describe aims of the learning process itself, for example by answering questions such as, "*What makes up a well-performed presentation?*" or, "*Which features should an e-learning platform possess?*"

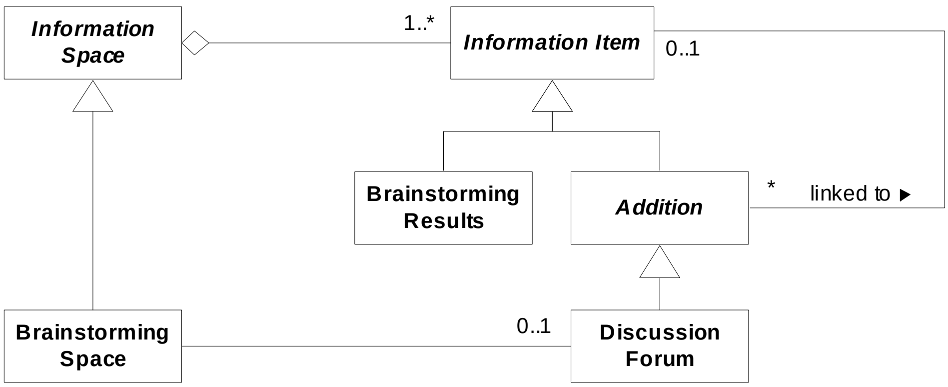
Sequence



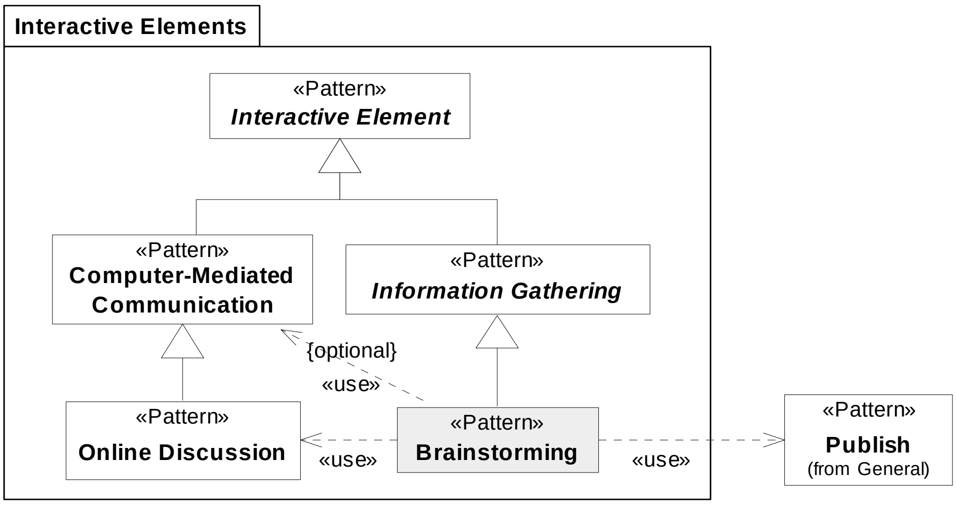
Activity	Description
Create brainstorming space	The type of platform space allocated for the brainstorming session and its results is restricted by its intended use. If the brainstorming session is conducted face-to-face, a space has to be created that is capable of uploading and holding brainstorming results. This can even be a simple page. If the brainstorming session is otherwise conducted as a moderated CHAT, the initialization of the chat facility is done in the sequence of CHAT.
Brainstorming session	The brainstorming session itself is subject to the variation described in the *Motivation* section above and has to be chosen and conducted accordingly. As already mentioned, the brainstorming session may optionally be conducted as a moderated CHAT. However, after the brainstorming sessions, the resulting ideas have to be published in the allocated platform space.
Discuss brainstorming results	To allow for subsequent exchange of opinions and ideas, an ONLINE DISCUSSION forum may be initiated and attached to the brainstorm-

Activity	Description
	ing results.
Add to brainstorming results	To enable participants to make additions to the initial brainstorming results, they have to be supplied with a facility to upload additional notes and ideas.
Publish structured results	The results are structured and evaluated in terms of how often each item has been mentioned

Structure



Taxonomy/Dependencies



Parameters

- Primary pattern author: Michael Derntl
- Primary pattern source: Research Lab for Educational Technologies (University of Vienna)
- Pattern categories: Utility, Motivational
- Level of abstraction: Low

- Scope: Activity
- Primary presence type: Present
- Flexibility: High
- Level of confidence: 4
- Number of participants: Unrestricted
- Application effort: Low
- Level of expertise required: Low
- Target skills: Collaboration, Communication, Problem solving
- Input: Brainstorming topic, Guidelines and rules, Brainstorming space in online scenario
- Output: Brainstorming results and additions

Web Template

Inherited: This pattern reuses the Web template from parent pattern INFORMATION GATHERING.

Examples

In the first lecture of the ***Web Engineering*** modules, different topics of interest to the participants as well as goals and expectations were collected in a plenary brainstorming session: while participants raised their comments and ideas, one assistant wrote them down in real-time on a laptop computer, whose display was projected on the front wall of the lecture room. The collected ideas were then posted as an electronic document on the learning platform, and each participant was asked to provide additional comments and complements.

Evaluation

Not available.

References

Clark, D. (2000). *Brainstorming*. Retrieved Dec 14, 2003, from http://www.nwlink.com/~donclark/perform/brainstorm.html

Cotton, J. (1995). *The Theory of Learning*. London: Kogan Page.

Infinite Innovations Ltd. (2001). *Definitions of Brainstorming* [Online]. Retrieved Dec 14, 2003, from http://www.brainstorming.co.uk/tutorials/definitions.html#brainstorming

Steege, S. (1999). *What is Brainstorming?* Retrieved Dec 14, 2003, from http://www.buffalostate.edu/centers/creativity/Resources/Reading_Room/Steege-99.html

Wideman, M. R. (2002). *Project Brainstorming - What is brainstorming?* Retrieved Dec 14, 2003, from http://www.maxwideman.com/issacons4/iac1446/tsld002.htm

CHAT

Package: *Interactive Elements*

Intent

Provide facilities for synchronous communication among participants, instructors, tutors, and/or guests.

Motivation

Chat is one possible form of COMPUTER-MEDIATED COMMUNICATION (CMC) and is used for synchronous online interaction. Chat communication is normally anchored to a certain learning activity (e.g., problem-solving activities such as PROJECT-BASED LEARNING) or other special purpose (e.g., CONSULTATION), as chat facilities are rarely used without any need/desire for communication between participants. There are several dimensions that are worth considering when employing a CHAT scenario[539]:

- The *task* that is set for learners. It is important to accurately specify what is expected from participants in their online communication. For example, it is not sufficient to say “discuss this topic”. Instead, desired outcome and *rules* for discussions have to be specified.
- Synchronous CMC usually works best with only a *small* number of participants involved, unless the communication is moderated.
- In many cases it may be reasonable or required to *moderate* the chat to keep the discussion on track or to lead it in desirable directions.

A special form of moderation would be a CHAT with an expert, who may be invited to share his know-how with participants and to answer their questions.

Chat communications also entail some drawbacks, as it does not always lead to topic-centric[540] or very reflective conversations, which is due to the fact that synchronous CMC favors fast thinking and typing over careful thought[541]. However, when appropriately coached (by moderators) and embedded into learning activities, CHAT is certainly capable of adding value.

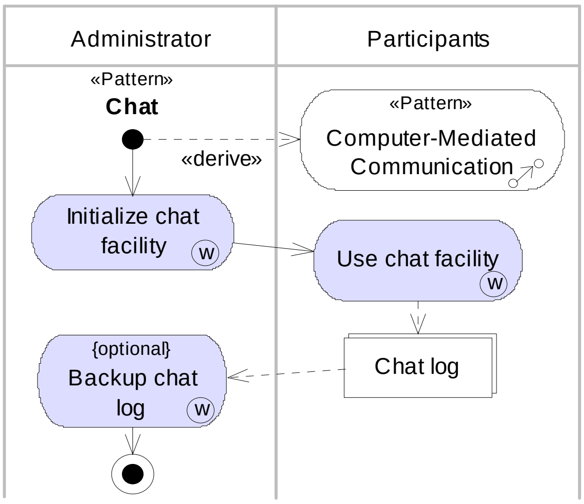
---

[539] Cf. Ingram, Hathorn and Evans (2000, p. 30-32)

[540] Orvis et al. (2002) revealed in a recent, major study that only about half of the conversation in synchronous CMC is task-related, even in communication-intensive problem-solving scenarios. The remaining share is distributed among social and technology-related exchange.

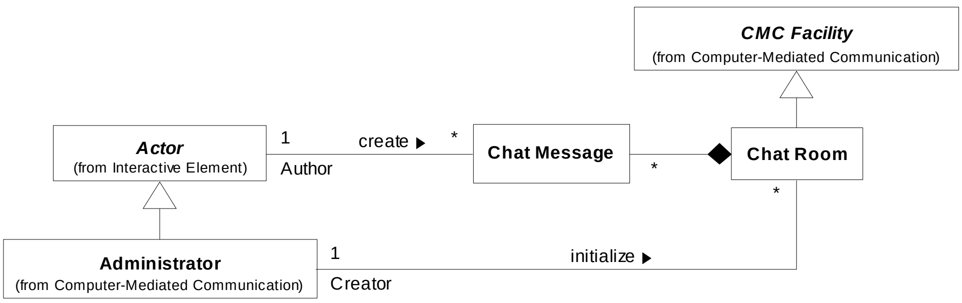
[541] Ingram, Hathorn and Evans (2000, p. 33)

Sequence

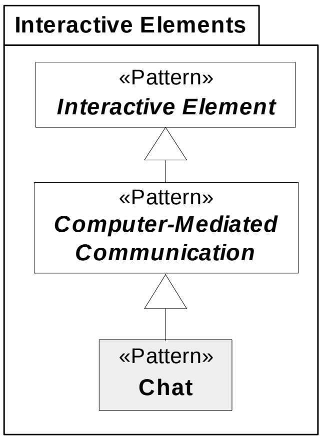


Activity	Description
Initialize chat facility	The chat facility is initialized, possibly along goals, guidelines, and rules for communication.
Post chat message	Participants may log into the chat channel and post their chat messages. Participant is a generic role name and can also refer to an administrator, moderator, or expert.
Backup chat log	If desired, the chat log may be backed up as a resource for later lookup by participants.

Structure



Taxonomy/Dependencies



Parameters

- Primary pattern author: Michael Derntl
- Primary pattern source: Research Lab for Educational Technologies (University of Vienna)
- Pattern categories: Utility, Collateral
- Level of abstraction: Low
- Scope: Activity
- Primary presence type: Online
- Flexibility: High
- Level of confidence: 2
- Application effort: Low (open chat) or Medium (moderated chat)
- Level of expertise required: Low (open chat), High (moderated chat)
- Suggested assistance: Administrator, Expert (chat with expert)
- Target skills: Communication, Collaboration
- Input: Chat facility, Guidelines and rules
- Output: Chat log

Web Template

Not available. Chat facilities are complex applications. There is a wealth of free chat software and ready-to-use websites available that can be linked to. Additionally, chats have not yet been used in any course underlying this repository.

Examples

This pattern has been identified from personal experience outside educational environments and by review of pertinent literature. It has not yet been applied in any of the courses underlying this repository.

Evaluation

Not available.

References

Ingram, A. L., Hathorn, L. G., & Evans, A. (2000). Beyond chat on the internet. *Computers & Education, 35*, 21-35.

Orvis, K. L., Wisher, R. A., Bonk, C. J., & Olson, T. M. (2002). Communication patterns during synchronous Web-based military training in problem solving. *Computers in Human Behavior, 18*, 783-795.

COMPUTER-MEDIATED COMMUNICATION

Package: *Interactive Elements*

Intent

Asynchronous and synchronous means of online communication allow for online interaction and exchange among participants independent of time and location.

Motivation

Student-to-student communications become increasingly important when the teacher is no more the sole director of educational experience[542], but rather a facilitator of learning as in Person-Centered learning scenarios. Providing participants with means of computer-mediated communication (CMC) gives them the opportunity to stay connected with each other even when being locally or chronologically dispersed. There are two distinct forms of CMC: synchronous and asynchronous. *Synchronous* CMC resembles direct communication with each other (i.e., a dialog) and offers facilities for immediate response (e.g., online CHAT). *Asynchronous* CMC, on the other hand, rather resembles the posting principle: messages are posted, and responses (if any) arrive with initially unknown delay (e.g., in ONLINE DISCUSSIONS, or e-mailing). CMC, as opposed to face-to-face communication, has the advantage of always having the option of recording the communication messages by backing up the communication logs.

However, it is often hard to initiate and especially to sustain online collaboration and communication. Some research threads argue that it is necessary to tightly couple online activity with learner assessment to ensure the desired degree of participation[543] while others stress the importance of appropriate coaching, mentoring, and facilitation practices[544]. A recent study revealed, that any form of CMC (either asynchronous or synchronous) has positive effects on subsequent face-to-face discussions, as CMC is considered more enjoyable, uninhibited, and it

[542] Ingram, Hathorn and Evans (2000, p. 28)

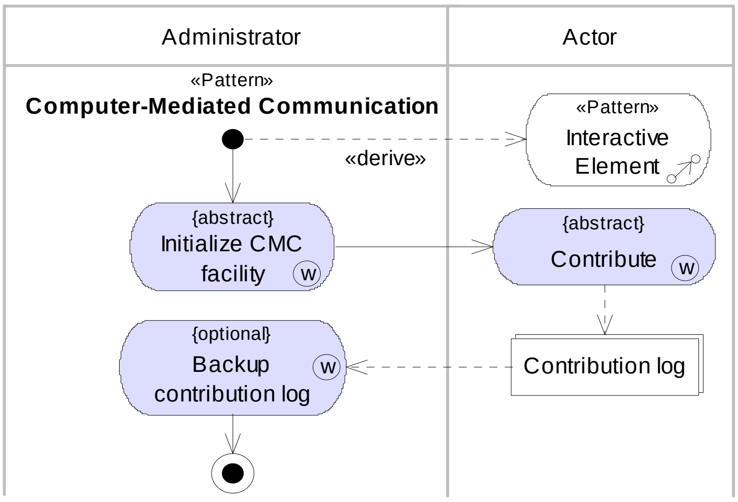
[543] For example, Macdonald (2003)

[544] For example, Ensher, Heun and Blanchard (2003), McNeil, Robin and Miller (2000)

is capable of producing a greater diversity of perspectives.[545] This is just perfect for use in blended learning scenarios. In our context this is substantiated by the fact that, for example, in the *Person-Centered Communication* as well as in the *Project Management/Soft Skills* courses, we observed that face-to-face sharing of some subject matter motivated students to discussing in ONLINE DISCUSSION forums. However, when issues that had not been attended to face-to-face were set up in online forums students tended to ignore them.

This pattern provides a shared, generic CMC sequence to be refined by more concrete patterns.

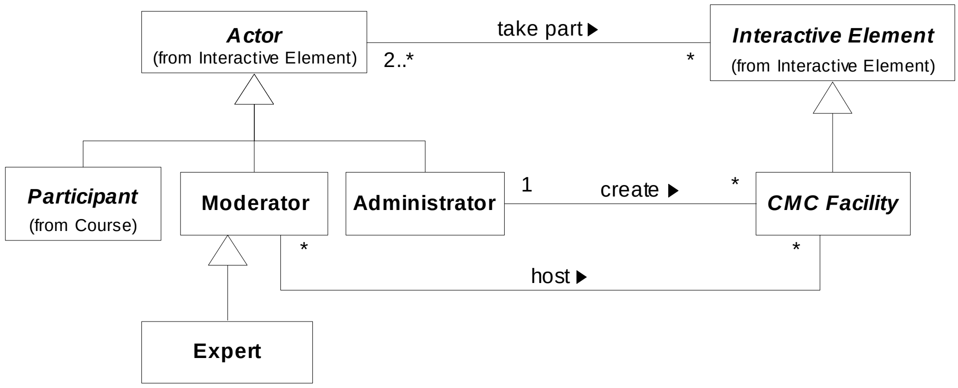
Sequence



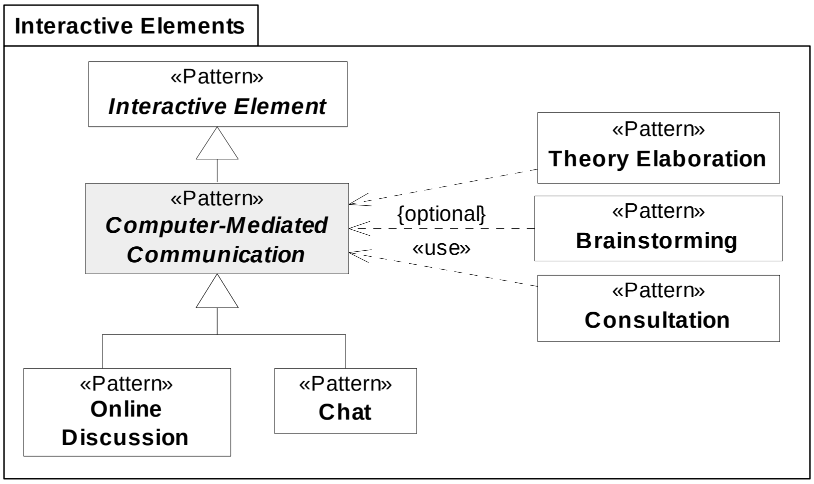
Activity	Description
Initialize CMC facility	For CMC to take place, an appropriate facility has to be initialized.
Contribute	The CMC facility provides participants with the option to contribute; the actual means of contributing have to be defined by concrete sub-patterns.
Backup CMC log	Communication logs may be backed up for later lookup or dissemination.

[545] Dietz-Uhler and Bishop-Clark (2001)

Structure



Taxonomy/Dependencies



Parameters

- Primary pattern author: Michael Derntl
- Primary pattern source: Research Lab for Educational Technologies (University of Vienna)
- Pattern categories: Utility, Generic, Collateral
- Level of abstraction: Medium
- Scope: Activity, Phase
- Primary presence type: Online
- Flexibility: High
- Level of confidence: 4
- Application effort: Low
- Level of expertise required: Medium (providing appropriate stimulation of online interactivity may be difficult)
- Suggested assistance: Administrator
- Target skills: Communication, Collaboration
- Input: CMC facility and configuration

- Output: Communication logs

Web Template

Not available: see *Remarks*.

Examples

Not available: see *Remarks*.

Evaluation

Not available: see *Remarks*.

Remarks

Web Templates, *Examples*, and *Evaluation* may be presented by more concrete patterns.

References

Dietz-Uhler, B., & Bishop-Clark, C. (2001). The use of computer-mediated communication to enhance subsequent face-to-face discussions. *Computers in Human Behavior, 17* (2001), 269-283.

Ensher, E. A., Heun, C., & Blanchard, A. (2003). Online mentoring and computer-mediated communication: New directions in research. *Journal of Vocational Behavior, 63* (2003), 264-288.

Ingram, A. L., Hathorn, L. G., & Evans, A. (2000). Beyond chat on the internet. *Computers & Education, 35*, 21-35.

Macdonald, J. (2003). Assessing online collaborative learning: process and product. *Computers & Education, 40* (2003), 377-391.

McNeil, S. G., Robin, B. R., & Miller, R. M. (2000). Facilitating interaction, communication and collaboration in online courses. *Computers & Geosciences, 26* (2000), 699-708.

Consider Conventional Style

Package: *Interactive Elements*

Intent

Offer participants who dislike self-initiated scenarios the option to switch to a more conventional, directive course style.

Motivation

Participants may initially feel uneasy with the high degree of freedom and self-responsibility provided by Person-Centered learning scenarios. Those participants, who prefer a higher degree of instruction and guidance, should be provided with a more conventional, programmed style of teaching. Carl Rogers calls such an option *division of group*[546] or *choice of a group*[547], while acknowledging the fact that concurrent teaching threads employing different degree of freedom may not always be feasible, but should at least be considered by instructors. One of the biggest problems with this is that preparation of multiple teaching threads requires the instructor to afford additional time and resources. Thus it seems appropriate to obtain a majority vote. Even if the majority votes for a conventional style, the instructor has the option of giving those opting for the person-centered style more freedom in the learning process.

An apt time to offer such an option would be in the INITIAL MEETING of a course. Nevertheless, even if the majority of participants initially opt for the Person-Centered style, the instructor should be aware of uncomfortable participants throughout the course.

Sequence

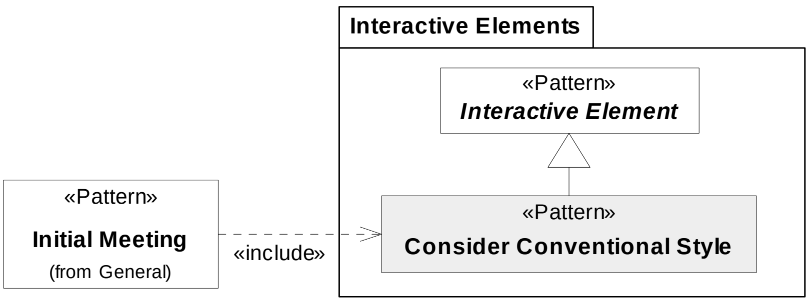
No formal sequence is available. There are different options of obtaining a decision regarding the course style:

- Ask the participants in the INITIAL MEETING (e.g., obtain majority vote)
- Collect initial QUESTIONNAIRE where this issue is addressed.
- Letting participants rate (e.g., on a numerical scale) the conventional and person-centered style, respectively.
- ...

Structure

Inapplicable.

Taxonomy/Dependencies



Parameters

- Primary pattern author: Michael Derntl

[546] Rogers (1983, p. 154)

[547] Rogers and Freiberg (1994, p. 202)

- Primary pattern source: Research Lab for Educational Technologies (University of Vienna)
- Pattern categories: Utility, Generic, Collateral, Motivational, Administrative
- Level of abstraction: Low
- Scope: Activity
- Primary presence type: Present
- Flexibility: Low
- Level of confidence: 4
- Number of participants: Unrestricted
- Application effort: Low
- Level of expertise: Medium
- Person-Centered variables addressed: Acceptance, Transparency, Understanding
- Input: Provision of different teaching styles
- Output: Decision on style

Web Template

Inapplicable.

Examples

In most of the courses underlying this pattern, the option of switching to a more directed style was provided in the INITIAL MEETINGS. However, those few participants who initially raised objections soon perceived the benefits of the Person-Centered style, so there was never any need to switch to traditional mode.

Evaluation

Not available.

Remarks

The following sections are unavailable in this pattern:

- There is no meaningful generalizable *Sequence* as well as *Structure* for this pattern
- As this pure face-to-face interaction is described, there is no *Web Template.*

References

Rogers, C. R. (1983). *Freedom to Learn for the 80's*. Columbus, OH: Charles E. Merrill Publishing Company.

Rogers, C. R., & Freiberg, H. J. (1994). *Freedom to Learn* (3rd ed.). Columbus, Ohio: Charles E. Merrill Publishing Co.

CONSULTATION

Package: *Interactive Elements*

Intent

Provide options for participants to seek synchronous (CHAT) or asynchronous (ONLINE DISCUSSION) consultation from teaching staff or experts regarding specific questions, topics or problems.

Motivation

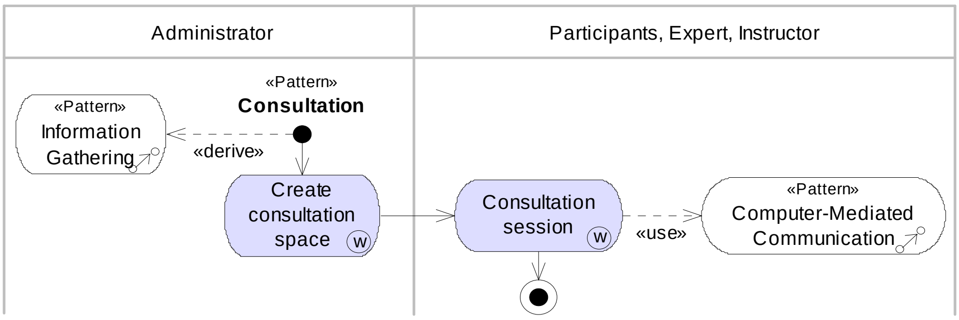
Availability of instructors and tutors is highly appreciated by participants and a critical part of the educational process[548]. Usually, instructors are available for face-to-face consultation during their office hours. Additionally, the instructor can make herself available online in *distant* communication scenarios for questions regarding administration, certain topics, problems, or general questions:

- The instructor may host CHAT sessions at defined hours and intervals. This allows for synchronous online consultation. To make conversations of general interest available to other participants not attending the chat, a chat log should be published.
- Alternatively, an ONLINE DISCUSSION forum may be initiated by the instructor for asynchronous consultation. Thereby it is important that the instructor frequently reviews the forum and dedicates herself to answer questions within a certain justifiable time span.
- Of course there are many additional options of online availability of teaching staff, such as e-mail, mailing lists, newsgroups, video conferencing tools, application sharing, etc. However, e-mailing and mailing lists are considered to be common practice and the other options seem too cumbersome and/or specialized to be incorporated as patterns at this time.

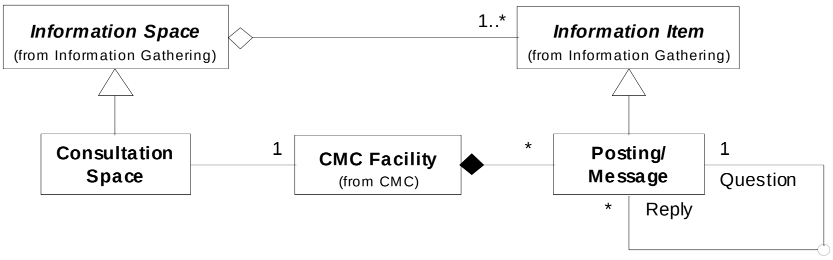
However, this scenario is not restricted to instructor-participant interaction. It may also be provided by tutors, or external experts invited to host CHATS or ONLINE DISCUSSIONS on specific topics.

[548] Cf. Wallace and Wallace (2001, p. 195)

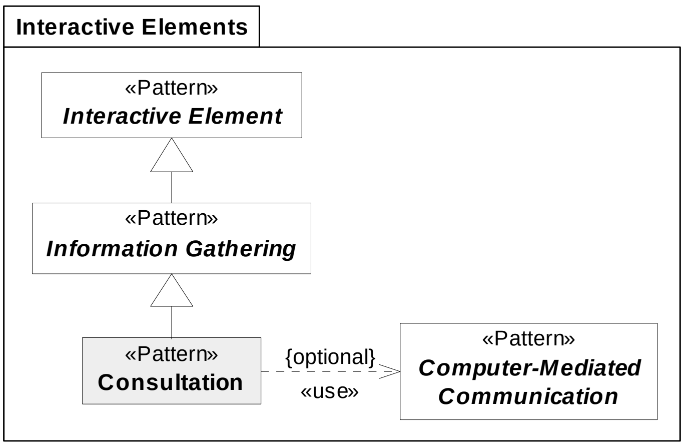
Sequence



Structure



Taxonomy/Dependencies



Parameters

- Primary pattern author: Michael Derntl
- Primary pattern source: Research Lab for Educational Technologies (University of Vienna)
- Pattern categories: Utility, Administrative, Collateral, Motivational
- Application effort: Low
- Scope: Activity, Phase
- Primary presence type: Online
- Flexibility: Low

- Level of confidence: 3
- Number of participants: Unrestricted
- Level of abstraction: Low
- Level of expertise required: Low
- Input: Consultation forum
- Output: Consultation logs

Web Template

This pattern simply uses some form of Computer-Mediated Communication for consultation purposes. Only the consultation mode has to be announced (PUBLISH) accordingly.

Examples

In ***Web Engineering 2003***, a general ONLINE DISCUSSION forum was installed for exchange on organizational issues, which was frequently reviewed for new questions and postings by instructors. In the subsequent years ***2004*** and ***2005*** dedicated discussion forums were created for each instructor, where participants could post questions and comments directly to their instructor. See Figure 110 for a screenshot of my Web Engineering consultation forum in 2005.

Figure 110: Online CONSULTATION forum in Web Engineering 2005.
The heading and introductory text of the forum were translated from German. Participants' surnames have been blurred to maintain anonymity.

Evaluation

Not available.

References

Wallace, F. L., & Wallace, S. R. (2001). Electronic office hours: a component of distance learning. *Computers & Education, 37* (2001), 195-209.

Elaborate Goals and Expectations

Package: *Interactive Elements*

Intent

Elaborate and PUBLISH participants' goals and expectations (and also fears) for the course or for specific learning activities.

Motivation

In traditional settings, goals of the course or of the learning activities are solely set by the instructor. In Person-Centered settings, the learner should be actively involved in defining overall as well as individual goals for the course, such as, "*What do I want to achieve in the course?*" It seems to be a common misconception that the only goal of students is to get the final grade right. Getting occupied with and acquiring comprehensive knowledge in the course's subject matter and improving professional as well as soft skills is – or can in proper settings become – something that complies by far more with students' goals and motivations than just getting the final grade, as the histogram in Figure 111 underlines.

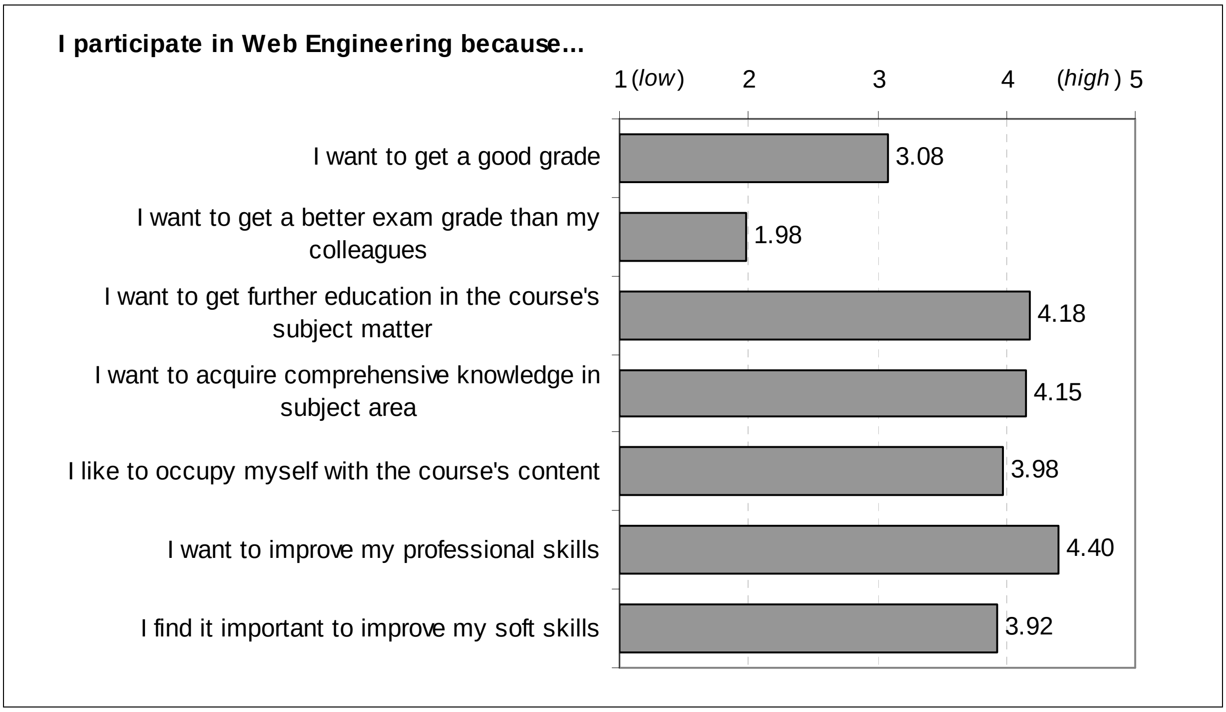
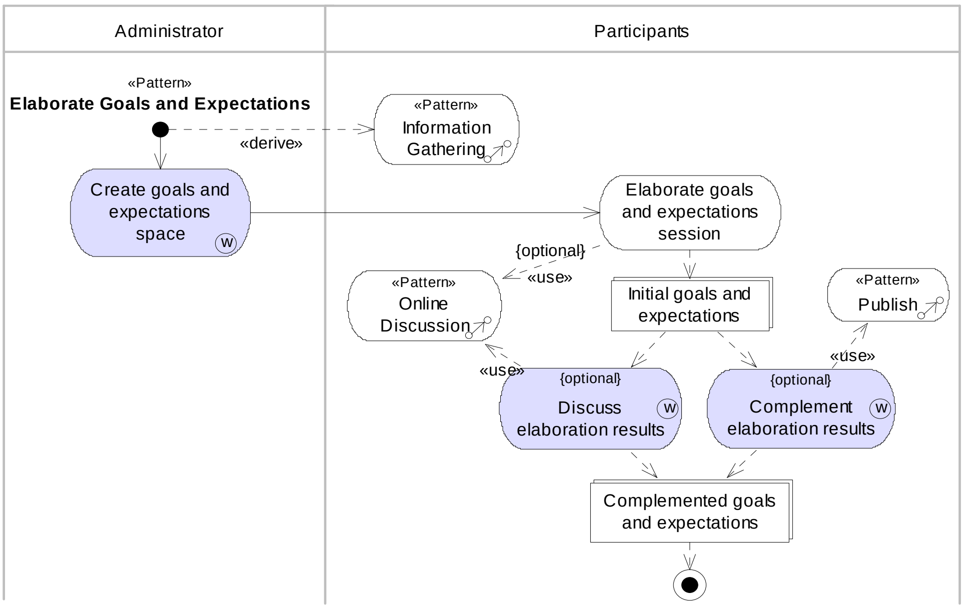


Figure 111: Initial motivation to participate in the Web Engineering course (n=131).
At the beginning of the Web Engineering course, a questionnaire was distributed among participants where 22 motivational items were surveyed.

Thus, inquiring and collecting goals (and possibly fears) from participants should raise motivation to learn, as the instructor can – at least to certain degree – adapt the learning process to the goals and fears formulated by participants. Additionally, participants bring along different expectations for the course, either thematically or regarding what is demanded from them. However, incorporating expectations in course design requires a high degree of flexibility of the instructor, e.g., when providing additional content or elaborating/lecturing on additional topics. Nevertheless it is certainly a worthwhile effort, as cognitive psychology has shown that furthering learner orientation by having the participants formulate questions and goals first is an effective strategy for improving their learning[549]. Elaborating *and* considering goals and expectations of participants, letting them speak first, and thereby letting them open communications channels is one option of being *acceptant* toward participants by considering their concerns.

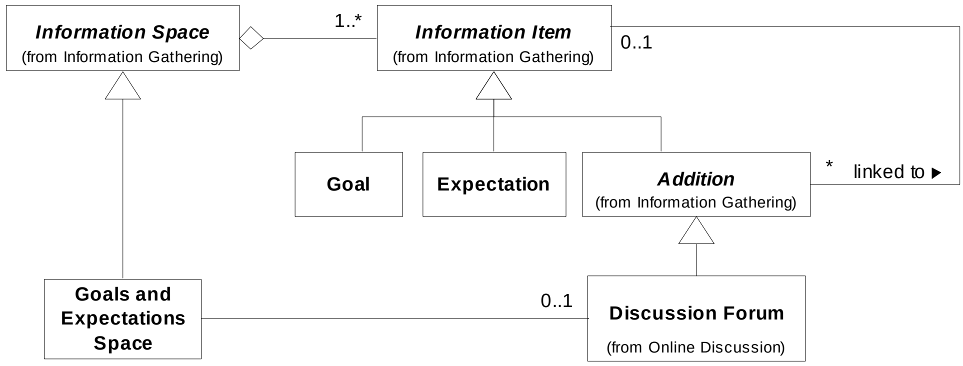
[549] Anderson (1991)

Sequence

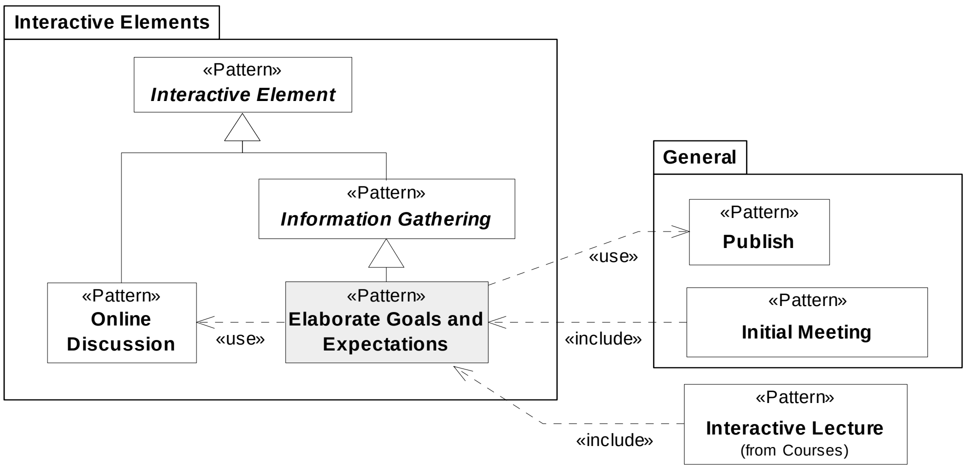


Activity	Description
Create goals and expectations space	Space has to be allocated on the learning platform, where results from the elaborate goals and expectations sessions can be published to participants.
Elaborate goals and expectations session	For the elaboration session there are different variations: • The elaboration may take place face-to-face. Subsequently, results are published on the platform. • The elaboration takes place online, e.g., in a moderated ONLINE DISCUSSION scenario. The results may be condensed and published in a formatted way after the asynchronous collection of goals and expectations has ended.
Discuss elaboration results	Facilities for ONLINE DISCUSSION of elaborated items are provided.
Complement elaboration results	Facilities for adding (i.e., uploading or posting) goals and expectations are provided.

Structure



Taxonomy/Dependencies



Parameters

- Primary pattern author: Michael Derntl
- Primary pattern source: Research Lab for Educational Technologies (University of Vienna)
- Pattern categories: Utility, Motivational
- Level of abstraction: Low
- Scope: Activity, Phase
- Primary presence type: Online
- Flexibility: High
- Level of confidence: 5
- Number of participants: Unrestricted
- Application effort: Medium
- Level of expertise required: Medium
- Target skills: Collaboration, Communication, Self-directedness, Questioning, Expressing own interest
- Input: Guidelines, space for results
- Output: Goals and expectations of participants

Web Template

Inherited: This pattern reuses the Web template from parent pattern INFORMATION GATHERING.

Examples

- Elaboration of goals and expectations was applied in a ***PhD Research Seminar*** in the summer term 2003 by providing a dedicated ONLINE DISCUSSION forum, where the participants could post their expectations for the seminar.
- In ***Web Engineering*** (2003-2005) goals and expectations were elaborated face-to-face in the first lecture session in a BRAINSTORMING-like manner. While the participants raised their comments, the assistant wrote them down in real-time on a laptop, whose screen was projected on the wall. The instructor has prepared about 10 major topics for the course, and the gathering session revealed that thematic expectations of participants were mainly congruent with the instructors' topics. However, several additional topics of special interest to the participants were taken up and presented during the lectures.
- In ***Project Management/Basics and Techniques*** (winter term 2003), a three-step procedure was employed to elaborate goals and expectations: First discussion in small teams, then compilation and collection, and finally a protocol published on the platform.

Evaluation

Not available.

References

Anderson, J. R. (1991). *Cognitive Psychology and its Applications* (3rd ed.). New York: Freeman.

EXCHANGE OF CONTRIBUTIONS

Package: *Interactive Elements*

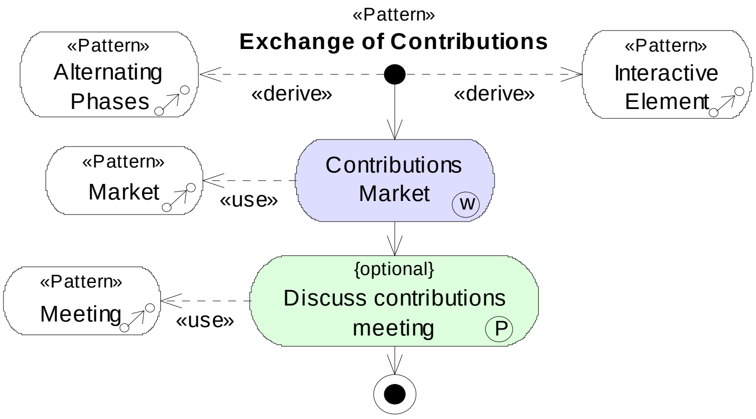
Intent

Let participants exchange and discuss their contributions and ideas online.

Motivation

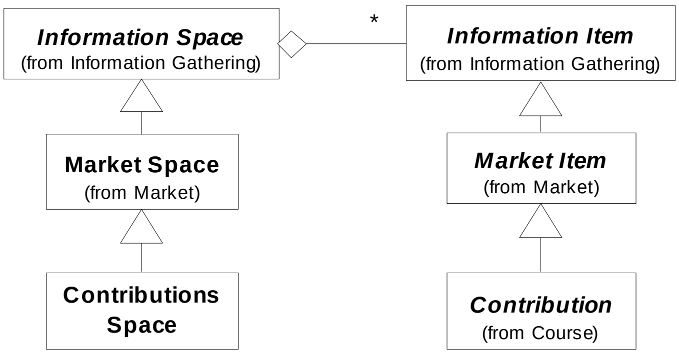
When participants or teams elaborate their contributions, it often happens that there are various overlaps and points of contact in the different contributions. Therefore it is useful to provide the participants with space for exchange of contributions, resources, and ideas. This way they can profit and learn from the work of each other. This experience may be deepened additionally by hosting a face-to-face discussion meeting following the online exchange phase.

Sequence

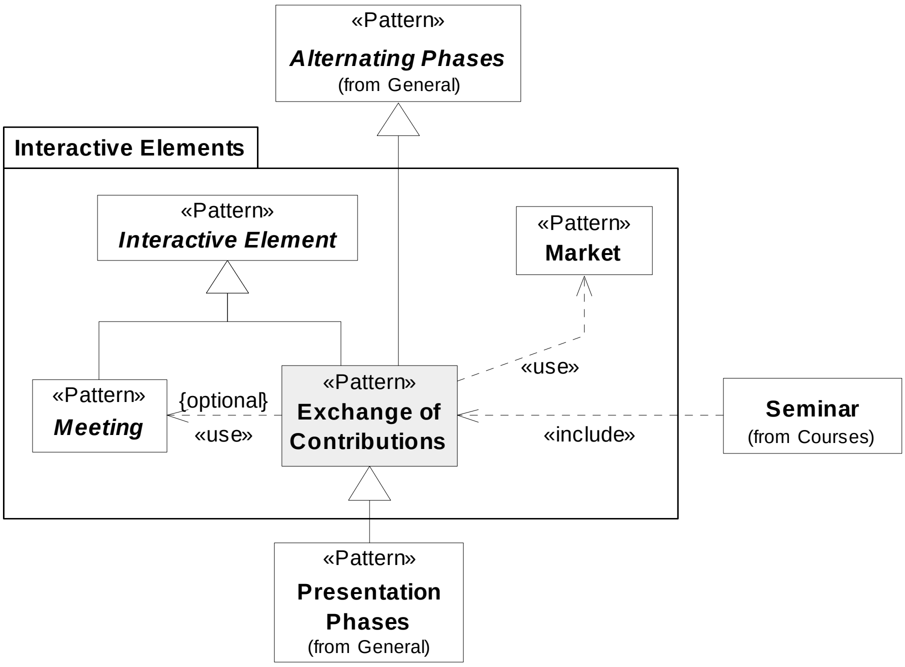


Activity	Description
Contributions market	A space for exchange is provided, which basically resembles a MARKET for upload, download, and discussion contributions.
Discuss contributions meeting	Optionally, to continue and further online exchange and experiences, a discussion meeting is held.

Structure



Taxonomy/Dependencies



Parameters

- Primary pattern author: Michael Derntl
- Primary pattern source: Research Lab for Educational Technologies (University of Vienna)
- Pattern categories: Utility, Generic, Collateral, Motivational
- Level of abstraction: Medium
- Scope: Phase
- Primary presence type: Online
- Flexibility: High
  Other forms of exchange can help as an alternative to using Markets.
- Level of confidence: 4
- Number of participants: Unrestricted
- Application effort: Medium
- Level of expertise required: Low
- Suggested assistance: Tutor
- Target skills: Communication, Collaboration
- Input: Exchange guidelines, relevant content, market space
- Output: Populated market space

Web Template

For Web-support of Exchange of Contributions, a simple Market with appropriate settings suffices. See the Market Web template for details. For minimal implementation, even an Online Discussion forum might be appropriate.

Examples

This pattern is applied on a regular basis in ***PhD Seminars***: In literature seminars, where literature from specific subject areas is reviewed, and in research seminars, where PhD theses are presented, the participants exchange their reports, topics of interest, and related resources in an online forum. Additionally, the instructor offers the opportunity to host a subsequent discussion MEETING on demand. These meetings are usually well attended and commonly turn out to be very enriching through exchange of ideas, opinions, and expert contacts on each participant's topic of focus.

Evaluation

Not available.

INFORMATION GATHERING

Package: *Interactive Elements*

Intent

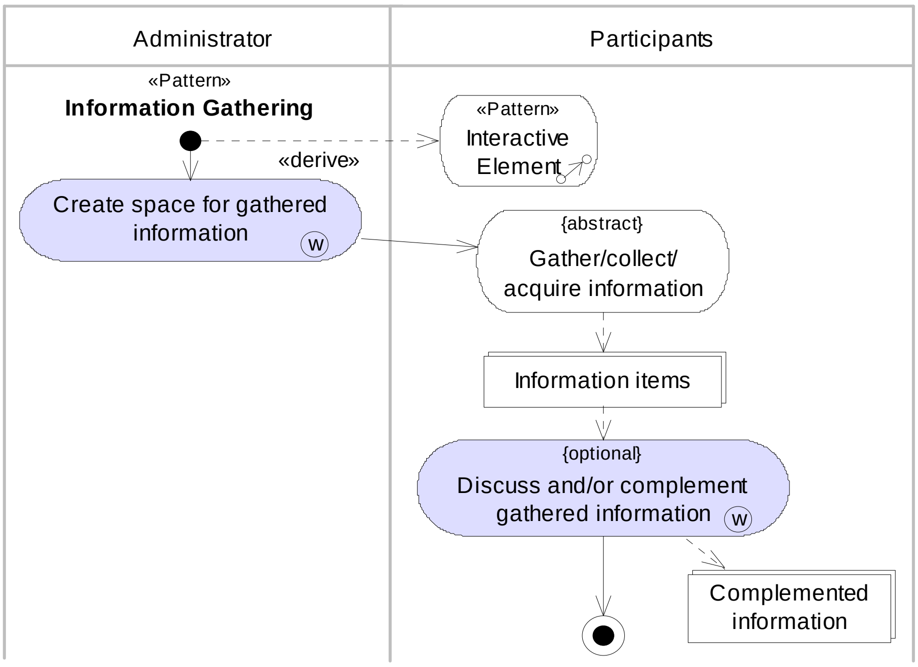
Participants and instructors interact with the primary target to collect information which shall be gathered collaboratively and shared among all participants.

Motivation

INFORMATION GATHERING is a generic pattern aiming to actively involve participants in the learning process by letting them collect, discuss, and add to collaboratively gathered information, such as theories, examples, brainstorming results, etc. Additionally, inherited motivational aspects from the parent INTERACTIVE ELEMENTS pattern apply.

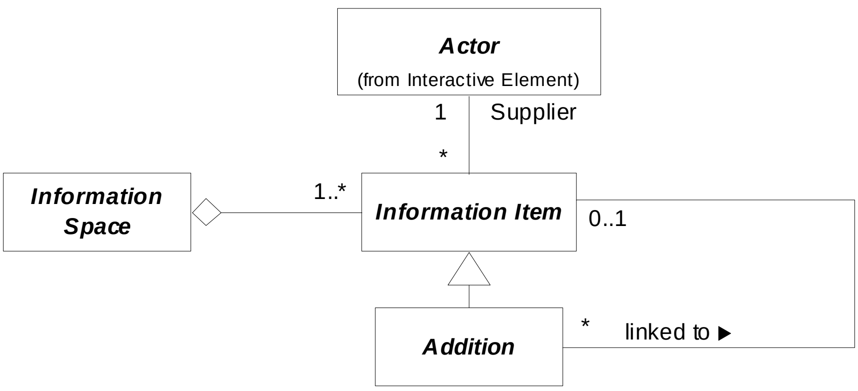
Gathering of information seems like a typical face-to-face scenario, e.g., when thinking of BRAINSTORMING sessions. However, this pattern lays foundation for more specific information gathering patterns to allow for *online support* of such activities. This requires allocation of space on the learning platform to publish gathered information. Such space may either be a folder, a discussion forum or just a collection of documents. Subsequently, participants get the opportunity to review, discuss, and add to information that was collected in face-to-face or online sessions.

Sequence

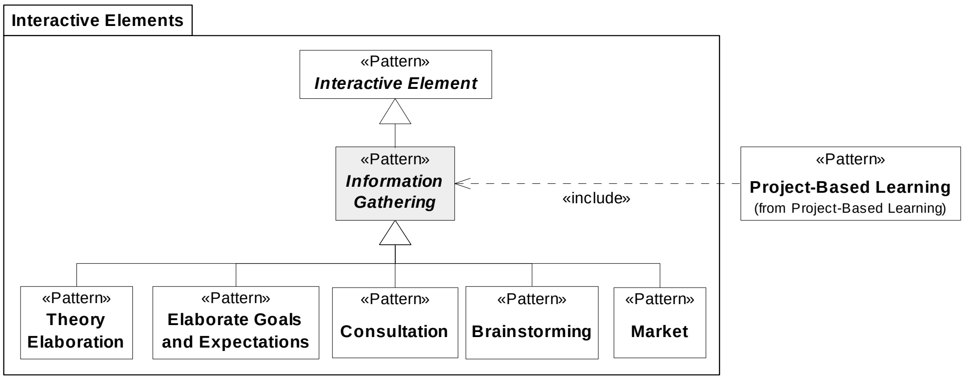


Activity	Description
Create space for gathered information	A Web template for creating space for gathered information has to be provided by concrete sub-patterns, as different gathering methods may require different types of Web spaces.
Gather/collect/acquire information	This is an abstract placeholder which has to be refined by sub-patterns. It represents the actual method of gathering information.
Discuss and/or complement gathered information	Depending on the information gathering method it may be useful to provide means for discussing and adding to existing information.

Structure



Taxonomy/Dependencies



Deriving from INTERACTIVE ELEMENTS, this pattern is located in the *Interactive Elements* package. It has a number of concrete sub-patterns showing specific methods of gathering information.

Parameters

- Primary pattern author: Michael Derntl
- Primary pattern source: Research Lab for Educational Technologies (University of Vienna)
- Pattern categories: Utility, Generic, Collateral, Motivational
- Scope: Activity, Phase
- Level of abstraction: Medium
- Flexibility: High
- Target skills: Interpersonal Skills, Collaboration, Problem solving, Practical skills
- Input: Method of information gathering, space on the platform
- Output: Gathered and optionally complemented information

Web Template

Even though this is an abstract pattern, it provides a generic Web template that may be reused and/or overridden by sub-patterns. According to the *Sequence* section, the two basic activities in this pattern are as follows:

- First, the actual information gathering session: Regardless of what kind of information is gathered, we may always use a form of COMPUTER-MEDIATED COMMUNICATION in case that activity proceeds online. If that session does not proceed online, there will be an option to add the session's results manually as a structured text or document.

- Second, the subsequent activity "*Discuss and/or add to gathered information*" allows participants to complement the information gathering session's results, which is also possible online through COMPUTER-MEDIATED COMMUNICATION.

Therefore, this pattern assumes that for each of the two main activities we use a form of COMPUTER-MEDIATED COMMUNICATION.

Administration View

The following wizard steps are required to configure an instance of this pattern, based on the assumption:

1) Configuration of the main information gathering page, which is specific to the approach used (e.g., Brainstorming, Theory Elaboration, etc.) This includes setting a *heading* and an *introductory text* for that page. Additionally, it has to be specified whether the actual information gathering session is already completed, which would mean that the *results are already available*. (See Figure 112.)

2) Depending on whether results are already available as specified in step 1, one of the following steps follows:

   a) *Session coming up*: In this case, it has to be specified which method will be used for the information gathering session. Typically, this will be some form of Computer-Mediated Communication (Discussion Forum or Chat), or a simple Market. (See Figure 113.)

   b) *Results already available*: In that case, a textbox is offered to enter the results in structured form. (See Figure 114.)

3) Configuration of result complementation, which can be activated after the information gathering session has completed, allowing participants to discuss the results in an attached Online Discussion forum. If this is activated, *guidelines for complementation* can be entered. (See Figure 115.)

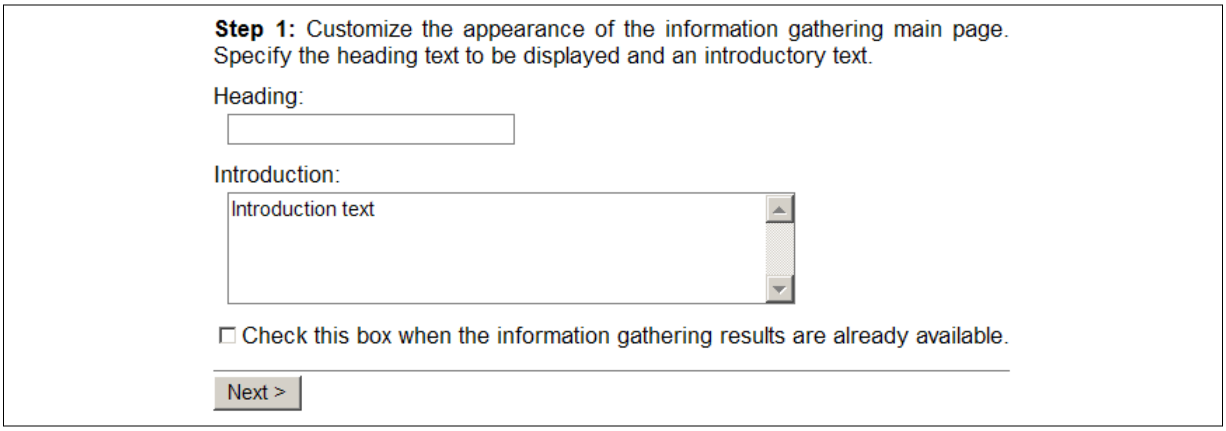


Figure 112: General configuration of Information Gathering.

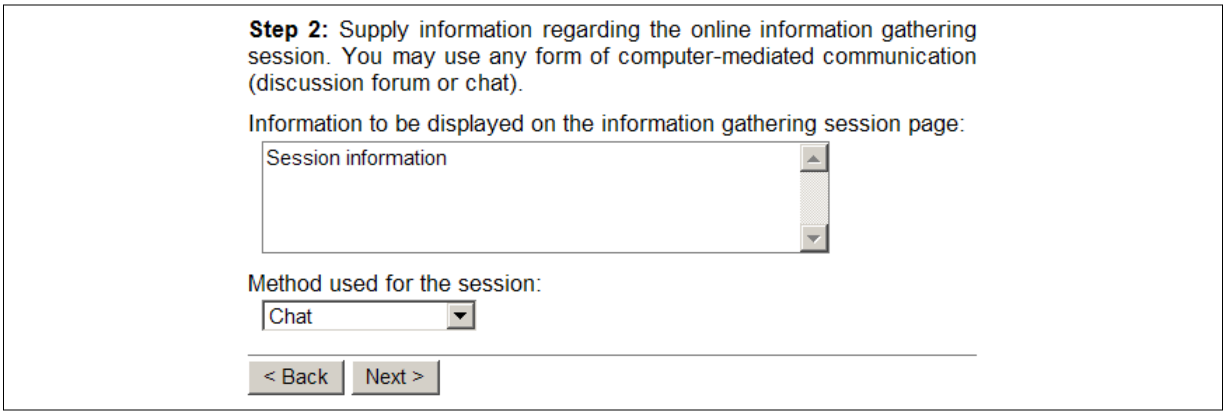


Figure 113: Configuration of the actual INFORMATION GATHERING session.

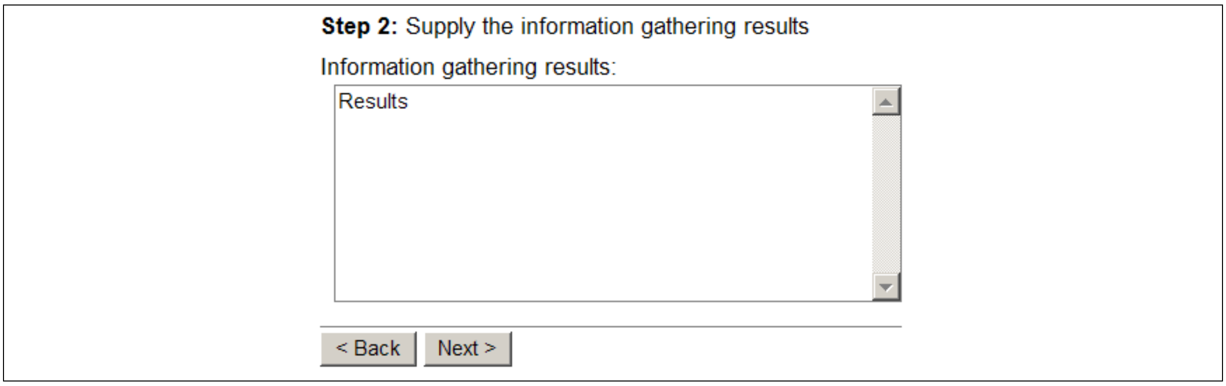


Figure 114: Configuration of INFORMATION GATHERING results.

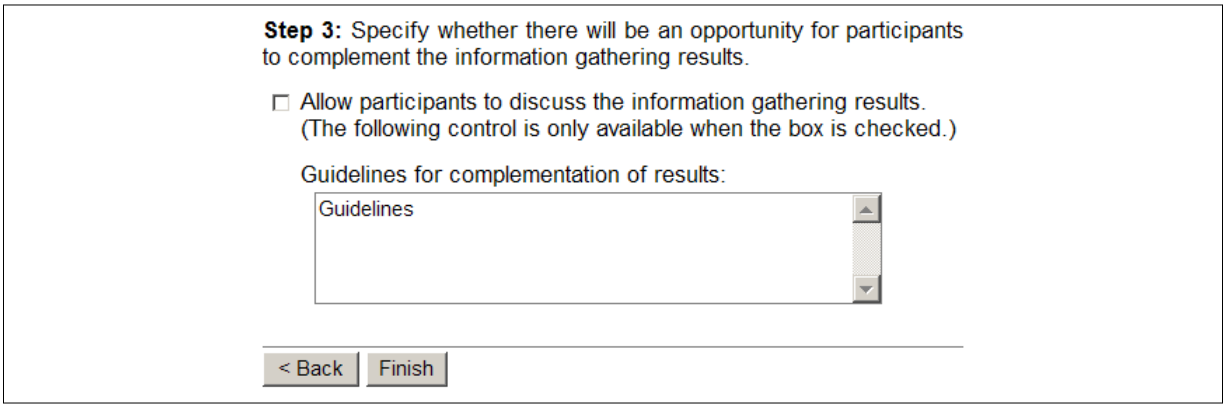


Figure 115: Configuration of complementation options for INFORMATION GATHERING.

Participant View

The participant view of this pattern is as generic as the pattern itself. It just displays the structure of each respective page. There are two pages in this view:

1) ***Information gathering session page***: This page (see Figure 116) displays the configured *heading*, *introduction text*, and *information* for the gathering session. The last part contains the actual participant view of the employed *gathering method*, which was configured in step 2 of the administration view (e.g., MARKET or ONLINE DISCUSSION).

Note that this page structure only applies to scenarios where the actual gathering session lies ahead or is currently in progress. Otherwise, the following page applies:

2) ***Results and complementation page***: After the gathering session is over, and the results have been structured in the administration view, the page structure includes the *session results* along with a *link* to a page where the participants can complement the results (see Figure 117). The complementation method itself depends on the method of the actual session. For example, if an ONLINE DISCUSSION forum was employed for the session, the complementation page would typically offer the participant to post a message to that forum.

[Heading]

[*Introduction text*]

[*Session information*]

[*Participant view of employed information gathering method*]

Figure 116: INFORMATION GATHERING page structure of the actual gathering session.

[Heading]

[*Introduction text*]

[*Session results*]

[*Link to result complementation*]

Figure 117: INFORMATION GATHERING page structure of the session results.

Report View

No separate report view for this pattern is provided here, as all activities are traceable via the participant view. However, to ease reporting for the instructor, it might be useful to provide an overview of *who* contributed *what* in the actual information gathering session and in the complementation phase, respectively. That information may also be provided directly in the participant view.

Examples

Not available: see concrete sub-patterns for examples.

Evaluation

Not available.

INTERACTIVE ELEMENT

Package: *Interactive Elements*

Intent

Set of learning activities involving active participation and interaction among participants, instructors, and/or tutors.

Motivation

Interactivity, collaboration, cooperation, and other forms of active participation are central goals of any Person-Centered learning scenario. Aspects of interactivity such as between participants and instructor, among participants, or between learner and software, are essential ingredients of making Web-based learning networks effective[550]. This pattern lays foundations for execution and online support of concrete interactive scenarios. When learners are actively involved in the learning process, several benefits may arise[551]:

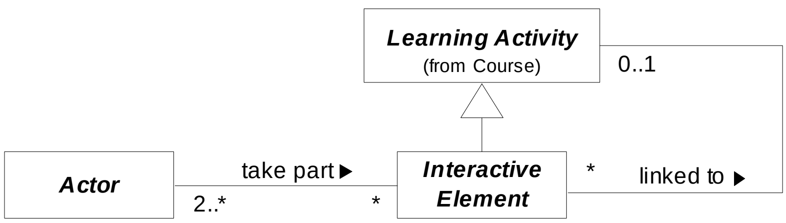
- It helps learners to maintain *attention* of participants in the learning process.
- Contributing actively may increase *long-term learning effects* as participants have personal stake in learning.
- *Motivation* rises as students have to take higher responsibility in the process.

This pattern does not supply a concrete scenario, but the following proposition must be true for any derived sub-pattern: An interactive element is a scenario where a kind of interaction among participants, instructors, and/or tutors takes place that supports the learning process, e.g., BRAINSTORMING or ONLINE DISCUSSION.

Sequence

Not available.

Structure

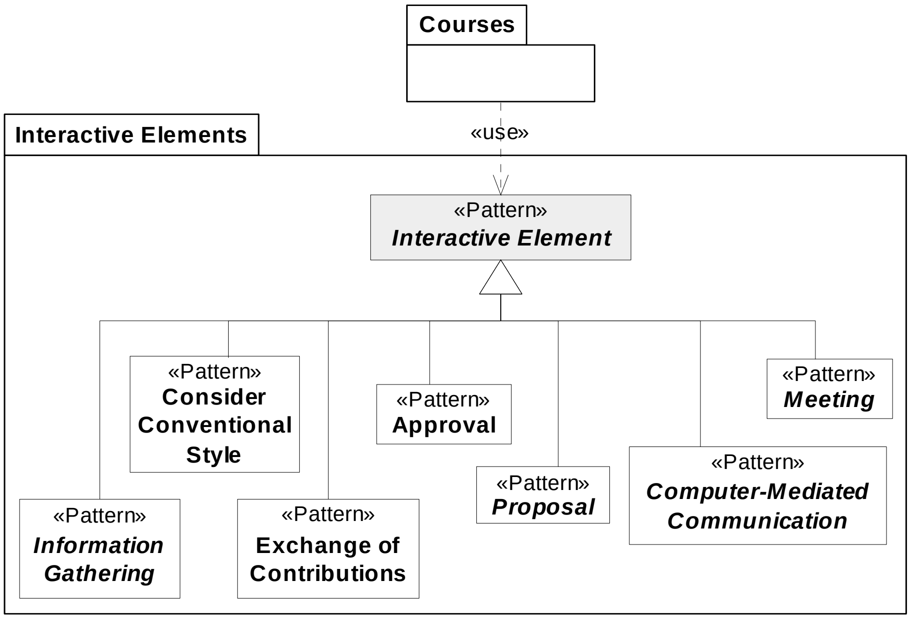


[550] Cf. Hiltz and Turoff (2002)

[551] Cf. Cotton (1995, p. 110f)

For interaction to take place at least two *actors* must be involved in an *interactive element*, which itself is a kind of a *learning activity*. The actor, the interactive element, and the kind of participation of the actors in the interactive element are abstract concepts and have to be specialized by sub-patterns.

Taxonomy/Dependencies



INTERACTIVE ELEMENT is one of the central patterns in the repository. It is currently used by all courses in the *Courses* package, and is direct parent to a host of other generic as well as concrete interactive patterns: APPROVAL, CHAT, CONSIDER CONVENTIONAL STYLE, DISCUSSION, EXCHANGE OF CONTRIBUTIONS, INFORMATION GATHERING, MEETING, and PROPOSAL.

Parameters

- Primary pattern author: Michael Derntl
- Primary pattern source: Research Lab for Educational Technologies (University of Vienna)
- Pattern categories: Utility, Motivational, Generic
- Level of abstraction: High
- Scope: Activity, Phase
- Flexibility: High

Web Template

Not available.

Examples

Not available. See concrete sub-patterns for examples.

Evaluation

Inapplicable.

Remarks

The following sections are omitted due to the abstract, generic nature of this pattern:

- *Sequence*
- *Web Template*: Typically, interactive elements (at least partly) take place in separate sections on the learning platform; however, as platform sections and platform features used for interactive elements (e.g., folder, ONLINE DISCUSSION forum, market, etc.) differ significantly, it is left to sub-patterns to specify web templates.
- *Examples*
- *Evaluation*

References

Cotton, J. (1995). *The Theory of Learning*. London: Kogan Page.

Hiltz, S. R., & Turoff, M. (2002). What makes learning networks effective? *Communications of the ACM, 45* (4), 56-59.

MARKET

Package: *Interactive Elements*

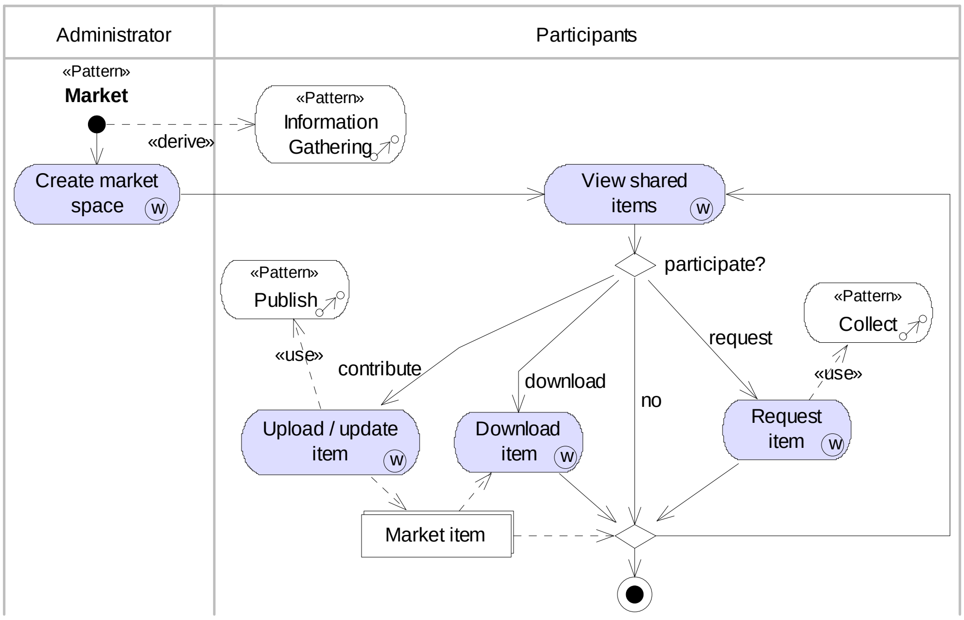
Intent

Provide facilities for participants and instructor to offer and exchange any kind of useful documents or resources.

Motivation

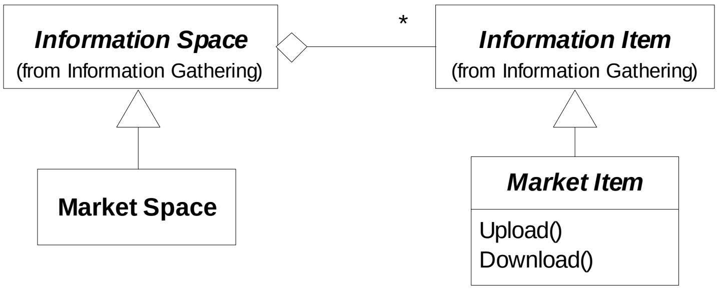
MARKET provides a reusable scenario for exchange of all kinds of information items (e.g., text or documents). It is a generic pattern that defines three general actions for markets: *uploading*, *downloading*, and *requesting*. Basically, any space where users can upload and/or download any material is a kind of a market. However, the central intention of a market is the active *exchange* of resources. The market actions are described in the following section.

Sequence

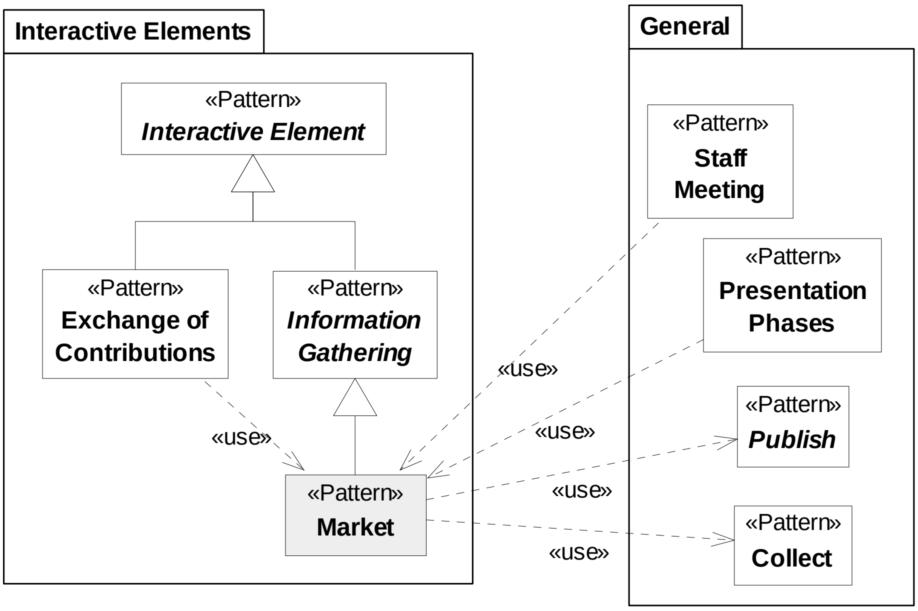


Activity	Description
Create market space	The administrator creates and initializes market space is on the platform. Authorized users are assigned to the new market.
View shared items	Markets are all about resource sharing. Any previously uploaded item can be viewed and downloaded by authorized users (in open markets there exist no access restrictions).
Upload / update item	Users can upload items onto the market space to make them available for download. If the same item is uploaded repeatedly, it is being *updated*.
Download item	When browsing through the market space, users may download available items.
Request item	This extra feature enables users to request specific items from other users, which is a uses a form of Collect. In turn, the requested items are uploaded by the holder of the item.

Structure



Taxonomy/Dependencies



Concrete usages of MARKET include EXCHANGE OF CONTRIBUTIONS (market for contributions), PRESENTATION PHASES (presentations market) or STAFF MEETING (market for preparatory exchange of agenda items).

Parameters

- Primary pattern author: Michael Derntl
- Primary pattern source: Research Lab for Educational Technologies (University of Vienna)
- Pattern categories: Utility, Generic, Collateral
- Level of abstraction: Medium
- Scope: Phase
- Primary presence type: Online
- Flexibility: High
- Level of confidence: 2
- Number of participants: Unrestricted
- Application effort: Low

- Level of expertise: Medium
- Input: Market space, guidelines
- Output: Market items

Web Template

Administration View

The administration view of this pattern offers the following options:

- A *heading* and an *introductory text* to be displayed in the participant view (see configuration step 1 in Figure 118.)
- A number of *sections*, which are used to group the exchanged documents according to some predefined criteria (see Figure 119.) For each section the administrator may specify whether each team/participant is allowed to upload *multiple documents*. Additionally, for each section there is a hyperlink to the *link management* page.
- Additional options in step 2 include:
    - o Whether *document management* within section is based on teams or single participants;
    - o Whether sections are *grouped* by teams/participant, which may be useful when MARKETS are used to collect contributions from teams/participants.
- A collection of *hyperlinks* that are displayed for each section, which may either point to internal (learning platform) or external (Web) locations (see Figure 120.)

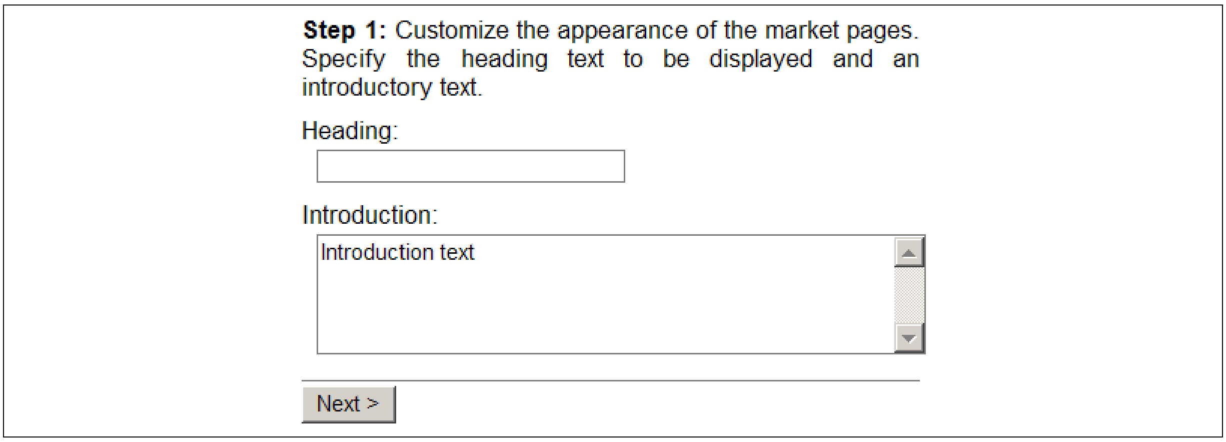


Figure 118: General configuration of MARKET.

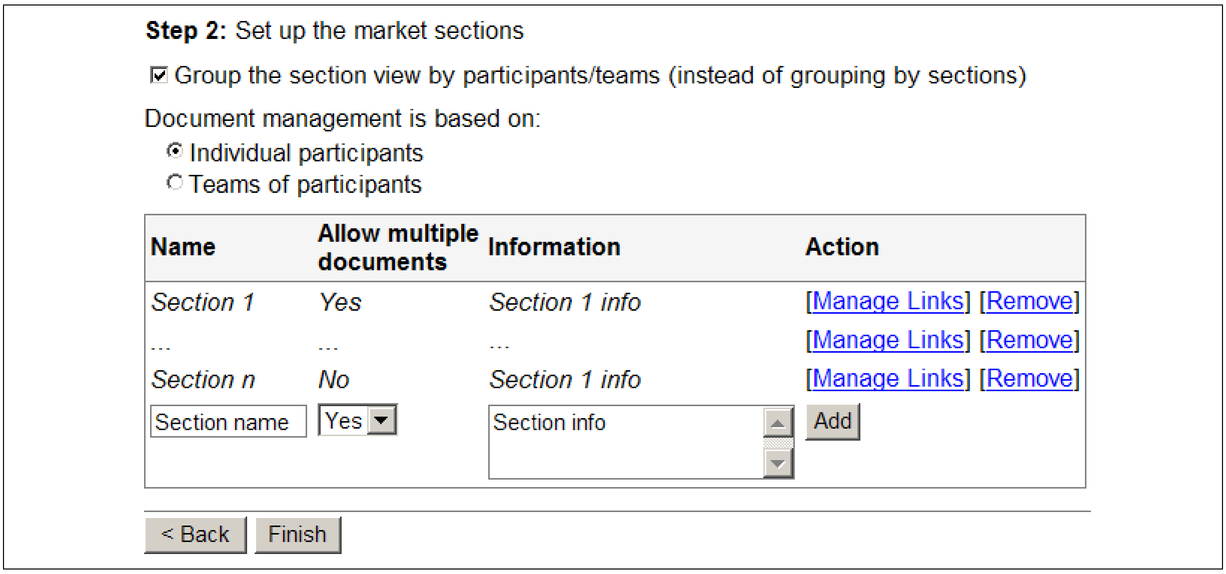


Figure 119: Section configuration for MARKET.

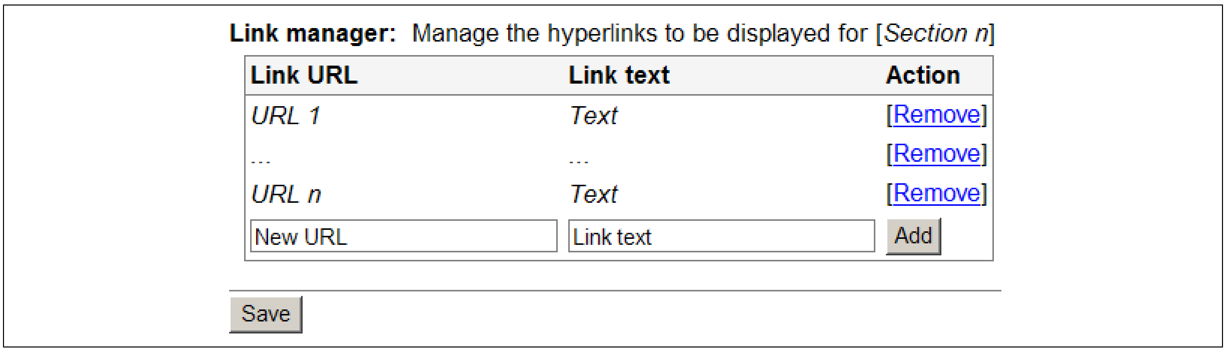


Figure 120: Link management for each MARKET sections.

Participant View

Market space: The initial view shows the *heading* and *introduction* for the market space as configured in step 1 of the administration view. This is followed by *links to section management* where documents can be managed (uploaded/deleted). The main part of the space is occupied by the *market sections*, which are optionally *grouped* by teams (as in the example in Figure 123) and filled with *links* and uploaded *documents*. Clicking on some document will initiate download of that document. Clicking on a section link leads the logged in participant to the document management page for that section (see Figure 122). This simple page includes a *list* of currently available documents in the respective section, along with the option to *delete* single documents from the section. The preceding *section information* is taken from the step 2 of the market administration. Finally, there is a file *upload control* for adding (i.e., uploading) documents to the current section.

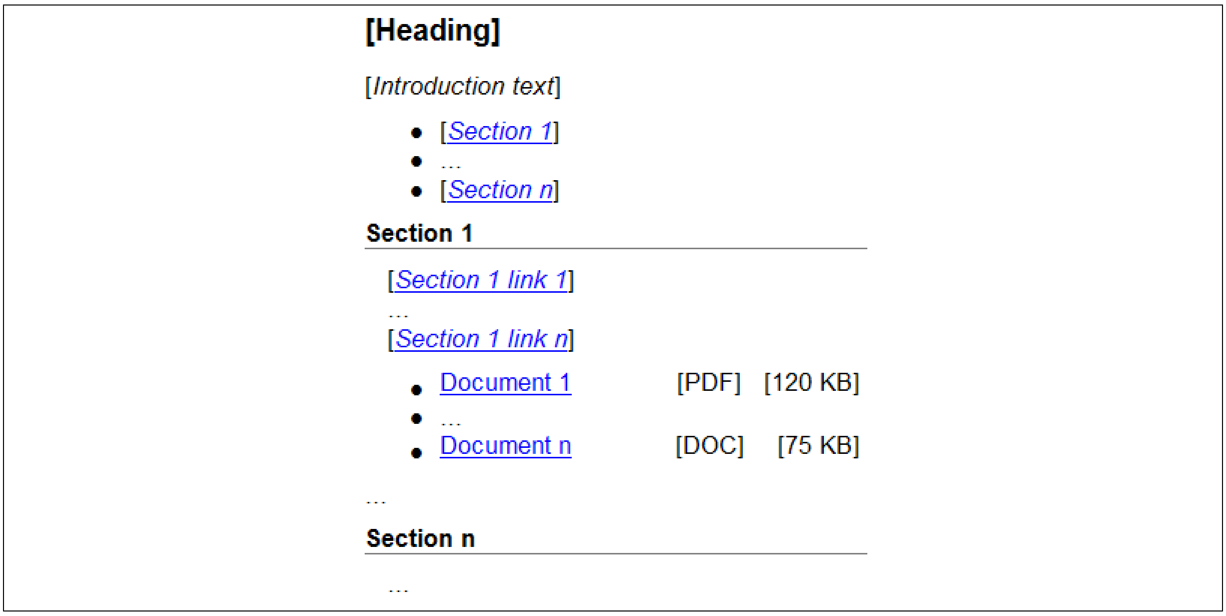


Figure 121: Participant view of the actual MARKET space.

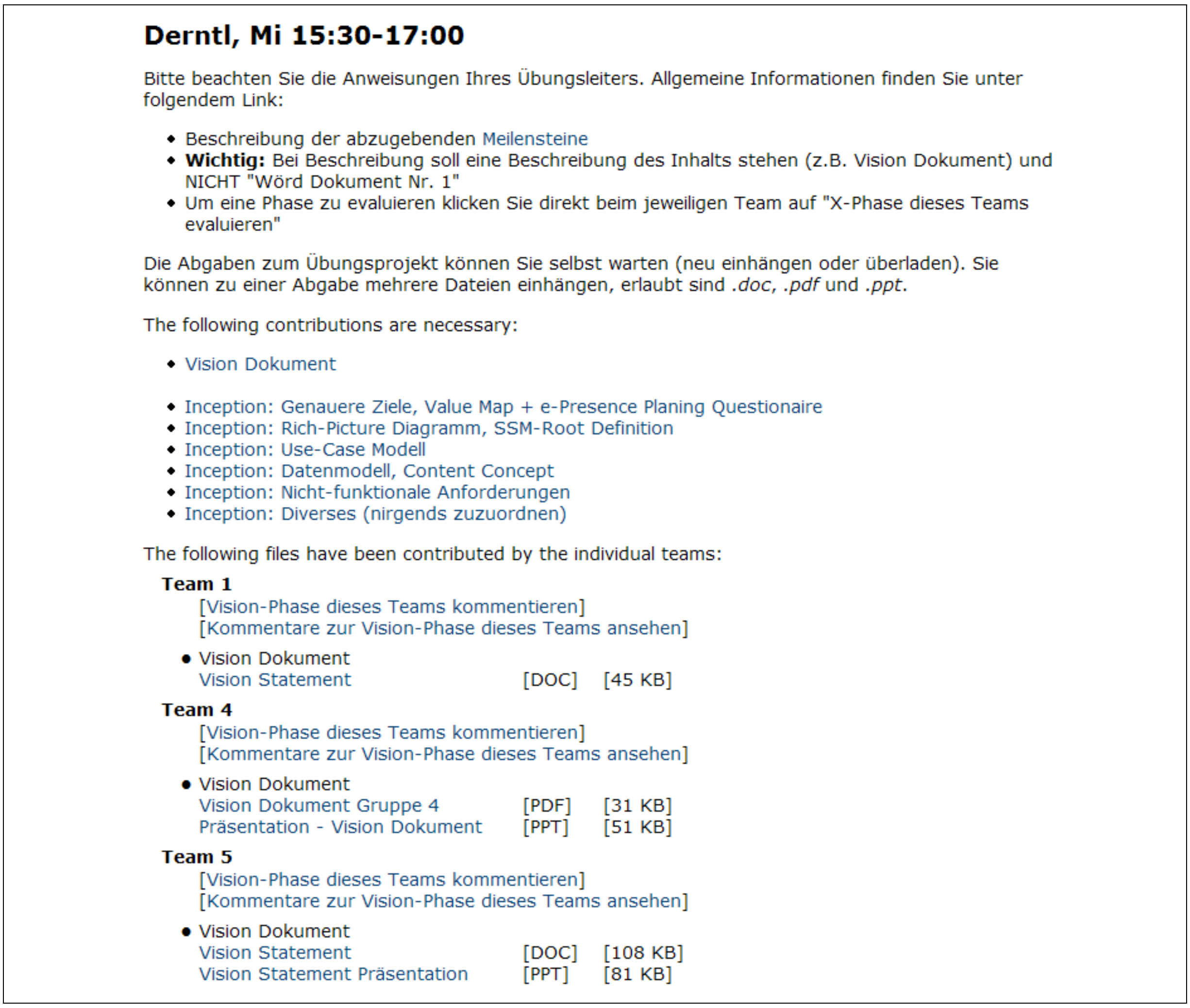
Figure 122: Section management for MARKET participants.

Report View

Market needs no dedicated report view, as all contributions are visible in the participant view (see Figure 121). However, some simple report to provide a basic overview ("what was contributed by whom?") could be useful.

Examples

The following Figure 123 shows a sample market space for lab course contributions from the ***Web Engineering 2005*** course, grouped by teams. The teams were asked to upload their project documents, while peers would be able to download/review and subsequently PEER-EVALUATE the documents.



Derntl, Mi 15:30-17:00

Bitte beachten Sie die Anweisungen Ihres Übungsleiters. Allgemeine Informationen finden Sie unter folgendem Link:

- Beschreibung der abzugebenden Meilensteine
- **Wichtig:** Bei Beschreibung soll eine Beschreibung des Inhalts stehen (z.B. Vision Dokument) und NICHT "Wörd Dokument Nr. 1"
- Um eine Phase zu evaluieren klicken Sie direkt beim jeweiligen Team auf "X-Phase dieses Teams evaluieren"

Die Abgaben zum Übungsprojekt können Sie selbst warten (neu einhängen oder überladen). Sie können zu einer Abgabe mehrere Dateien einhängen, erlaubt sind *.doc*, *.pdf* und *.ppt*.

The following contributions are necessary:

- Vision Dokument

- Inception: Genauere Ziele, Value Map + e-Presence Planing Questionaire
- Inception: Rich-Picture Diagramm, SSM-Root Definition
- Inception: Use-Case Modell
- Inception: Datenmodell, Content Concept
- Inception: Nicht-funktionale Anforderungen
- Inception: Diverses (nirgends zuzuordnen)

The following files have been contributed by the individual teams:

Team 1
[Vision-Phase dieses Teams kommentieren]
[Kommentare zur Vision-Phase dieses Teams ansehen]

- Vision Dokument
Vision Statement [DOC] [45 KB]

Team 4
[Vision-Phase dieses Teams kommentieren]
[Kommentare zur Vision-Phase dieses Teams ansehen]

- Vision Dokument
Vision Dokument Gruppe 4 [PDF] [31 KB]
Präsentation - Vision Dokument [PPT] [51 KB]

Team 5
[Vision-Phase dieses Teams kommentieren]
[Kommentare zur Vision-Phase dieses Teams ansehen]

- Vision Dokument
Vision Statement [DOC] [108 KB]
Vision Statement Präsentation [PPT] [81 KB]

Figure 123: Web Engineering 2004 Market for lab project documents.

See additional examples in other patterns that use markets.

Evaluation

In ***Web Engineering 2004***, students rated the *relevance* of online document sharing facilities with a mean value of $M_{\mathrm{R}} = 4.0$ on a scale from 1 = *very low* to 5 = *very high*. On the same scale, the overall *quality* of the Market as depicted in Figure 123 was rated with a mean value of $M_{\mathrm{Q}} = 3.94$. Note that this value showed high differences in mean values of students of different instructors (ranging from 3.45 for Instructor 5 to 4.38 for Instructor 1.)

MEETING

Package: *Interactive Elements*

Intent

Use meetings for face-to-face interaction and for preparation and conclusion of online phases.

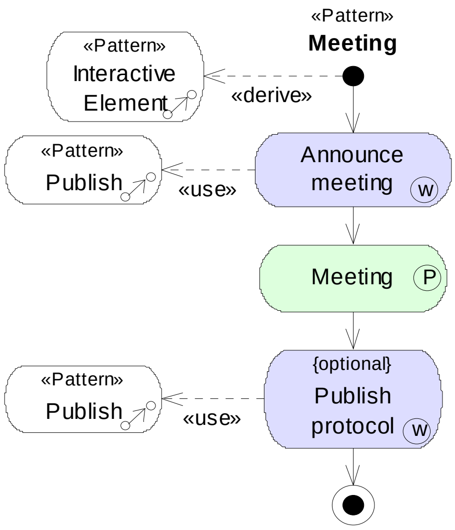
Motivation

Meetings are INTERACTIVE ELEMENTS that may be used for several purposes in blended learning settings, such as:

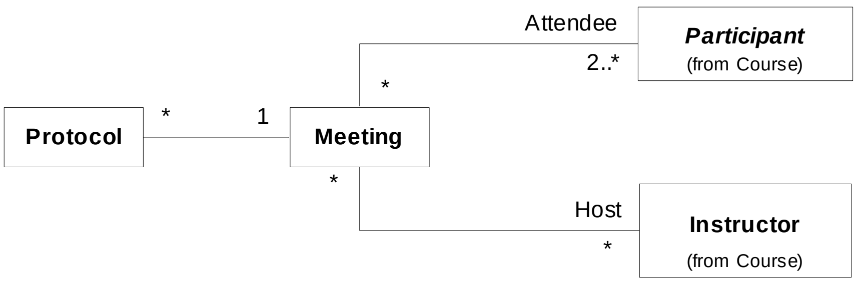
- ***Lecturing***: Especially in traditional educational scenarios, lecturing is one of the main forms of face-to-face delivery of information to learners.
- ***Interaction***: This is, or better should be, one of the main purposes of an educational meeting. Learners and instructors congregate to interact productively and to proceed in the learning process.
- ***Initiation*** of a blended learning process by the preparation of online phases: Requirements, assignments, and tasks to be done in online phases have to be prepared accordingly. In addition to publishing relevant information online it is useful to convey the intent of following online phases to participants.
- ***Conclusion*** of online phases: Alternatively, instead of preparing for online phases, meetings can also be seen as "finishers" of online phases in blended learning scenarios, depending on viewpoint and concrete scenario employed. Meetings finishing online phases may be used to discuss, exchange experiences, and to reflect on issues evolved during the preceding online phase. In other cases, meetings are explicitly the focal part of the blend, e.g., the presentations meeting in PRESENTATION PHASES.

A concrete scenario for the meeting itself is not described here, as this has to be specified by users of this utility pattern. However, as meetings take place at certain locations and time slots, this information is usually published prior to the meeting. After the meeting, it may be appropriate to publish a meeting protocol to provide a briefing and a lookup resource and furthermore, means to catch up for anyone who could not attend the meeting.

Sequence

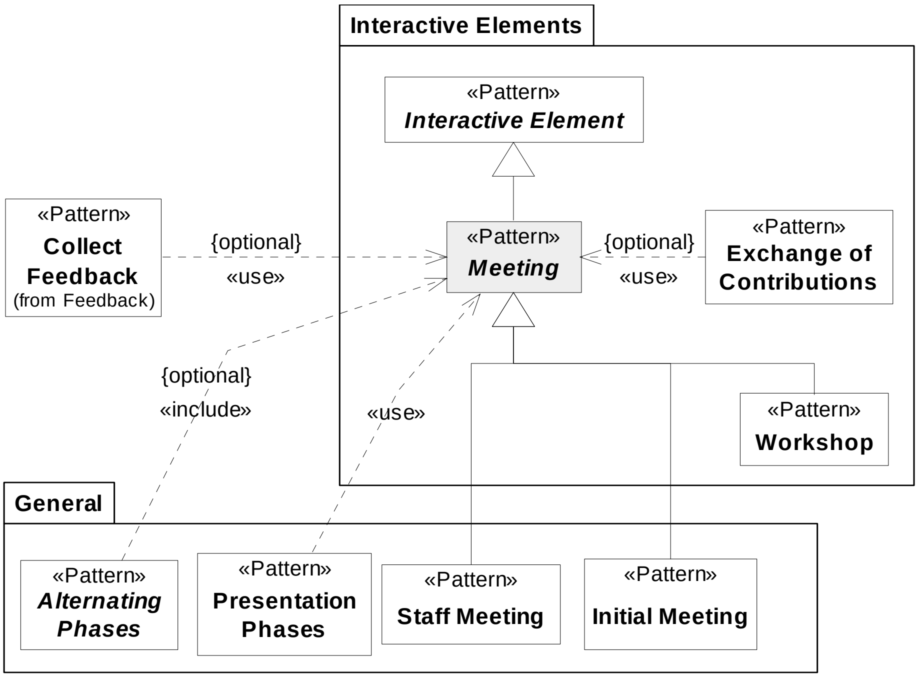


Structure



A meeting is *attended* by at least two *participants*, and potentially *hosted* by a subset of the participants. Optionally, meeting *protocols* may be created for each meeting.

Taxonomy/Dependencies



MEETING is located in the *Interactive Elements* package. It derives from INTERACTIVE ELEMENTS, as interaction between meeting participants takes place. There are a number of more specialized patterns describing meetings: As an informal STAFF MEETING, as a kick-off PRELIMINARY MEETING, and as a practice-oriented WORKSHOP or TUTORIAL. Additionally, MEETING is used in the sequences of COLLECT FEEDBACK as a final, optional feedback meeting, and by EXCHANGE OF CONTRIBUTIONS and ALTERNATING PHASES blended with online phases.

Parameters

- Primary pattern author: Michael Derntl
- Primary pattern source: Research Lab for Educational Technologies (University of Vienna)
- Pattern categories: Utility, Generic
- Level of abstraction: Medium
- Scope: Activity
- Primary presence type: Present
- Flexibility: High
- Application effort: High
- Level of expertise required: Low
- Input: Preparation of meeting agenda, invitations,
- Output: Meeting protocol (optional)

Web Template

Not available. This pattern does not include any central Web-supported activities. For the meeting announcement and protocol dissemination phases, simple means of PUBLISHING suffice.

Examples

Not available.

Evaluation

Inapplicable.

ONLINE DISCUSSION

Package: *Interactive Elements*

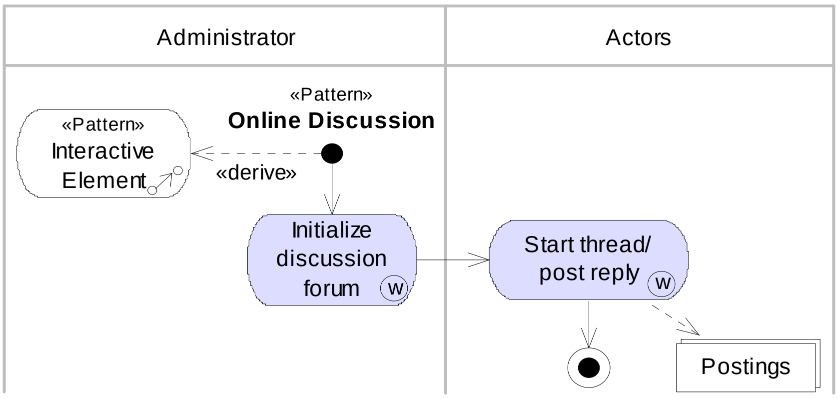
Intent

Provide facilities for asynchronous online communication among participants, instructors, tutors, and/or guests.

Motivation

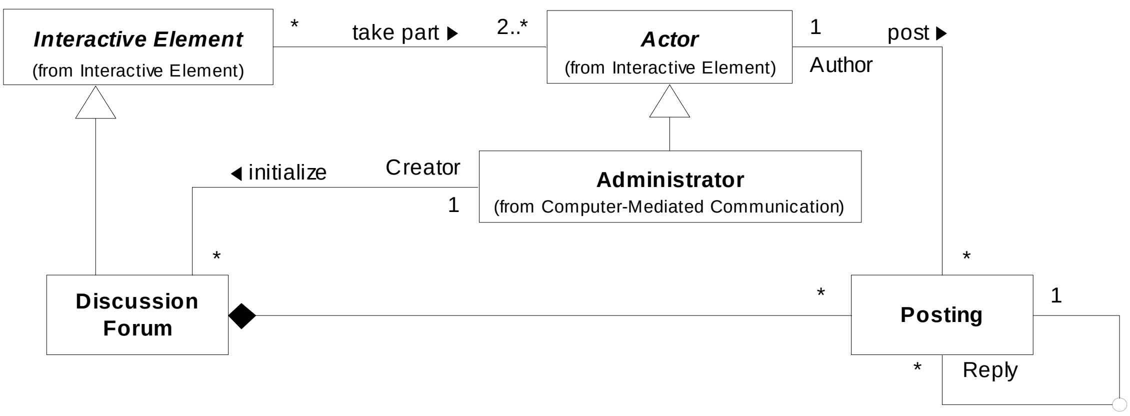
For many practitioners, aside from providing online content, employing ONLINE DISCUSSION forums is basically the entrance to blended learning or e-learning. In COMPUTER-MEDIATED COMMUNICATION (CMC), ONLINE DISCUSSION can be considered the asynchronous sibling of CHAT. For sustaining meaningful online communication activity, discussion forums are usually linked to specific tasks, documents, learning activities, or other interactive scenarios where communication is central or useful to achieve a goal. Discussion forums can also be used for general questions and organizational or administrative issues, e.g., as described in CONSULTATION. As a complement, task- and topic-independent discussion forums may be provided for open exchange. As such, these online forums may be used for communication and cooperation between distant participants, e.g., when face-to-face MEETINGS are impossible. Additionally, online discussion may be employed for initiating contact with distant experts, guests, or students.

Sequence

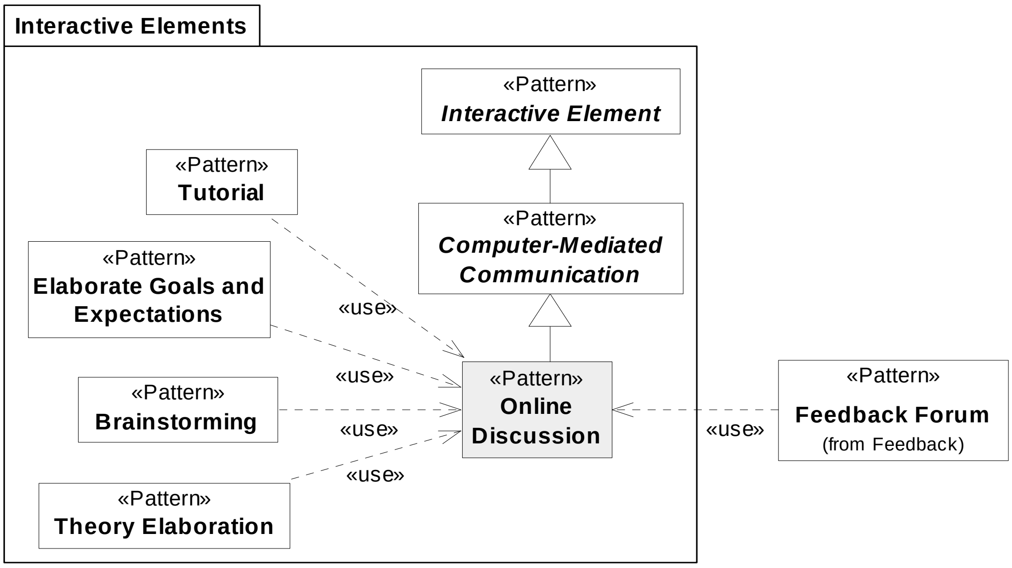


Activity	Description
Initialize discussion forum	A discussion forum is initialized along with instruction, guidelines and rules for usage.
Start thread/post reply	Depending on the concrete form of ONLINE DISCUSSION, participants may freely start new discussion threads, post messages, and reply to existing messages.

Structure



Taxonomy/Dependencies



Parameters

- Primary pattern author: Michael Derntl
- Primary pattern source: Research Lab for Educational Technologies (University of Vienna)
- Pattern categories: Utility, Generic, Collateral, Motivational, Administrative
- Level of abstraction: Low
- Scope: Activity, Phase
- Primary presence type: Online
- Flexibility: Low
- Level of confidence: 5
- Number of participants: Unrestricted
- Application effort: Low (open discussion) or Medium (moderated discussion)
- Level of expertise: Low (open discussion) or Medium (moderated discussion)
- Suggested assistance: Administrator, Tutor
- Target skills: Communication, Collaboration, Problem solving
- Input: Discussion forum, guidelines and rules
- Output: Populated discussion forum

Web Template

As the Online Discussion pattern is used as a means of Computer-Mediated Communication (CMC) by many other patterns, it should be designed flexibly enough to allow for different asynchronous CMC scenarios.

Administration View

The administration view of Online Discussions requires the following configuration steps:

- *General configuration*: Heading and introduction (step 1; see Figure 124).
- *Forum list*: Each Online Discussion space may host several different forums, each acting as a main topic thread. The forum list is configured using a tabular display for creating, editing, and removing forums (step 2; see Figure 125).
- *Forum options*: When editing an existing forum, or adding a new forum, the following configuration items are available for that forum (see Figure 126):
    - *General*: *Caption* and *introductory text*
    - *Status*: Flags that indicate whether that forum is currently *active* (i.e., usable) and *visible* in the forum list in the participant view.
    - *Usage type*: A choice whether this forum is open to all participants or restricted to be used by participants of groups/ instructors.
    - *User rights*: Are users allowed to *create threads*, to *edit messages* they posted (if yes, in what time span), and/or to post *anonymously*?
    - *Notification*: Specifies whether the message author receives e-mail notification when someone posts a reply to his/her message. Note that, depending on the learning platform solution employed or envisioned, notifications may also be sent via some platform-internal messaging system or via *RDF Site Summary* (RSS)[552] feeds.

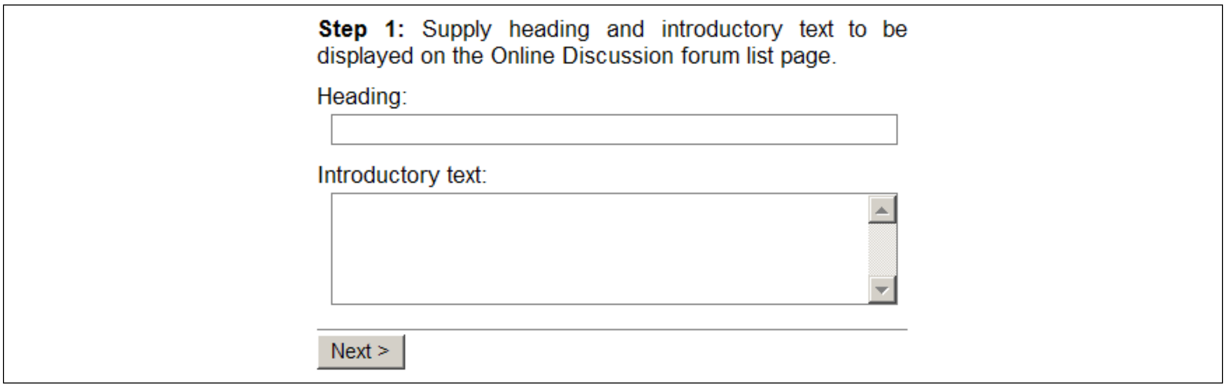


Figure 124: General configuration of Online Discussion.

[552] RSS-DEV Working Group (2001)

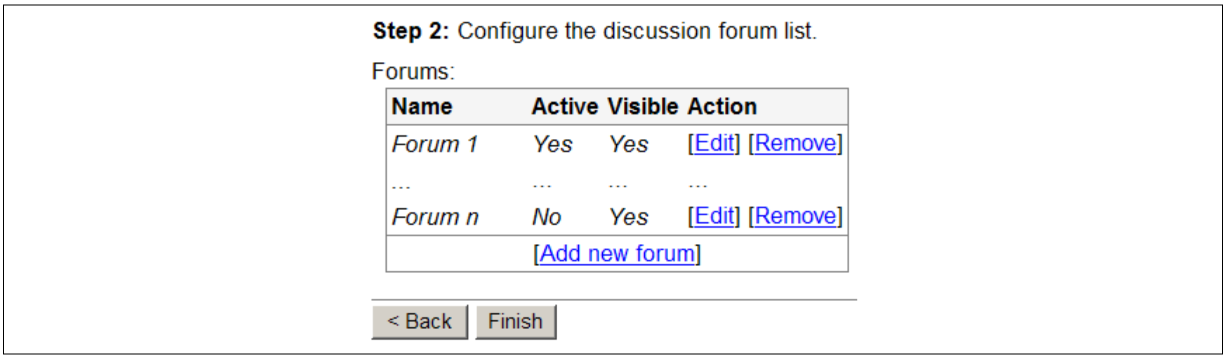


Figure 125: ONLINE DISCUSSION forum list configuration.

Note that clicking *edit* or *add new forum* leads the user to the forum configuration page, which is depicted in the following Figure 126. Also note that the remove option should only be available for empty forums, i.e. forums in which no messages were posted yet. Removing populated forums is more safely achieved by just making them *invisible* in the participant view.

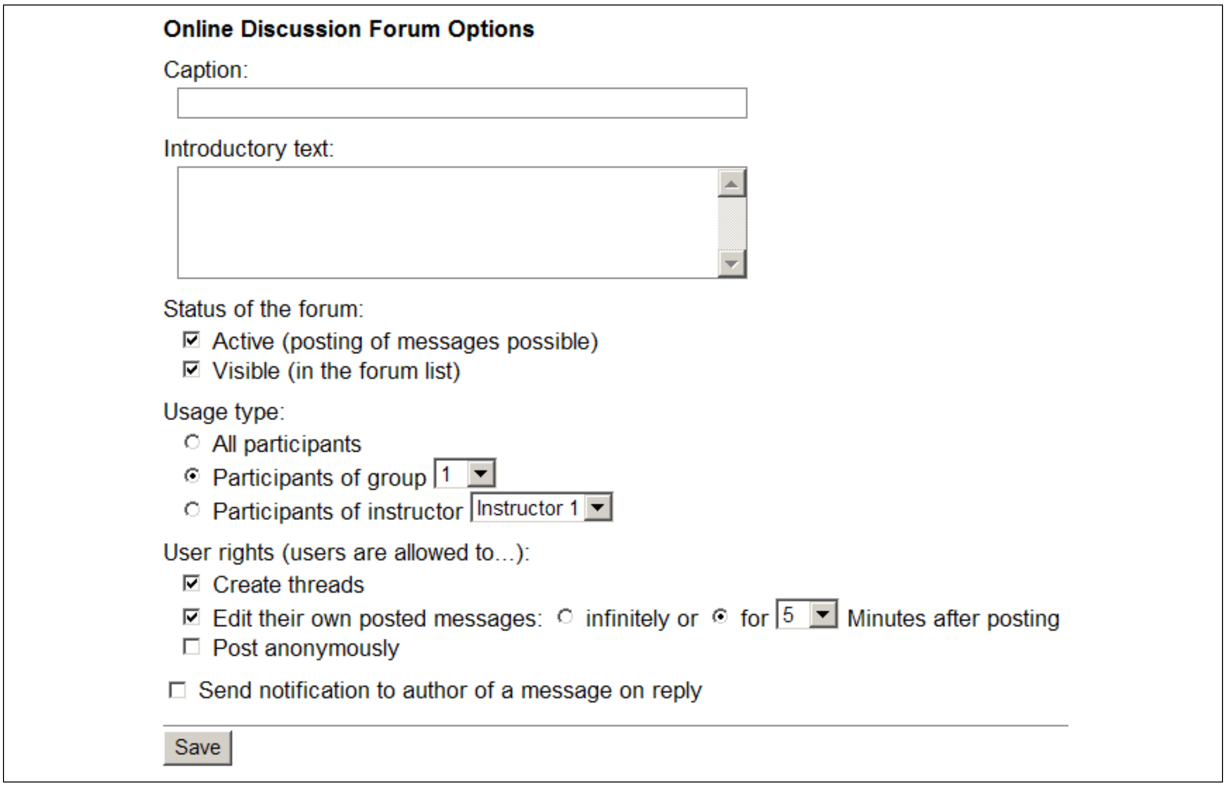


Figure 126: ONLINE DISCUSSION forum options.

Clicking the *save* button returns control to the forum list configuration page (Figure 125), and updates the forum name in the table there (only if the name was modified here).

Participant View

The participant view occupies three pages:

- *Main entry page* (Figure 127): This page shows the general heading and introduction to the online discussion (see administration step 1), as well as a list of configured forums (see administration step 2). Note that if only one single forum has been specified, it might be more useful to directly forward the user to that forum's page.

- *Forum overview* (Figure 128): This page shows the forum-specific caption and introduction as well as all message threads within the current forum. On top of the page, the currently selected message (if any) is displayed along with links to edit the message and to reply to that message.
- *Message posting form* (Figure 129): This form is used for posting a message/reply to the forum using a message *subject* and a message *body*.

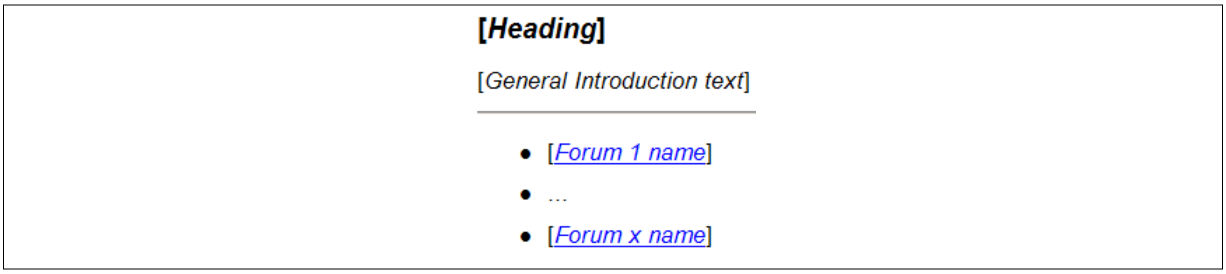


Figure 127: ONLINE DISCUSSION forums main page.
See also the example in Figure 131.

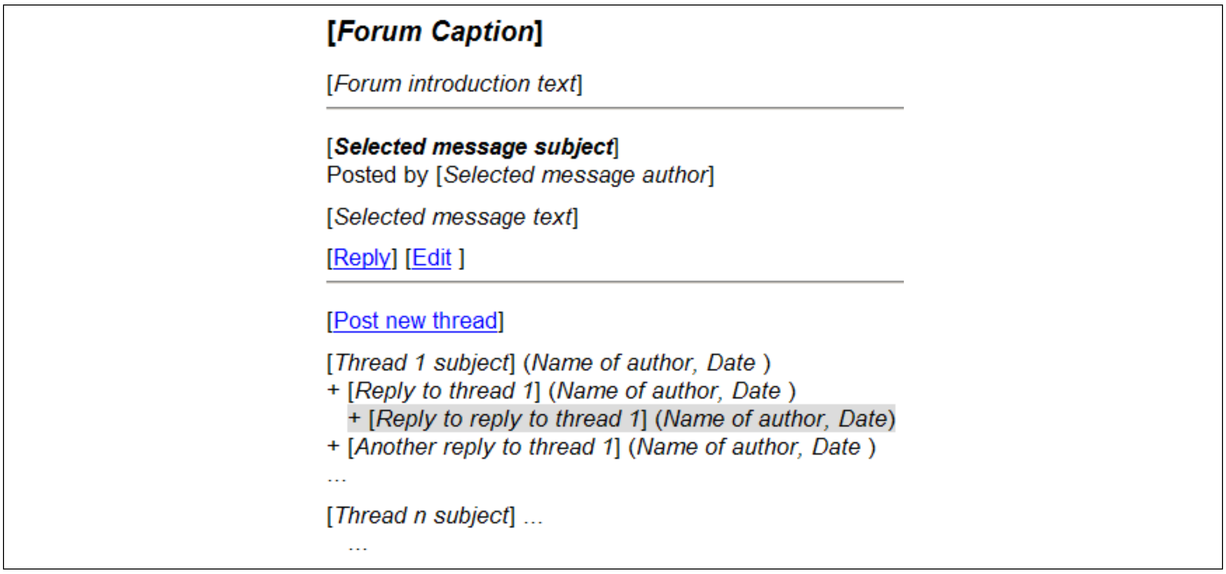


Figure 128: ONLINE DISCUSSION forum view.
Note that the selected message is marked with light gray background in the thread view. Clicking the "*reply*" link for that message, as well as clicking on "*post new thread*" redirects the user to the message composition page, which is depicted in Figure 129. See also the example in Figure 133.

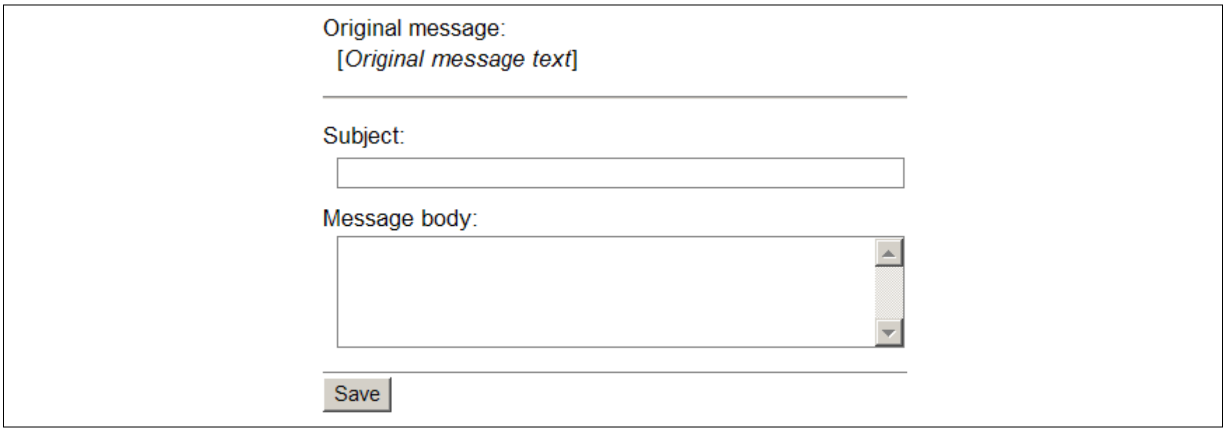


Figure 129: New/edit message form in ONLINE DISCUSSION.
Note that the original message text is only displayed when this form is reached via the "reply" link. When posting a new thread, no original message is available. See also the example in Figure 133.

Report View

Generally, the pages in Figure 127 and Figure 128 of the participant view may serve as the main report sources for discussion content. Additional reports may be provided for more detailed reports, for example:

- ***Overall statistics***: Overview for all participants regarding number of messages posted, number of replies received, etc. This report should allow for filtering the statistics by participant.
- ***Detailed report***: Including all threads/messages per forum; allow for filtering by participant. See an example below in Figure 130.

Ummeldungsmöglichkeit (**Derntl Michael** - 28.03.2005, 19:48)
Aufgrund der ungleichen Verteilung meiner Übungsgruppen (17 Teilnehmer in Gruppe 3, und 8 Teilnehmer in Gr. 4) biete ich den Teams in Gruppe 3 an, sich in Gruppe 4 (Mi 15:30-17 Uhr) umzumelden. Diese Ummeldung würde ich übernehmen, dh es wäre Ihrerseits sonst kein Aufwand erforderlich. Bei Interesse melden Sie sich bitte bei mir.

Nächste Übung ([redacted] - 27.03.2005, 18:08)
Sehr geehrter Herr Derntl,
auf der WE-Seite steht, dass die nächste Übungseinheit am 30.03. stattfindet. Soweit ich weis sind zu diesem Zeitpunkt aber noch Ferien.
Ist dieser Übungstermin korrekt?
Mfg [redacted]

\+ Nächste Übung (**Derntl Michael** - 28.03.2005, 17:30)
Der 30.3. im Bild war ein Fehler meinerseits. Ich dachte ich hatte ihn schon korrigiert, dürfte aber nicht funktioniert haben ;) Natürlich ist in den Ferien keine Übung, die nächste Einheit ist am 6.4.!
Sorry für die fehlerhafte Info
LG

ORACLE Datenbank ([redacted] - 23.03.2005, 17:05)
Sehr geehrter Herr Derntl, Michael
Ich besuche die WI/WE Übung bei Ihnen am Mittwoch (14:00 15:30) und heiße [redacted]
Heute habe ich versucht mich in den ORACLE Datenbank auf dem ALMIGHTY einzuloggen, aber leider ohne Erfolg.
Um mit unserem Teamprojekt weiter zu kommen brauche ich Zugang zu dieser Datenbank.
Ich habe vor einem Jahr die Übung IT/SK Künstliche Intelligenz gemacht und damals haben wir die selbe ORACLE Datenbank benutzt. Es ist möglich, dass ich damals mein Passwort geändert habe und jetzt nicht mehr weiß wie es lautet. Können Sie mir bitte irgendwie helfen?
Ich danke Ihnen vorzeitig.

\+ ORACLE Datenbank (**Derntl Michael** - 23.03.2005, 17:21)
Wie bereits angekündigt wird als Datenbanklösung heuer ausschliesslich SQLite angeboten. Oracle Datenbank gibt es heuer nicht.
Der Grund ist simpel:
* Die PHP Anbindung für SQLite ist die einfachste (weil mit wenig Befehlen viel erreicht wird, genauso wie bei allen PHP Datenanbindungen). Siehe Beispiel unter [http://almighty.pri.univie.ac.at/~mangler/sql.txt], sowie Dokumentation unter [http://at.php.net/manual/de/ref.sqlite.php]

Figure 130: Example of a detailed report (excerpt) from my groups' Web Engineering forum.

Examples

Online discussions have already been used in almost all courses for different scenarios, such as for:

- Organizational and administrative issues
- Discussion on lectures, course modules, and learning activities
- Various EXCHANGE OF CONTRIBUTIONS and PRESENTATION PHASES scenarios in PhD seminars
- Collect students' requirements for learning platforms and posting error reports for the learning platform used
- Moderated forums for feedback on specific tools, courses, and learning activities (see FEEDBACK FORUM)
- ELABORATING GOALS AND EXPECTATIONS for a course
- Open communication among participants
- *etc.*

Evaluation

In ***Web Engineering 2003***, online discussion forums were mainly used for open exchange and discussion of organizational and administrative issues. Discussion with peers was a highly appreciated learning aspect of the course (M = 4.20 on a scale with a maximum value of 5). However, the perceived support that the learning platform provided for online discussion was valued rather low (M = 1.85, SD = .94). This seems to suggest that the technical realization of discussion forums is highly important for supporting exchange and discussion among participants. This argument is supported by the fact that in the subsequent year 2004 – when the home-grown CEWebS platform replaced the commercial platform that was used in the year before – the discussion forum facility was valued much higher ($M = 3.36$, $SD = 1.15$) than in 2003. A two-sample t-test shows that this mean value is significantly higher than the year before ($p < .001$). The major reason for this increase is presumably the fact that the general usability and particularly the discussion forum usability of the 2004 platform were much better than that of the 2003 platform in terms of navigation, ease-of-use, simplicity, and clarity.

The following figures show examples for the different pages in the participant view of Online Discussion in Web Engineering ***2004***.

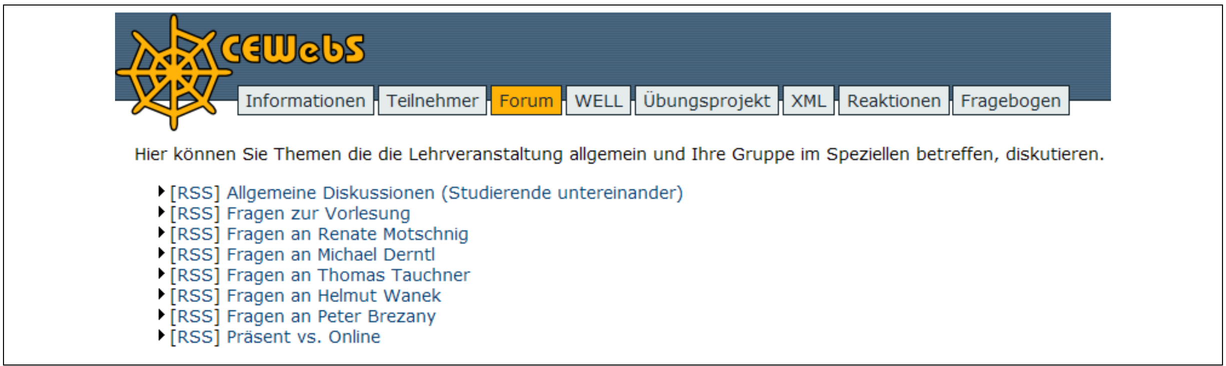


Figure 131: Example ONLINE DISCUSSION forum list.

Note that for each discussion forum, as separate RSS feed is offered, where anyone can subscribe to. New postings are published in the RSS feed, and subscribers are automatically notified.

Figure 132: Example ONLINE DISCUSSION thread/message view.

Figure 133: Example ONLINE DISCUSSION message composition form.

References

RSS-DEV Working Group. (2001). *RDF Site Summary (RSS) 1.0 Specification* [Online]. Retrieved Mar 29, 2004, from http://web.resource.org/rss/1.0/spec

PROBLEM PROPOSALS

Package: *Interactive Elements*

Intent

Let participants choose and solve problems of personal or particular professional interest to make learning processes more self-initiated, more authentic, and learning effects more persisting.

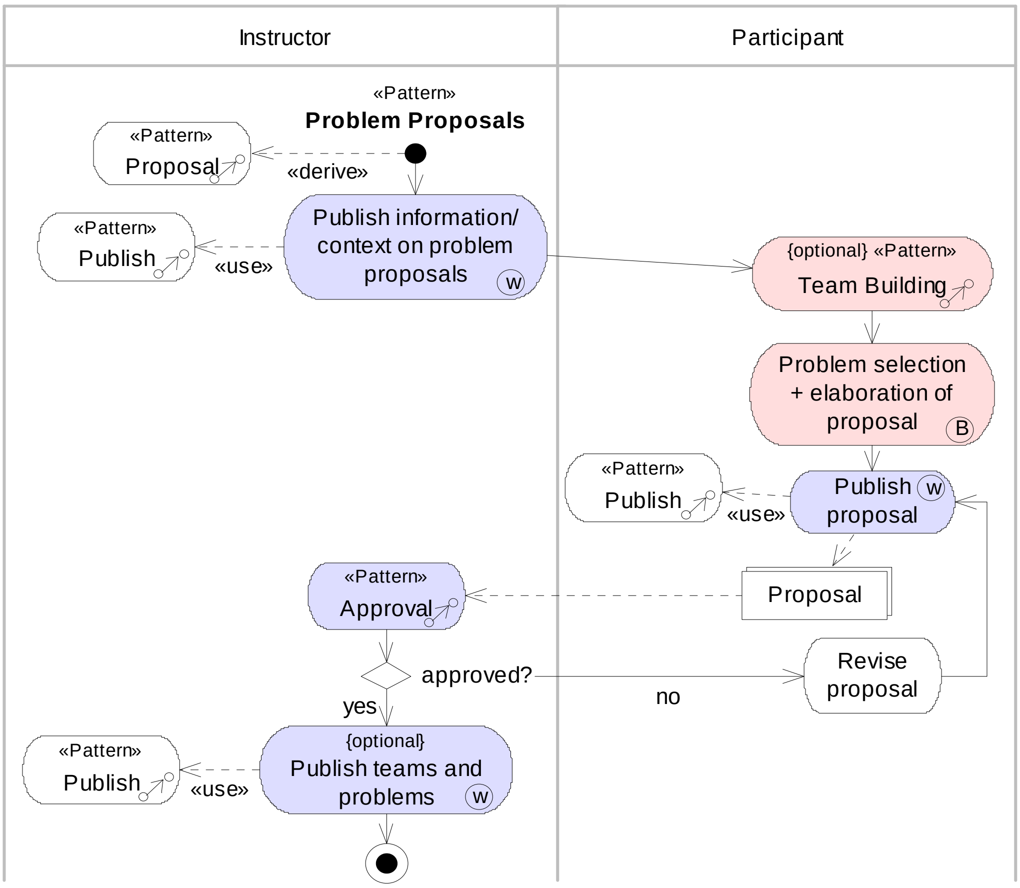
Motivation

The traditional assignment-based system has shown to be not very motivating for students, as all assignments are preset by the instructor, and there is no way for participants to bring in their personal interests and/or experience. Giving the participants a certain degree of freedom of choice facilitates more active involvement and dedication of participants in the problem-solving process (see parent pattern PROPOSAL) and approximates whole-person learning more closely.

Therefore, in a Person-Centered setting, assignments set solely by the instructor are minimized. Instead, the participants are provided with some problem context (e.g., a software project employing the Rational Unified Process) as well as with relevant content and resources, and may propose their own specific problem of interest.

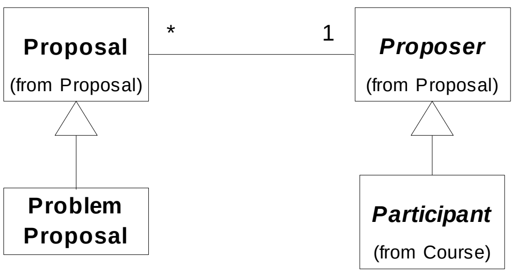
The term *problem* implies a problem-solving process and shall refer to generic tasks, so it was chosen deliberately for the name of the pattern.

Sequence

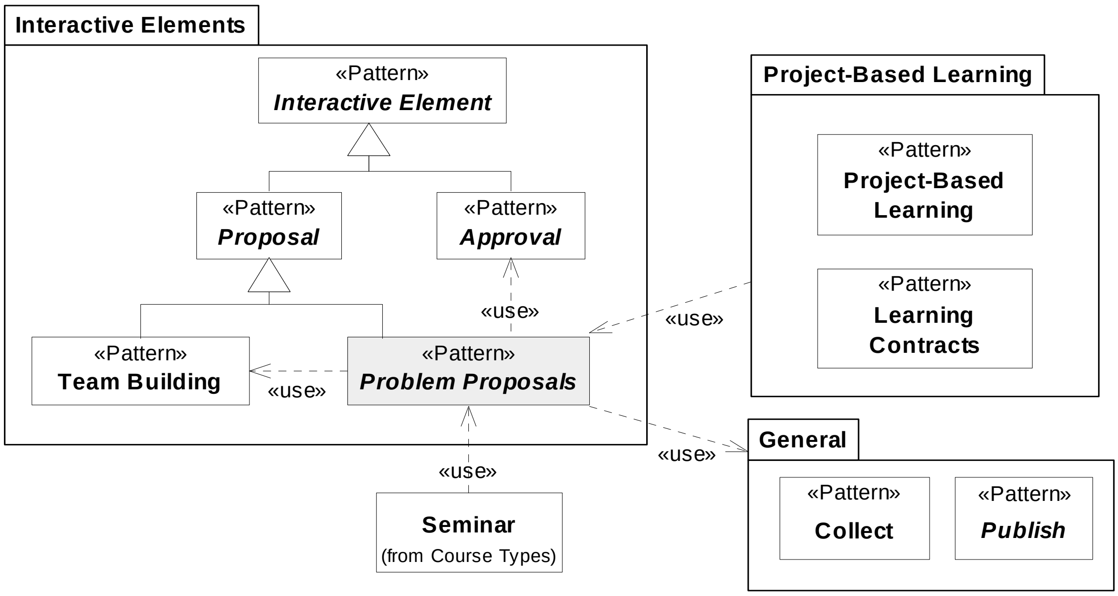


The sequence is analogous to that of parent pattern PROPOSAL, so no special description is provided here.

Structure



Taxonomy/Dependencies



Parameters

- Primary pattern author: Michael Derntl
- Primary pattern source: Research Lab for Educational Technologies (University of Vienna)
- Pattern categories: Utility, Generic, Motivational, Administrative
- Level of abstraction: Medium
- Scope: Activity, Phase
- Primary presence type: Blended
- Flexibility: High
- Level of confidence: 5
- Number of participants: up to 30
- Person-Centered variables addressed: Acceptance, Transparency, Understanding
- Application effort: High
- Level of expertise: High
- Suggested assistance: Tutor; Expert
- Target skills: Problem solving, Negotiation, Communication, Collaboration, Creativity
- Input: Problem context and proposal guidelines
- Output: Approved problem proposals

Web Template

Inherited from PROPOSAL.

Examples

Examples include:

- Collecting proposals for projects, where the project process model and subject context is given, and the specific purpose/subject of the projects are proposed by participants (individually, in teams, or in the whole group) – see PROJECT-BASED LEARNING.
- Problem proposals may also be applied for elaboration of contributions or reports, where topics can be chosen within a thematic scope that is relevant to the course or learning activity (refer to SEMINAR for a usage example).

Evaluation

Not available.

PROPOSAL

Package: *Interactive Elements*

Intent

Use PROPOSAL and subsequent APPROVAL scenarios in contexts where participants are encouraged to freely choose or to propose, for example in PROBLEM PROPOSALS or TEAM BUILDING.

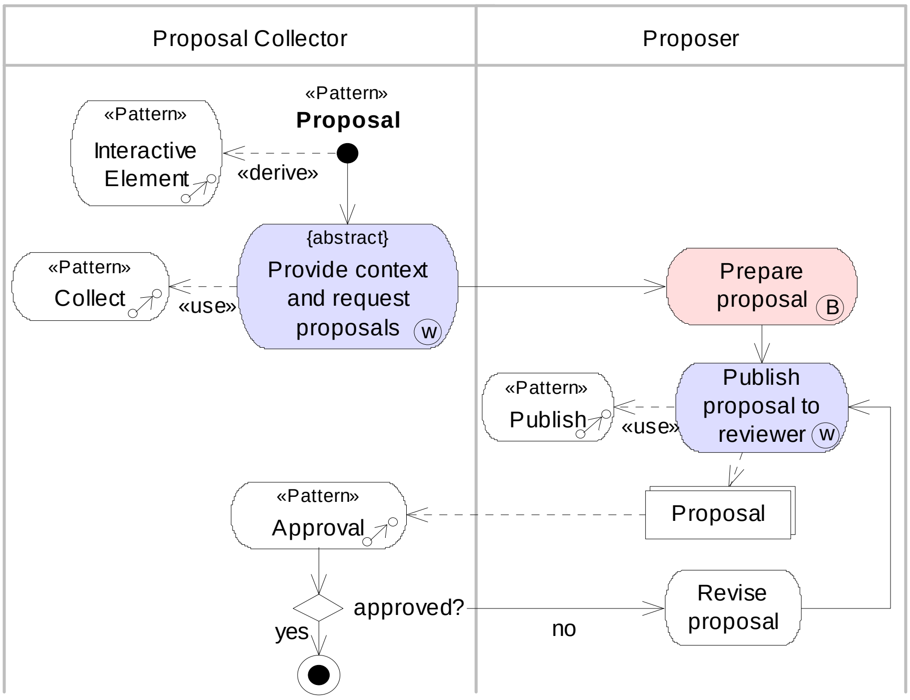
Motivation

Proposal is a pattern that, if set in action properly, perfectly embodies Person-Centered attitudes in any scenario where learner-initiated tasks have to be accomplished, or contributions have to be elaborated by letting participants choose their specific topics of interest within a given thematic context (situational *authenticity*). A certain degree of freedom of choice allows the participants to select topics in which they are intrinsically interested, and thus higher motivated to contribute and participate actively.

On the side of the instructor this scenario requires as well as allows expressing all Person-Centered attitudes. Acceptance and understanding are involved in letting participants choose their particular topics of interest, openness, and transparency are especially required in the APPROVAL of proposals, as typically not all proposals formally and/or thematically meet the requirements. In such a case the instructor should ask questions, clarify ambiguities, and point out problems from his or her perspective. Consecutively, a transparent statement of the project proposal serves as the foundation for a successful project.

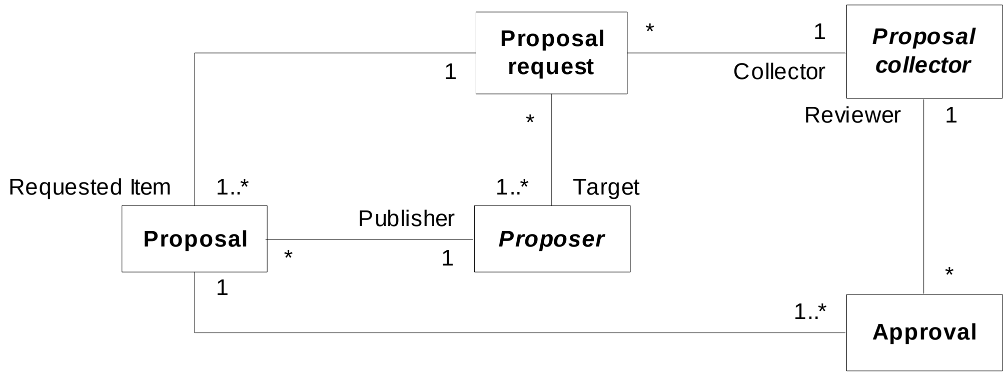
Proposal is an INTERACTIVE ELEMENT and provides a simple, generic flow: First, the proposal collector (usually the instructor) publishes the context in which proposals are solicited. This comprises guidelines for proposals and thematic context, if appropriate. Subsequently the proposal is PUBLISHED to the collector by the proposers (usually participants), and finally approved by the collector, who is then acting as a reviewer.

Sequence



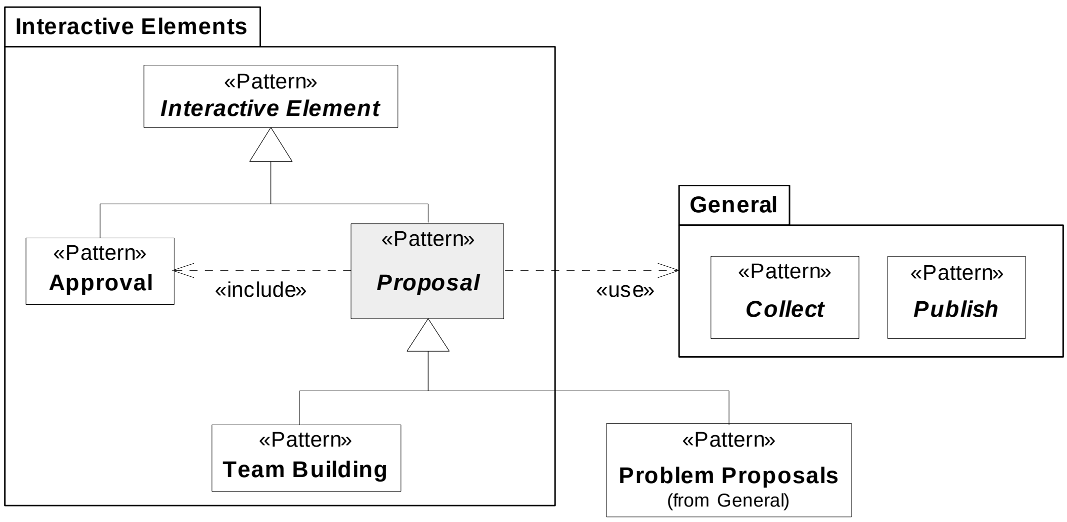
Activity	Description
Provide context and request proposals	This activity uses COLLECT by initially issuing a proposal request to potential proposers, whereas context, guidelines, and deadline for proposal submission are set by the collector (in most cases the instructor). The collector may provide some predefined choices and relevant content/resources to support participants in creating proposals.
Prepare proposal	Proposers prepare their proposals according to the context provided by the proposal collector.
Publish proposal to reviewer	Prepared or revised proposals are published to the collector/reviewer for approval.
APPROVAL	
Revise proposal	If the APPROVAL outcome was negative (i.e., the proposal was rejected), the proposal is revised.

Structure



Proposals are published by *proposers* in response to a *proposal request* that is issued by the *proposal collector*. The collector acts as a reviewer in *approval* scenarios for each proposal. The entities involved in approval are sketched in detail in the APPROVAL pattern.

Taxonomy/Dependencies



PROPOSAL derives from INTERACTIVE ELEMENT, as interaction between proposal collector/reviewer and proposers takes place in the form of a request/response scenario as generically specified by COLLECT. PROPOSAL depends on APPROVAL and vice versa, as any proposal scenario without approval as well as any approval scenario without prior proposals is useless. Specialized forms of PROPOSAL are TEAM BUILDING, where team proposals are collected from participants, and TOPIC PROPOSALS, where participants propose their own topics of interest within a certain thematic scope.

Parameters

- Primary pattern author: Michael Derntl
- Primary pattern source: Research Lab for Educational Technologies (University of Vienna)
- Pattern categories: Utility, Generic, Motivational, Composite

- Scope: Activity, Phase
- Level of abstraction: High
- Person-Centered variables addressed: Acceptance, Transparency, Understanding
- Primary presence type: Blended
- Target skills: Communication, Collaboration, Problem solving
- Input: Proposal context and guidelines, proposals
- Output: Approvals

Web Template

This pattern is mainly composed of three other patterns, namely COLLECT for issuing proposal requests and subsequent PUBLISHING of proposals by participants, as well as a concluding instance of APPROVAL. As the scope of potential concrete proposal scenarios is almost unlimited, and the nature of the aforementioned patterns is generic, it is impossible to define one single, comprehensive Web template here. However, we can make suggestions here what the proposal request should encompass:

- A clear outset of what is the *expected content* of the proposals. For example, when proposing project topics, this can be a simple vision statement for the project itself. On the other hand, as evident from the TEAM BUILDING Web template, the proposals may also implicitly unfold as a result of a more complex, cooperative process. In such a case it is sufficient to outline the process rules.
- Clear guidance on what the *criteria* for approval or rejection are. For example, minimum and maximum team sizes in TEAM BUILDING.
- If there is some broader context for the proposals, it can be useful to provide some *examples* from that context for participants. It is also possible to provide a number of predefined proposals that participants can just adopt, if they feel reluctant to propose their own problems or solutions. For example, in a LEARNING CONTRACTS scenario the instructor may predefine a number of topics of interest to choose from.
- Links to related *resources*, if appropriate.

Examples

PROPOSAL is used in any scenario where participants work in teams, as team members should never be assigned, but proposed by participants. Additionally, a specialized form of PROPOSAL is used whenever participants are free to choose topics of personal interest in a certain context. More concretely, proposal has been used in the following contexts:

- In ***Web Engineering*** for TEAM BUILDING and proposing LEARNING CONTRACT topics (TOPIC PROPOSALS).
- In ***Project Management*** courses for projects and/or thematic proposals for reports and presentations.
- In the ***PhD Seminars*** for proposing topics of reports and presentations.

Evaluation

Not available.

Team Building

Package: *Interactive Elements*

Intent

In teamwork scenarios, let participants choose their team partners. Restrict only team size to about 2 – 5 members, as appropriate for the current learning activity.

Motivation

Many learning activities are accomplished through collaborative teamwork, especially in complex tasks such as projects. Working together collaboratively in teams has the potential of more adequately resembling real-life situations in most of today's businesses. Solitary work in separated chambers is not quite what is expected by managers, as a recent study in Austria revealed: according to this survey, personnel managers value social skills, teamwork, and communication skills of graduates much higher than professional skills and factual know-how[553]. Additionally, work in small teams is a proven option to further self-initiated, active learning as well as constructive personal development through mutual support and information exchange, collaborative elaborations, reciprocal inspiration, and social contact with peers[554].

However, the first step on the way to teamwork is building a team. Following the Person-Centered style, participants are free to propose their preferred team composition rather than being assigned with randomly instructor-chosen peers. Depending on the task or problem to be tackled, small teams of about 2–5 members are particularly suited to further collaboration, problem-solving, and communication skills, while keeping organizational and communication overhead within acceptable boundaries. By deriving from PROPOSAL, this pattern allows participants to propose their team compositions, while the instructor takes over the role of a supervisor, assisting in the process when needed.

Note that, as described in TEAM WORKSPACES, teams are "linked" to learning activities, as building teams without any task assignments is needless.

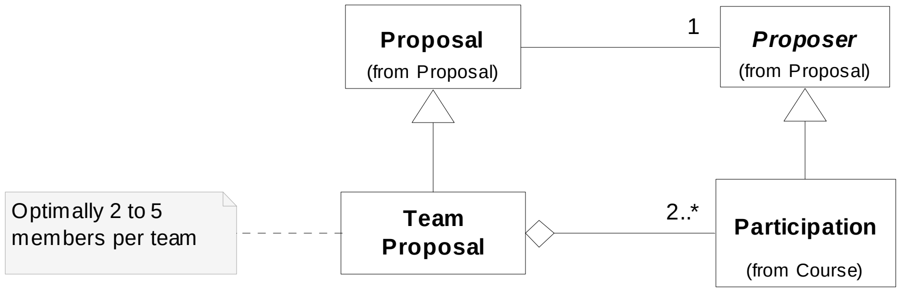
Sequence

Inherited from PROPOSAL.

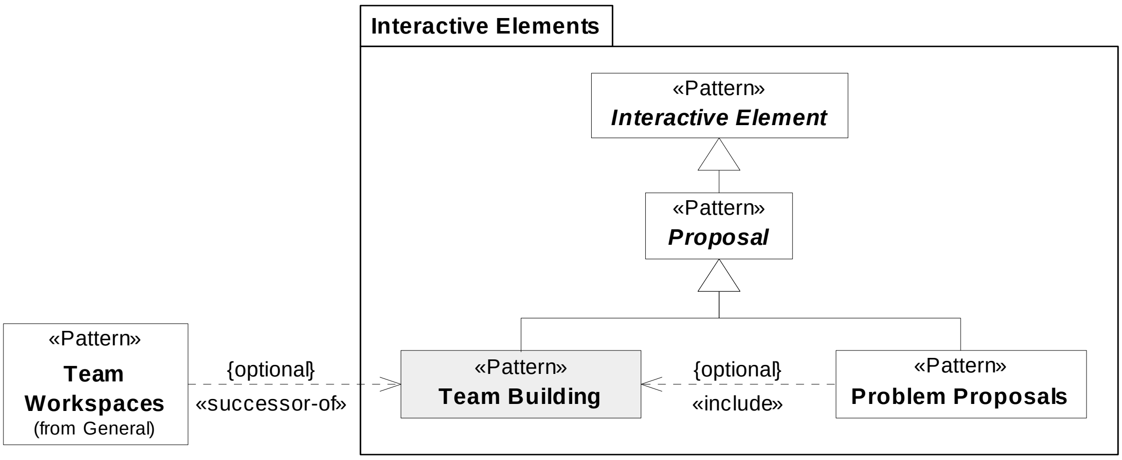
[553] Motschnig-Pitrik (2002a)

[554] Tausch and Tausch (1998, p. 253-264)

Structure



Taxonomy/Dependencies



Parameters

- Primary pattern author: Michael Derntl
- Primary pattern source: Research Lab for Educational Technologies (University of Vienna)
- Pattern categories: Utility, Collateral, Motivational, Administrative
- Level of abstraction: Low
- Scope: Activity, Phase
- Primary presence type: Blended
- Flexibility: High
- Level of confidence: 4
- Number of participants: Unrestricted
- Person-Centered variables addressed: Acceptance
- Application effort: Medium
- Level of expertise required: High, Medium, Low
- Suggested assistance: Expert, Administrator, Tutor
- Target skills: Technical skills, Interpersonal Skills, Communication, Collaboration, Problem solving, Practical skills
- Input: Team building guidelines and expected team size; Maximum and Minimum team size; Deadline for team building

- Output: Team constellations; Approval by instructor

Web Template

Figure 134 shows a diagram depicting the basic use cases that are supported in this Web template.

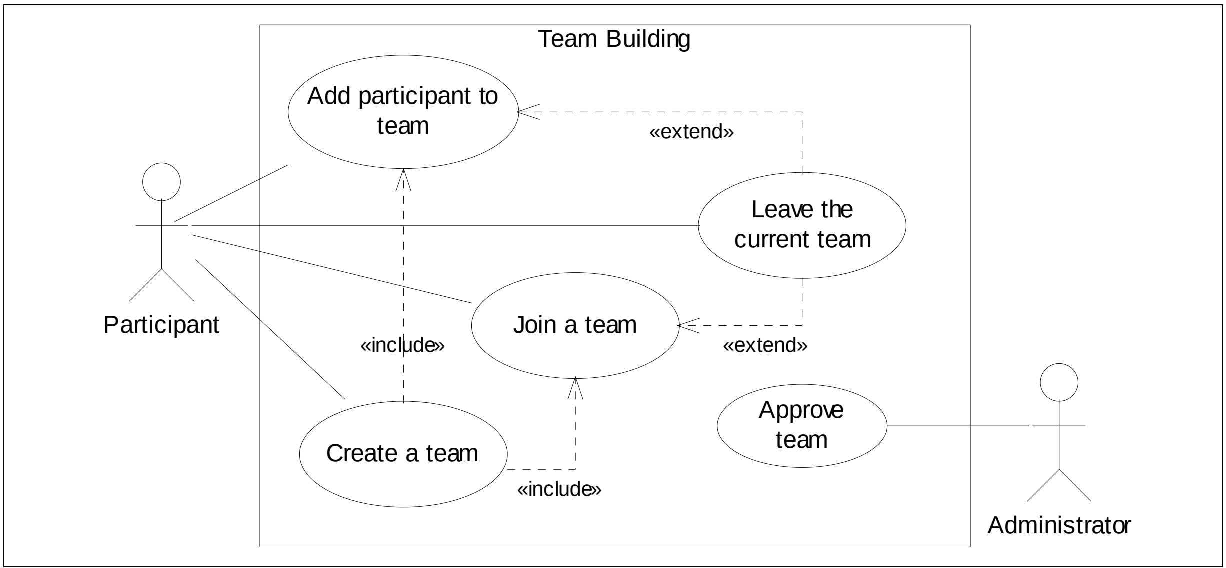


Figure 134: Use cases in the TEAM BUILDING Web template.

Participant View

The Web template for the team building process is very simple, as there are only a small number of scenarios for course participants:

A) ***Join a team***: Any participant that is currently not member of any team may join a team. Basically, this can be done in two ways:

   1) Selecting another participant from the *pool* of available participants – i.e., all participants that are currently not members of any team – to create a new team (cf. use case '*Add participant to team*' in Figure 134).

   2) Joining an existing team.

B) ***Create a team***: The fastest way of building a team is letting one participant set up the complete list of team members. This is just a minor variation of use case *A*: The current participant (who has to be already member of a team) may select additional participants from the pool to join his or her team.

C) ***Leave the team***: Any participant that is currently member of a team may leave that team by (re)joining the participant pool.

The above cases can all be projected on one single Web page, whereby there are many different options for visual presentation. Adhering to simplicity, the page template consists of three major sections (see Figure 135):

1) *General information* on the team building process, as well as a contextual help text depending on the current status of the participant (i.e., team member or pool member).

2) The *participant pool*, which is just a list of participants that has not yet joined any team. Thereby, the pool is amended with a hyperlink that allows any current team member to rejoin the pool by clicking on it (i.e., leaving his or her current team as described in case *C* above). Additionally, each participant in the pool is shown as a hyperlink that executes the actions described in cases *A1* and *B* above, depending on the status of the currently logged in participant (i.e., the *current user*):

   a) If the current user is not yet member of any team, clicking on a participant in the pool will create a new team including the selected participant from the pool as well as the current user (see case *A1*).

   b) If the current user is already member of a team, the pool participant is added to the already existing team (see case *B*). Basically, this resembles an invitation to join a team. If the new team member does not agree on joining the team, he or she may leave the team as described in case *C* above.

3) A list of currently existing teams (*team constellation*). Each team is shown as a hyperlink that, when clicked, makes the current user join that team. Potentially, this may imply that the current user leaves his or her current team.

Note: to make the consequences of any action/click more obvious, buttons or separate text hyperlinks with appropriate captions may be used instead of hyperlinks on the actual items. In Figure 135, the pool and each teams has a separate text hyperlink "*Join*", whereas participants in the pool are shown as 'normal' hyperlinks.

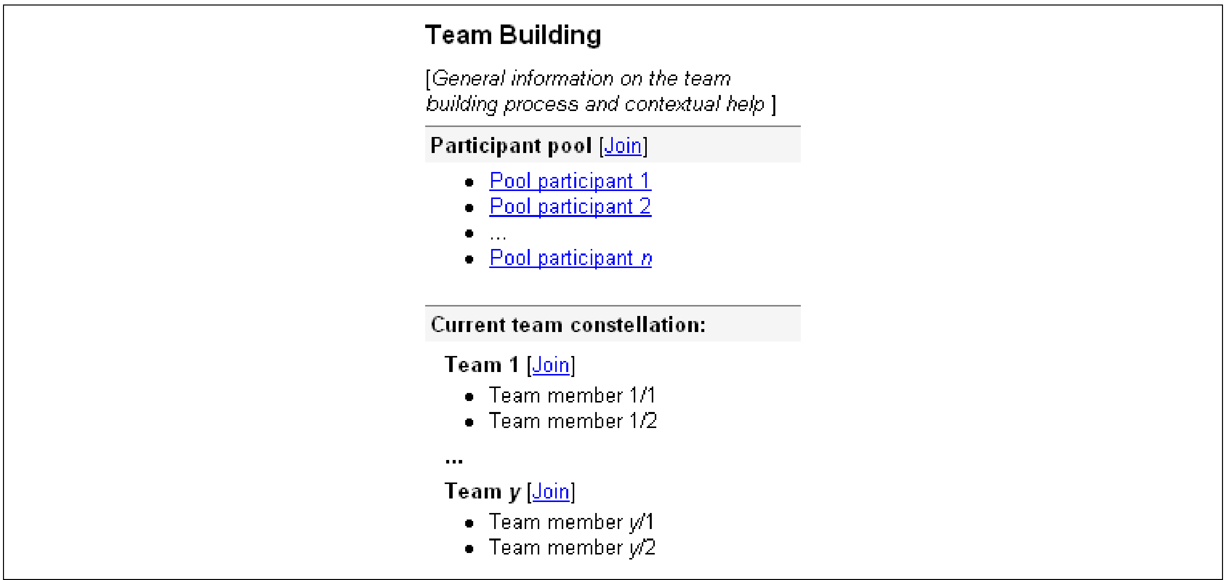


Figure 135: Participant view in the TEAM BUILDING Web template.

Administration View

As asserted in the *Parameters* section, the administration has to allow for supplying the following in the pattern initialization process (see Figure 136):

1) *General information* on the team building process as well as additional useful information such as expected *team size*.
2) *Minimum and maximum team sizes*. An administrator (typically impersonated by an instructor or tutor) can only approve a team proposal when the minimum team size restriction is met. On the other hand, a participant may only join a team or add another participant to his/her current team when that action will not exceed the maximum allowed team size.
3) A *deadline* for the team building process. However, such a deadline should not be implemented as a restriction that will generally prevent the continuation of the team building process. Rather, it could serve as the date when notifications are (automatically) sent to participants, who have not yet joined a team.

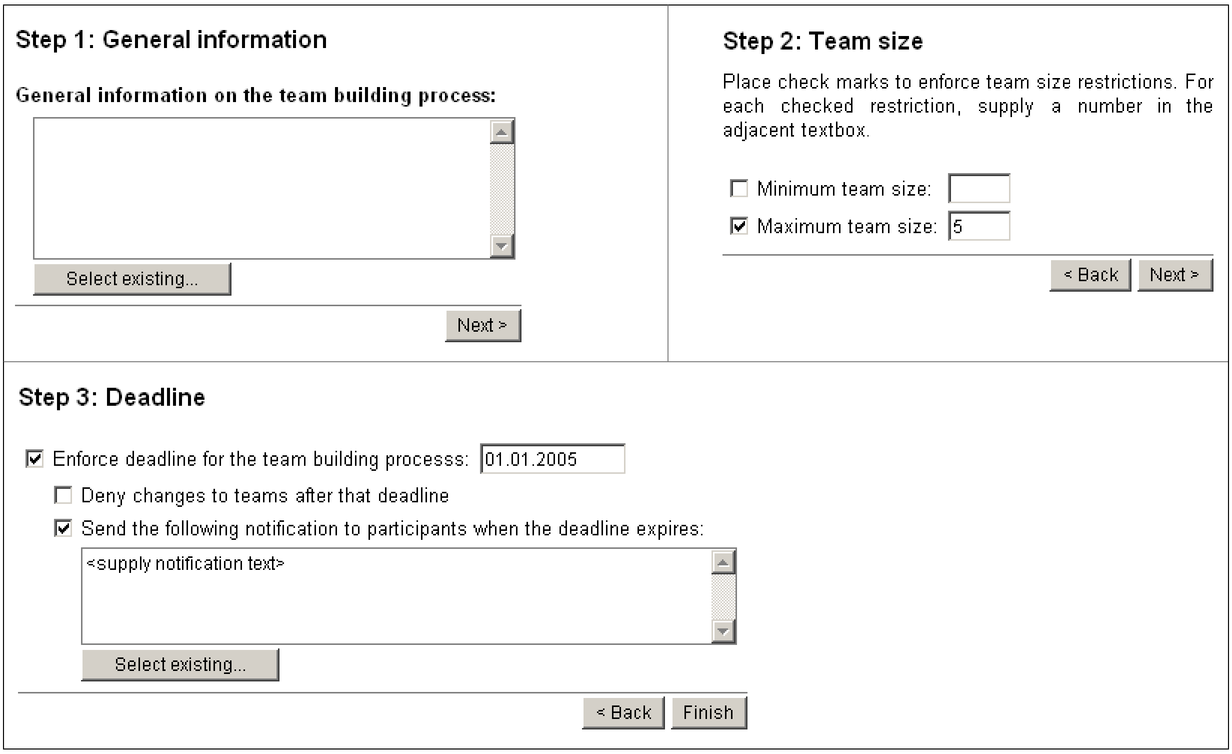


Figure 136: Configuration wizard in the TEAM BUILDING administration view.

Regarding the approval of team constellations by the administrator, the participant view is basically sufficient with one simple extension: Instead of showing the "*join*" hyperlink the administrator is presented with a hyperlink labeled "*approve*".

Report View

The report view outputs the current team constellation just like in the participant view. Additionally the report should offer the possibility of e-mail notification of participants who have not yet joined a team.

Examples

This pattern is used by a number of learning scenarios where teams perform tasks, e.g., in PROJECT-BASED LEARNING or LEARNING CONTRACT:

- In ***Web Engineering***, teams of 2 – 5 participants were built for elaborating LEARNING CONTRACTS contributions as well as for the PROJECT-BASED LEARNING scenario in the Web Engineering LAB COURSE (cf. Figure 137).
- In ***Project Management***, teams were engaged in PROJECT-BASED LEARNING and homework scenarios.
- In several other courses, teams were built for collaboratively writing reports or for elaborating and presenting theories.

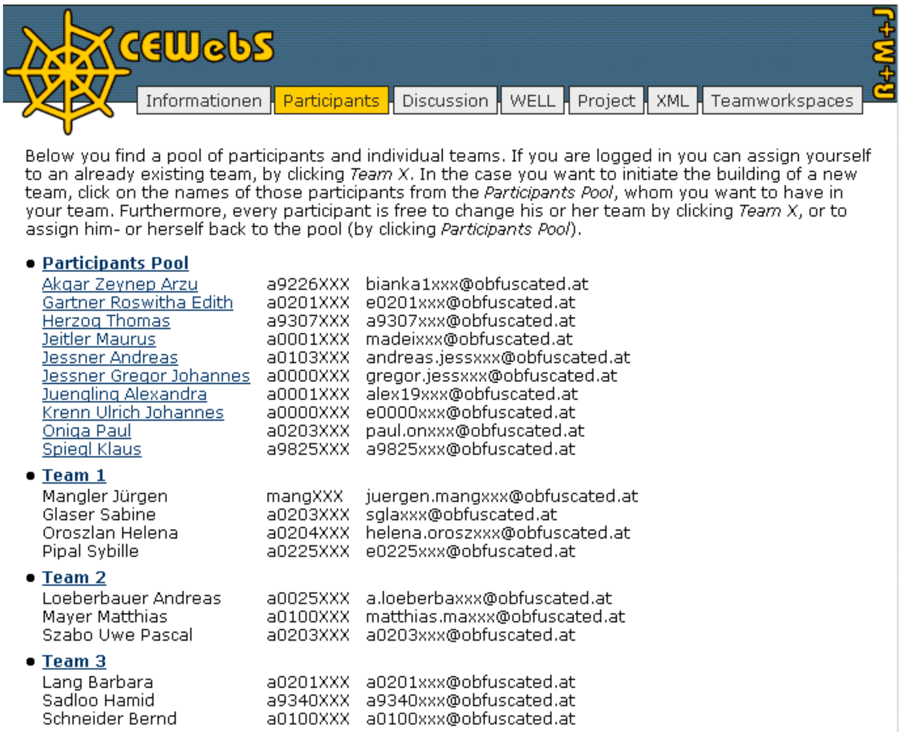


Figure 137: Screenshot from the team building page in Web Engineering 2003. Three teams have already been built (lower part), almost half of the participants is still in the participants pool (compare the *Web Template* section). Note that the page content was translated form German, and participant ID's and e-mail addresses were obfuscated.

Evaluation

During the initial application of the pattern in the ***Web Engineering*** lab courses (summer term 2004), there was neither negative feedback nor suggestions for improvement from participants or instructors. One of the Web Engineering instructors wrote that, "*I like to say that the team building module worked perfectly for me and that students managed to assign themselves to teams without any questions.*" Anyway, there were no items in the final QUESTIONNAIRE to explicitly survey participants' opinion on the TEAM BUILDING module. Possible questionnaire items for future investigations could be:

- Comparison of the online team building process with conventional team building practices (e.g., on sheets of paper). Teaching staff as well as participants could be asked for their preference [*interval scale*] with regard to the following:
  a) Ease of use
  b) Comfort
  c) Effort required
  d) Transparency
  e) General preference
- Suggestions for improvement [*open question*]

Remarks

This pattern does not specify an own *Sequence* section, as all activities are taken over from PROPOSAL.

References

Motschnig-Pitrik, R. (2002). getProfile: Anforderungsanalyse an Wirschaftsinformatiker(innen) aus der Sicht der Wirtschaft. *OCG Journal, 1*, 8-11.

Tausch, R., & Tausch, A.-M. (1998). *Erziehungs-Psychologie*: Hogrefe.

THEORY ELABORATION

Package: *Interactive Elements*

Intent

Certain (aspects of) topics or subject areas are elaborated and subsequently PUBLISHED and/or presented by participants.

Motivation

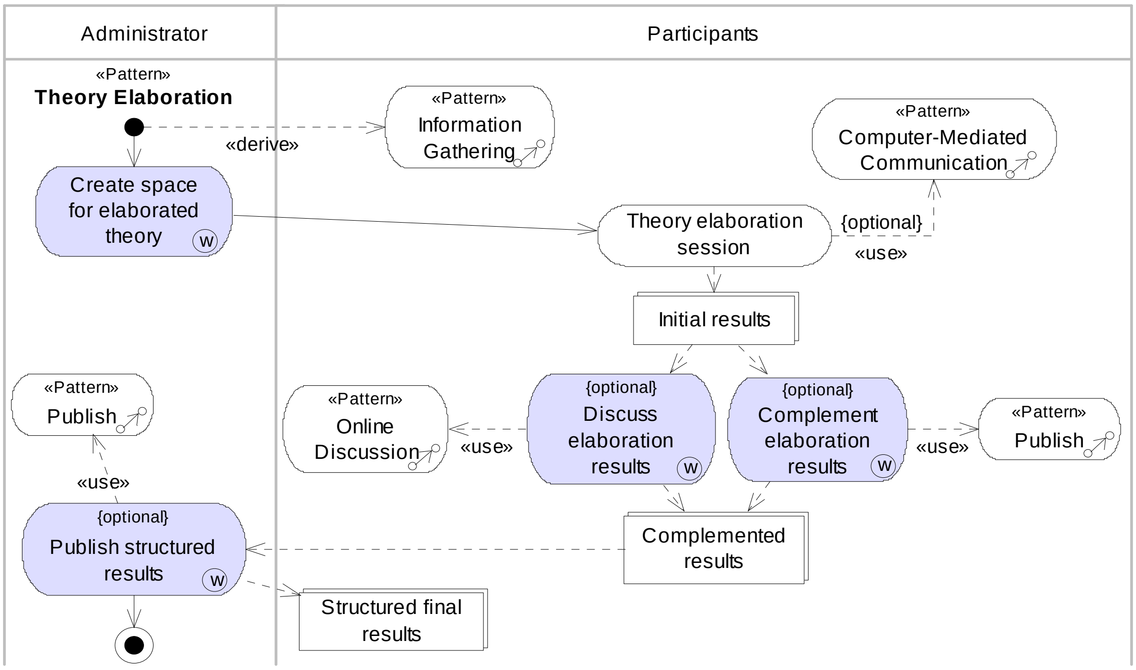
Instead of passively receiving factual information about a theory in a lecture-style manner, theories can be (co-)elaborated by participants, which requires them to explore certain subject areas on their own or in teams and thus furthers active participation and problem solving skills. The instructor just has to provide some key facts, references, (e-)content, and aspects of the theory to be elaborated.

This pattern shares quite a lot with sibling pattern BRAINSTORMING, as material and information related to the theory to be elaborated is gathered. However, the main differences are:

- THEORY ELABORATION usually spans across a longer timeframe and is conducted location-independent and may even be conducted asynchronously (e.g., in ONLINE DISCUSSION forums).
- Not *any* idea is collected, but theories and related literature are elaborated more thoughtfully and possibly cooperatively (in the sense of synchronous and managed collaboration).
- The "deferred judgment" guideline specified in BRAINSTORMING does not apply. Irrelevant facts or materials are eliminated immediately.

Most of the learnflow and general motivation of THEORY ELABORATION is already predefined in parent pattern INFORMATION GATHERING.

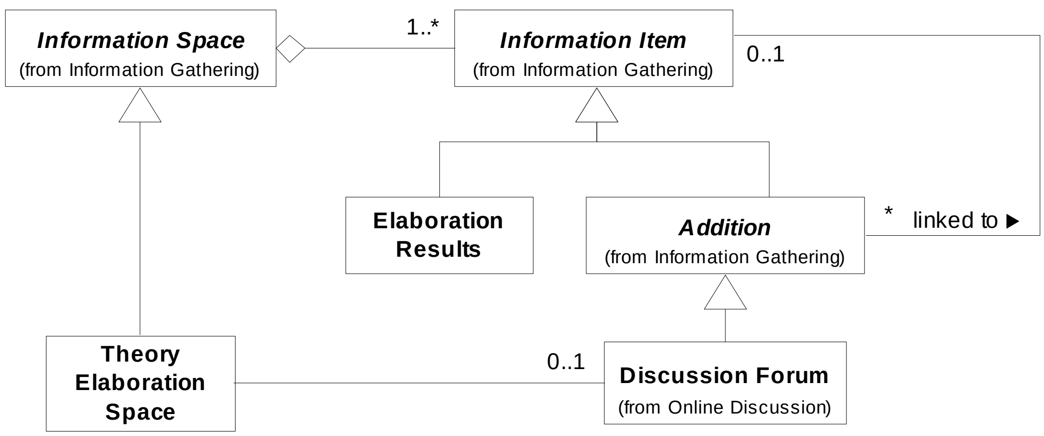
Sequence



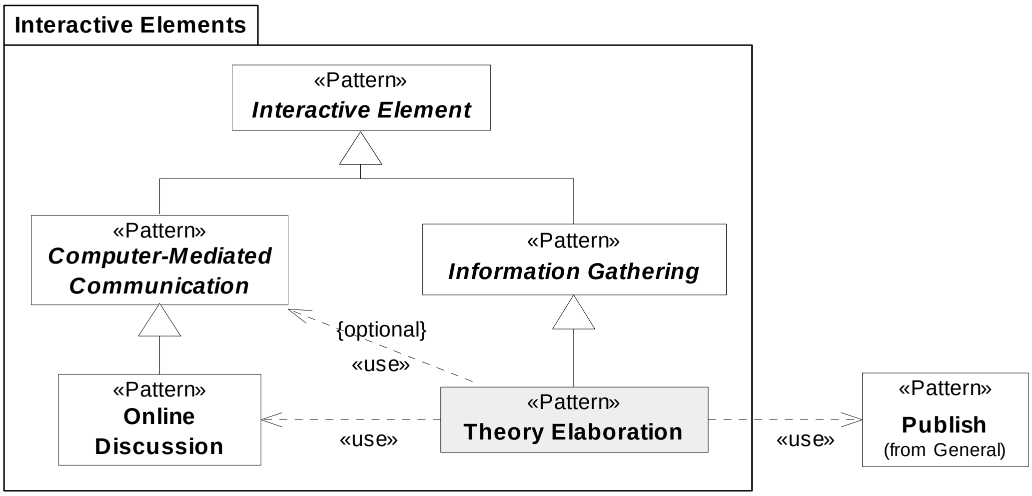
Activity	Description
Create space for elaborated theory	Depending on how theory is elaborated, space has to be provided for publishing elaborations on the platform. If the elaboration session is conducted as a moderated ONLINE DISCUSSION, no additional space is necessary. Otherwise, space (e.g., a folder or document collection) has to be initialized where elaborated

Activity	Description
	documents can be uploaded.
Theory elaboration session	Theory is elaborated as specified by the instructor and the resulting contributions are published within the allocated platform space.
Discuss elaboration results	To allow for subsequent exchange of opinions and ideas, an ONLINE DISCUSSION forum may be initiated and attached to the elaboration results.
Complement elaboration results	To enable participants to make additions to elaborations, they have to be supplied with a facility to upload additional information.
Publish structured results	Optionally, the instructor may revise the initial/complemented results and PUBLISH them as structured content.

Structure



Taxonomy/Dependencies



Parameters

- Primary pattern author: Michael Derntl
- Primary pattern source: Research Lab for Educational Technologies (University of Vienna)
- Pattern categories: Utility, Motivational
- Level of abstraction: Low
- Scope: Activity, Phase
- Primary presence type: Blended
- Flexibility: Low
- Level of confidence: 4
- Number of participants: Unrestricted
- Application effort: Medium
- Level of expertise required: Medium
- Target skills: Collaboration, Communication, Problem solving
- Input: Theory to be elaborated, guidelines and rules, optionally an elaboration space
- Output: Elaboration results and additions

Web Template

Inherited: This pattern reuses the Web template from parent pattern INFORMATION GATHERING. For simple realization of the THEORY ELABORATION scenario a MARKET with appropriate settings or TEAM WORKSPACES may suffice.

Examples

In ***Project Management/Soft Skills*** the question "*what motivates me?*" was elaborated face-to-face using moderation cards that were pinned on a board and subsequently clustered in categories. One team prepared presentation slides on the motivation topic that were discussed based on the individuals' experiences. Or another question was, "*what are the most* *important reasons for conflicts?*' Theory was elaborated in small teams with following presentation of the results on a flipchart. This was done *prior* to discussion of existing theories of conflict management by the facilitator.

In ***Web Engineering***, navigation strategies and usability issues in student-supplied, commercial Web applications were collaboratively elaborated during a lecture prior to presentation of usability and navigation theories and guidelines by the instructor.

Evaluation

Not available.

TUTORIAL

Package: *Interactive Elements*

Intent

For complex technical or application-oriented scenarios involving new or sophisticated tools and methods, let tutors do introductory technical tutorials.

Motivation

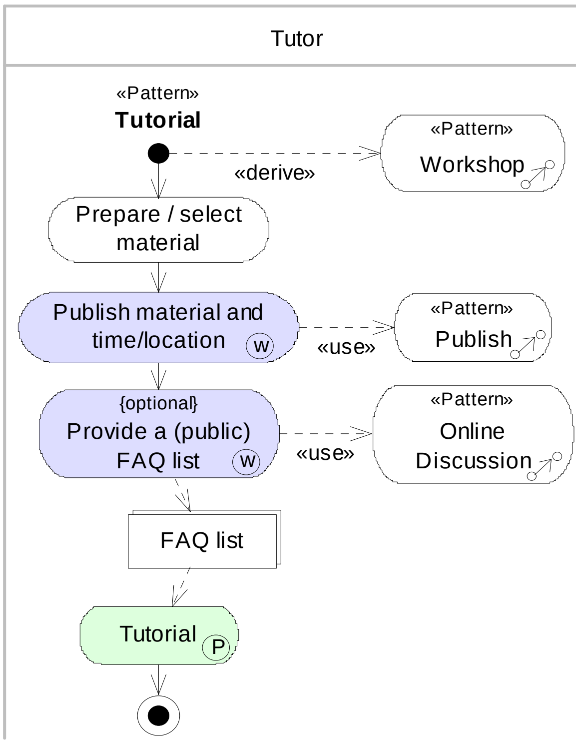
Especially (but not only) in complex, application-oriented subjects (e.g., programming languages or statistics) where practical and technical skills are essential for fulfilling the targets of the learning process, no theoretical lecture can ever replace or be as valuable as personal or hands-on experience with respective tools and methods. Tutorials, where more advanced students help or teach their peers, are excellent subsidiaries for providing example-driven introductions to complex technical subjects, making background, problems, and application scenarios perceptible to learners.

There are also fundamental personal aspects: Evaluation of tutorials has shown that technical skills are increased and that tutees tend to show "*greater confidence, more motivation to work, and an improved attitude*" toward the subject[555]. Crucial aspects of tutorials are preparation of relevant material and examples, and availability of real-world environments and tools.

In addition, it may be useful to provide collateral open tutorials, where participants can contact tutors on demand, or just to make lab facilities available to participants.

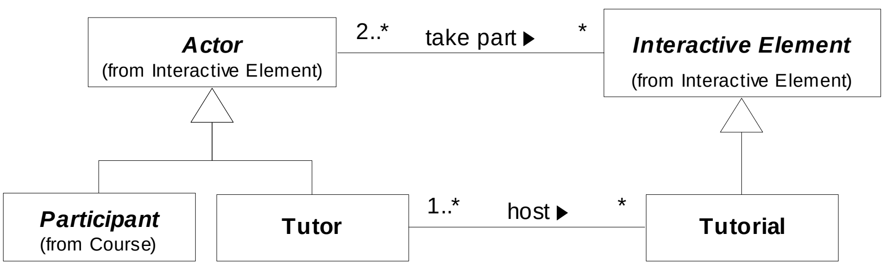
[555] Rogers (1983, p. 154); Rogers refers to tutorials as *peer teaching*

Sequence

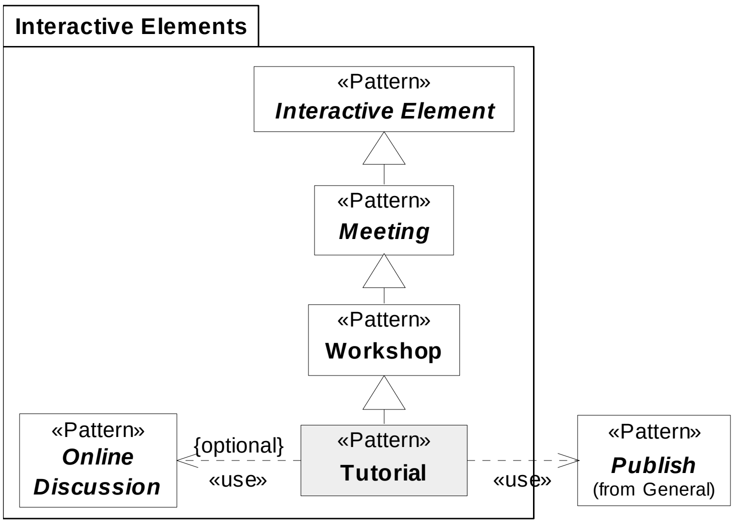


Activity	Description
Prepare / select material	The tutor selects and prepare relevant material (resources, links, examples, introductions) for the tutorial sessions
Publish material and time/location	Prepared material is PUBLISHED on the platform, along with tutorial times and locations.
Provide a (public) FAQ list	Optionally, it may be worthwhile to collect public frequently asked questions (FAQ) prior to a tutorial session. This allows the tutor to focus on specific aspects raised by participants. Collecting the FAQ may simply be done in an appropriately configured ONLINE DISCUSSION forum.
Tutorial	The tutorial MEETING.

Structure



Taxonomy/Dependencies



Parameters

- Primary pattern author: Michael Derntl
- Primary pattern source: Research Lab for Educational Technologies (University of Vienna)
- Pattern categories: Utility, Collateral, Motivational, Administrative
- Level of abstraction: Low
- Scope: Activity, Phase
- Primary presence type: Blended, Online, Present
- Flexibility: High
- Level of confidence: 4
- Number of participants: up to 30
- Person-Centered variables addressed: Acceptance, Understanding
- Application effort: High
- Level of expertise: High
- Suggested assistance: Administrator, Tutor
- Target skills: Technical skills, Practical skills, Hands-on experience
- Input: Tutorial content, resources, and samples

Web Template

The Web-based activities of this pattern can be supported by related patterns PUBLISH and ONLINE DISCUSSION. No additional Web template information is provided here.

Examples

In ***Web Engineering*** introductory tutorials were offered for practical aspects and presentation of examples in PHP[556] (a web programming language) and XML[557] (a meta-language used in web technologies). However, REACTION SHEETS revealed that participants only appreciate the tutorials when sufficient time is spent to consider questions and problems of participants.

In any other course where lab practice is involved, open tutorials (1.5 hours per week) are offered for each course group. Additionally, tutorials are used to exemplarily show the usage of required tools (e.g., usage of Microsoft® Project® in ***Project Management***, or using Rational Rose® for UML modeling in ***Software Engineering***).

Evaluation

For the Web Engineering tutorials that are mentioned in the above example, participants have provided some feedback in their reaction sheets, e.g.:

- "*The tutorials during the lab hours were very beneficial.*"
- "*The PHP tutorial took only 10–15 minutes time. The tutor seemed to be in a hurry.*" This reaction shows that it is important for the tutor to hold the tutorial in an appropriate pace and to take time to consider students' questions: "*At least, the XML tutorial was well prepared and the tutor got into the students' questions.*"

References

Rogers, C. R. (1983). *Freedom to Learn for the 80's.* Columbus, OH: Charles E. Merrill Publishing Company.

WORKSHOP

Package: *Interactive Elements*

Intent

Use workshops as application-oriented, particularly interactive MEETINGS, where the focus is on collaboration and/or sharing among participants.

---

556 recursive acronym for PHP Hypertext Processor

557 eXtensible Markup Language

Motivation

According to SIL International[558], a workshop may be defined as "*a series of educational and work sessions [where] small groups of people meet together over a short period of time to concentrate on a defined area of concern.*" The focal point in this definition from our point of view is the term *work sessions*, which implies a highly practical, "hands-on" orientation of workshops, where practical and interpersonal skills are trained, problem-solving processes are employed, and theory is put into practice. This is exactly what differentiates a workshop from a casual MEETING that may also be used for lecturing, which is definitively not practical. A workshop concentrates on practical, (inter)active, and/or collaborative cooperation and involvement of participants and instructor.

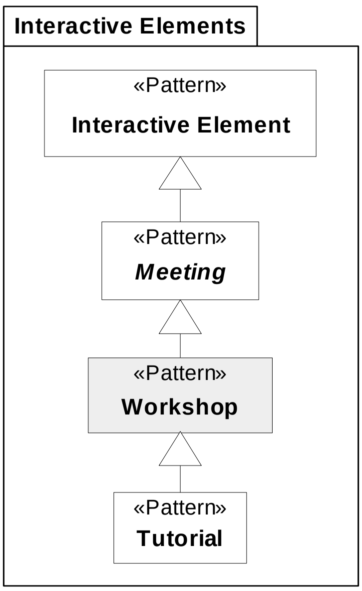
Sequence

Inherited.

Structure

Inherited.

Taxonomy/Dependencies



Parameters

- Primary pattern author: Michael Derntl
- Primary pattern source: Research Lab for Educational Technologies (University of Vienna)
- Pattern categories: Utility, Generic, Motivational
- Level of abstraction: Low

[558] SIL International (1999a)

- Scope: Activity
- Primary presence type: Present
- Flexibility: High
- Number of participants: optimally, less than 20
- Application effort: High
- Level of expertise required: High
- Target skills: Technical skills, Interpersonal Skills, Communication, Collaboration, Problem solving, Practical skills
- Input: Preparation of workshop agenda, Invitations
- Output: Workshop protocol (optional)

Web Template

Not available.

Examples

Workshops are used in any practice-oriented course: In ***Web Engineering***, ***Person-Centered Communication***, and ***Project Management***, workshops were used to present, to elaborate, and to *apply* theories of computer science, information systems, project management, education and psychology, as well as to acquire presentation and communication skills.

Evaluation

Not available.

References

SIL International. (1999). *What is a workshop?* [Online]. Retrieved Oct 27, 2002, from http://www.sil.org/linguistics/GlossaryOfLinguisticTerms/WhatIsAWorkshop.htm

5.7 Project-Based Learning

KNOWLEDGE BASE CONSTRUCTION

Package: *Project-Based Learning*

Intent

Use LEARNING CONTRACTS in a way to advance the construction of a knowledge base in a specific subject area from single contributions and knowledge fragments.

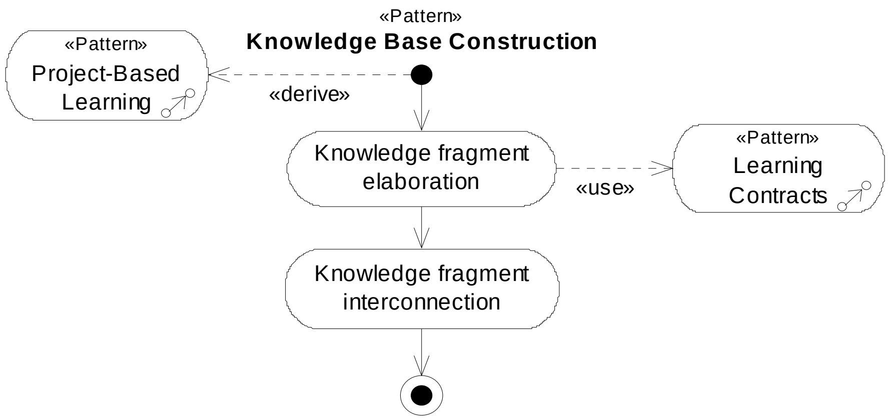
Motivation

Though there are many more or less technical definitions of the term *knowledge base* (KB), we think of it conceptually as a collection of knowledge *fragments* (e.g., participants' contributions). Typically a knowledge base is not just a loose collection of fragments, but rather organized by different criteria, topics, entities, and relations among them. Additionally, what makes knowledge bases so valuable to users is that their structuring should allow for targeted querying and locating of desired, relevant information. Employing KNOWLEDGE BASE CONSTRUCTION within an educational course setting has several advantages:

- Elaborating single contributions (= the fragments) serves a shared aim, namely to participate in the construction of a super-ordinate whole.
- Contributions of participants are not discarded after the end of the course (typically, participants' contributions end up as dust catchers in instructors' cupboards and drawers).
- The knowledge base can be made available as a valuable resource for participants of subsequent, similar courses.

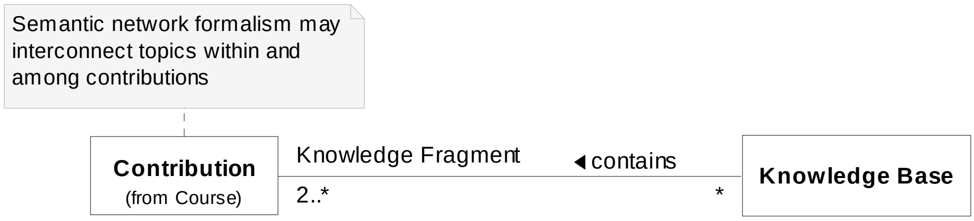
This pattern describes the process of constructing a knowledge base by interconnecting knowledge fragments contributed in a LEARNING CONTRACTS setting. However, different ways of collecting and elaborating fragments are thinkable.

Sequence

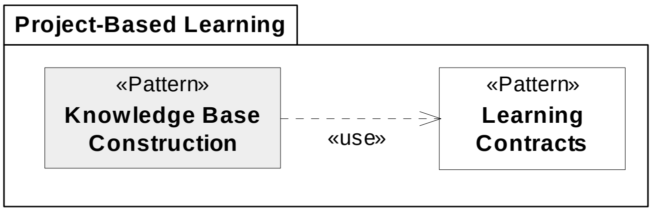


Activity	Description
Knowledge fragment elaboration	A LEARNING CONTRACT scenario is employed for letting participants elaborate knowledge fragments in a specific subject area.
Knowledge fragment interconnection	The knowledge fragments need to be incorporated or compiled into a knowledge base by interconnecting the fragments following a semantic network formalism or structuring concept.

Structure



Taxonomy/Dependencies



Parameters

- Primary pattern author: Michael Derntl
- Primary pattern source: Research Lab for Educational Technologies (University of Vienna)
- Pattern categories: Supplement, Composite, Utility
- Level of abstraction: Low
- Scope: Phase, Course
- Primary presence type: Blended, Concurrent
- Flexibility: High
  Knowledge fragments and bases can also be constructed in many other ways.
- Level of confidence: 4
- Application effort: High
  This is largely influenced by the number of LEARNING CONTRACT contributions to be considered.
- Level of expertise required: High
  This is due to inclusion of the LEARNING CONTRACT scenario.
- Target skills: Technical skills, Problem solving, Practical skills
- Input: LEARNING CONTRACT input, semantic network formalism to be used for constructing the knowledge base
- Output: Knowledge fragments, knowledge base

Web Template

Not available.

Examples

Knowledge base construction was used in ***Web Engineering***. As in 2003 was the first time this course was conducted (due to a revised curriculum), the aim was to construct an initial Web Engineering knowledge base by interconnecting individual LEARNING CONTRACT contributions (i.e., the knowledge fragments). This was achieved by investigating and applying different formalisms and methods for knowledge/information structuring, which was done by a master student after the end of the course.

Evaluation

For initial results, see the *Evaluation* section of LEARNING CONTRACTS.

LEARNING CONTRACTS

Package: *Project-Based Learning*

Intent

Let teams/participants propose topics they want to elaborate and sign contracts defining learning targets and expected contributions for each team/participant.

Motivation

The use of learning contracts is one of Carl Rogers' proposed methods of building freedom in the classroom[559]: They allow students to define and follow their own learning plans and targets while providing them with a substantial degree of both security *and* responsibility. Usually, learning contracts are employed as an alternative form of evaluation of students' achievements, while the contracts act as signed agreements about desired learning outcomes[560].

Using LEARNING CONTRACTS helps to:

- Develop and evaluate/assess those skills that shall be acquired in the course more directly than by a conventional EXAMINATION. While exams only demand passive reconstruction of previously transmitted information, learning contracts allow students to explore and elaborate areas of interest in an active, self-directed way that significantly leverages learner's motivation by developing an *inquiring state of mind*[561].
- Further deeper interest/knowledge in a special application/context of the course's subject matter.

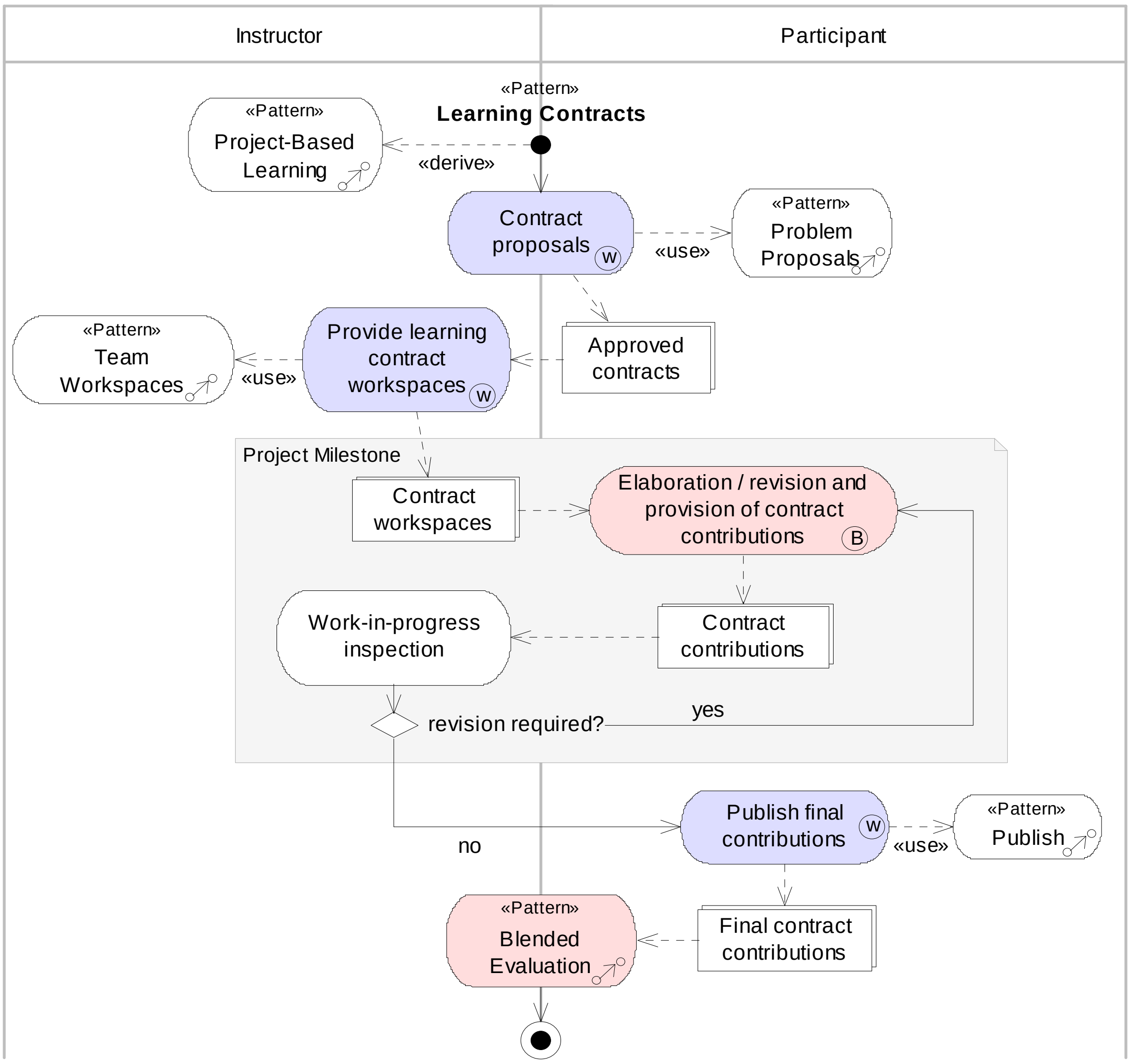
In terms of the pattern repository, LEARNING CONTRACTS describe a specific form of PROJECT-BASED LEARNING, including three major steps:

1) PROPOSAL of learning contracts by students/teams and approval by the facilitator.
2) Elaboration of contributions defined in the contracts.

[559] Cf. Rogers (1983, p. 149)
[560] See also Atherton (2003)
[561] Cf. Rogers (1983, p. 156-157)

3) BLENDED EVALUATION of contributions.

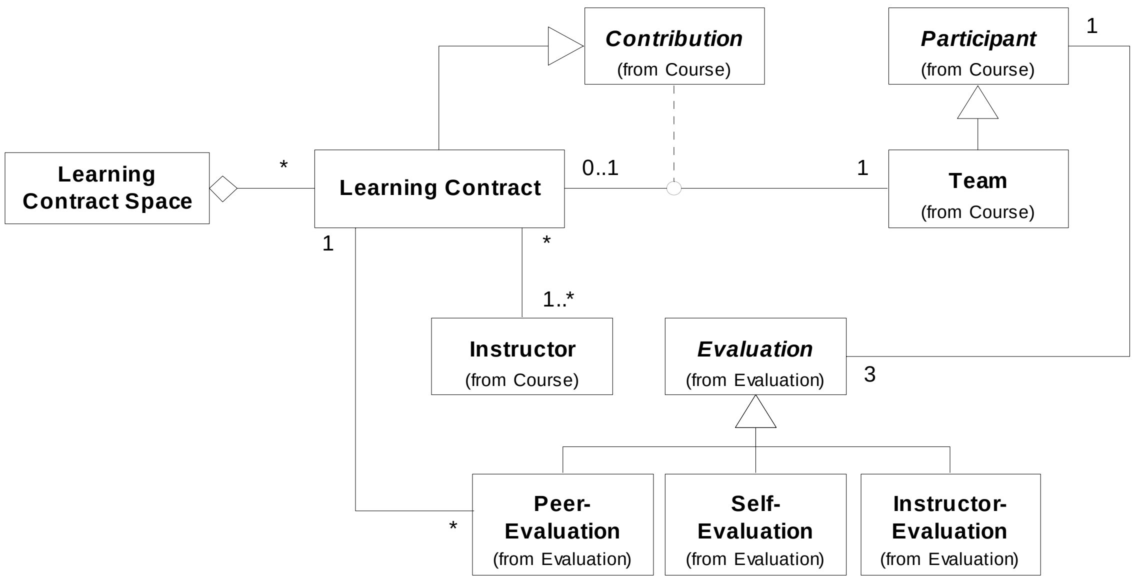
Sequence



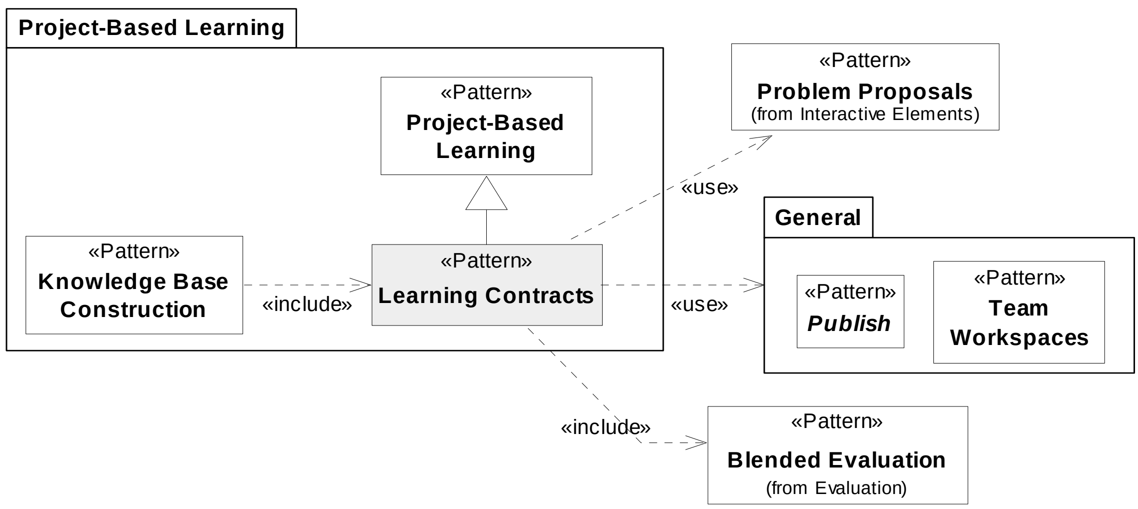
Activity	Description
Contract proposals	Uses a PROBLEM PROPOSALS scenario for collecting learning contract proposals from participants.
Provide learning contract workspaces	Each learning contractor, i.e. a person or team, is provided with a private team workspace on the learning platforms for storage and management of documents and contributions relevant to their contracts.
Elaboration / revision and provision of contracts	Participants elaborate deliverables as defined in their learning contract and publish them onto their workspaces. Following a work-in-progress inspection that revealed required modifications, this can also mean revision of existing deliverables.

Activity	Description
Work-in-progress inspection	During the work on the contracts, it is advisable to do a work-in-progress inspection to avoid major deviations from the contracts and to give the teams guidance on work that is still required.
Publish final contributions	Final contract contributions are published on the learning platform to be available to other participants.
BLENDED EVALUATION	Learning contracts are elaborated by the teams, their, peers, and by the instructor in a BLENDED EVALUATION scenario.

Structure



Taxonomy/Dependencies



Parameters

- Primary pattern author: Michael Derntl
- Primary pattern source: Research Lab for Educational Technologies (University of Vienna)
- Pattern categories: Utility, Collateral, Motivational
- Level of abstraction: Low
- Scope: Phase, Course
- Primary presence type: Blended
- Flexibility: Low
- Level of confidence: 5
- Number of participants: up to 30 (per group)
- Person-Centered variables addressed: Acceptance, Transparency, Understanding
- Application effort: High
  This is depending heavily on the number of involved teams and teaching staff.
- Level of expertise required: High
- Suggested assistance: Administrator, Tutor
- Target skills: Collaboration, Technical skills, Interpersonal Skills, Communication, Problem solving, Practical skills, Creativity, Self-directedness, Teamwork
- Input: Learning contract workspaces and guidelines, relevant content, evaluation criteria and guidelines
- Output: Learning contract contributions
  These may be interconnected to construct a knowledge base by applying KNOWLEDGE BASE CONSTRUCTION.

Web Template

Inherited. This pattern reuses the parent pattern's PROJECT-BASED LEARNING Web template. The specialization is just that *contracts* are the "projects". See the examples section for an example of a contract proposal form that was used in the previous Web Engineering LEARNING CONTRACT projects.

Examples

This pattern was used in the ***Web Engineering*** courses 2003 and 2004: Instead of passing a conventional written EXAMINATION at the end of the lectures, participants were offered the option to engage in the *Web Engineering Learning License* (WELL) project that was based on the LEARNING CONTRACTS pattern. Thereby, teams of 2 – 5 students had to elaborate contributions for a self-chosen topic (PROBLEM PROPOSALS). Different types of contributions were possible: Written reports, exam questions and answers, presentation slides, or any useful combination thereof. WELL teams were coached by their lab course instructor, who approved the contract proposals and who conducted the work-in-progress and final inspections. Subsequently, each student participating in the WELL project had to PEER-EVALUATE at least three other WELL contracts. Additionally, each WELL team had to provide a written SELF-EVALUATION, and during the final inspection, there was an oral EXAMINATION on the elaborated WELL topic as well as on surrounding subject matter. Of the 355 Web Engineering participants, nearly 300 participated in the WELL project (i.e., 84%). The WELL contract structure each team had to fill out in the PROBLEM PROPOSAL phase is given in Table 11.

Group / team number:
Instructor:
Team members with e-mail address:
Topic:
Goals:
Activities and documents:
Significant changes and their dates:
Intermediate version accepted on:
Final version due:
Signature team representative:
Signature instructor:

Table 11: WELL contract structure.
http://elearn.pri.univie.ac.at/patterns/examples/learningcontracts/contract.html shows an example of a completed WELL contract (in German).

During the WELL evaluation phase in the year 2003, 903 peer-evaluations were collected. Each peer-evaluation consisted of a written comment and of 0 to 5 assigned bonus points. This way, it was possible to compile rankings of WELL contracts based on bonus points and number of evaluations. Table 12 shows the 20 most often evaluated WELL contracts along with their assigned bonus points.

Table 12: The 20 most often evaluated WELL contracts in 2003.

Topic	Number of Evaluations	Bonus Points
PHP	62	186
JavaScript	39	83
Internet Exchange Markets	32	83
XML	25	70
Usability and User Interface Design	23	84
Battlenet	23	25
Elaboration of Web-Engineering Exam Questions	22	67
Intrusion Detection Systems	22	65
Formulating Questionnaires	22	38
Web Services (Interoperability)	20	80
Cyberlaw	20	60
IPv6 – The Next-Generation Internet Protocol	20	54
Analyzing Webshops in the USA and in the EU	19	46
Social and Security Aspects of Web Communication	19	44
Security in E- and M-Commerce	18	60
Usability of Webpages und Webapplications	17	76
Virtual Communities	17	55
Websecurity	16	63
Web Services – Basics, Benefits, Architecture, Security...	16	50
SSL – Secure Socket Layer	15	53

The analysis of the collected QUESTIONNAIRES (see the following section) showed that participants highly appreciated this form contract-based learning.

Evaluation

Evaluations on the WELL project in ***Web Engineering 2003*** (see *Examples* section) have produced some interesting results:

- The long term learning effect of contributing to WELL is considered higher than that of passing a conventional exam (cf. Figure 138).
- The time investment required for a learning contract project is perceived as being considerably higher than learning for a conventional exam (cf. Figure 138). Nevertheless, more than 80% of 355 students engaged in the learning contracts.
- Regarding the alternative form of assessment (i.e., BLENDED EVALUATION), WELL participants found it very meaningful to read their peers' evaluations of their own work (cf. the *Evaluation* section of PEER-EVALUATION).

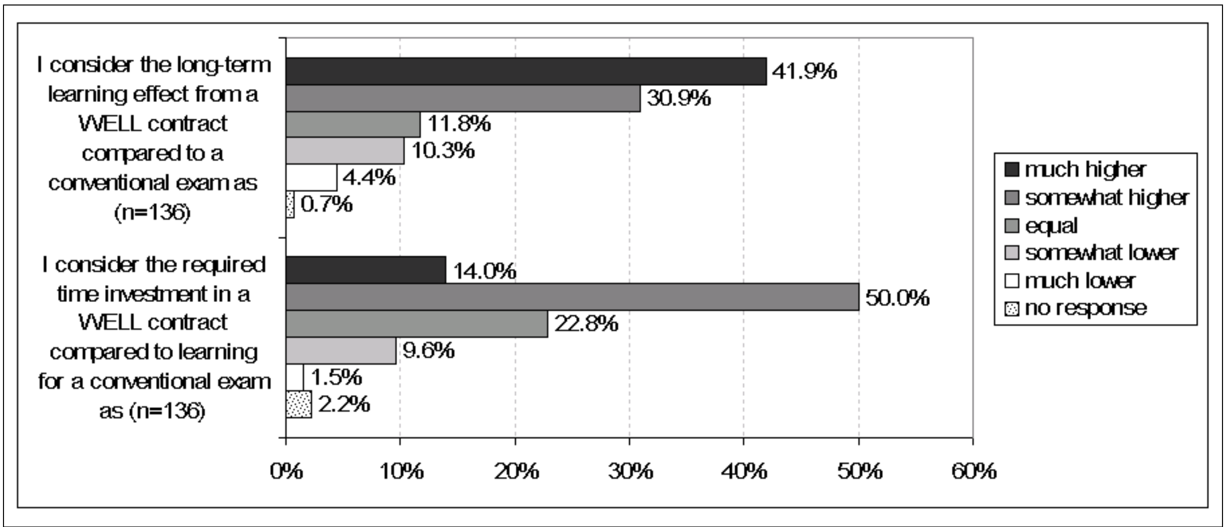


Figure 138: Long-term learning effect and time investment required in the WELL project (Web Engineering 2003).

Remarks

If LEARNING CONTRACTS are used as a substitute for EXAMINATIONS, care has to be taken on how to assess participants' knowledge of subject matter (e.g., a brief oral EXAMINATION of each team on related/remaining content is advisable).

There are many Web resources on learning contracts. A short list of valuable sites is given below:

- http://www.pri.univie.ac.at/~derntl/papers/mICTE03-LC.pdf
  "Web-Support for Learning Contracts: Concept and Experiences," by Motschnig-Pitrik R., Derntl M., and Mangler J.

- http://www.dmu.ac.uk/~jamesa/teaching/learning_contracts.htm
  By TRACE – Training Resources and Continuing Education at the Univ. of Waterloo
- http://iteslj.org/Articles/Schwarzer-Contracts.html
  "Learning Contracts and Team Teaching in a University ESL Writing Class," by David Schwarzer, Robert E. Kahn, and Kristi Smart (Univ. of Texas)
- http://www.csd.uwa.edu.au/altmodes/to_delivery/learning_contracts.html
  "Alternative Modes of Teaching and Learning: Learning Contracts"
- http://www.msu.edu/user/coddejos/contract.htm
  "Using Learning Contracts in the College Classroom," by Joseph R. Codde, Michigan State Univ.
- http://www-distance.syr.edu/contract.html
  Paper on learning contracts by Syracuse University

References

Atherton, J. A. (2003). *Learning and Teaching: Learning Contracts* [Online]. Retrieved Mar 5, 2004, from http://www.dmu.ac.uk/~jamesa/teaching/learning_contracts.htm

Rogers, C. R. (1983). *Freedom to Learn for the 80's*. Columbus, Ohio: Charles E. Merrill Publishing Company.

PROJECT-BASED LEARNING

Package: *Project-Based Learning*

Intent

Participants elaborate projects iteratively and incrementally in several successive PROJECT MILESTONES. Participants may work out individual projects, may be organized in teams, or may collaborate collectively on a single group/course project.

Motivation

Project-based learning (PBL) is a well-established and widely employed approach of teaching and learning in complex subject domains that is capable of enhancing the quality of student learning when compared with traditional instructional approaches[562]. PBL is a perfect field for applying Person-Centered principles, as it "*is a model for classroom activity that shifts away from the classroom practices of short, isolated, teacher-centered lessons and instead emphasizes learning activities that are long-term, interdisciplinary, student-centered, and integrated with real world issues and practice.*"[563] Instead of focusing on detached elaborations on certain topics, theories, or technologies, students engage in integrated problem-solving

[562] Cf. Thomas (2000, p. 35)

[563] San Mateo County Office of Education (2001)

efforts to tackle more complex, authentic problems that are – particularly in Person-Centered settings – of personal relevance. In this respect, PBL is likely to leverage students' motivation and encourages the "development" of life-long learners[564].

Optimally, PBL activities are characterized by[565]:

- *Real world* orientation: learning has value beyond the demonstrated competence of the learner.
- Utilization of *hands-on approaches* and various modes of communication.
- Students' *responsibility for their own learning*: projects promote meaningful learning, connecting new learning to students' past performances.
- The instructor becoming a *facilitator* of learning.
- Encouragement of students' SELF-EVALUATION of learning. This is frequently complemented by including PEER-EVALUATION and reflection[566] (e.g., project DIARIES).

A very valuable outline is given by Brunner[567], who stresses the essences of PBL:

- Engaging learning experiences that involve students in complex, real-world projects through which they develop and apply skills and knowledge.
- A strategy that recognizes that significant learning taps students' inherent drive to learn.
- Learning in which curricular outcomes can be identified up-front, but in which the outcomes of the student's learning process are neither predetermined nor fully predictable.
- Learning that requires students to draw from many information sources and disciplines in order to solve problems.
- Experiences through which students learn to manage and allocate resources such as time and materials.

A valuable article on PBL by John Thomas is available online[568] and including a thorough review on research on this approach. He reports benefits of PBL claiming that,

> [t]here is direct and indirect evidence, both from students and teachers, that PBL is a more popular method of instruction than traditional methods. Additionally, students and teachers both believe that PBL is beneficial and effective as an instructional method. Some studies of PBL even report unintended and seemingly beneficial consequences associated with PBL experiences. Among these consequences are enhanced professionalism and collaboration on the part of teachers and increased attendance, self-reliance, and improved attitudes towards learning on the part of students.[569]

[564] Cf. Buck Institute for Education (2002)

[565] Cf. Kraft (2003)

[566] See also San Mateo County Office of Education (2001)

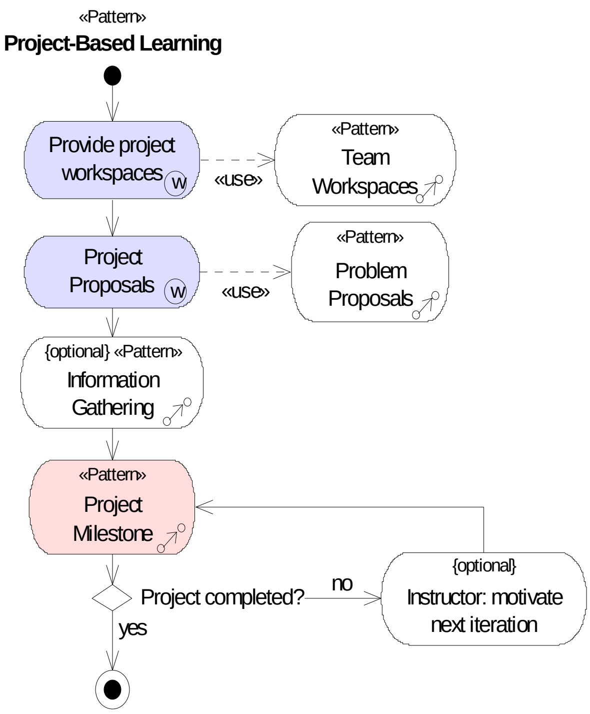
[567] Adapted from Brunner and Polin (1999)

[568] See http://www.bobpearlman.org/BestPractices/PBL_Research.pdf (accessed Jan 9, 2004)

[569] Thomas (2000, p. 34)

In terms of the pattern repository, PROJECT-BASED LEARNING is mainly composed of a number of consecutive PROJECT MILESTONES, resulting in a model of an iterative problem-solving process. It is a generic pattern, as it does neither define a specific type of a project, nor the concrete number or targets of its milestones.

Sequence

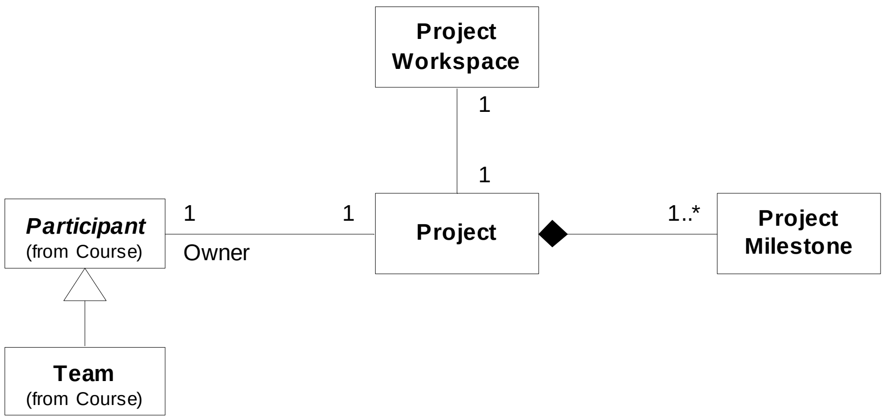


Activity	Description
Project proposals	As in any Person-Centered setting, participants are entitled to propose their own projects (see PROBLEM PROPOSALS).
Create project workspaces	For project teams to manage their documents, they are provided with TEAM WORKSPACES for elaborations.
INFORMATION GATHERING	When required by the concrete project setting, any concrete form of INFORMATION GATHERING can be used to elaborate theories, techniques, and content underlying the project work.
PROJECT MILESTONE	Depending on the concrete project, the complex work process is arranged in a number of consecutive PROJECT MILESTONES. For example, when employing the *Rational Unified Process*[570], possible milestones are inception, elaboration, construction and transition. Note that, even though not explicitly modeled in the diagram, each

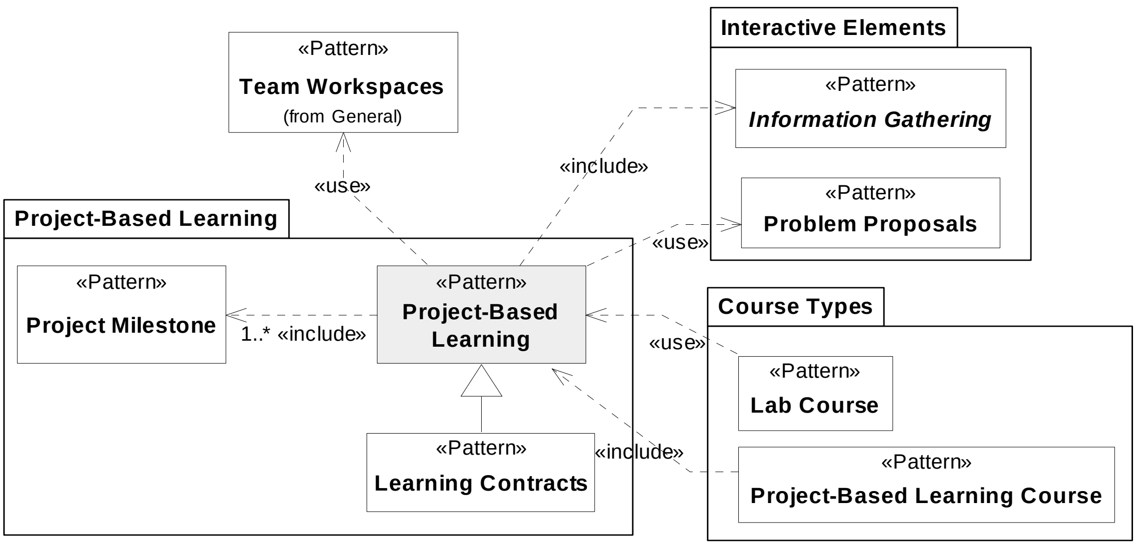
[570] Jacobson, Booch and Rumbaugh (1999)

Activity	Description
	PROJECT MILESTONE or the whole project can be accompanied by a number of collateral patterns, such as DIARY, ACHIEVEMENT AWARD, or any INTERACTIVE ELEMENT.
Instructor: motivate next iteration	The instructor should recapture the current status of the project works and provide an outlook on the next iteration.

Structure



Taxonomy/Dependencies



Parameters

- Primary pattern author: Michael Derntl
- Primary pattern source: Research Lab for Educational Technologies (University of Vienna), Literature
- Pattern categories: Utility, Generic, Collateral, Motivational, Traditional
- Level of abstraction: Low

- Scope: Phase, Course
- Primary presence type: Blended
- Flexibility: High
- Level of confidence: 4
- Number of participants: up to 30
  Note that this number applies for the basic scenario described here. For more participants derived scenarios and/or student facilitators might be required.
- Application effort: High
- Level of expertise required: High
- Person-Centered variables addressed: Acceptance, Transparency, Understanding
- Suggested assistance: Administrator, Tutor
- Target skills: Collaboration, Problem solving, Interpersonal Skills, Communication, Technical skills, Practical skills
- Input: Project resources and guidelines, Relevant content, Process model to be employed, Expected activities and artifacts
- Output: Elaborated projects and documents

Web Template

Generally, the Web template for PBL has to accommodate all Web modules for supporting the various PBL phases. Therefore, the PBL page on the participants' side is a "living" page that is extended with respect to the current phase. The PBL Web template fully exploits the modular design of the pattern repository, by including/linking other patterns in the PBL main page.

Please note that a very simple form of PBL may be realized just by providing a shared MARKET for project documents. A more sophisticated implementation of PBL may also be realized using TEAM WORKSPACES as the main project page, with appropriate information on the workspace overview page, and a dedicated folder for each PROJECT MILESTONE.

Administration View

The administration view of PBL allows for configuring general settings as well as other patterns which are plugged in as the project proceeds:

- *General settings*: *Heading* and *introductory text* to be displayed on the main PBL page (step 1; see Figure 139). The introductory text typically includes links to relevant material for the projects.
- *PBL phases*: For each phase of the PBL process, the administrator may add a section to the main PBL page which controls the respective phase (step 2; see Figure 140). Each of these phases/sections includes:
  - *Name*: Short name for the section (e.g., Project proposals)
  - *Description*: Short paragraph describing the intent/rationale of the current phase

- *Type*: Specifies the type of the section which is used by the configuration wizard to forward the administrator to another pattern's administration, in case the current section refers to a pattern. The section type is one of the following (cf. the *Sequence* section): TEAM BUILDING, DIARY, TEAM WORKSPACES, PROBLEM PROPOSALS, MARKET, GENERIC EVALUATION, or *custom link* in cases a phase does not rely on a particular pattern.
- *Link text*: Text to be displayed as hyperlink to the respective section's Web page.

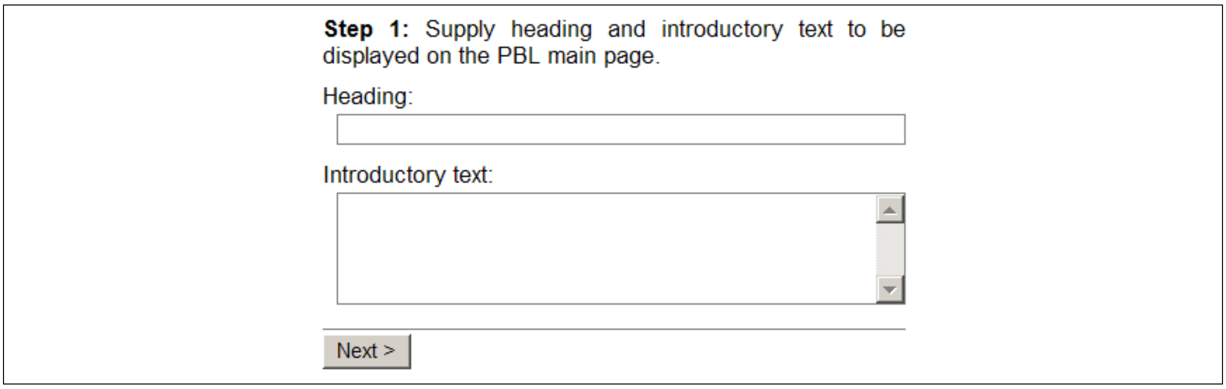


Figure 139: General configuration of PROJECT-BASED LEARNING.

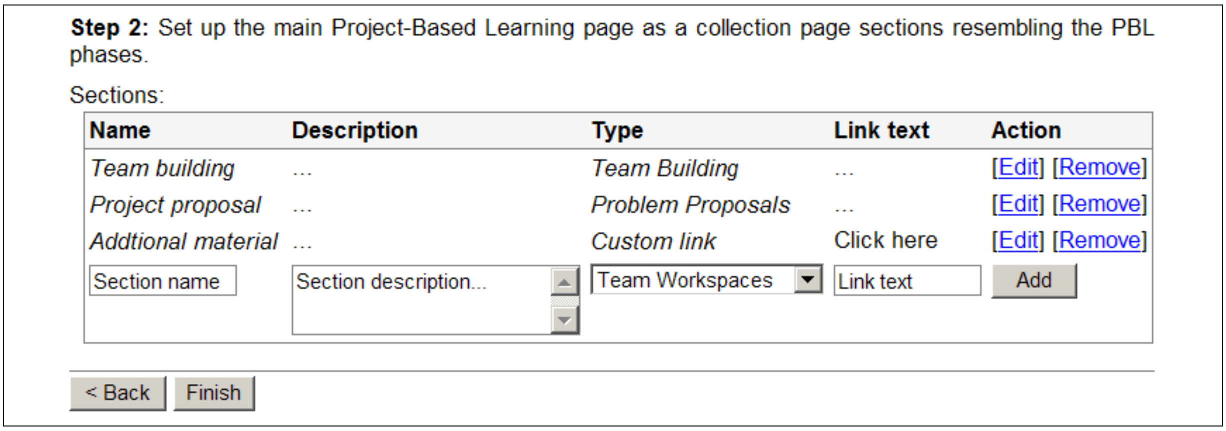


Figure 140: PROJECT-BASED LEARNING phases configuration.

Participant View

Even though a PBL scenario is a quite complex setting, one single main entry page for participants suffices. This page carries succinct descriptions and links for each of the PBL phases, which manage the actual PBL tasks/steps, such as TEAM BUILDING. This page is depicted in Figure 141, showing the configured heading and introductory text, followed by a number of sections that resemble the PBL phases. An example PBL main page can is depicted in the *Examples* section.

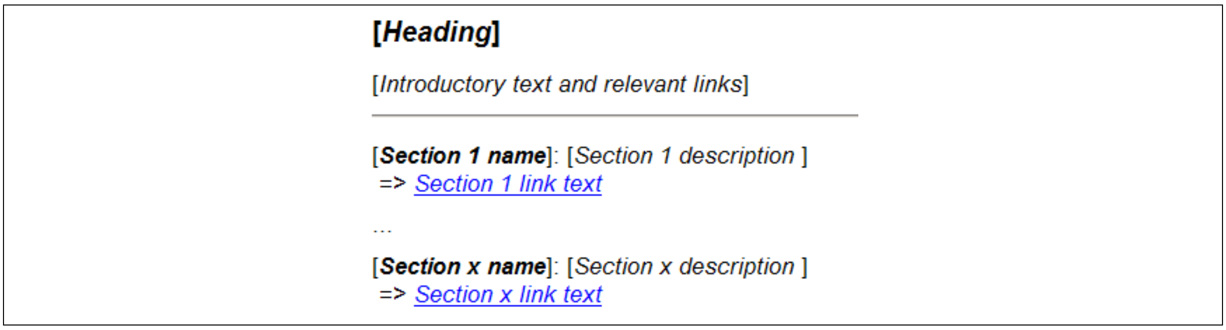


Figure 141: The main PROJECT-BASED LEARNING page.

Report View

There is no specific report provided by this pattern; rather, reports for individual steps in the project-based learning scenario can be requested via the included patterns' report interfaces.

Examples

There are thre basic options of organizing project work:

Team Projects

Every participant or team has his/her/its own project, i.e., there are multiple objects employing the same process/approach in one single course group. This kind of project work is applied in almost any of the lab courses at the author's institution. Typically, some complex technology or elaboration process (or a combination of these two) builds the foundation of the project work. Figure 142 shows the participant view of the Project-Based Learning main page of the *WELL project* thread in ***Web Engineering 2004***. The screenshot shows four configured sections, i.e. TEAM BUILDING, PROPOSALS (via online forms), TEAM WORKSPACES, and PEER-EVALUATION.

Figure 142: WELL contract projects main page in Web Engineering 2004.

Group Project

There is one central project that is elaborated by the whole group. Thereby it is possible that participants focus on different aspects of the same project (it is still possible that participants form small teams to elaborate their contributions). Such a PBL approach was employed in ***Software Engineering*** (summer term 2002), where the whole group engaged in planning and building a prototype of a Web application that would support them in certain aspects of their studies. The following gives a list of functional requirements that were collaboratively elaborated[571]:

- Supply of information: relevant links, information about and contacts to other students, course descriptions, etc.
- Materials: e-content, lecture notes, etc.
- Exchange of information: organized chats and discussion forums
- Markets: trading of lecture notes, books, jobs and other offers
- Cooperative learning: tutoring, training, etc.
- Organizational issues: schedulers, alerts, etc.
- News: notification of changes, cancellations, etc.

[571] Cf. Motschnig-Pitrik and Derntl (2002); a copy of the paper describing the case study is available at http://elearn.pri.univie.ac.at/patterns/examples/project-basedlearning/ICL-SCeL02.pdf

Personal Project

For Masters or PhD students writing their thesis personal projects are equally valid scenarios for this pattern. The same may apply to seminars with very few participants. The only difference is that the TEAM WORKSPACES are used by single participants.

Evaluation

In Web Engineering (2003), the general QUESTIONNAIRE included some items that addressed the collaborative work with colleagues, the self-directed learning style, and the building of practical skills that are central to PBL. Figure 143 and Figure 144 show histograms with these items scores as compared to a hypothetical conventional course.

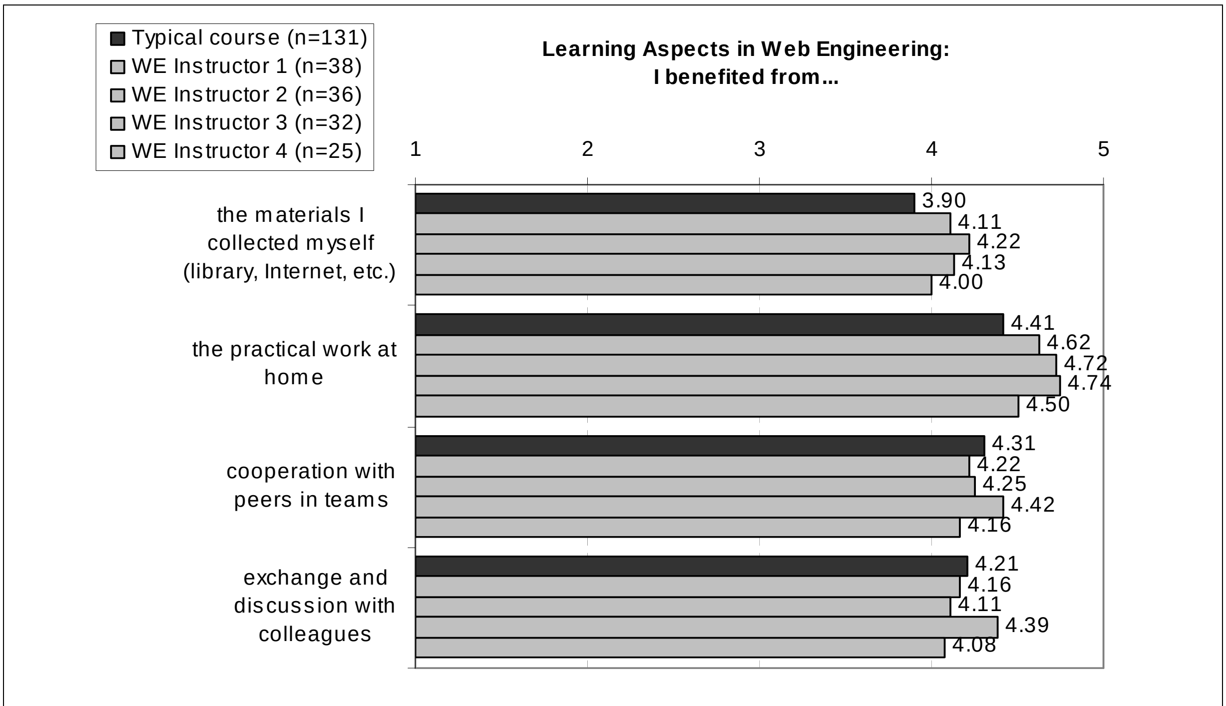


Figure 143: Learning aspects related to PBL in Web Engineering 2003.
The black bars show the items' values for a hypothetical, typical course that was surveyed in the beginning of the Web Engineering course. The grey bars show values for Web Engineering instructors 1 – 4, respectively. The scale reaches from 1 = *not at all* to 5 = *very much.*

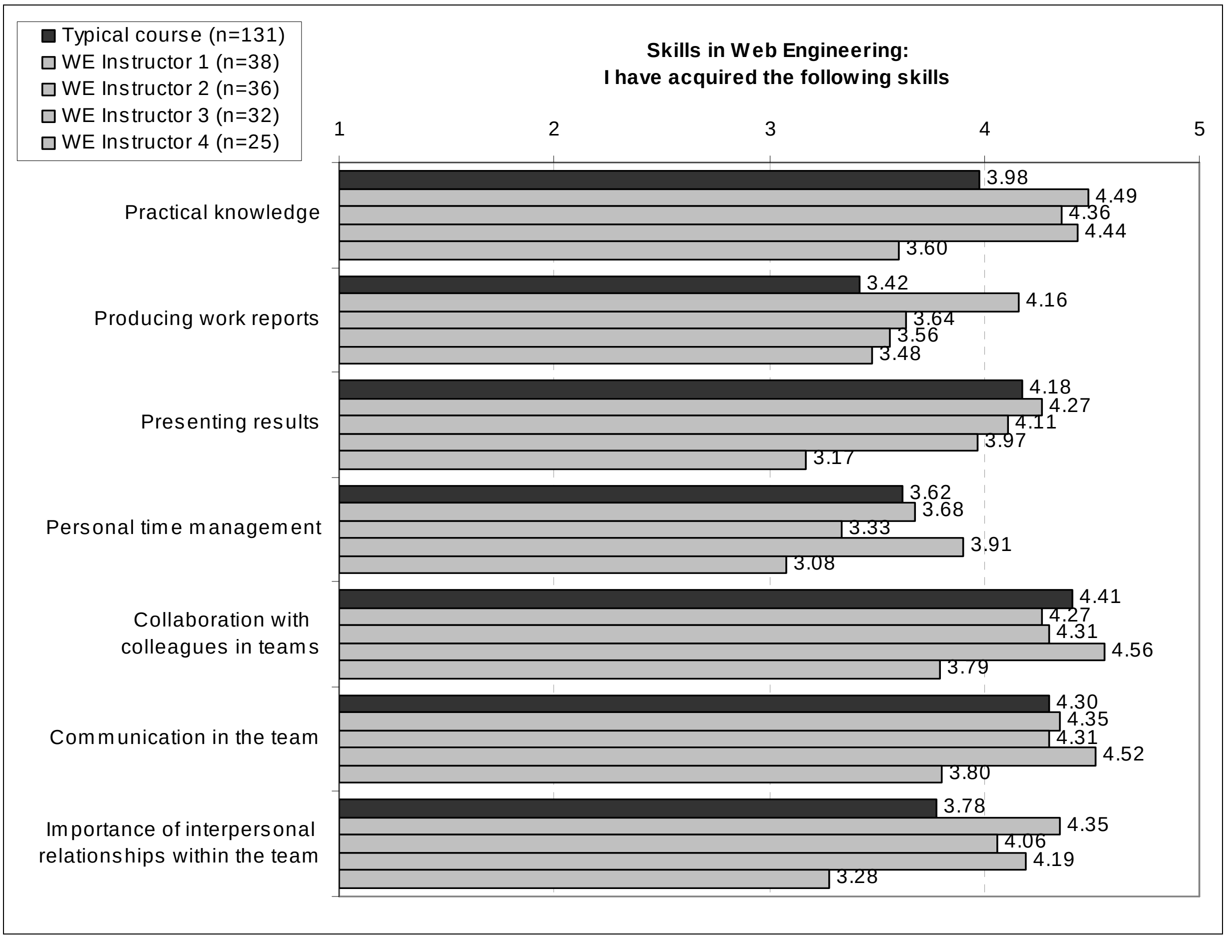


Figure 144: Skills related to PBL in Web Engineering 2003.
Layout of the histogram and scale of the items are the same as in the above figure.

References

Brunner, J., & Polin, L. (1999). *Definition of Project-Based Learning*. Retrieved Oct 25, 2003, from http://hale.pepperdine.edu/~jabrunne/wpgrid.html

Buck Institute for Education. (2002). *Project Based Learning Handbook: Introduction to Project Based Learning*. Retrieved Jan 9, 2004, from http://www.bie.org/pbl/pblhandbook/intro.php

Jacobson, I., Booch, G., & Rumbaugh, J. (1999). *The Unified Software Development Process*. Reading, MA: Addison-Wesley.

Kraft, N. (2003). *Criteria for Authentic Project-Based Learning*. Retrieved Dec 22, 2003, from http://www.rmcdenver.com/useguide/pbl.htm

Motschnig-Pitrik, R., & Derntl, M. (2002). *Student-Centered e-Learning (SCeL): Concept and application in a students' project on supporting learning*. Proceedings of International Workshop on Interactive Computer-Aided Learning (ICL) 2002, Villach, Austria.

San Mateo County Office of Education. (2001). *Why Do Project-Based Learning?* Retrieved Dec 22, 2003, from http://pblmm.k12.ca.us/PBLGuide/WhyPBL.html

Thomas, J. W. (2000). *A Review of Research on Project-Based Learning*. Retrieved Jan 9, 2004, from http://www.bobpearlman.org/BestPractices/PBL_Research.pdf

PROJECT MILESTONE

Package: *Project-Based Learning*

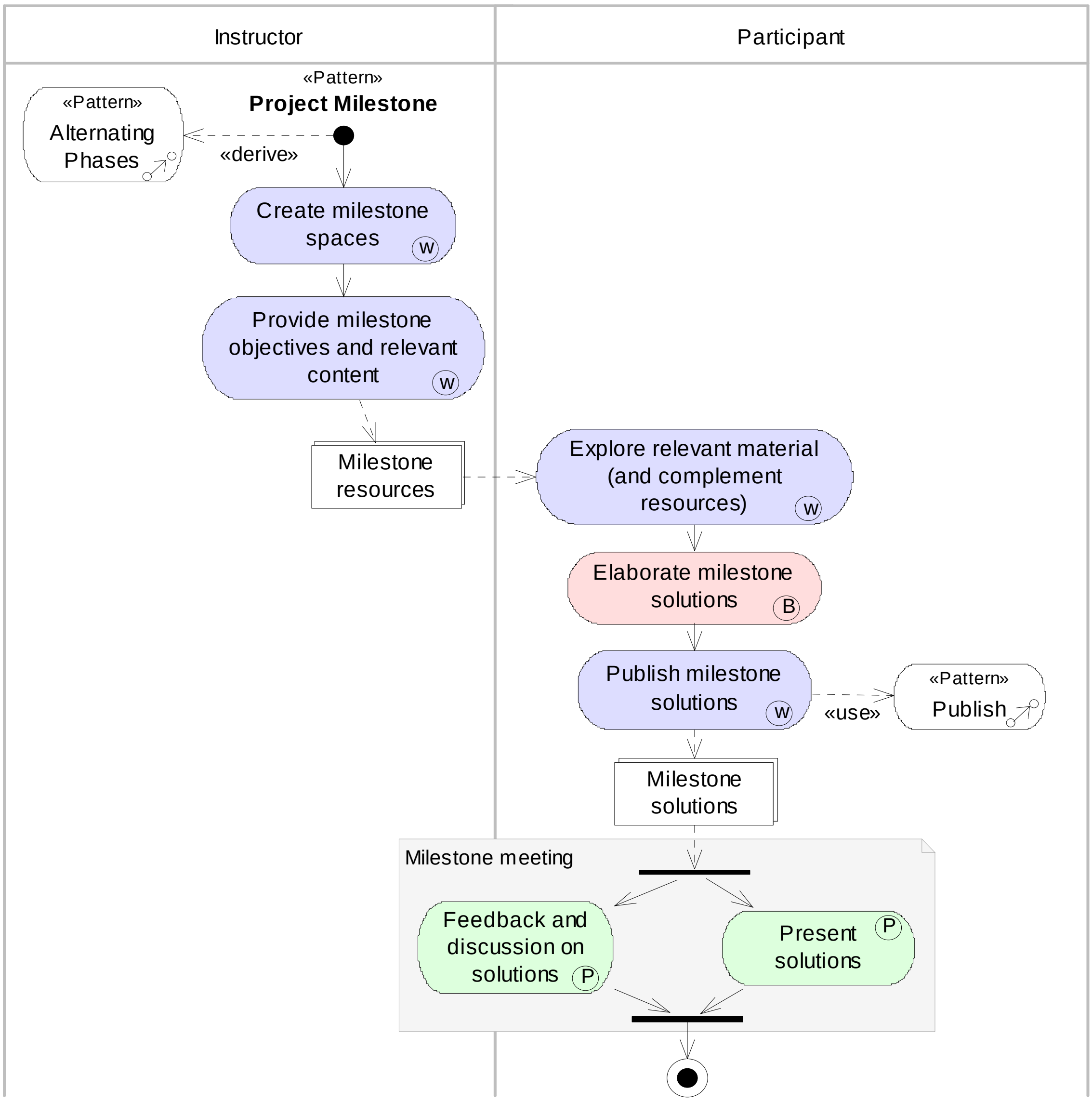
Intent

Project work is accomplished in a series of well-defined phases, each producing a certain set of artifacts. These milestone solutions are PUBLISHED and presented by the project owners.

Motivation

Each project is separated into a number of milestones which can be used to decompose and to structure a certain overall project goal/target. Arranging project fragments this way helps one to derive work packages and thus giving a more modular, compact view on the whole elaboration process. In terms of this pattern repository, the main learning thread of each PROJECT-BASED LEARNING (PBL) scenario is divided into a number of consecutive PROJECT MILESTONES. Each milestone has well-defined objectives and participants elaborate a set of milestone artifacts that give a measure of the degree of achievement in their work. Thereby, the learning platform provides workspaces for the project teams and allows them to manage their milestone artifacts online, which allows for distant collaboration among team members. The facilitator coaches the project teams and guides them in their work. In PBL settings, presenting and discussing the milestone solutions (artifacts) has the potential to (a) to improve presentation skills and (b) to inform the whole group about the current project status and the practices/techniques/tools that were employed. This way, the whole group can profit and learn from more than one single example, from problems and their solutions, from good practices, and generally from the situated handling of new and unforeseen situations.

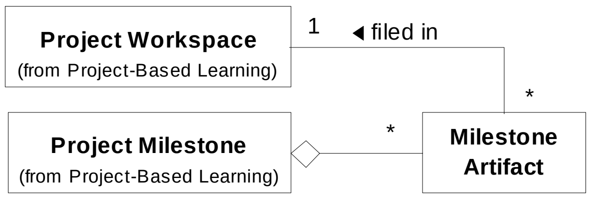
Sequence



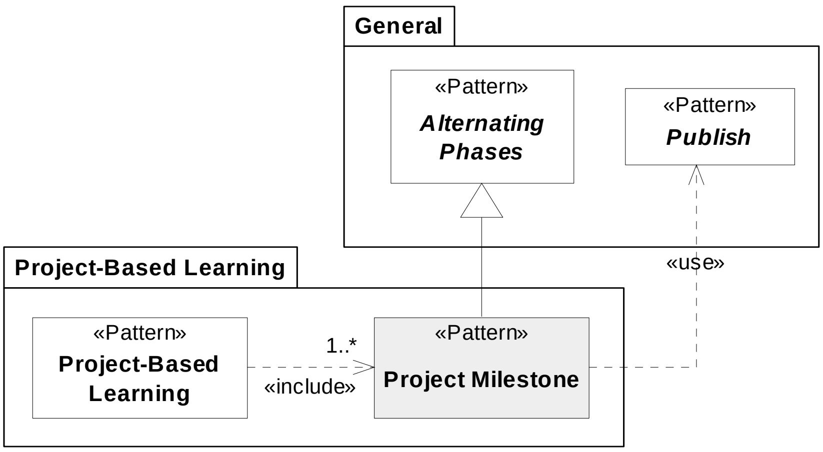
Activity	Description
Provide milestone spaces	Within the project workspaces (TEAM WORKSPACES), a separate space for each milestone should be provided for participants to upload their milestone artifacts (documents, diagrams, notes, etc.).
Provide milestone objectives and additional relevant content	The instructor outlines the current milestone's objectives to provide the participants with proceeding/elaboration guidelines. Content relevant to the current project milestone is prepared by the instructor. Relevant content may be elaborated by using any of the INFORMATION GATHERING patterns.
Explore relevant material (and complement resources)	Participants explore and apply material relevant to their current work. They may also complemented the resources if they are provided with means to do so.

Activity	Description
Elaborate milestone solutions	Milestone solutions are elaborated by participants in a self-directed mode, whereas the facilitator makes himself/herself available for answering questions and for providing guidance when needed.
Publish milestone solutions	Solutions (milestone artifacts) are uploaded to the milestone workspace on the learning platform.
Present solutions	Milestone solutions are presented in the plenum. If there are many project teams in the same group, it is advisable *not* to let all teams present their solutions. A well-proven alternative approach is to let project teams select partners to whom they present their solutions to prevent excessive, exhausting presentation sessions.
Feedback and discussion of solutions	Providing feedback from the instructors and from the peers' points of view to project teams is essential to approve as well as to criticize their work. Discussions on project solutions in the group additionally further interaction among participants.

Structure



Taxonomy/Dependencies



Parameters

- Primary pattern author: Michael Derntl
- Primary pattern source: Research Lab for Educational Technologies (University of Vienna)

- Pattern categories: Utility
- Level of abstraction: Low
- Scope: Phase
- Primary presence type: Blended
- Flexibility: High
- Level of confidence: 4
- Application effort: High
- Level of expertise required: High
- Suggested assistance: Tutor
- Target skills: Collaboration, Problem solving, Interpersonal Skills, Communication, Technical skills, Practical skills
- Input: Milestone objectives, Relevant content
- Output: Milestone artifacts

Web Template

Not available. Web support for this pattern is addressed in PROJECT-BASED LEARNING.

Examples

In ***Web Engineering***, participants plan and elaborate Web application projects in four consecutive milestones, basically following the *Unified Process*[572]: Inception, Elaboration, Construction, and Transition. For example, in the Inception phase, the following objectives and deliverables are defined:

- Vision Statement
- Goal analysis, value map (cf. Web IS Development)
- E-presence planning questionnaire (cf. Business & Social Webs)
- Requirements analysis
  - Use case model
    - Overview diagram
    - Use case relationships
    - Description of default and alternative flows for each use case
    - Non-functional requirements for use cases, if relevant
  - Optional: document specifying the application of an alternative requirements analysis technique, including a justification of the selected technique
- Non-functional, use case independent requirements
- Data model
- Class diagram or ER model
  - XML description, XML schema (to be delivered per person)
  - Rich picture diagram including comments
- SSM root definitions (cf. Web IS Development)

[572] Jacobson, Booch and Rumbaugh (1999)

- Content concept
- Accompanying project diary (on a separate space on the learning platform)

Evaluation

Not available: see pattern PROJECT-BASED LEARNING.

References

Jacobson, I., Booch, G., & Rumbaugh, J. (1999). *The Unified Software Development Process*. Reading, MA: Addison-Wesley.

6 Case Study and Pattern Management

In this Chapter we apply the PCeL pattern approach (Chapter 3) and the PCeL pattern repository (Chapter 5) to initialize and construct a Web-based support environment for a hypothetical course. First, the course and its objectives are described verbally. Then we derive an initial activity model from the verbal description and subsequently derive a PCeL pattern based model from that activity model. To provide a Web support environment for the course, the Web templates of selected patterns were implemented on top of the *Cooperative Environment Web Services* (CEWebS) learning platform. Thereby, the administration view is covered by the "Pattern Manager" that was implemented and plugged into the administration Web application of CEWebS. Also, the report view and of course the participant view are available in CEWebS.

We will "walk through" the pattern activity model of the course by configuring the respective Web services via the "Pattern Manager" and by populating the participant view with data from a real course. This way, we experience the application of the BLESS model in combination with the PCeL pattern approach for a whole course.

This Chapter is structured as follows:

- In Section 6.1 (p. 403) the CEWebS learning platform is presented to provide the reader with relevant information on the implementation and Web support aspects. Also, the patterns that were implemented on top of CEWebS are listed and the implementations of the Web templates are briefly described.
- In Section 6.2 (p. 410) the hypothetical course is verbally described and modeled using the PCeL patterns. Subsequently we "step through" the course's sequence of patterns and activities, considering the administration, participant, and report views.
- In Section 6.3 (p. 438) we conclude the Chapter with a discussion on the case study.

6.1 PCeL Patterns on CEWebS

6.1.1 CEWebS Architecture and "Philosophy"

The CEWebS (Cooperative Environment Web Services) architecture was developed during a Masters thesis[573] at the Faculty of Computer Science, University of Vienna. It was developed based on the experience that none of the currently existing learning platforms was fully

[573] Mangler (2005); see also Mangler and Derntl (2004)

capable of supporting thoughtfully designed blended learning scenarios in an optimally situated way. Consider the following example: Students elaborate projects and you want them to evaluate the milestone solutions of their peers online, which basically resembles a simple peer-evaluation scenario. How do you project such a scenario on a learning platform that offers, for example, chat, discussion forum, and workspaces? There is no optimal solution, but there are feasible *compromises* such as providing evaluations as discussion postings, uploading evaluations as text files onto the teams' workspaces, or simply circumnavigating the learning platform by sending evaluations per e-mail. However, refusing to think solely in terms of available functionality, we push to think more in terms of *desired scenarios*, which are captured by the PCeL pattern repository. In this context, CEWebS is targeted at providing learning technology support at layer 5 of the BLESS model.

CEWebS is a distributed architecture that is completely based on interacting Web Services[574]. CEWebS defines *interfaces* by specifying Web Service Description Language (WSDL) contracts[575] that each Web Service has to comply with. The guiding philosophy of CEWebS is to *keep functionality where it is needed and to keep it simple.* A conceptual sketch of the architecture is given in Figure 145. Screenshots of the visual appearance of CEWebS pages are available in Section 6.2.3.

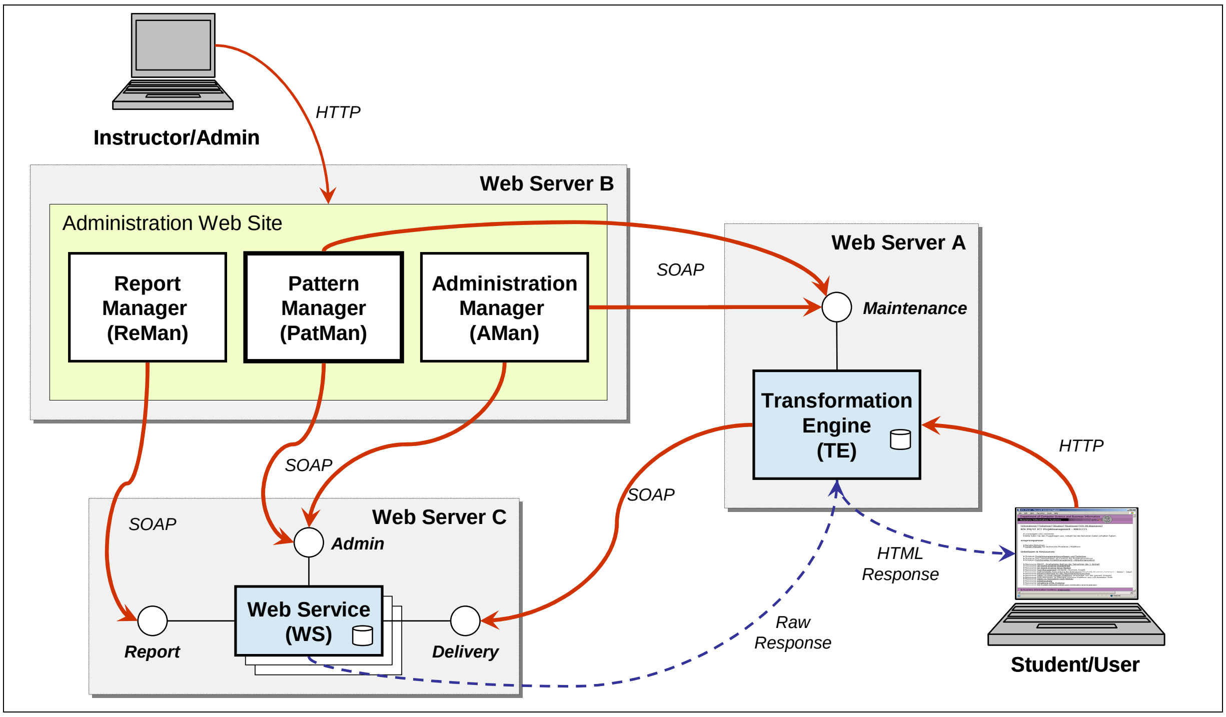


Figure 145: The CEWebS architecture.

574 Curbera, Nagy and Weerawarana (2001), W3C (2003b)

575 W3C (2001)

In the following, the main components of CEWebS are briefly explained:

Web Services (WS). WS are the central components of CEWebS. Each WS implements a self-contained learning object, whereby the term "learning object" is regarded in its broadest sense here: It may refer to a simple, static Web page including some content, to interactive means of online communication (e.g., discussion forums), to more complex learning processes (e.g., peer-evaluation, project-based learning), or to any useful composition thereof. Our current set of WS was implemented based on frequently used learning scenarios as captured by the PCeL pattern repository, whereby each WS implements the functionality needed to support one single scenario or pattern. Through implementing the *Delivery* interface, each WS supports a number of delivery commands that control the visual output of the WS. The *Report* interface defines a number of operations to generate reports (e.g., learner activity within the WS) that are output to the requesting browser using HTML or any type of binary data such as PDF, or that may send messages to learners (e.g., notification of those who have missed a deadline). Finally, the *Administration* interface defines functionality needed to control the administrative parameters of a WS, e.g., involved students and teams, whereby each WS is capable of hosting multiple instances of that data. This way it can be used concurrently by more than one Transformation Engine. By implementing all of the described WS interfaces, *any* provider may contribute a WS that encapsulates some Web-based learning content or process.

Transformation Engine (TE). The TE can actually be considered as the *learning platform* or, alternatively, the Web Service container: It receives the connection properties (i.e., server URL, port, name, etc.) of a set of distributed WS, which are configured by the Administration Manager via the *Maintenance* interface. The TE keeps that list of WS instances and users in its database. Any TE is hosted on a Web server and is publicly accessible through standard HTTP requests: Upon receiving an initial (empty) request, the TE calls the first WS in its list of services over SOAP[576] via the *Delivery* interface to display itself by sending its configured default command along with the unique instance identifier of that WS. The WS in turn returns a *raw response* (which is basically a subset of HTML) that represents its contents for the requested command. The TE transforms the WS response and constructs a complete *HTML page* including page headers, footers, and other "surrounding" elements that make up a user-friendly Web application. That HTML page is returned to the requesting Web client, which is typically a standard Web browser used by a student to access the learning platform. Note that the TE in combination with the WS implements the ***participant view*** of the patterns' Web templates.

Administration Manager (AMan). To configure the TE and WS, administrators are provided with the AMan, which is basically a Web application that interacts with WS

[576] *Simple Object Access Protocol*; see W3C (2004b)

through the *Administration* and *Notification* interfaces as well as with TE through their *Maintenance* interfaces. The core administrative task is to configure each TE along with its WS, which represents the learning platform for a particular course. Additionally, the *Notification* interface empowers the AMan to keep shared data (e.g., user data) consistent among a TE's services.

Pattern Manager (PatMan). Previously, the Web services had to be configured by more or less manually editing their configuration data. However, to be able to address a broader, technically less experienced audience, the PatMan was implemented as part of this thesis to provide a wizard-based WS configuration application, leading the administrator or instructor through any required step in the WS configuration process. The PatMan is described in more detail in the following Section 6.1.2. Note that the PatMan implements the ***administration view*** of the patterns' Web templates.

Report Manager (ReMan). The ReMan, which resides within the same Web application as the AMan and the PatMan, is used by instructors to retrieve reports through the WS's *Report* interfaces (compare the WS description above). Actually, supporting the instructor in generating meaningful reports of students' activities was initially one of the driving factors for designing CEWebS, as none of the learning platforms we used so far was able to produce useful, customizable reports. For example, when assessing a grade for a student's contribution, the ReMan can assist the instructor in generating evaluation reports for this contribution (e.g., peer-evaluation and self-evaluation reports). Note that the ReMan implements the ***report view*** of patterns' Web templates.

6.1.2 The Pattern Manager Prototype

The Pattern Manager (PatMan) is intended to provide an easy-to-use implementation of the administration view of the PCeL patterns in the pattern repository. It was plugged into the administration Web site of the CEWebS architecture.

The primary aim of using the PatMan is to guide the instructor or administrator in the process of instantiating patterns and learning processes on the learning platform. This was achieved by creating a prototypical, working implementation of the administration view of each pattern for which a CEWebS Web Service exists. That is, in alphabetical order: BLENDED EVALUATION, COLLECT FEEDBACK, COURSE, DIARY, FEEDBACK FORUM, ONLINE DISCUSSION, PEER-EVALUATION, PRELIMINARY PHASES, PROJECT-BASED LEARNING, PROJECT-BASED LEARNING COURSE, PROJECT MILESTONE, QUESTIONNAIRE, REACTION SHEETS, SELF-EVALUATION, TEAM BUILDING, AND TEAM WORKSPACES. Additionally, the participant view of the TEAM WORKSPACES pattern was also developed in the scope of this thesis.

The main Web page of the PatMan for a particular course is divided into four sections, as depicted in Figure 146:

Currently Active Patterns. This section provides a general overview of the current course's structure, containing a list of currently active patterns in the course. The pattern instances are displayed hierarchically, which means that a pattern that was instantiated by a pass-through pattern is listed below its "host" pattern (e.g., PEER-EVALUATION was instantiated by the BLENDED EVALUATION pattern wizard). Each pattern shows two links: "Edit", which allows the instructor to modify an active pattern instance (e.g., adding information or changing its configuration) in a wizard, and "Info", which displays configuration information for the respective pattern.

Employ a Pattern. This section shows a dropdown box for choosing a pattern to instantiate. By selecting a pattern from the list and by clicking "Go" the instructor is directed to the selected pattern's administration wizard, which guides the instructor through the necessary configuration steps. Multiple examples of such wizards are given in Section 6.2.3.

Web Form Manager. The Web form manager is used to configure interactive Web forms for the current course. Its implementation is based on the Web template of the QUESTIONNAIRE form editor (see p. 233). Web forms are used/required by a number of patterns, such as QUESTIONNAIRE or PEER-EVALUATION. Figure 147 shows a series of screenshots of the Web form manager: First, a list of forms is presented, with the option of adding a new form. When editing a form, its included blocks are presented. These blocks can be rearranged and edited. The final part of the figure shows the configuration screen of the "Peer Evaluation" form's "Rating" block which is used in the case study course by the PEER-EVALUATION pattern to rate projects of peers (see the resulting Web form in the bottom part of Figure 177, p. 435.)

Participants Data. The final section allows the instructor or administrator to edit the course's structural data, which includes its groups, instructors, and most importantly, the participants. All data can be displayed and modified conveniently by clicking on one of the links in this section.

Note that the PatMan can be seen in action in the case study course in Section 6.2.3, p. 420.

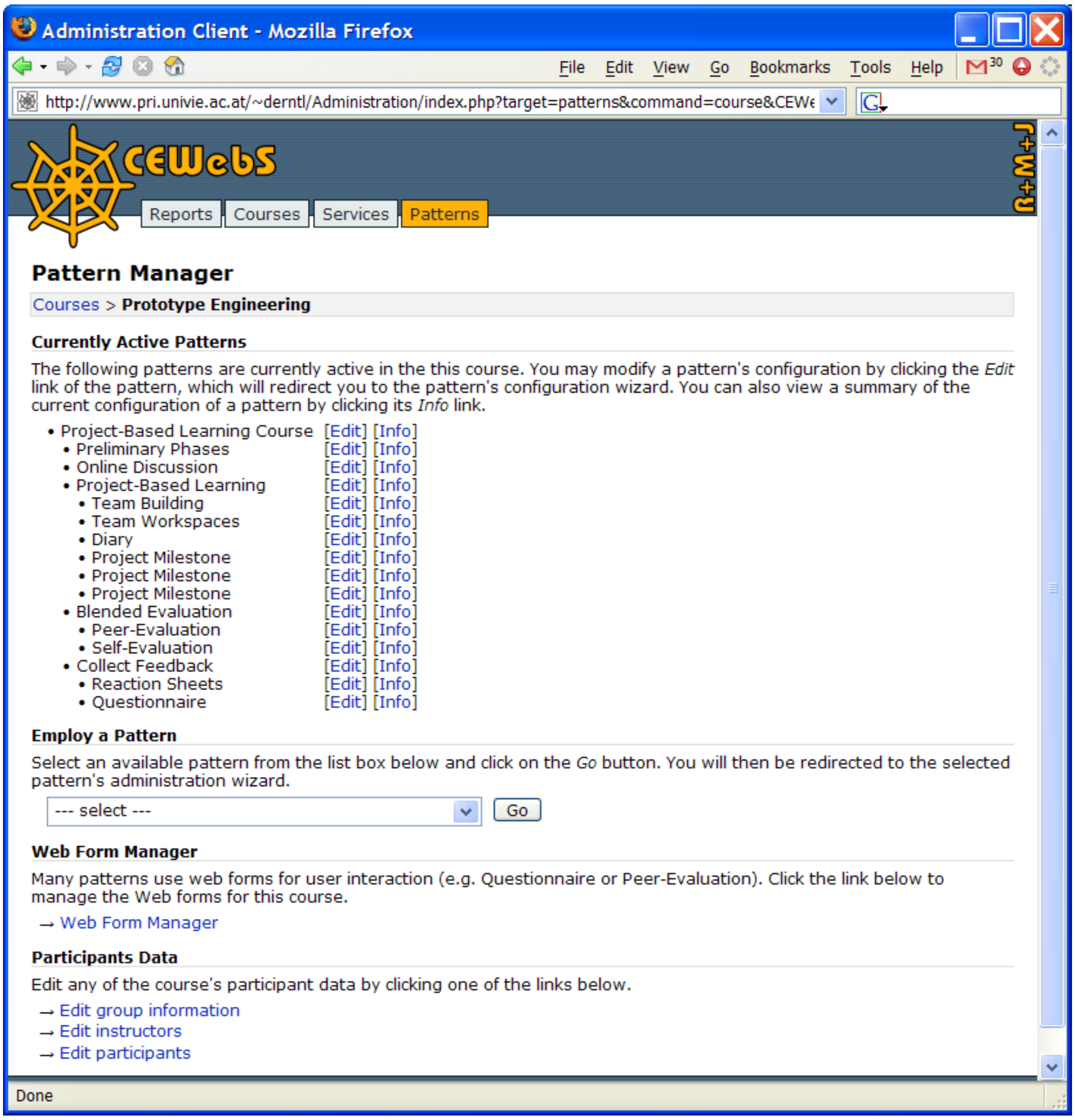
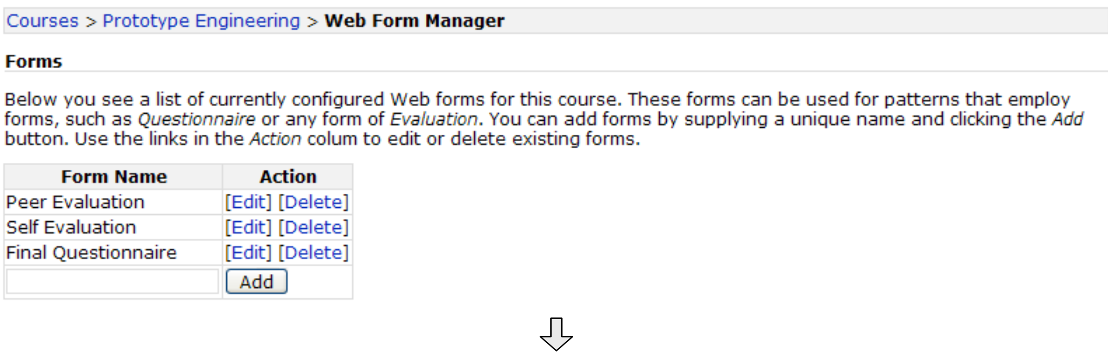
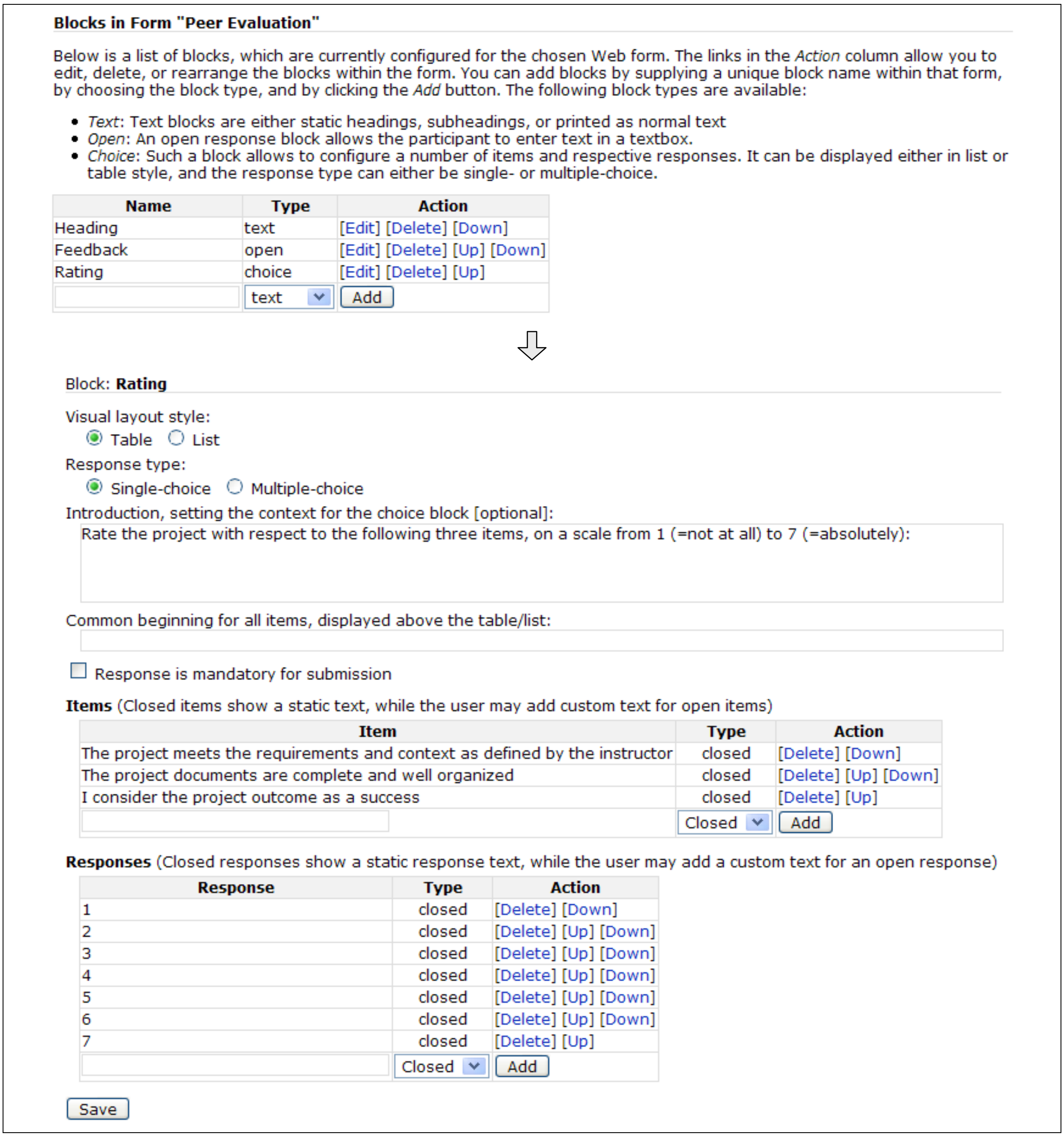


Figure 146: The Pattern Manager main page.





Blocks in Form "Peer Evaluation"

Below is a list of blocks, which are currently configured for the chosen Web form. The links in the *Action* column allow you to edit, delete, or rearrange the blocks within the form. You can add blocks by supplying a unique block name within that form, by choosing the block type, and by clicking the *Add* button. The following block types are available:

- *Text*: Text blocks are either static headings, subheadings, or printed as normal text
- *Open*: An open response block allows the participant to enter text in a textbox.
- *Choice*: Such a block allows to configure a number of items and respective responses. It can be displayed either in list or table style, and the response type can either be single- or multiple-choice.

Name	Type	Action
Heading	text	[Edit] [Delete] [Down]
Feedback	open	[Edit] [Delete] [Up] [Down]
Rating	choice	[Edit] [Delete] [Up]
	text	Add

Block: **Rating**

Visual layout style:
Table List

Response type:
Single-choice Multiple-choice

Introduction, setting the context for the choice block [optional]:
Rate the project with respect to the following three items, on a scale from 1 (=not at all) to 7 (=absolutely):

Common beginning for all items, displayed above the table/list:

Response is mandatory for submission

Items (Closed items show a static text, while the user may add custom text for open items)

Item	Type	Action
The project meets the requirements and context as defined by the instructor	closed	[Delete] [Down]
The project documents are complete and well organized	closed	[Delete] [Up] [Down]
I consider the project outcome as a success	closed	[Delete] [Up]
	Closed	Add

Responses (Closed responses show a static response text, while the user may add a custom text for an open response)

Response	Type	Action
1	closed	[Delete] [Down]
2	closed	[Delete] [Up] [Down]
3	closed	[Delete] [Up] [Down]
4	closed	[Delete] [Up] [Down]
5	closed	[Delete] [Up] [Down]
6	closed	[Delete] [Up] [Down]
7	closed	[Delete] [Up]
	Closed	Add

Save

Figure 147: The PatMan's Web form manager.

Source Code

The source code of the Pattern Manager and the TEAM WORKSPACES service, which were implemented in the scope of this thesis, is available for download as open source at http://elearn.pri.univie.ac.at/derntl/diss/. Note that the code and software are provided for demonstration/documentation purposes only, and will be useless in source or executable form without appropriately configured server software and a working CEWebS installation.

The prototype's components were written in C#, using the .NET Framework SDK v.1.1. The source code includes 118 C# files with a net total of 13,114 lines of code[577].

6.2 The Course "Prototype Engineering"

The course for the case study is called "***Prototype Engineering***". The name for this course was chosen for two reasons: First, its intent is to provide a context for the application of the Pattern Manager *Prototype*, and second, the data for populating the learning platform is taken from a real course on "Web *Engineering*".

6.2.1 Course Goals, Didactics, and Description

Overall Goals

Students acquire skills and familiarity in employing Web and software engineering methods and tools for iteratively and incrementally elaborating authentic Web projects. They learn and practice how to present their project work and results to a group of peers. Students work in small teams and are required to employ theories and methods as presented in the weekly lectures in their projects with the aim of producing a working prototype Web application for their projects.

Didactic Baseline

The overall course goals shall be achieved by a blend of online and face-to-face phases as proposed by the Person-Centered e-Learning approach.

Course Description

At the beginning the instructor populates the learning platform (LP) with initial information and resources for the course. In the first plenary meeting, the course and its goals are presented to participants. Goals and expectations are elaborated in the plenum and subsequently provided on the LP. Also, the instructor presents the overall and specific goals of the course and outlines the main teaching and learning method, i.e. project-based learning. After the first meeting, three concurrent threads of activities commence:

[577] The source code statistics were obtained using the Lines of Code Counter tool by Wolfinger (2004).

1) A series of ***weekly lectures***, where theories and background information on prototype engineering are presented and elaborated. The lecture slides are posted on the LP to make them available as learning resources.

2) A ***discussion forum*** is provided on the LP, where students can discuss general issues and consult their instructor for questions, comments, and concerns. The discussion forum is open until the end of the course.

3) The main thread of the course comprises the ***elaboration of projects***. Students are required to form teams of about 3 members each. Then, teams are encouraged to propose projects which they want to elaborate. The only constraint for project topics is that they have to be in the context of prototype engineering on the Web. In the first meeting the project visions are presented and approved by instructor. Projects are approved if they meet the contextual requirements and if their degree of complexity justifies three team members working on it for the whole term (i.e., about four months).

    a) The main part of the project-based learning (PBL) approach is accomplished in three consecutive ***project milestones***: Inception, elaboration, and construction. This milestone structure lends from the Rational Unified Process[578]. At the beginning of each milestone the instructor sets the milestone objectives and the context by providing relevant descriptions and resources. The project teams browse through these resources and elaborate their milestone solutions according to the requirements. Solutions and documents are uploaded onto a dedicated workspace on the LP. A presentation meeting, where solutions are presented and discussed, concludes each milestone.

    b) Concurrently to the PBL thread, teams are required to log their project activities in a ***Web-based diary***, which is provided on the LP. This way, the instructor can review and trace the activities and the distribution of work effort of project members.

6.2.2 Course Design

6.2.2.1 Draft Design

Based on the verbal course description above, the following Figure 148 depicts the draft of an overall activity model of the course. Such models are the primary means of knowledge communication at layer 2 (*course scenarios*) of the BLESS model. The milestones activities as referred to by the *Project milestone* subactivity are given in Figure 149. Each of these models uses a very simple notation based on the PCeL modeling approach. No references to patterns

[578] Jacobson, Booch and Rumbaugh (1999)

or other special modeling elements are used, to keep the draft design simple. But we have already colored the activities according to their presence type stereotype (present, blended, and Web-based.) Usually, speaking from the experience at our lab, most of these draft designs are first drawn by hand on paper to allow for easily erasing, modifying, and rearranging elements.

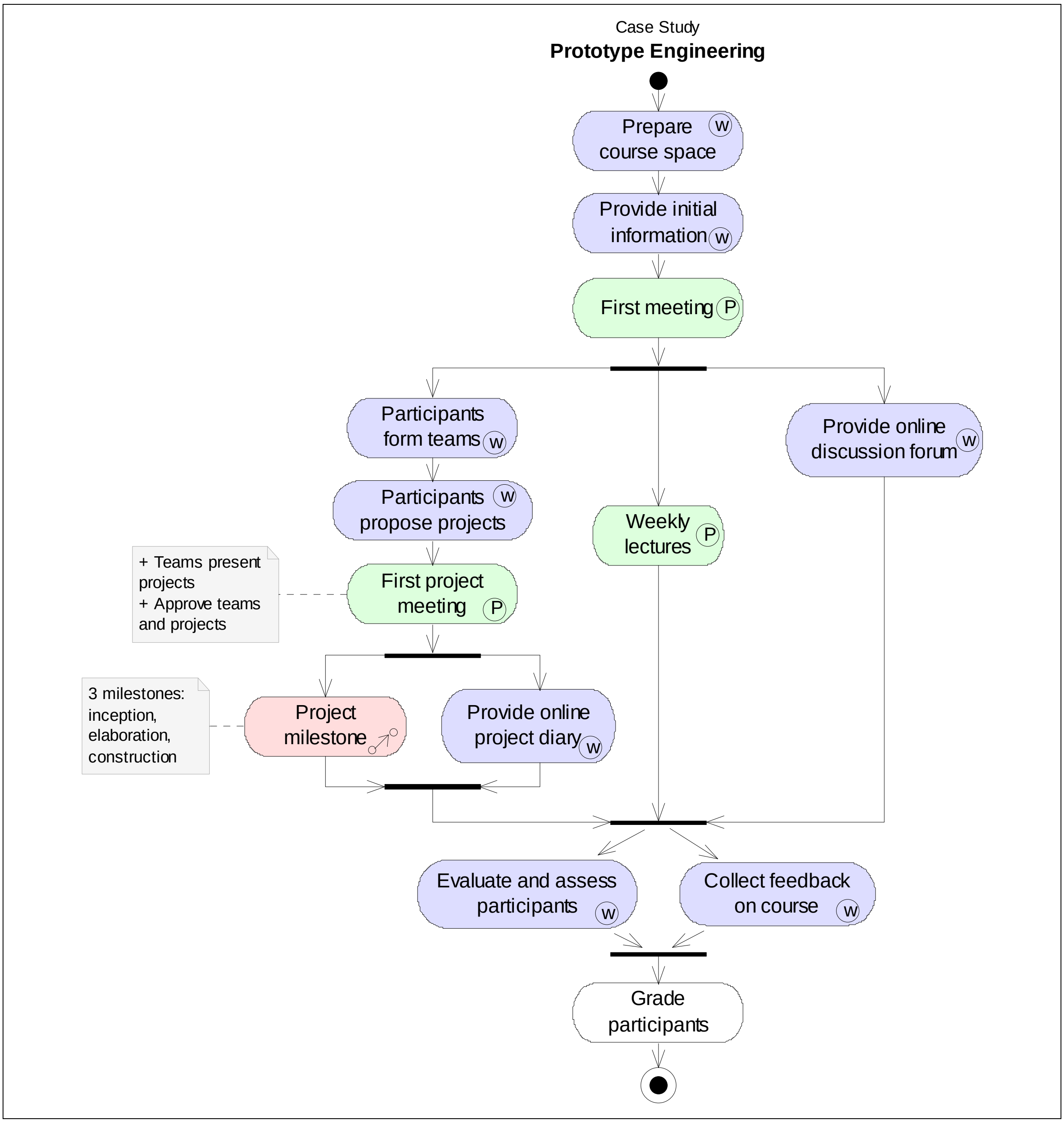


Figure 148: Overall draft design of the Prototype Engineering course.

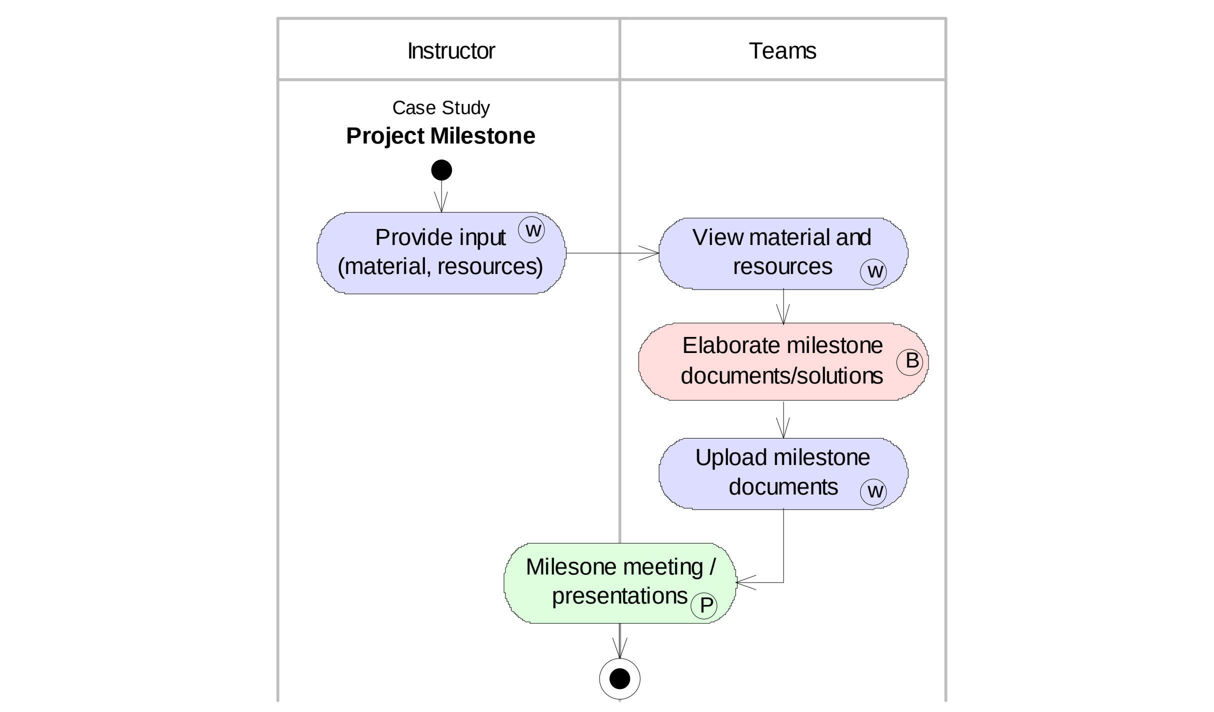


Figure 149: Draft design of project milestone activities.

6.2.2.2 PCeL-based Design

The design draft from the previous Section will now be expressed and modeled in terms of the PCeL pattern repository. Thereby, necessary steps are the following:

(1) Identification of PCeL Patterns

As a first step, which is located at the transition between layer 2 (*course scenarios*) and layer 3 (*blended learning patterns*) of the BLESS model, we identify PCeL patterns from the draft design by comparing the verbal description with the intents and motivations of the patterns in the repository and by comparing the draft model with the pattern models. The result is outlined in the following, and supported by the annotated diagram in Figure 150:

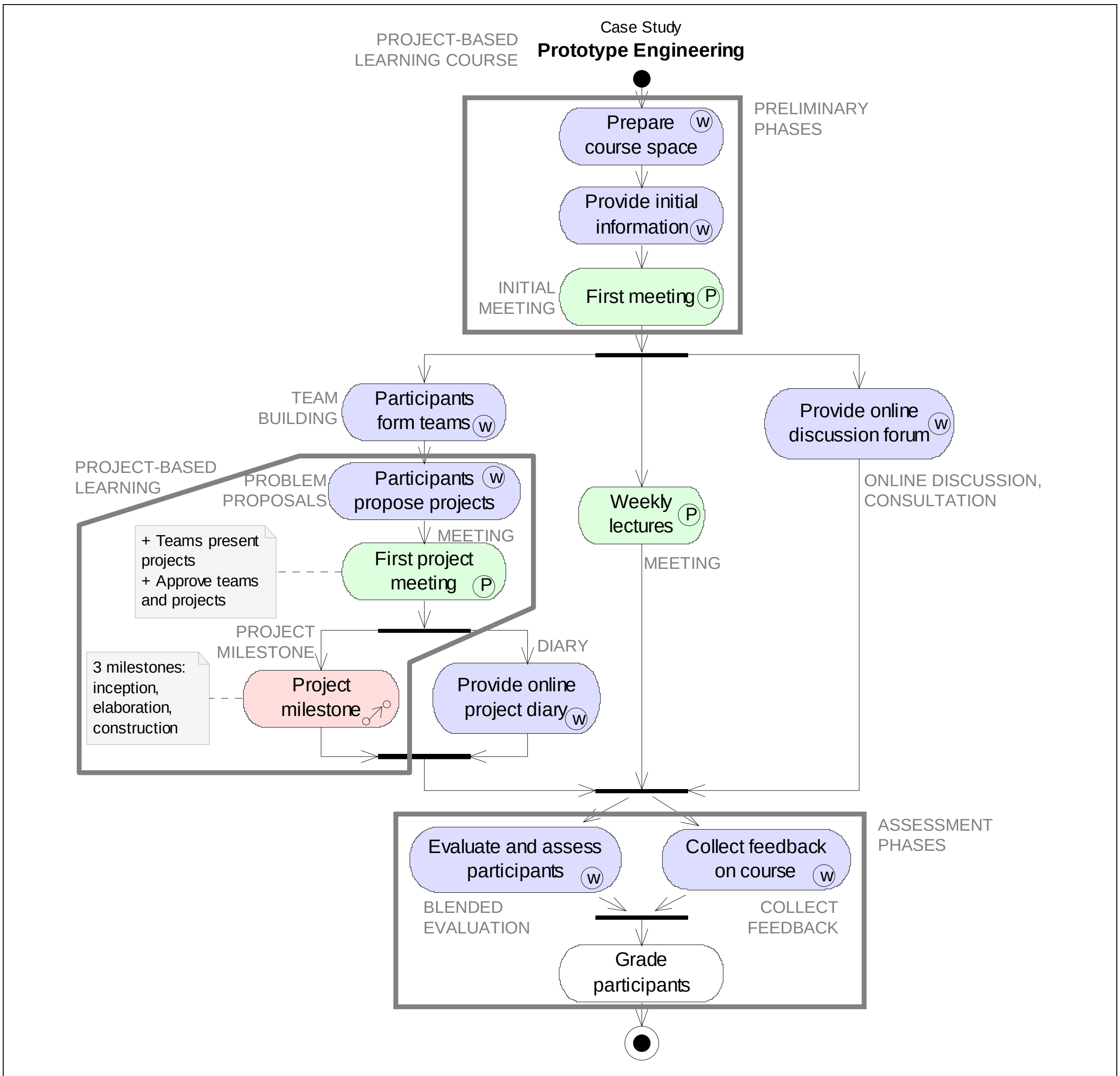


Figure 150: Annotated draft design of the Prototype Engineering course.
The thick dark-gray boxes/shapes enclose sets of activities that resemble some pattern's sequence. The pattern name is printed in gray capitals and attached to the respective box, or to a stand-alone activity that matches a pattern.

- The course starts off with initialization and by publishing initial information and resources on the LP. This Web-based thread is concluded with an initial face-to-face meeting. This sequence of activities pretty well resembles the sequence defined in PRELIMINARY PHASES (281).
- The first face-to-face meeting matches the intent of INITIAL MEETING (276). Each of the other meetings occurring throughout the course resembles a "normal" MEETING (344).
- The provision of an online communication facility for discussion and consultation as mentioned in the course description and modeled by the *Provide online discussion forum* activity matches the intents of ONLINE DISCUSSION (347) and CONSULTATION (319).

- The lecture thread matches no PCeL pattern. However, even in lectures, various derived forms of INTERACTIVE ELEMENT (335) may be used to foster communication and interaction, as well as collaborative work among participants and the instructor.
- The main teaching and learning thread of the Prototype Engineering course closely resembles a PROJECT-BASED LEARNING (387) scenario: It is preceded by TEAM BUILDING (363), where participants form teams of optimally 3 persons. Then participants propose projects as an application of PROBLEM PROPOSALS (356), and elaborate their projects in a series of consecutive instances of PROJECT MILESTONE (397).
- The project-based learning thread of the course is accompanied by the DIARY (272) pattern to allow teams and the instructor keeping track of project activities.
- The course is concluded by a form of ASSESSMENT PHASES (169) that replaces the use of GENERIC EVALUATION by BLENDED EVALUATION (186) of projects. Additionally, COLLECT FEEDBACK (221) is proposed as a form of collecting different types of written feedback.
- Finally, the whole Prototype Engineering course design represents a PROJECT-BASED LEARNING COURSE (178), as it includes PRELIMINARY PHASES at the beginning, PROJECT-BASED LEARNING in the main course phases, and concludes with ASSESSMENT PHASES based on BLENDED EVALUATION.

(2) Modeling of Course Activities

At this step, which is located at layer 3 (*blended learning patterns*) of the BLESS model, we model the course activity design using the identified PCeL patterns and extend these patterns with additional activities and learning/teaching threads as needed/required. Activity sequences that match pattern sequences are replaced by references to these patterns. If changes and/or additions to a pattern's sequence are necessary, the pattern's sequence is taken as the basic learnflow to form an activity diagram that derives from the pattern and extends it. If no matching pattern is found for a set of activities these activities can be included unchanged, or, if appropriate and justifiable, a new pattern candidate can be created. The main course model that shows the Prototype Engineering course in terms of the PCeL patterns identified before is given in Figure 151. It includes the PRELIMINARY PHASES at the beginning, followed by the main course phases comprising lectures, project-based learning, and collateral activities, and concluded by blended assessment phases. The diagram includes three subactivities that point to more detailed diagrams:

- ***Project-Based Learning*** (Figure 152): This diagram derives from PROJECT-BASED LEARNING. It starts with TEAM BUILDING and proceeds with the main PBL thread accompanied by a DIARY thread. In the PBL thread, TEAM WORKSPACES are provided, where the participants subsequently upload their PROBLEM PROPOSALS, and *Milestone* solutions. Before the main milestones commence, there is a project proposal MEETING where teams present their projects and the instructor approves teams and projects.

- *Milestone* (Figure 153): The PBL thread consists of three instances of this activity sequence, which is derived from PROJECT MILESTONE. First, milestone spaces are initialized on the TEAM WORKSPACES. The instructor publishes relevant material and descriptions there. The project teams explore that material as well as the information and theories presented in the lectures and use it for elaboration of milestone solutions for their projects. If required or desired, the instructor offers the option of hosting intermediate meetings during the elaboration phase as preparation for the concluding milestone MEETING. In this meeting, teams present their milestone solutions and the instructor as well as peers provide feedback and discuss the solutions.

- ***Collateral Online Activities*** (Figure 154): To provide means for general discussions among students and for online consultation of the instructor, the ONLINE DISCUSSION pattern is employed in combination with the CONSULTATION pattern, whereby the online consultation takes places in a dedicated forum of the online discussion facility.

- ***Blended Assessment Phases*** (Figure 155): This activity sequence is derived from ASSESSMENT PHASES; GENERIC EVALUATION was replaced by BLENDED EVALUATION including self-, peer-, and instructor-evaluation. Additionally it includes *Collect Feedback*:

  - ***Collect Feedback*** (Figure 156): This activity sequence derives from COLLECT FEEDBACK and replaces the abstract *feedback phase* with instances of QUESTIONNAIRE (structured, quantitative feedback) and REACTION SHEETS (unstructured, qualitative feedback).

All other elements of the diagram are either direct references to pattern sequences or normal activities and control flow elements.

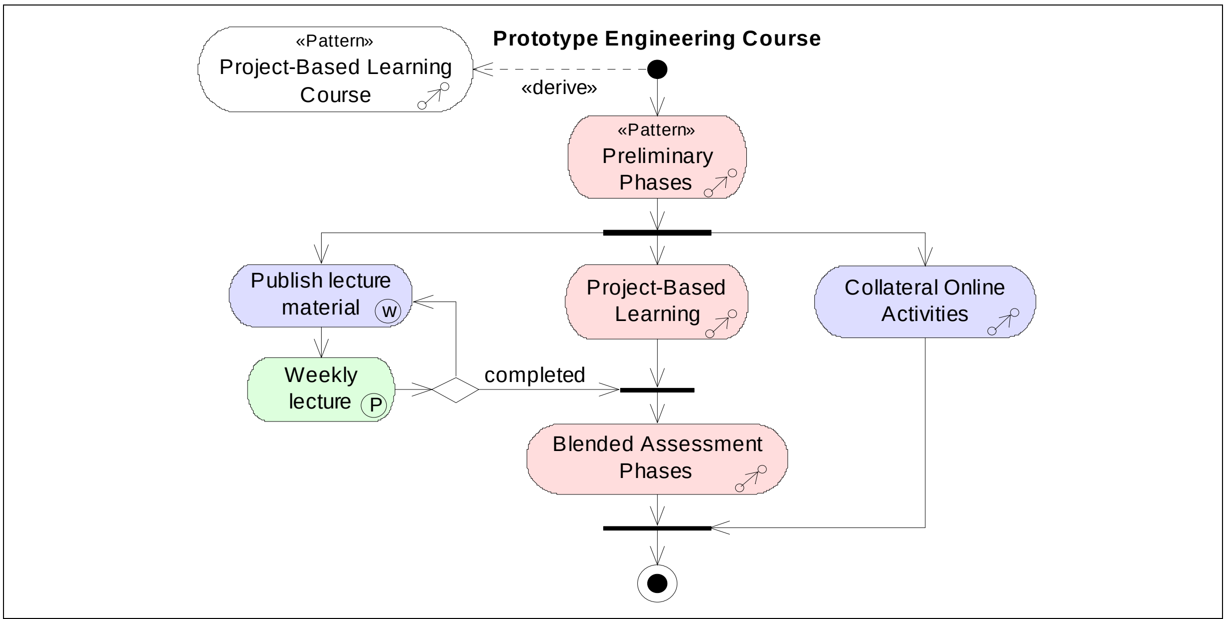


Figure 151: Main Prototype Engineering course model.

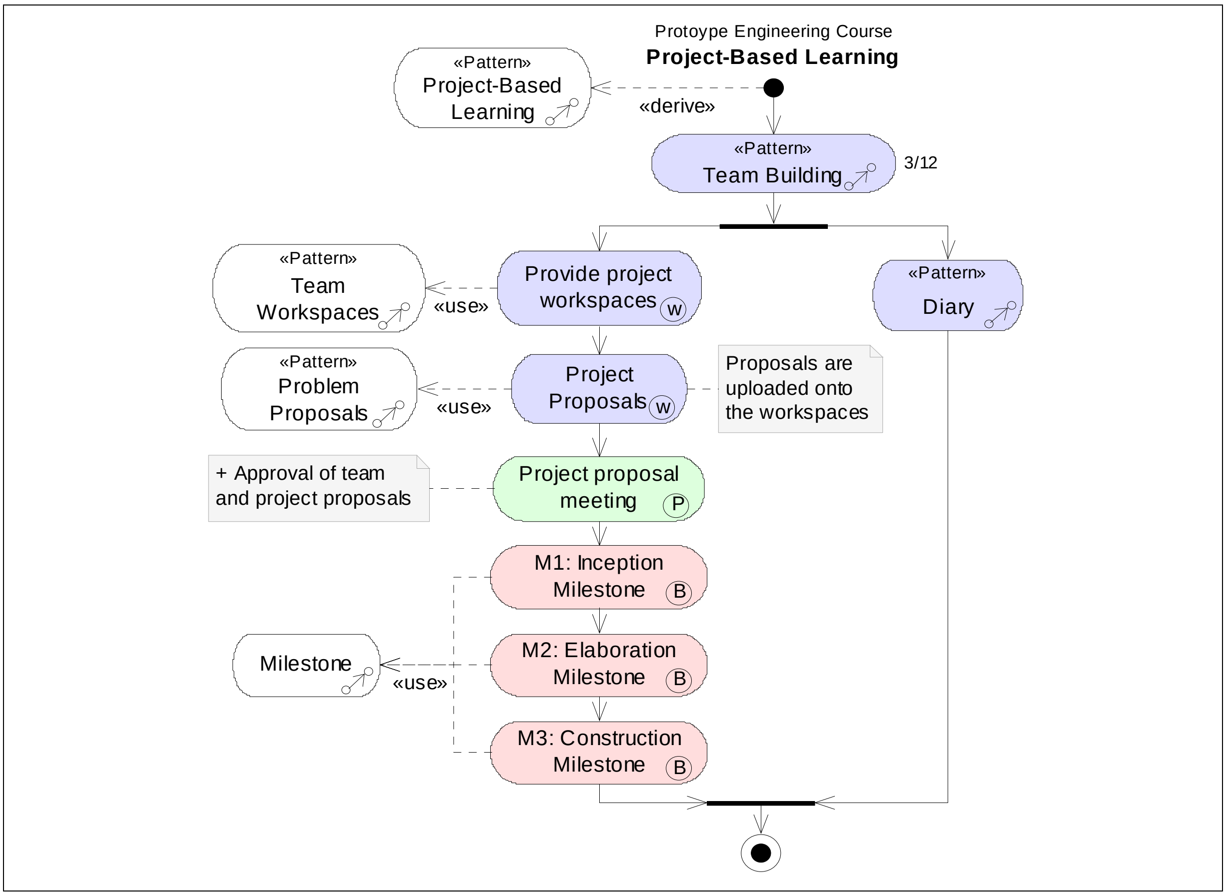


Figure 152: The "Project-Based Learning" thread of the Prototype Engineering course.

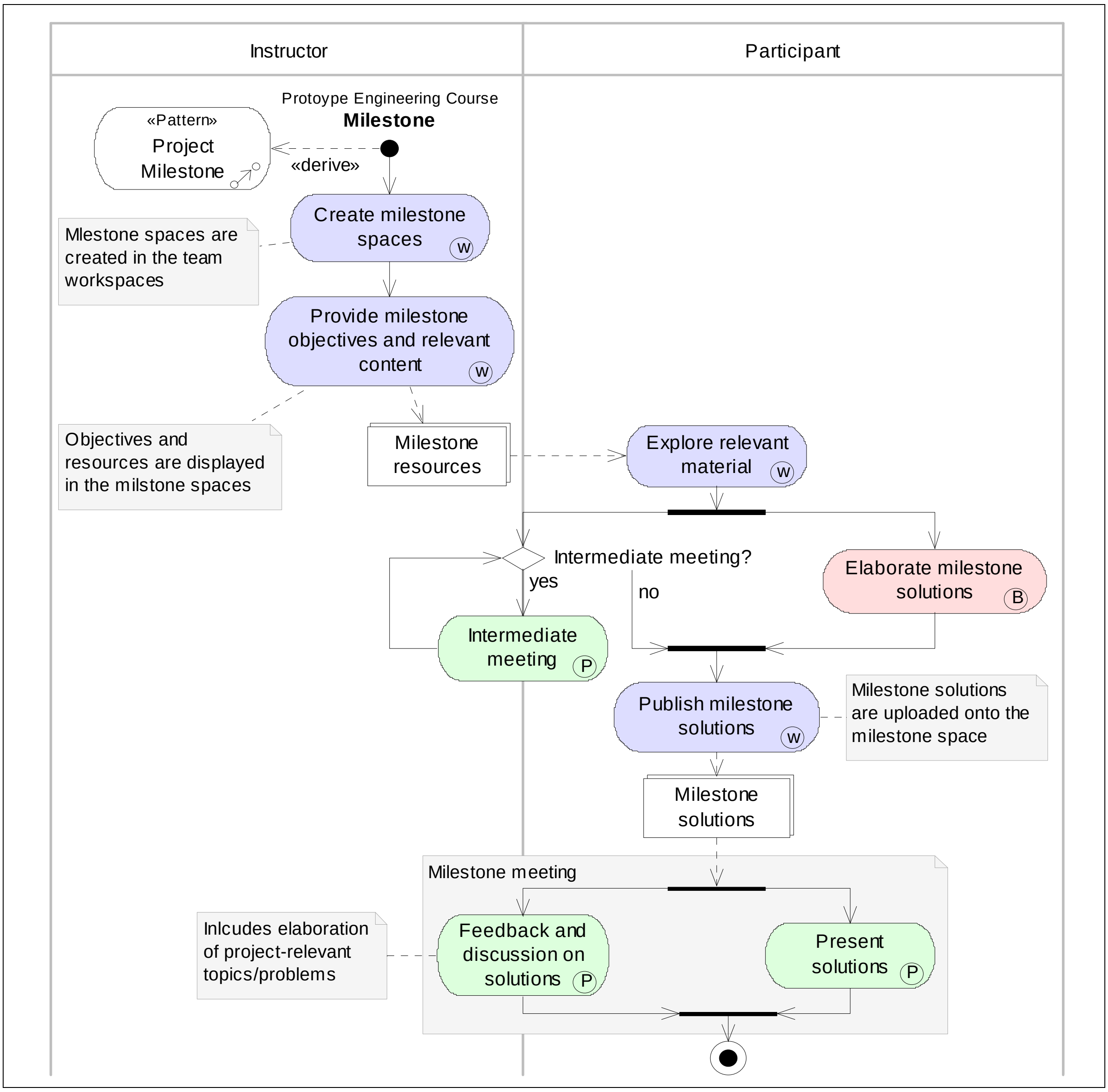


Figure 153: Project "Milestones" in the Prototype Engineering course.

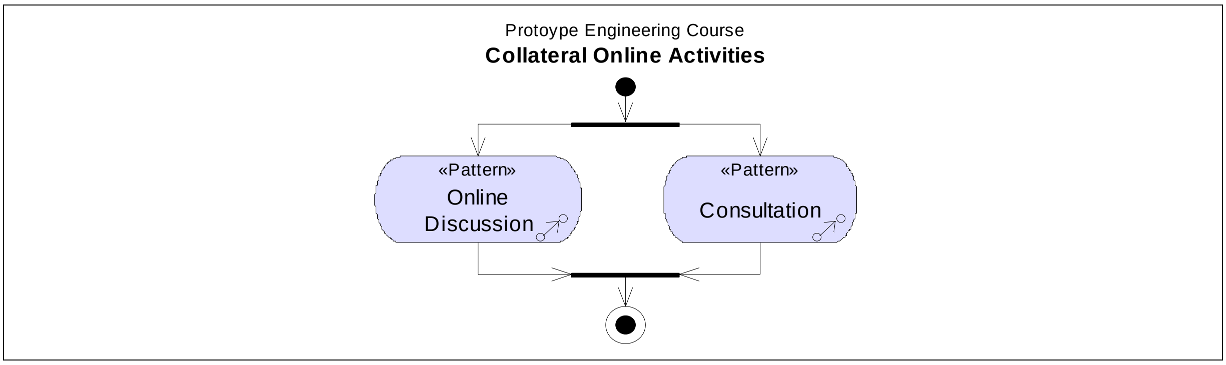


Figure 154: The "Collateral Online Activities" thread of the Prototype Engineering course.

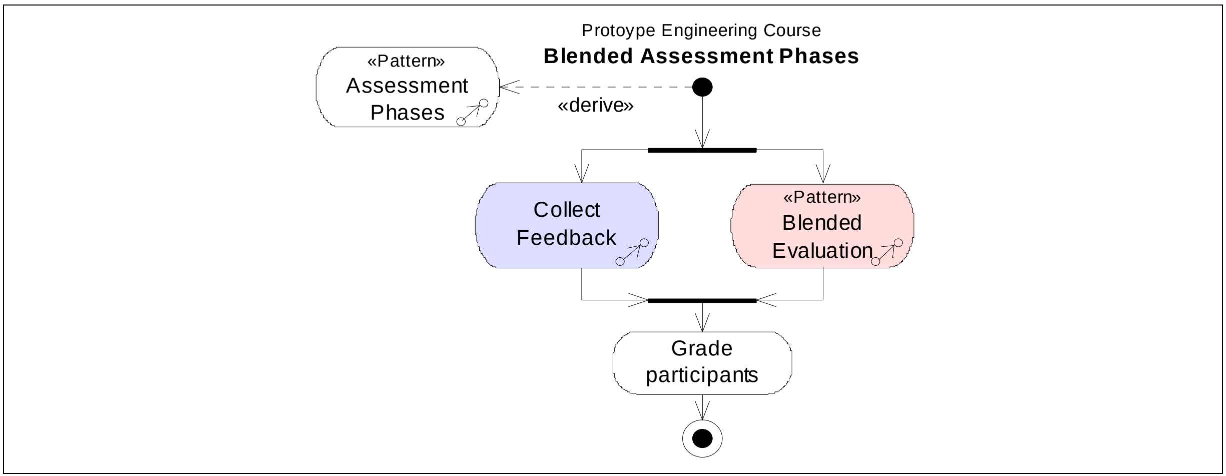


Figure 155: The "Blended Assessment Phases" thread of the Prototype Engineering course.

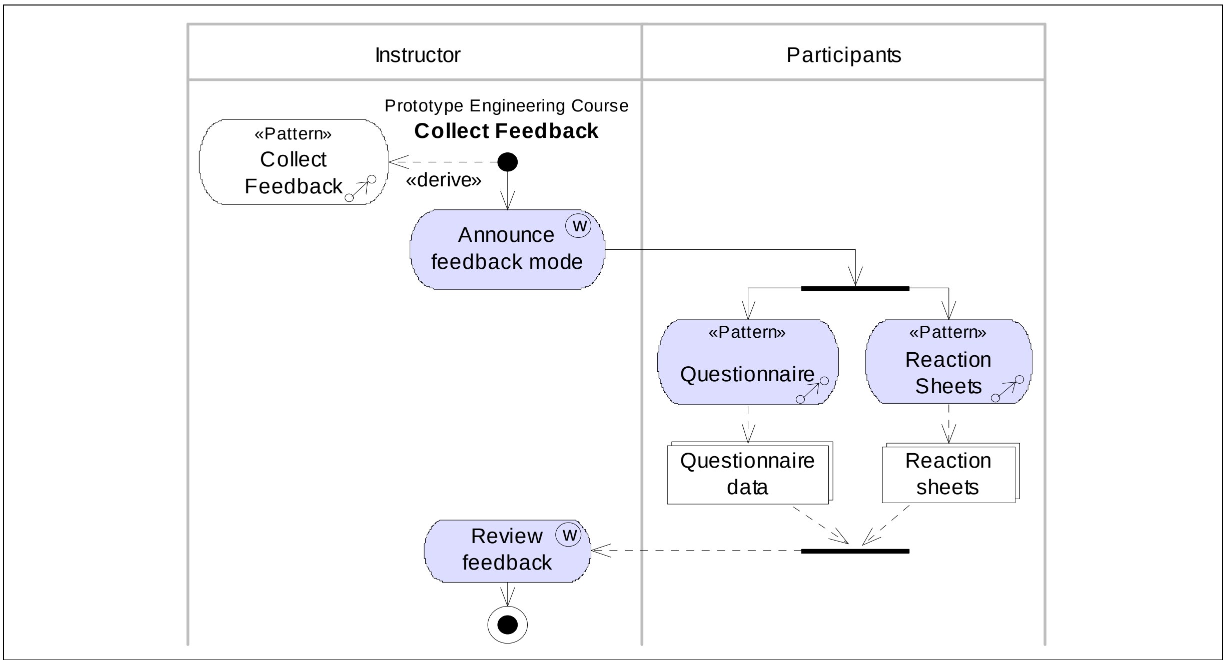


Figure 156: "Collect Feedback" in the blended assessment phases of the Prototype Engineering course.

(3) Structural Model of Diagrams and Patterns

At this step, we create a structural model of the diagrams and patterns involved to provide an overview for better understanding of activity models involved in the course. The structural model is given in Figure 157. It comprises a constrained view on the whole pattern repository, including the diagrams of the Pattern Engineering course and their structural relationships (dependency, generalization/specialization) with the repository patterns.

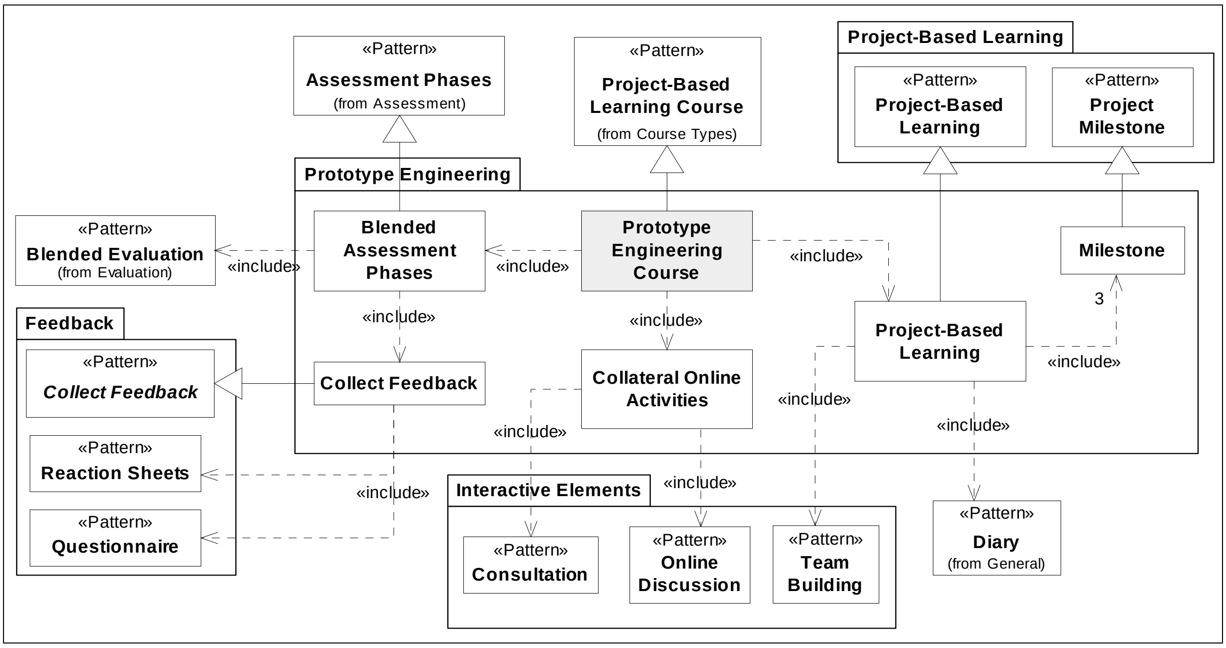


Figure 157: Structural model of the Prototype Engineering course.

6.2.3 Application of the Course Models

In this Section we show how the Pattern Manager Prototype (in the following simply referred to as "*PatMan*"), which implements the Web templates of the patterns in the PCeL pattern repository, is used to instantiate the patterns and the Web-based activities of the Prototype Engineering course. We show how the instructor is supported by simple Web-based wizards that allow for applying and instantiating patterns on the learning platform. In using the pattern models provided in the previous section (BLESS layer 3), we now show how to make the transition via layer 4 (*Web templates*) to layer 5 (*learning technology*) of the BLESS model. Finally, we reach the integration of the top-most BLESS layer (*learning theory and didactic baseline*) with the bottom-most BLESS layer. This way, we demonstrate how the BLESS model and the PCeL patterns can be used to "fill the gap" between learning theory and learning technology support for a particular course.

Course Initialization

First we need to initialize the course space on the learning platform. In our case, we employ the CEWebS learning platform, and the course data (including structure, participants, etc.) is provided by the faculty's information system called ISWI. So the first step is to create a course space on CEWebS using the CEWebS course manager (Figure 158) and to import course data from the ISWI system using the PatMan (Figure 159). After these steps we have an empty participant view at the course homepage.

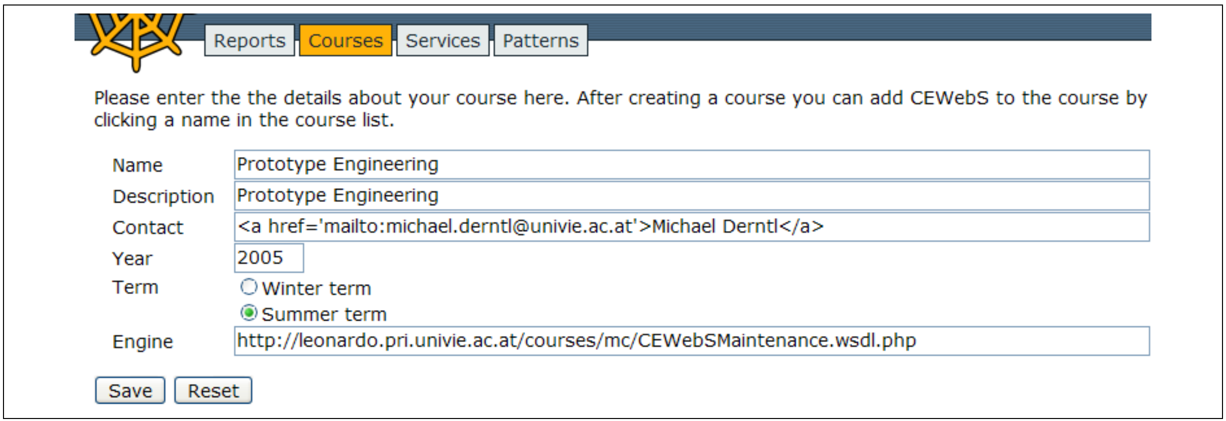


Figure 158: Creating the Prototype Engineering course space on CEWebS.

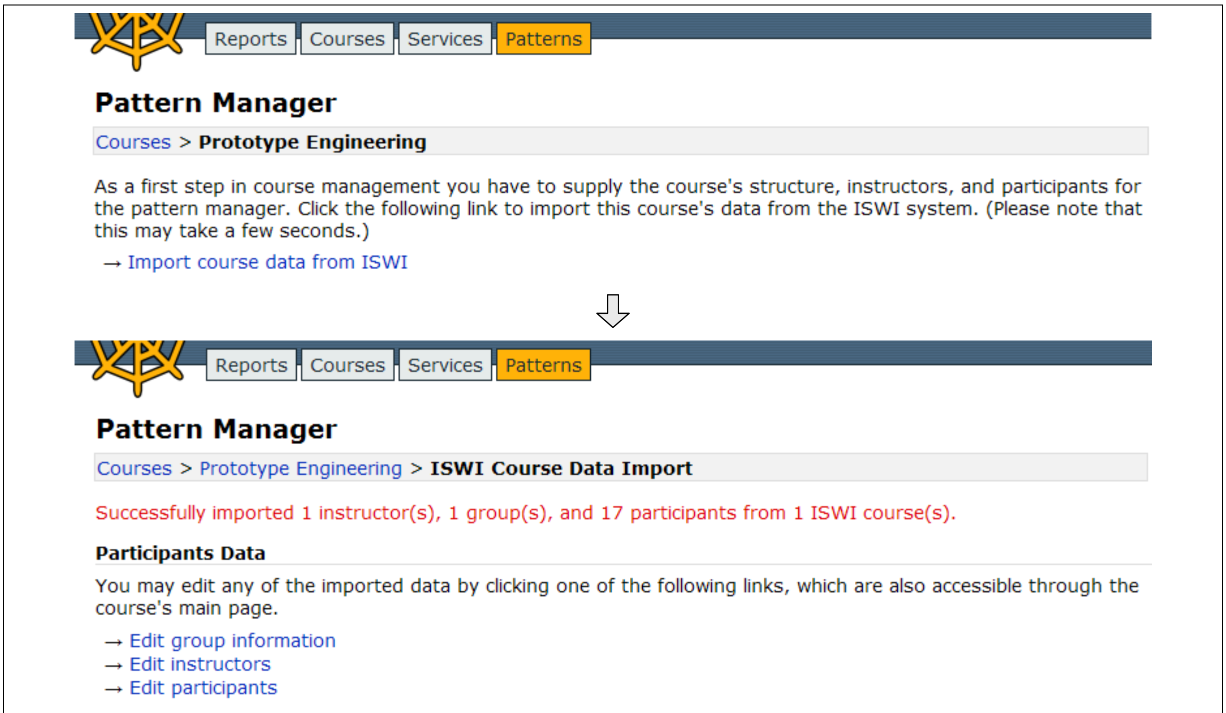


Figure 159: Importing course and participant data from the ISWI system.

Project-Based Learning Course

Now we are ready to kick off the course by starting a PROJECT-BASED LEARNING COURSE pattern instance. As this is a purely composite pattern, the PatMan shows a list of included patterns and suggests employing these in a stepwise fashion (Figure 160). Additionally all patters that include the category parameter “collateral” are available for use in the dropdown box at the bottom (e.g. ONLINE DISCUSSION, DIARY, etc.)

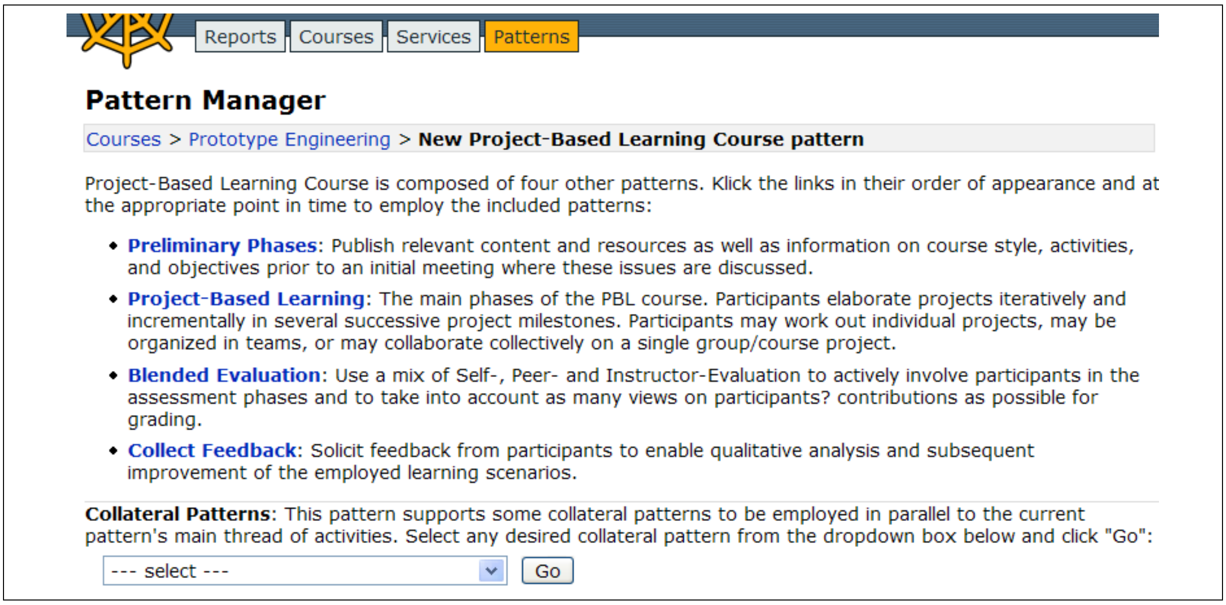


Figure 160: Instantiating the PROJECT-BASED LEARNING COURSE pattern.

Preliminary Phases

The PBL Course wizard first offers to instantiate the PRELIMINARY PHASES of the course. After submitting the simple form in Figure 161, the PatMan creates a course homepage, which is based on *WIKI* technology[579]. Afterwards, the PatMan immediately redirects the instructor to edit the homepage skeleton (Figure 162). After initial course information was provided and saved, the initial participant view of the homepage is depicted in Figure 163.

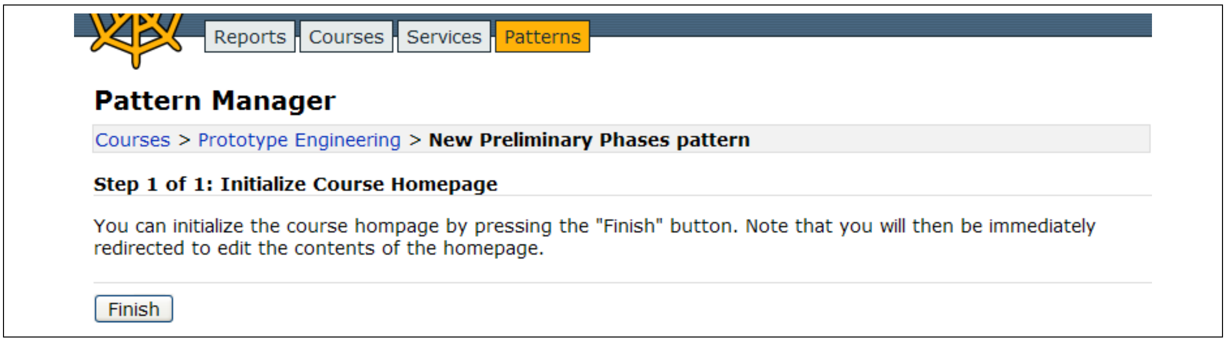


Figure 161: Instantiating PRELIMINARY PHASES in the PatMan.

Note that at the end of PRELIMINARY PHASES, the INITIAL MEETING is an important face-to-face encounter that is used by the instructor to prepare the participants for the following phases, which include the *Collateral Online Activities* thread, as well as first activities in the PROJECT-BASED LEARNING thread (such as TEAM BUILDING and uploading project proposals onto the TEAM WORKSPACES.)

[579] "Wiki" is the Hawaiian word for "quick". Wiki technology allows collaboratively editing Web pages based on a very simple, intuitive syntax. For one of the most popular Wikis refer to http://en.wikipedia.org

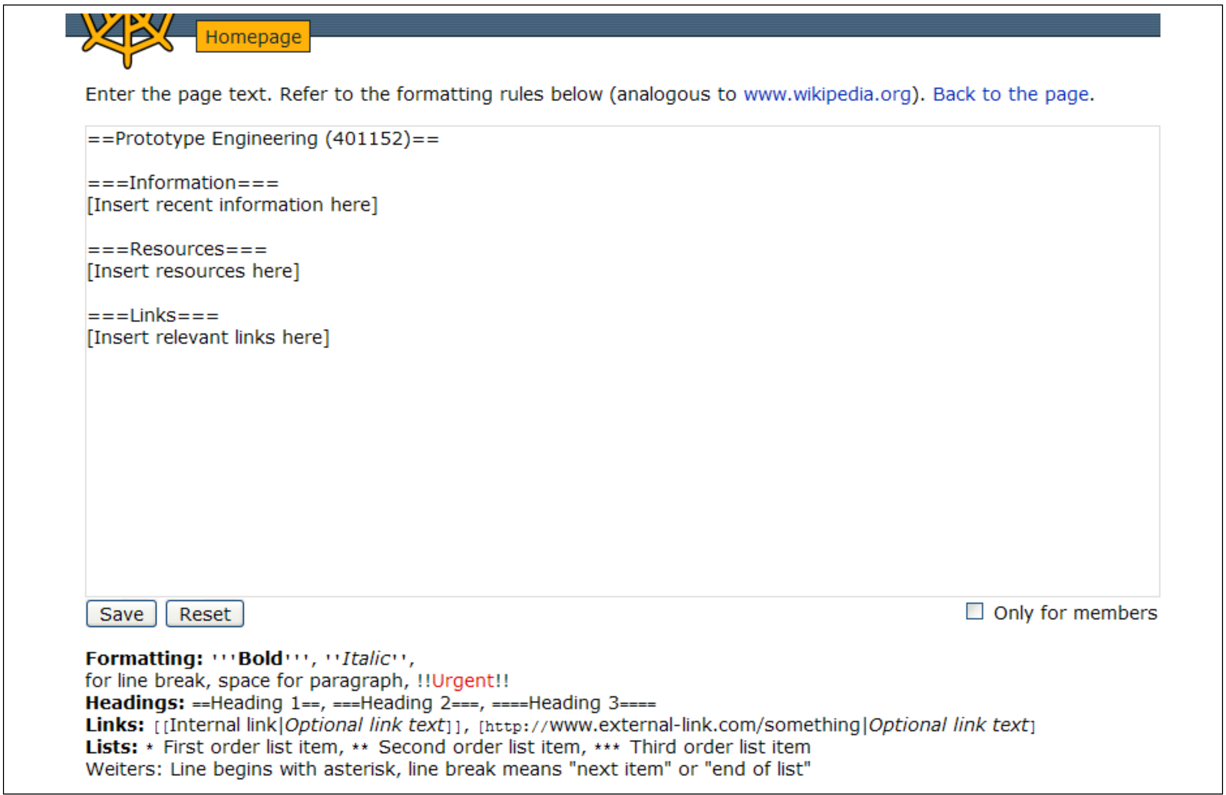


Figure 162: Initial edit screen of the Prototype Engineering course homepage.

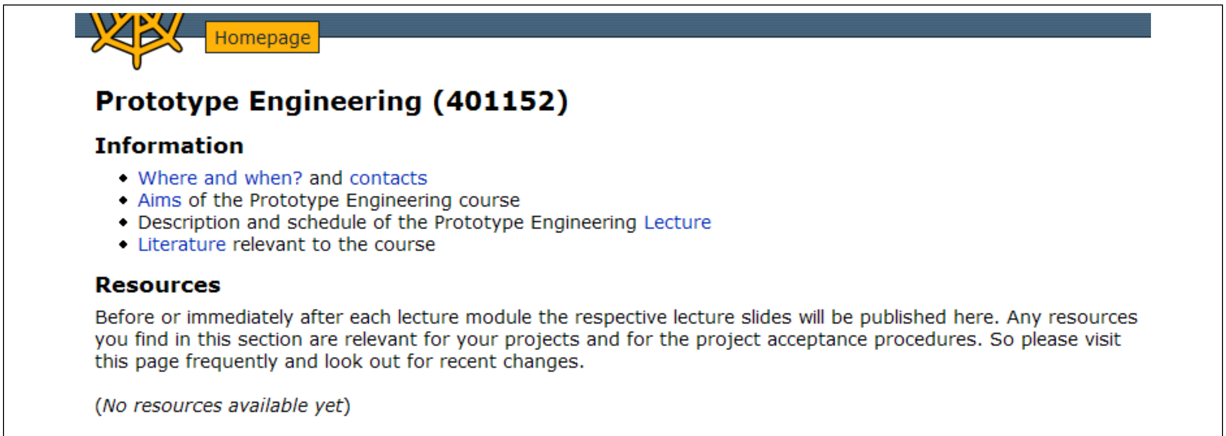


Figure 163: Initial information on the course homepage (participant view.)
Note that the “Links” section was removed because no links are (yet) available to be provided.

Collateral Online Activities

This is one of the three main activity threads in the course. It offers an Online Discussion forum for general discussions and for online Consultation of the instructor. As both forums were chosen to be hosted in the same discussion facility, we can instantiate both patterns using the pattern wizard which implements the Online Discussion Web template (see Figure 164).

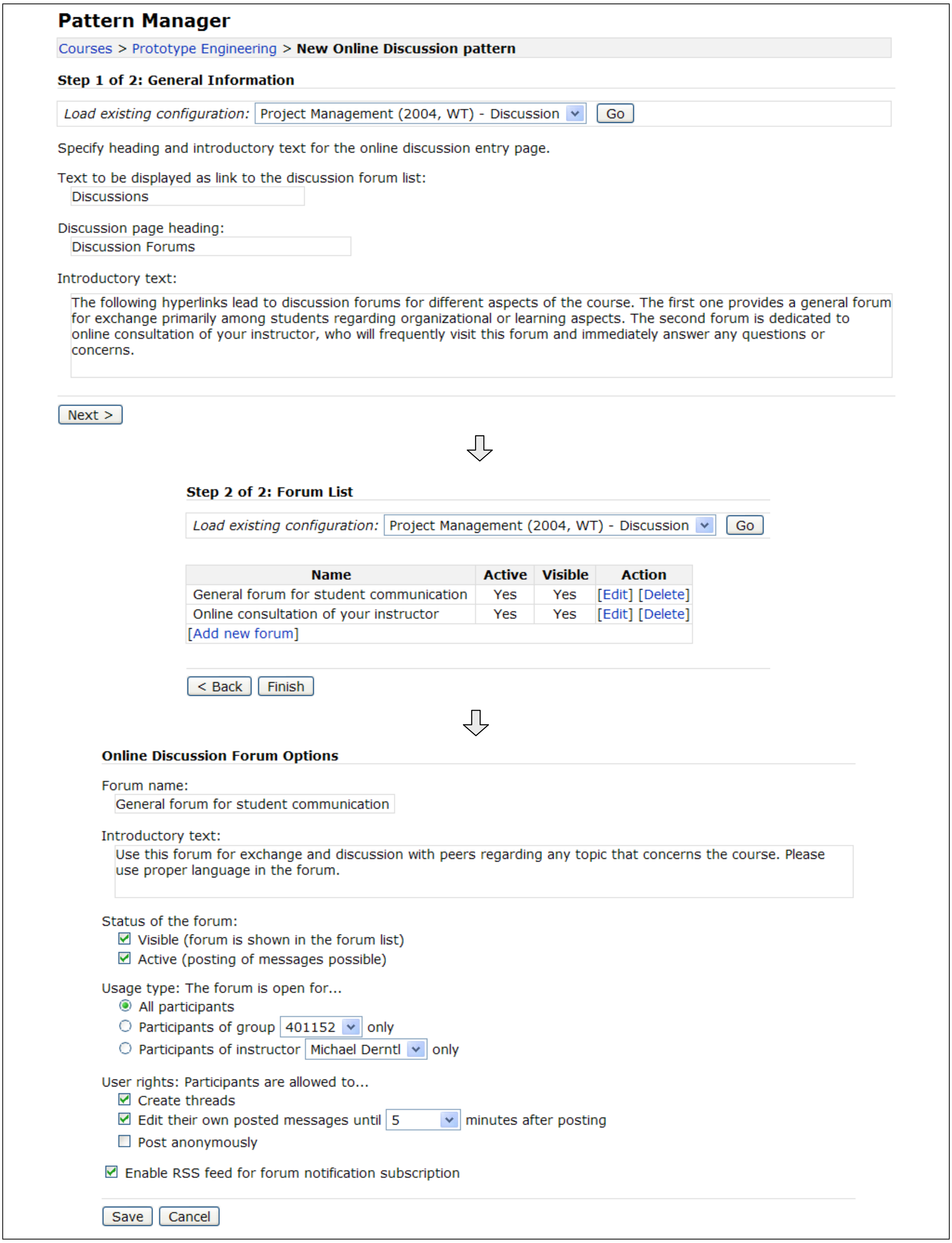


Figure 164: Administration wizard for ONLINE DISCUSSION.

After the wizard is finished, the PatMan creates a new page in the participant view of the course, called "Discussion". This page contains the introduction to the discussion forums as well as links to the general discussion forum and to the consultation forum. Additionally, the instructor posts an initial welcome message to the general forum (see the sequence of screenshots in Figure 165.)

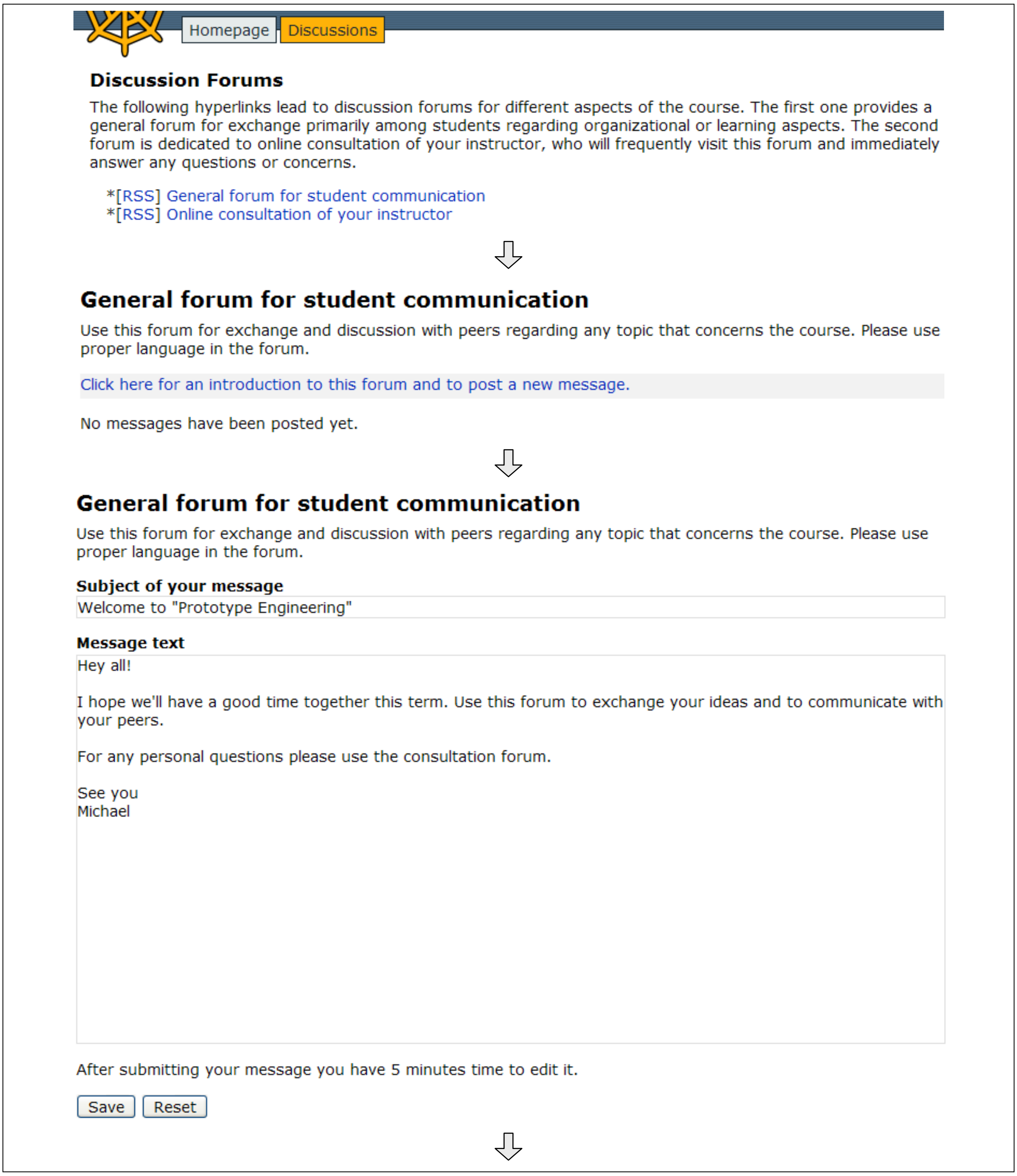


Figure 165: Participant view of the *Collateral Online Activities* thread.

Project-Based Learning

Now we have to initialize the main teaching/learning thread of the course by instantiating the PROJECT-BASED LEARNING pattern. The implementation of PBL in the PatMan is based on the TEAM WORKSPACES pattern, which is one feasible option as presented in the Web template section of PBL. For each project milestone a dedicated folder is provided in the workspaces. This folder is used by the instructor to publish relevant milestone information and by teams to upload their milestone documents. Following that approach, the composite PBL pattern shows links to included patterns in the administration wizard (Figure 166).

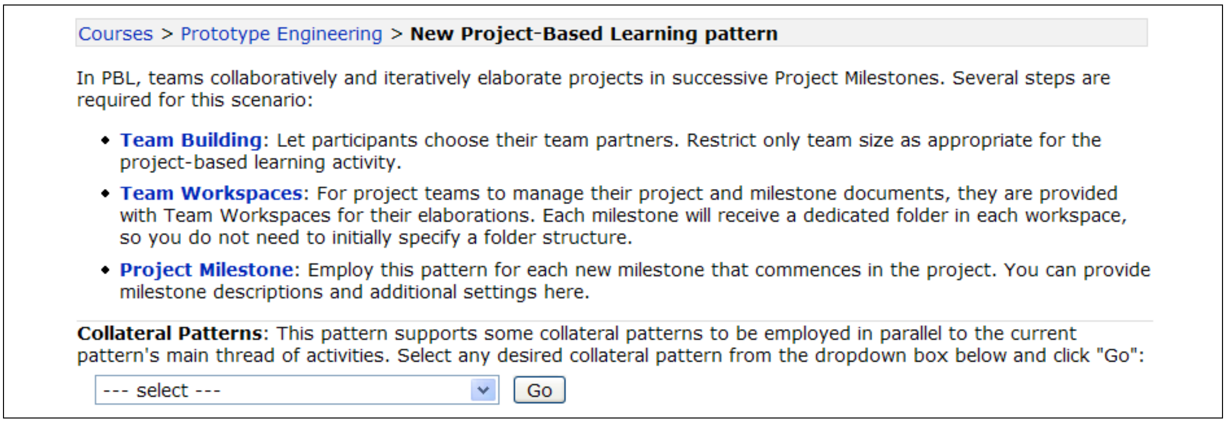


Figure 166: Administration wizard for Project-Based Learning.

Team Building

First, the instructor is redirected to the administration wizard of TEAM BUILDING as existing teams are a prerequisite for using TEAM WORKSPACES. So the first phase of PBL, which starts right after the INITIAL MEETING, requires participants to build teams in the TEAM BUILDING participant view on the LP. Both the participant view and the administration view are implemented as defined in the TEAM BUILDING Web template. In the administration wizard (Figure 167) relevant information is provided, team size is restricted to 2–4 members, and the team building deadline is set to one week after the Initial Meeting.

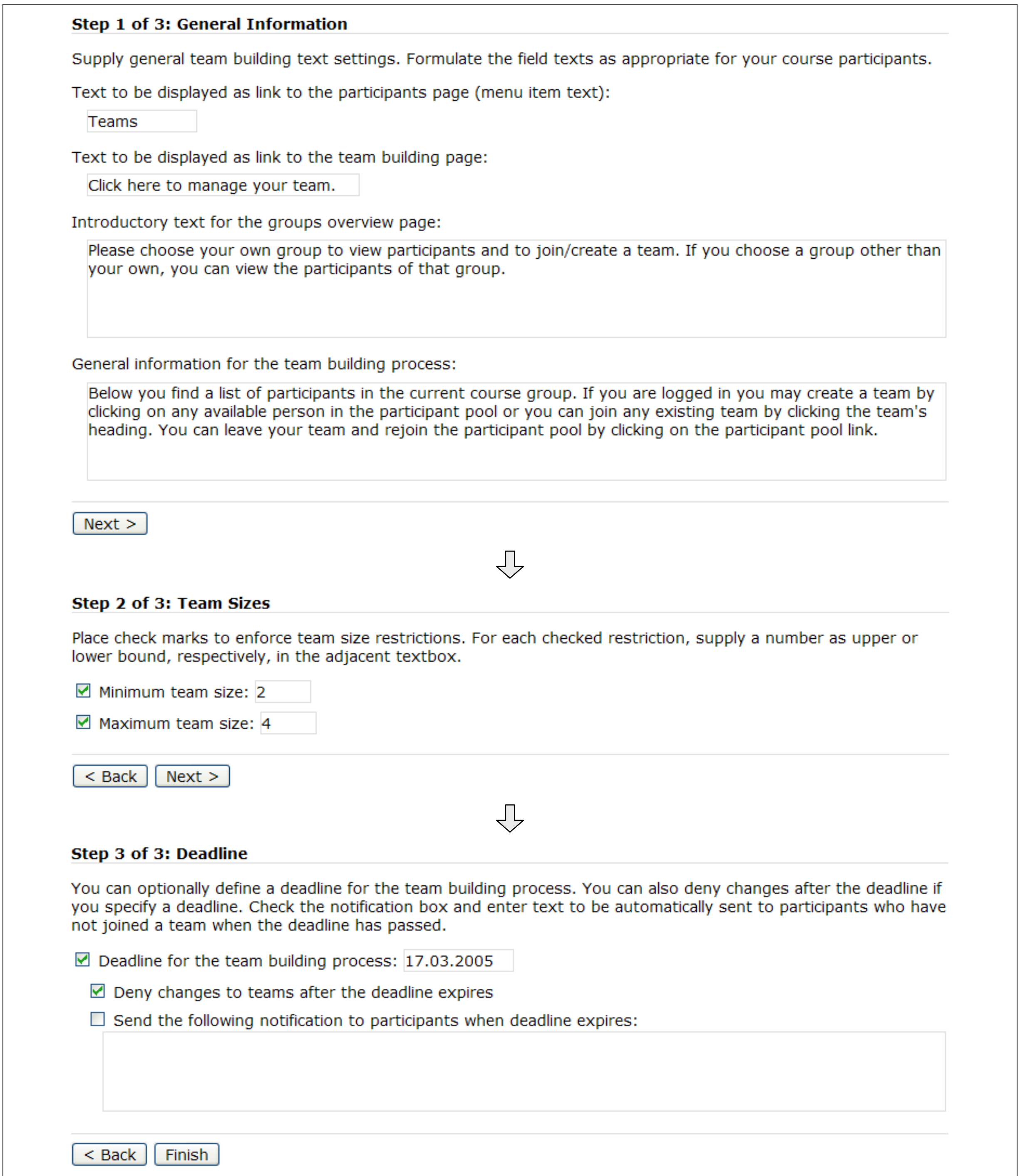


Figure 167: Administration wizard for Team Building.

The screenshot in Figure 168 shows the team building participant view after the deadline, when all teams are set. Note that the names, student ID’s, and e-mail addresses of participants have been changed to protect privacy of the students that were imported from the faculty’s information system by the PatMan.

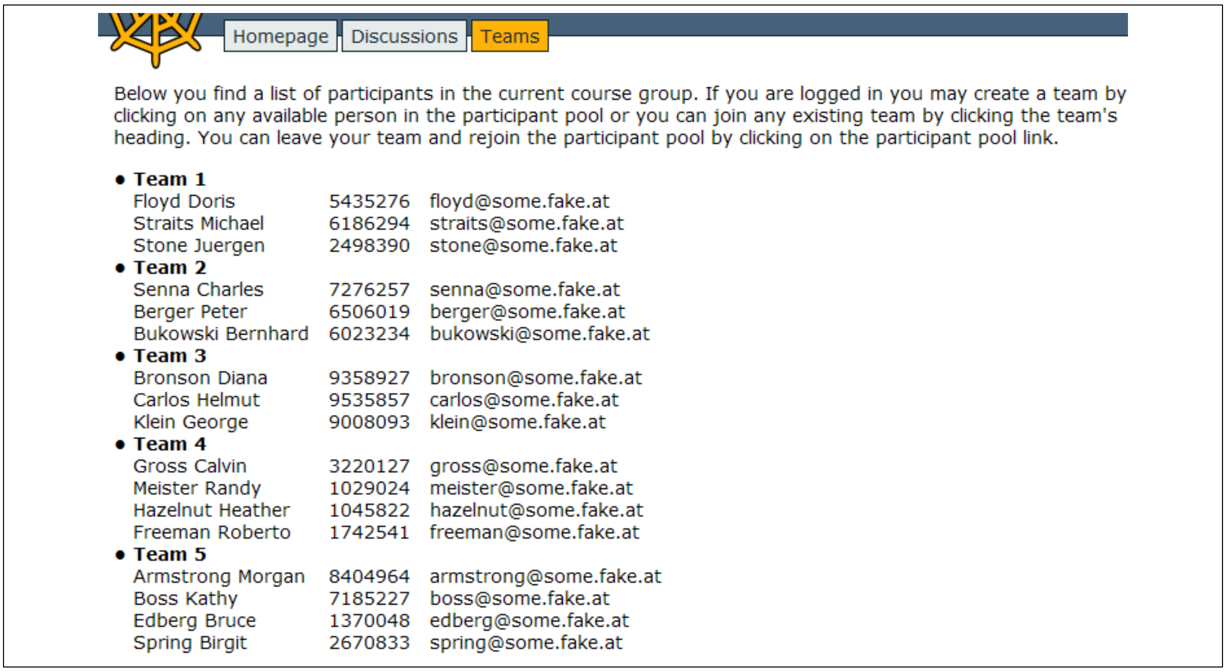
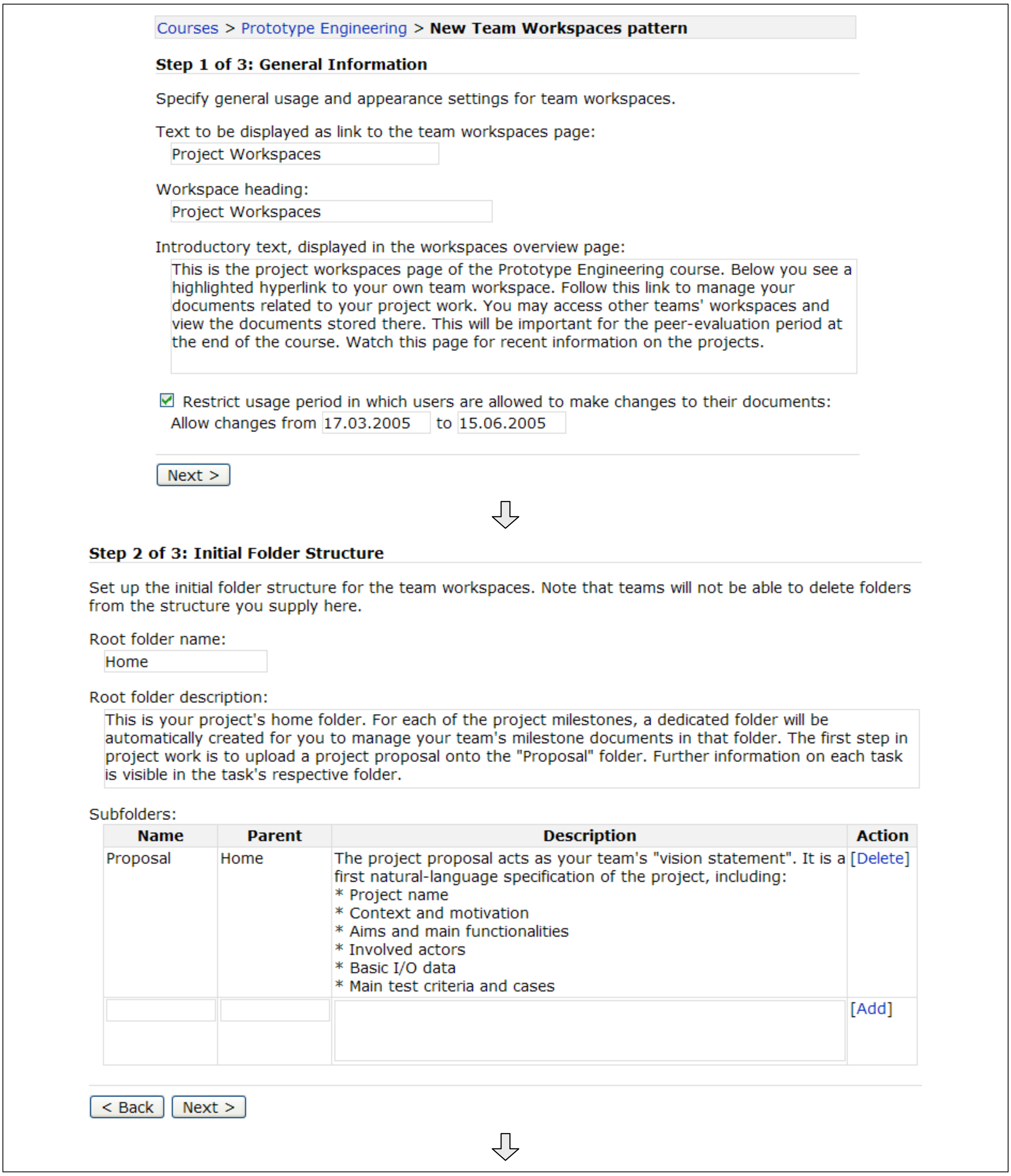


Figure 168: The TEAM BUILDING page on the learning platform after completion of the team building process.

Team Workspaces

After the teams are built, the instructor instantiates the TEAM WORKSPACES pattern as suggested by the PBL wizard. As the workspaces act as the main vehicle for Web-support of the PBL pattern the configuration of the workspaces is done as evident from the screenshots in Figure 169. Note that the *project proposal* activity is merged with the TEAM WORKSPACES initialization by supplying a folder called “Project Proposals” in the initial folder structure of the workspaces. Also note that the workspaces are “closed” about 2 weeks prior to the end of the course to deny changes to documents during the BLENDED EVALUATION period (see later). However, if there are any unexpected delays during the course, the instructor can edit the pattern’s configuration using the PatMan and set the end of the usage period to another date.


Courses > Prototype Engineering > New Team Workspaces pattern
Step 1 of 3: General Information
Specify general usage and appearance settings for team workspaces.
Text to be displayed as link to the team workspaces page:
Project Workspaces
Workspace heading:
Project Workspaces
Introductory text, displayed in the workspaces overview page:
This is the project workspaces page of the Prototype Engineering course. Below you see a highlighted hyperlink to your own team workspace. Follow this link to manage your documents related to your project work. You may access other teams' workspaces and view the documents stored there. This will be important for the peer-evaluation period at the end of the course. Watch this page for recent information on the projects.
Restrict usage period in which users are allowed to make changes to their documents:
Allow changes from 17.03.2005 to 15.06.2005
Next >
Step 2 of 3: Initial Folder Structure
Set up the initial folder structure for the team workspaces. Note that teams will not be able to delete folders from the structure you supply here.
Root folder name:
Home
Root folder description:
This is your project's home folder. For each of the project milestones, a dedicated folder will be automatically created for you to manage your team's milestone documents in that folder. The first step in project work is to upload a project proposal onto the "Proposal" folder. Further information on each task is visible in the task's respective folder.
Subfolders:
Name
Parent
Description
Action
Proposal
Home
The project proposal acts as your team's "vision statement". It is a first natural-language specification of the project, including:
* Project name
* Context and motivation
* Aims and main functionalities
* Involved actors
* Basic I/O data
* Main test criteria and cases
[Delete]
[Add]
< Back
Next >

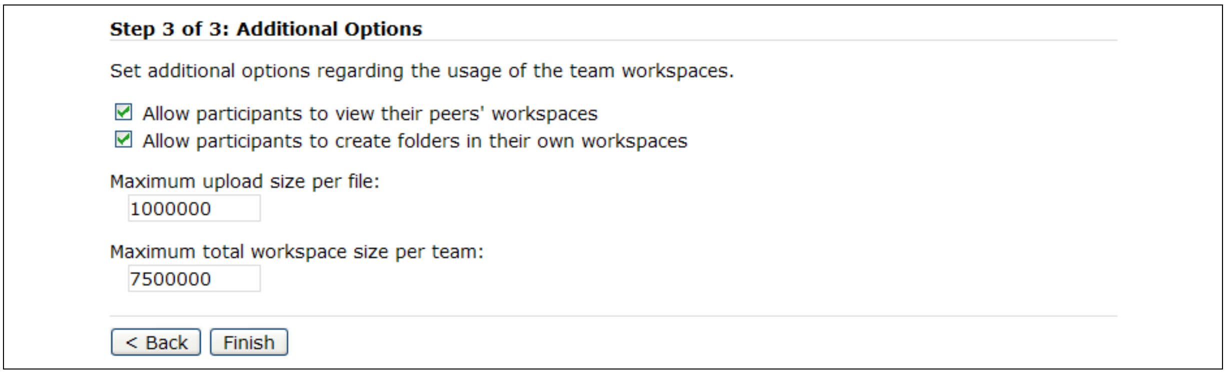


Figure 169: The Team Workspaces administration wizard.

In the participant view on CEWebS, the workspace overview page holds the general information on the projects, and links to the teams' workspaces. Workspaces are open to all participants, even those of other teams. However, modifications like uploading and deleting files, creating folders, etc., is only possible in one's own workspace. Figure 170 shows the overview page as well as the workspace of team 1.

After the workspaces are initialized, project teams can start using them for managing all project-related documents in the predefined folders.

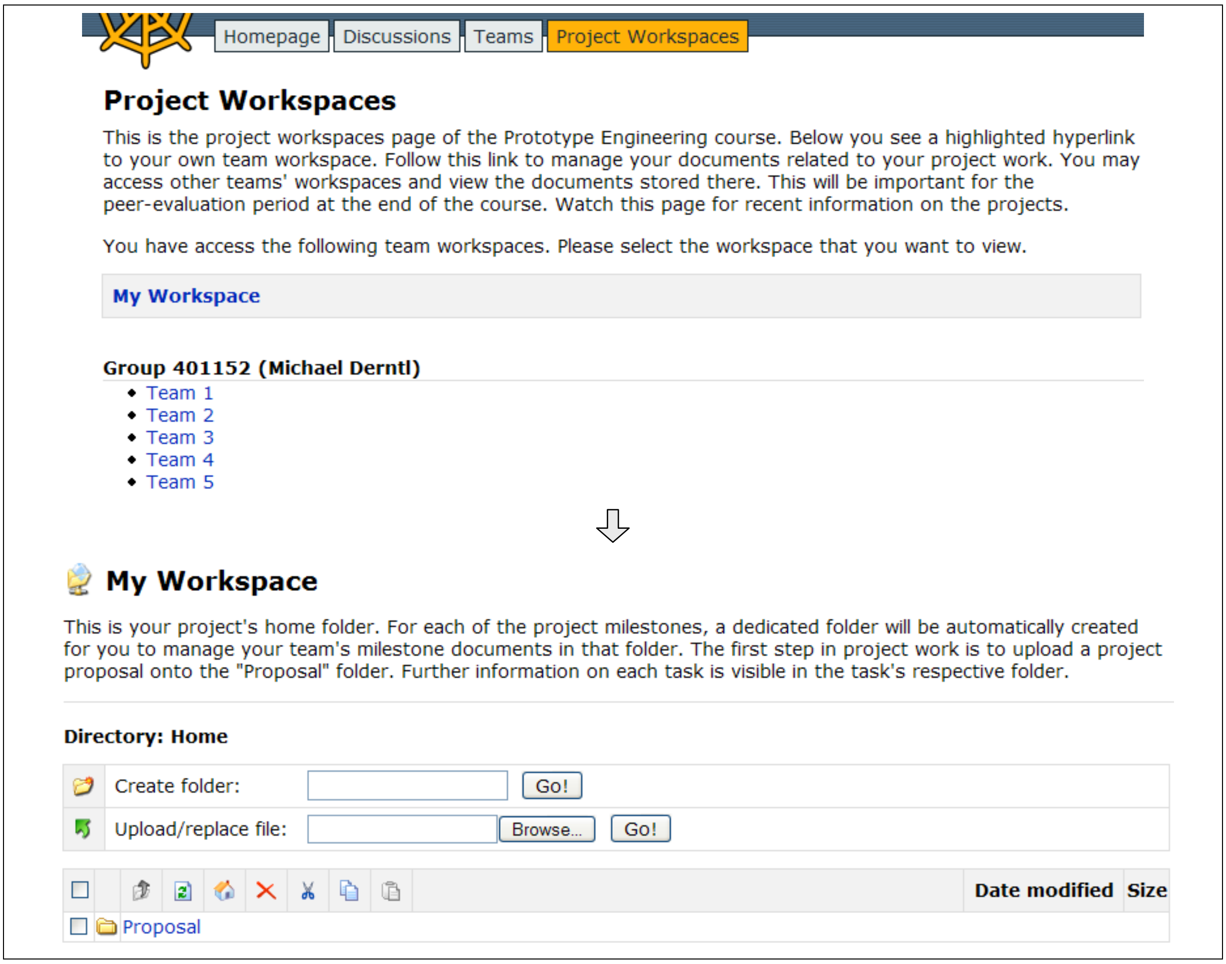


Figure 170: Team Workspaces participant view on the learning platform.

Diary

The course design includes an online DIARY thread that runs concurrently with the PROJECT-BASED LEARNING thread. Project participants are required to log their project-related activities in this Web-based diary. The administration wizard of the DIARY pattern is given in Figure 171, while the participant view, showing a sample diary including two entries, is given in Figure 172.

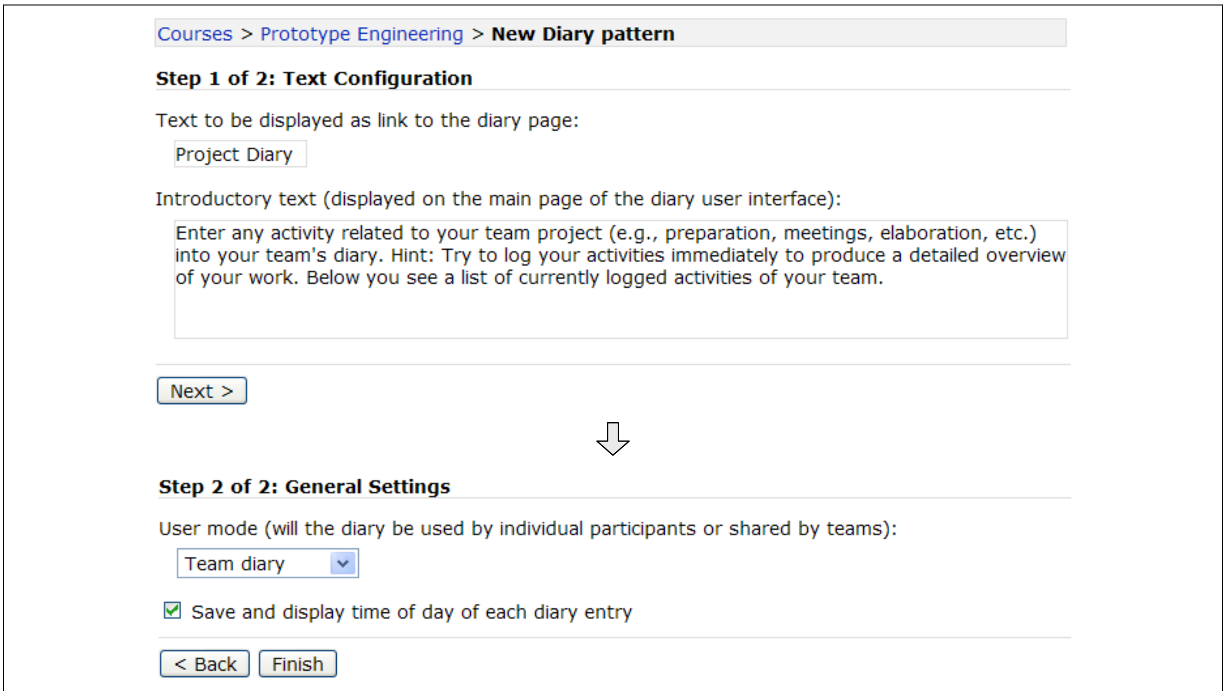


Figure 171: The DIARY administration wizard.

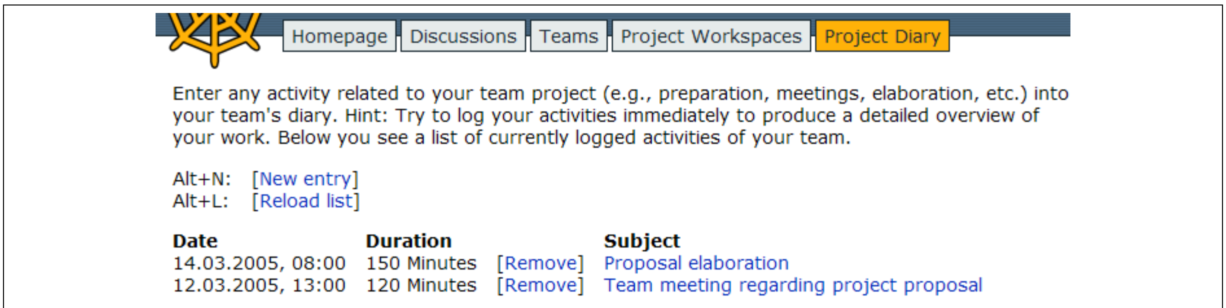


Figure 172: DIARY page of a team, including entries regarding the project proposal.

Project Milestone

As the project proceeds, the instructor initializes one PROJECT MILESTONE pattern for each of the milestones defined in the course design. Configuration of the milestone includes supplying a milestone name and relevant information and resources as text and/or links (Figure 173). After completing the one-step wizard for a milestone, the PatMan creates one dedicated folder for the milestone, including the information supplied by the instructor, in each team's workspace. The participant view of the project workspace of team #1 after the second milestone is given in Figure 174. The second part of the figure also shows the folder dedicated to the first milestone, including the milestone description and documents uploaded by that team.

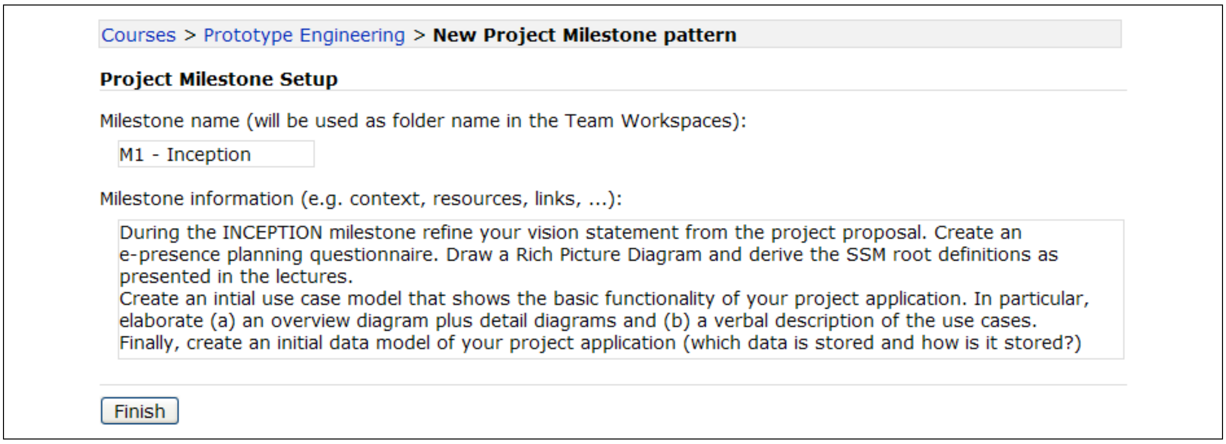


Figure 173: The PROJECT MILESTONE administration wizard.

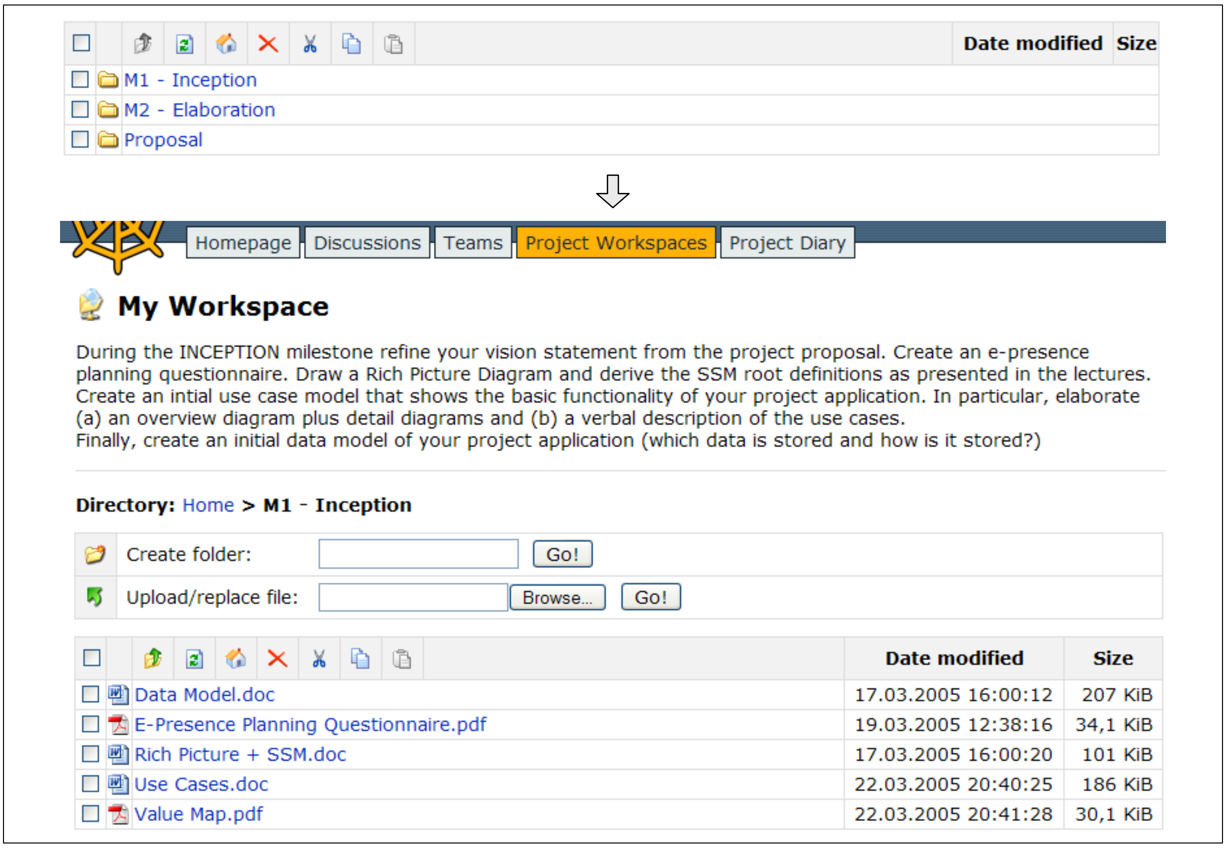


Figure 174: TEAM WORKSPACE of a project during the second PROJECT MILESTONE.

Lectures

In the meantime the lecture thread of the course proceeds with weekly lectures on relevant theories and techniques. The lecture slides and additional material are posted by the instructor on the homepage (in the “Resources” section). The screenshot in Figure 175 shows the homepage after the fifth lecture.

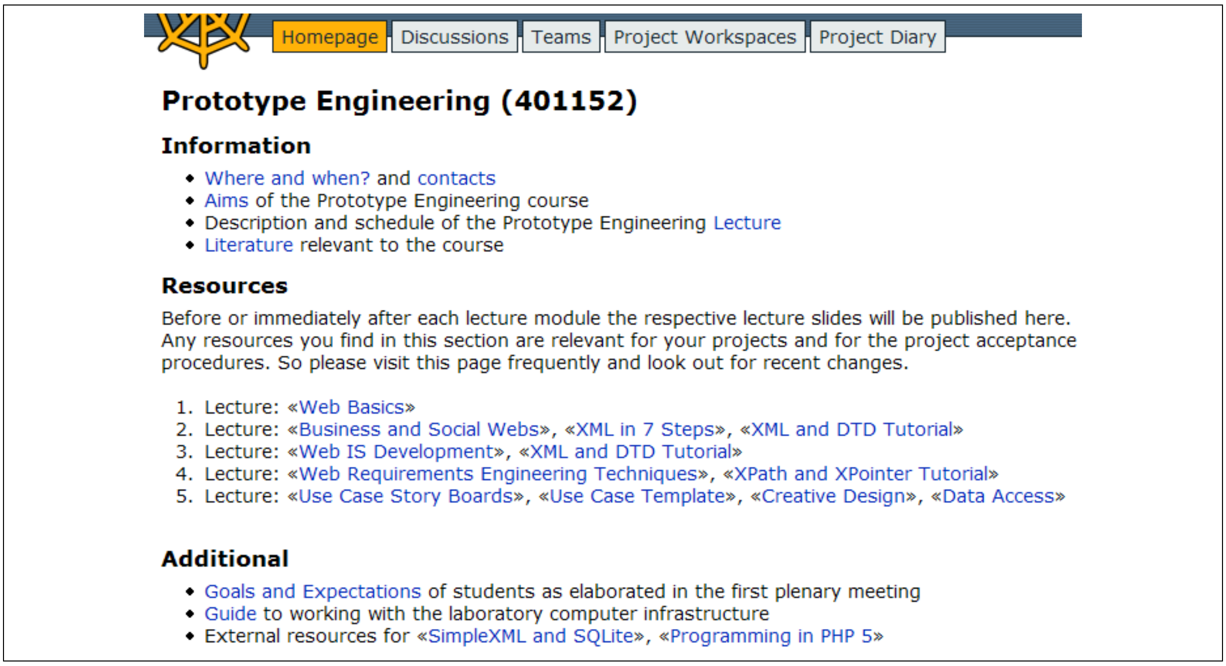


Figure 175: Prototype Engineering homepage after the fifth lecture.

Blended Assessment Phases

After the lecture and PBL threads are finished (about two weeks prior to the end of the course), the instructor starts the final course activities by instantiating the BLENDED EVALUATION and COLLECT FEEDBACK patterns:

Blended Evaluation

For evaluating and assessing the project achievements of the teams, a mix of SELF-EVALUATION, PEER-EVALUATION, and INSTRUCTOR-EVALUATION is used. The BLENDED EVALUATION wizard of the PatMan redirects the instructor to the administration wizards of these patterns. The wizard screenshot in Figure 176 shows the configuration of PEER-EVALUATION as an example. When finishing the wizard, the PatMan creates links to the PEER-EVALUATION and SELF-EVALUATION forms in the home folders in each team’s workspace. This way, participants can browse through the workspaces of their peers (or their own) and supply their evaluations directly where the underlying project documents are stored (Figure 177). The evaluation forms were designed using the PatMan’s *Web Form Manager*, which is implemented based on the Web template of the QUESTIONNAIRE form editor (see p. 233). The evaluation form simply consists of:

- A heading and an introductory text to be displayed on top of the form
- An open response block, where the evaluator can supply written comments on the current project

- A single-choice block, where the evaluator can assign 1 to 7 bonus points for the current project regarding three selected aspects.

The participant view of the Peer-Evaluation form is displayed in the bottom half of Figure 177. Note that after the evaluation period as configured in the wizard expires, the link to the evaluation form is replaced by a link to the evaluation results, such that each team can view the evaluations of their projects submitted by their peers.

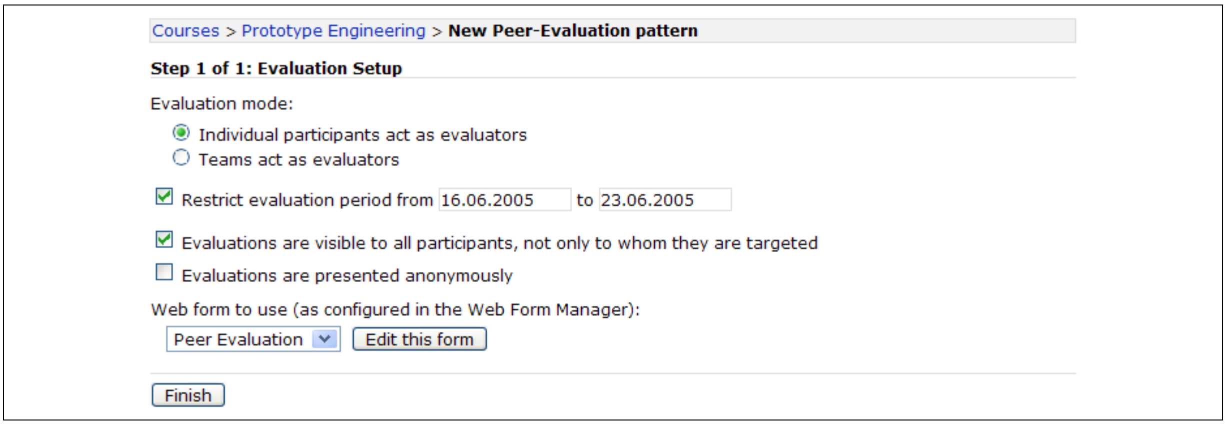
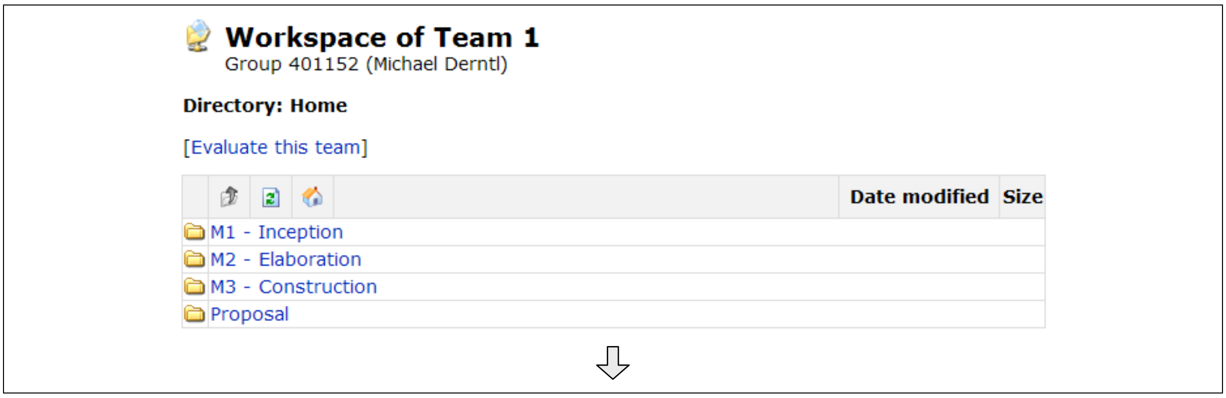
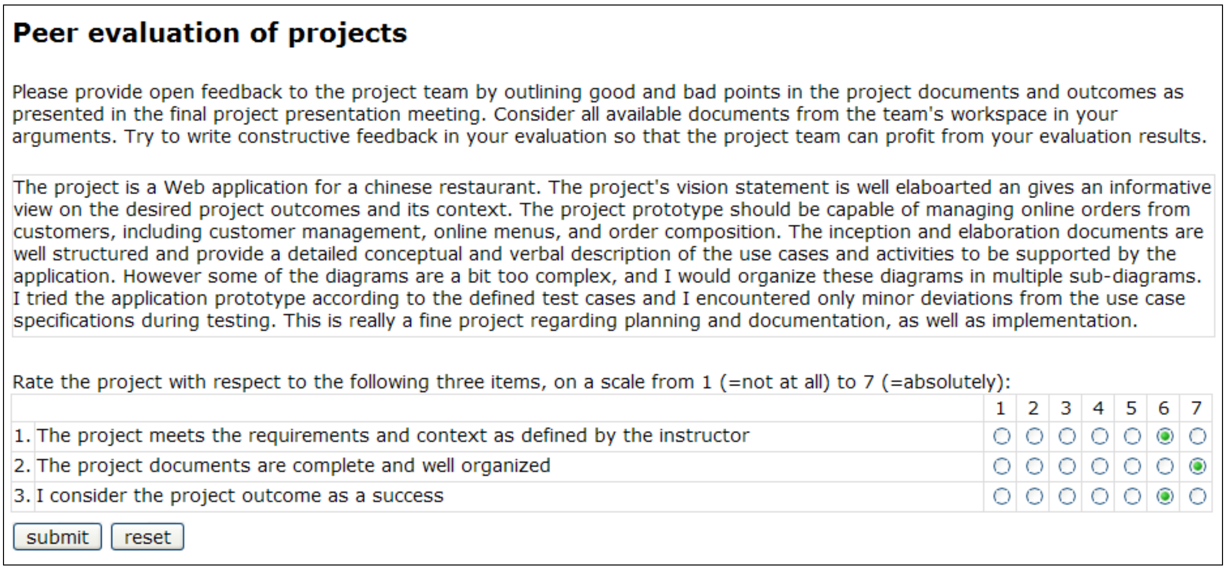


Figure 176: The Peer-Evaluation administration wizard.
Note that the administration wizards for Self-Evaluation and Instructor-Evaluation are exactly the same, but the instructor has to supply different values in the configuration form.





Peer evaluation of projects

Please provide open feedback to the project team by outlining good and bad points in the project documents and outcomes as presented in the final project presentation meeting. Consider all available documents from the team's workspace in your arguments. Try to write constructive feedback in your evaluation so that the project team can profit from your evaluation results.

The project is a Web application for a chinese restaurant. The project's vision statement is well elaboarted an gives an informative view on the desired project outcomes and its context. The project prototype should be capable of managing online orders from customers, including customer management, online menus, and order composition. The inception and elaboration documents are well structured and provide a detailed conceptual and verbal description of the use cases and activities to be supported by the application. However some of the diagrams are a bit too complex, and I would organize these diagrams in multiple sub-diagrams. I tried the application prototype according to the defined test cases and I encountered only minor deviations from the use case specifications during testing. This is really a fine project regarding planning and documentation, as well as implementation.

Rate the project with respect to the following three items, on a scale from 1 (=not at all) to 7 (=absolutely):

	1	2	3	4	5	6	7
1. The project meets the requirements and context as defined by the instructor							
2. The project documents are complete and well organized							
3. I consider the project outcome as a success							

submit reset

Figure 177: Team workspace including a link to PEER-EVALUATION, and the peer-evaluation submission form.

Note that in this case a member of team 2 browses the documents in the workspaces of team 1 and subsequently evaluates the project by clicking the “Evaluate this team” link in the workspace home folder of team 1.

Collect Feedback

As the final step in the Prototype Engineering course, the instructor collects feedback in the form of QUESTIONNAIRES and REACTION SHEETS as specified in the course design:

Reaction Sheets

For collecting reaction sheets, a separate page on the LP is created. This is done using the PatMan’s REACTION SHEETS wizard (Figure 178). The submission form just includes information on what kind of reaction is solicited and includes just one large text box where each participant can supply written, unstructured feedback to the instructor (Figure 179).

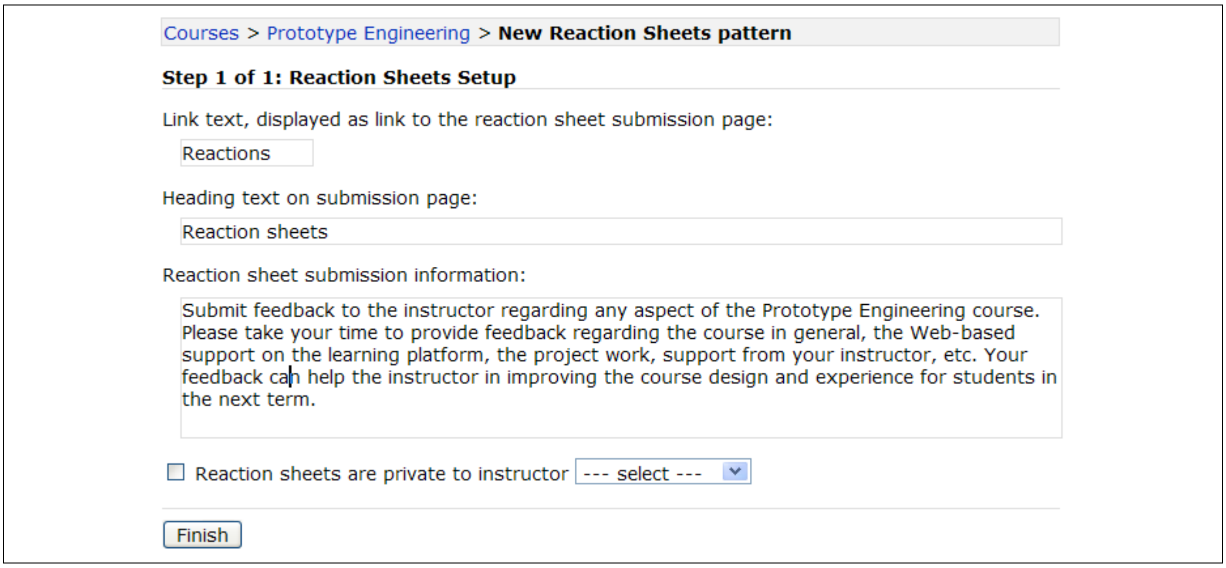


Figure 178: The REACTION SHEETS administration wizard.

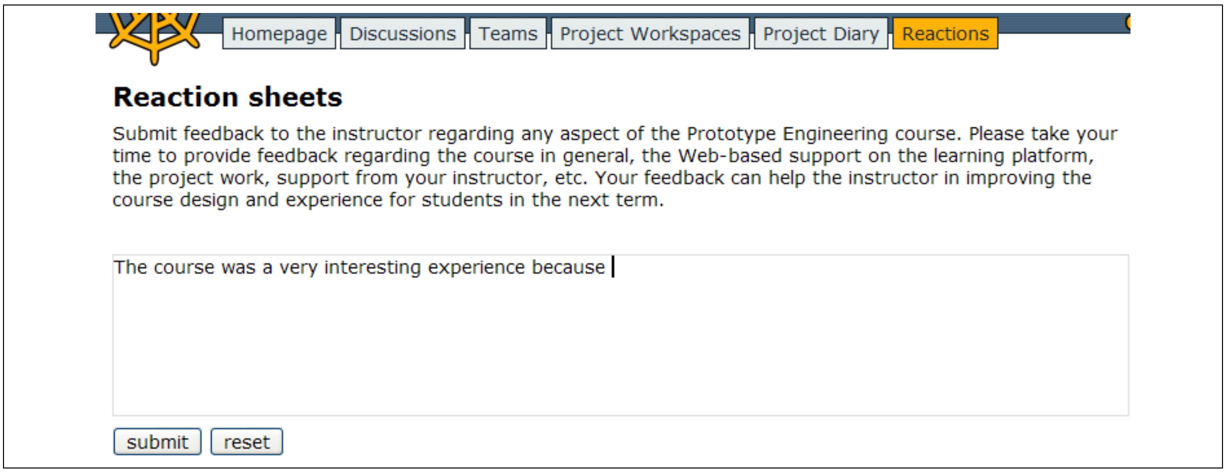


Figure 179: REACTION SHEETS submission form.

Questionnaire

Finally, we collect structured feedback in the form of a QUESTIONNAIRE. For demonstration purposes, the questionnaire form is kept very simple, including only two single-choice choice tables, which comprise items on *general aspects* and *learning aspects* taken from the *Evaluation* section of the COURSE pattern (see p. 260 and p. 265, respectively.)

The administration wizard of the QUESTIONNAIRE pattern is shown in Figure 180, while the questionnaire form on as it appears on the LP is given in Figure 181.

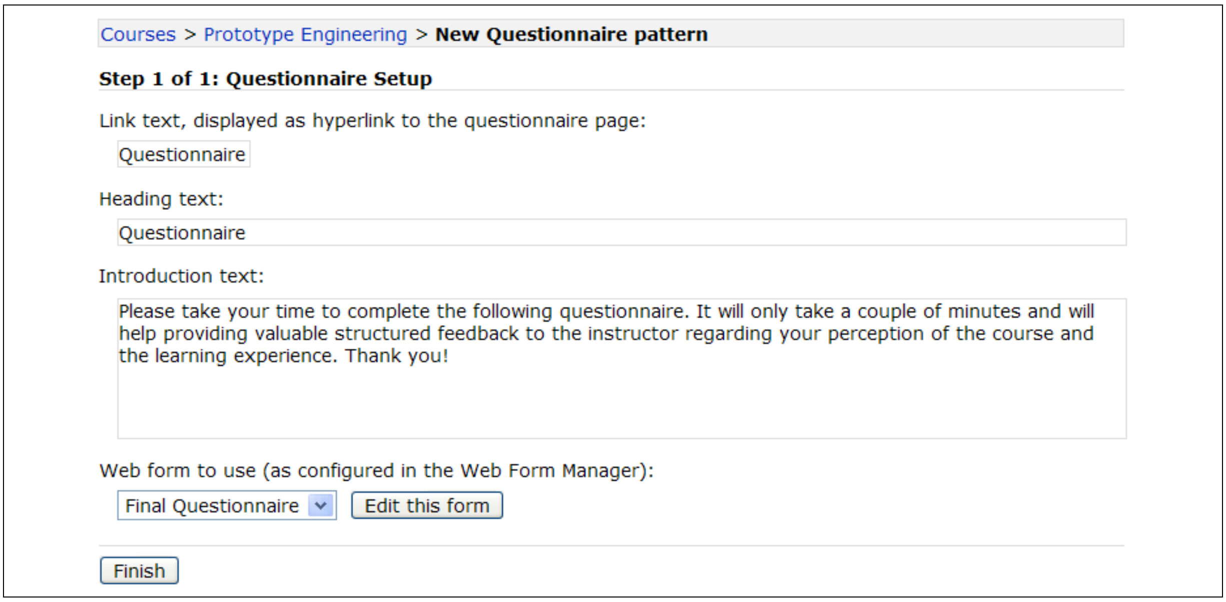


Figure 180: The QUESTIONNAIRE administration wizard.

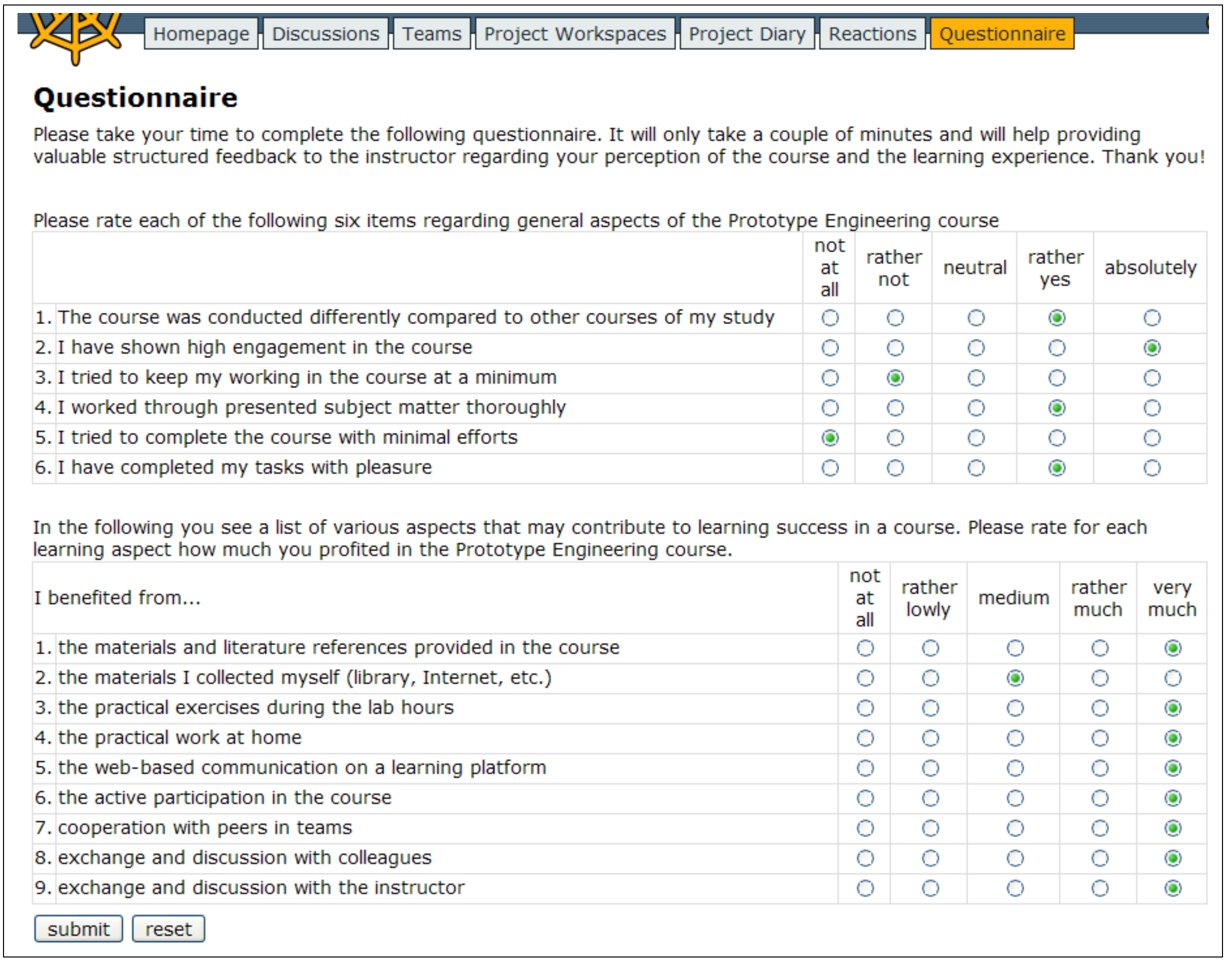
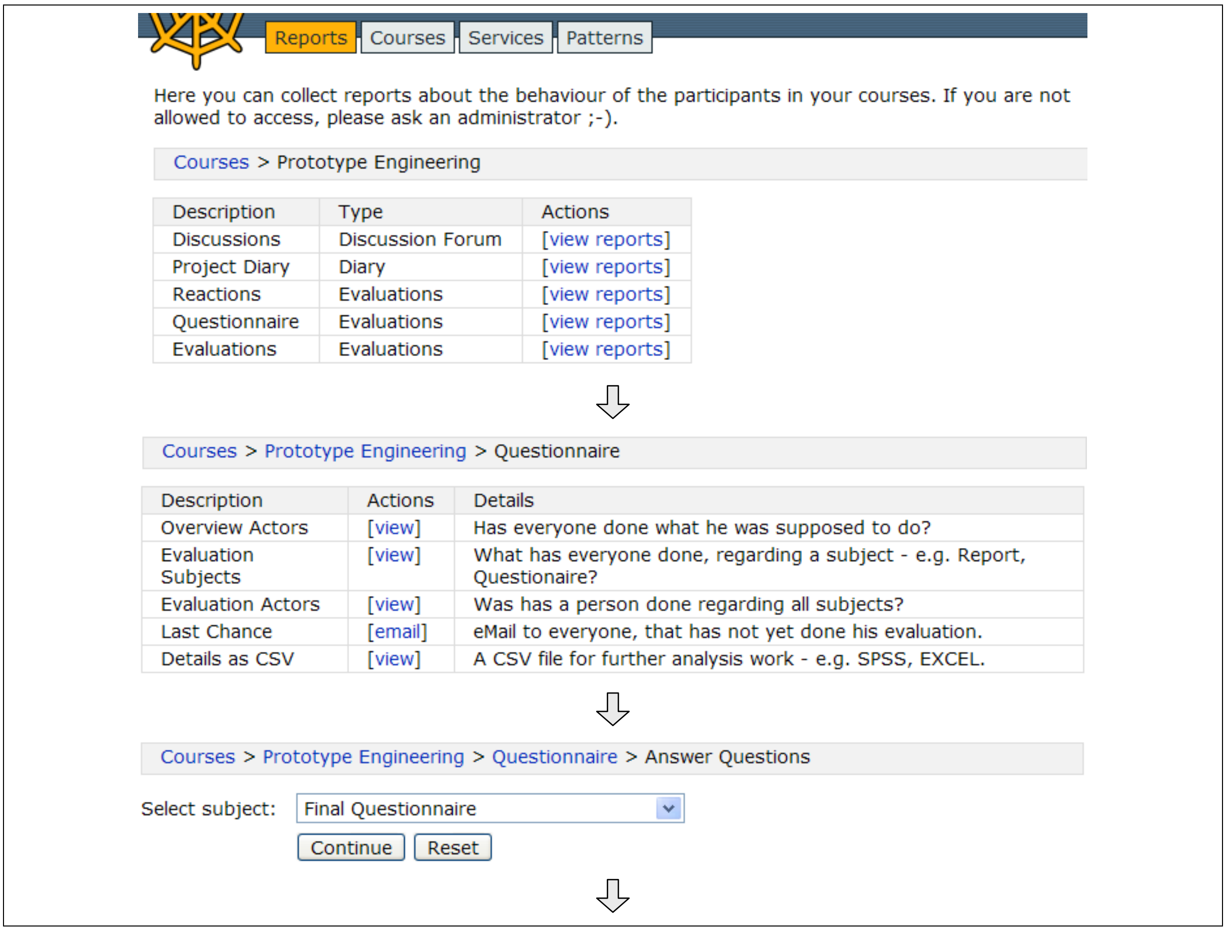


Figure 181: QUESTIONNAIRE submission form.

Reports

For the QUESTIONNAIRE pattern we also take a look at the report view of the patterns. The CEWebS Report Manager is capable of producing reports for any pattern, which offers reports for the course. Figure 182 shows a sample sequence of steps in report generation for the QUESTIONNAIRE instance in Prototype Engineering. The instructor first selects the Web service for which a report should be generated (“Questionnaire”), then selects the type of report (“Details as CSV”), and finally selects the subject or form as configured by the Web Form Manager (“Final Questionnaire”). The browser then automatically downloads the CSV file containing the detailed questionnaire results.

Similar steps of report generation are available in the Prototype Engineering course for TEAM BUILDING, all EVALUATIONS, REACTION SHEETS, and for the project DIARIES.



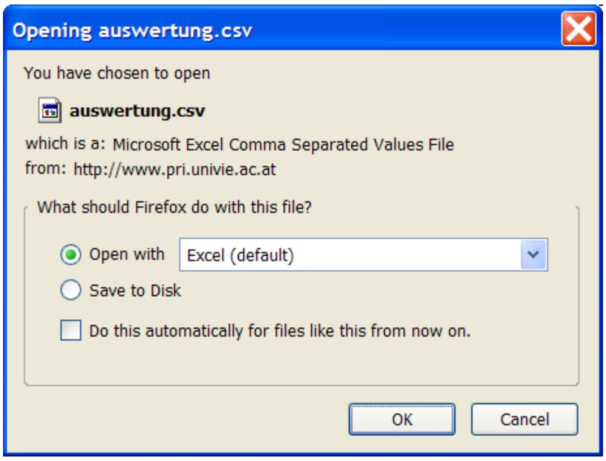


Figure 182: Report generation example for the Prototype Engineering QUESTIONNAIRE.

6.3 Case Study Discussion

The case study presented an application of the PCeL patterns and the BLESS model spanning all BLESS layers, from learning theory via course sequences via PCeL patterns via Web templates to the learning platform. Each step in this complex transition is supported by a piece of research or by a tool presented in this thesis. Starting from a verbal course description, the Prototype Engineering course represents a completely pattern-based course design and application case. The wizards implemented in the Pattern Manager are based on the Web templates of the patterns as presented in the PCeL pattern repository Chapter. The instructor or administrator is guided via a wizard-like series of Web pages that are designed to be simple, intuitive, and focused on the problem/process at hand.

To cope with the complexity inherent in many learning processes, the separation of different views (participant, administration, and report view) on the processes enables simple usage, instantiation, and tracking of employed scenarios, respectively. From a conceptual point of view, the pattern-based models and diagrams provide a clear overview as well as detailed activity flows (learnflows) of the whole course design.

However, while the Pattern Manager prototype and the pattern-based course design focus on the technological and conceptual points of view, we acknowledge that personal skills are equally needed in accordance with the pedagogy of PCeL to create a constructive and motivating learning environment. Put another way, "*in light of the fact that patterns as skeletons still need interpersonal skills as muscles to allow for real movement in the educational scene, appropriate staff-development strategies [are] vitally important as a key factor for cultivating the skills and attitudes required to use patterns in a way to promote [...] learning.*" [580]

[580] Derntl and Motschnig-Pitrik (2005, p. 129)

7 Conclusion

7.1 Summary

This thesis presented a novel approach of capturing, employing and researching Person-Centered e-Learning (PCeL) practices in higher education. In the first Chapter the motivation was presented based on:

- The fact that designing and conducting PCeL courses from scratch is more demanding with respect to time and effort than conventional teaching;
- The hypothesis stating that employing Person-Centered principles in teaching and learning is considered to add value to learning;
- The assumption that a conceptual toolkit and a repository of PCeL practices would greatly reduce the effort required by instructors and learning designers for constructing, conducting, and improving their courses, as well as for documenting and sharing their experiences.

In the second Chapter, an overview of the theories and research, on which the PCeL pattern approach is based, was presented. The approach is fundamentally based on the integration of three cornerstones, which are:

- Traditional and current learning theories and methods that are based primarily on a constructivist point of view, providing the theoretical basis of Person-Centered e-Learning, which essentially determines the educational value system underlying the PCeL pattern repository.
- The "classical" pattern approach as a means of capturing and uniformly documenting solutions to common design problems, which provides the historical and philosophical foundation for PCeL patterns.
- Conceptual modeling, acting as the main vehicle or tool of conveying the essence of PCeL designs by modeling their learnflows and structures using an extended subset of the Unified Modeling Language.

Basic principles and most relevant aspects of each of these cornerstones were elaborated and presented in that Chapter to provide the reader with necessary background information.

In the third Chapter, the pattern approach to PCeL was presented as the main contribution of this work. The pattern description and modeling method was explained in detail by presenting the PCeL pattern description template and the modeling approach used to specify the patterns' activity flows and structural relationships. The approach was embedded into a novel conceptual framework guiding the integrated application of learning theories and learning technologies in blended learning environments, i.e., the Blended Learning Systems Structure (BLESS) model. PCeL patterns and their modeling method substantially support the intermediate layers of the BLESS model, ranging from

- visual modeling of course sequences and teaching activities,
- expressing them in terms of PCeL patterns, and
- providing a means of supporting the patterns on learning platforms by specifying Web templates as visual specifications of Web support of PCeL practices underlying the patterns.

In the fourth Chapter, related approaches were presented that touch various aspects of the PCeL pattern approach. It was argued that none of these approaches is capable of supporting the most essential levels of blended learning design and practice. Either these approaches are targeted at the whole e-learning domain or they focus on very specific aspects only (e.g., e-content, technological aspects, pedagogical aspects). The real strength of the PCeL pattern approach was shown to lie in the dedicated focus on one consistent theoretically and practically founded pedagogical baseline, the utilization of object-oriented, conceptual modeling techniques for both structural and dynamical views on learning processes, and an underlying research and practice framework based on the BLESS model and participatory Action Research.

In the fifth Chapter, the pattern repository in its current version was presented, consisting of seven pattern packages and about 50 patterns at different levels of detail and abstraction. While the repository does comprise a substantial amount of patterns, it does not claim to cover all aspects and practices of PCeL, but the most basic and important ones along with required administrative processes.

To show the applicability and usefulness of the BLESS framework and the PCeL pattern approach, the sixth Chapter presented a case study of designing a course based on the research and results presented in this thesis. A prototypical Pattern Manager application was developed and plugged into the open-source CEWebS learning platform. The implementation of the prototype was completely based on the Web templates and sequences specified in the pattern repository. The case study showed that the PCeL pattern approach, in combination with appropriate tools, is capable of providing substantial support in all phases of course design and execution, making it a highly structured and almost "easy" effort to design a PCeL course from a basic, verbal course concept. Nevertheless, it was acknowledged that patterns in isolation do not suffice. They just provide a toolbox or skeleton that has to be complemented with interpersonal skills of educators to facilitate students' learning and to further innovative, person-centered learning scenarios.

7.2 Outlook

Even though the current state of research presented in this work can provide substantial support in designing, understanding, and researching PCeL scenarios, there are some aspects that have yet to be considered. The approach requires additional work on dissemination and adaptation of concepts. It is not clear why and when others tend to adopt pattern languages, and what factors further the adoption of patterns outside their original context. Clearly, the patterns were mined and specified by rather technically oriented computer science people, while potential users, e.g. in social science contexts, could be averse to some of the rather formal features of the approach such as object-orientation, conceptual modeling, using a formal modeling language that is rooted in computer science, etc. This is why each pattern comes with a detailed description in natural language and it is emphasized that, once adopted and installed, patterns can be applied without modeling expertise.

One of the central tasks in the near future will be the dissemination of the patterns and their usage by educators with different personal and professional background, and in different subject contexts. Initial cooperation with other faculties and work groups at the University of Vienna and with the Masaryk University in Brno has started in this respect.

The PCeL pattern approach and the pattern repository are currently employed in these local projects:

- *Technology-Enhanced Learning*: An internal R&D project funded by the University of Vienna, which aims to (1) revise the current version of the PCeL pattern repository and to apply it in practice in different institutions inside the University of Vienna to bring it closer to potential practitioners, and (2) extending the current set of patterns covered by Web service modules and to integrate them into the WebCT learning platform, which is the one centrally supported platform at the University of Vienna.
- *Knowledge Experts*[581]: Funded by the European Social Fund[582] (ESF). The project aims to develop two blended learning curricula for knowledge experts and e-tutors. The repository of blended learning courses previously and currently conducted by the consortium members was modeled using the PCeL pattern modeling approach, which in this case acts as a tool for communication and knowledge capturing/exchange.

Other research efforts, where the PCeL pattern approach plays a role, include:

- *Visual instructional design languages (VIDL)*: Researching the potential impact of VIDLs on e-learning design, and eliciting the underlying critical success factors, is one of the currently addressed issues of further interest. Thereby, three available VIDLs are compared, as a first step, in modeling different courses with different intents/settings. This is currently in progress at the Research Lab for Educational Technologies in cooperation with Luca Botturi (University of Lugano, Switzerland), Hannes Lischka and Kathrin Figl (University of Vienna).

581 http://leonardo.pri.univie.ac.at/communities/kex

582 http://europa.eu.int/comm/employment_social/esf2000/index-en.htm

- *Context modeling*: The PCeL modeling approach was recently adopted for modeling pervasive learning scenarios[583], which required the inclusion of learner context information in the activity models. This was achieved by proposing three simple extensions to the standard PCeL scenario modeling approach, and by explicitly including relevant context objects in the activity models.
- *Upcoming projects*: The extension of the current pattern repository and the adaptation of the modeling approach to specific needs and contexts is part of a number of project proposals that are currently in the review process.

The main future research thread in the PCeL approach will be the capturing, modeling, and implementation of further patterns from different contexts. Generally it can be stated that there is an internationally, fast growing interest in visual modeling and patterns of learning scenarios as more and more research papers and projects emerge in this area. This growing interest, in my opinion, underlines the proposition that blended learning and e-learning design, research, and practice, are too complex to be mastered in their entirety without appropriate tools and guidance. The bottom line is that we can use models and patterns as mediators between learning theory, practice, and learning technology as one powerful tool for this purpose.

583 Derntl and Hummel (2005)

8 References

ADL. (2003a). *About SCORM*. Retrieved Feb 1, 2004, from http://www.adlnet.org/index.cfm?fuseaction=AboutSCORM

ADL. (2003b). *ADL Overview*. Retrieved Feb 1, 2004, from http://www.adlnet.org/index.cfm?fuseaction=abtadl

ADL. (2004). *SCORM 2004 Overview*. Retrieved Feb 1, 2004, from http://www.adlnet.org/screens/shares/dsp_displayfile.cfm?fileid=992

AICC. (2001a). *AICC FAQ*. Retrieved Feb 1, 2004, from http://www.aicc.org/pages/aicc_faq.htm

AICC. (2001b). *Types of AICC Publications*. Retrieved Feb 1, 2004, from http://www.aicc.org/pages/aicc3.htm#PUB1

Akahori, K., & Kim, S. M. (2003). *Peer Evaluation Using the Web and Comparison of Meta-cognition between Experts and Novices.* Proceedings of ED-MEDIA 2003: World Conference on Educational Multimedia, Hypermedia & Telecommunications, June 23-28, 2003, Honolulu, Hawaii, pp. 1484-1487.

Alexander, C. (1979). *The Timeless Way of Building*. New York: Oxford University Press.

Alexander, C. (1981). *The Linz Café - Das Linz Café*. New York: Oxford University Press.

Alexander, C. (1983). *The Oregon Experiment*. New York: Oxford University Press.

Alexander, C., Ishikawa , S., Silverstein, M., Jacobson, M., Fiksdahl-King, I., & Angel, S. (1977). *A Pattern Language - Towns, Buildings, Construction*. New York: Oxford University Press.

Ambler, S. (1998). *An Introduction to Process Patterns* (White paper): Ambysoft, Inc.

Anderson, B. (1993). Workshop Report - Towards an Architecture Handbook. *OOPSLA Messenger, 4* (2), 109-114.

Anderson, J. R. (1991). *Cognitive Psychology and its Applications* (3rd ed.). New York: Freeman.

Appleton, B. (2000). *Patterns and Software: Essential Concepts and Terminology*. Retrieved Sept 2, 2002, from http://www.enteract.com/~bradapp/docs/patterns-intro.html

ARIADNE. (2002). *Presentation of the ARIADNE Foundation*. Retrieved Feb 1, 2004, from http://www.ariadne-eu.org/en/about/general/benefits/index.html

Arnold, F., & Podehl, G. (1999). *Features and Design Patterns - A Comparison.* Proceedings of Swiss Conference on CAD/CAM '99, Neuchâtel, Switzerland, pp. 140-148.

Aspy, D. N. (1972). *Toward a Technology for Humanizing Education*. Champaign, Illinois: Research Press Company.

Atherton, J. A. (2003). *Learning and Teaching: Learning Contracts* [Online]. Retrieved Mar 5, 2004, from http://www.dmu.ac.uk/~jamesa/teaching/learning_contracts.htm

Avgeriou, P., Papasalouros, A., & Retalis, S. (2003a). *Patterns For Designing Learning Management Systems.* Proceedings of European Pattern Languages of Programming (EuroPLOP), Irsee, Germany.

Avgeriou, P., Papasalouros, A., Retalis, S., & Skordalakis, M. (2003b). Towards a Pattern Language for Learning Management Systems. *Journal of Educational Technology & Society, 6* (2), 11-24.

Baggetun, R., Rusman, E., & Poggi, C. (2004). *Design Patterns For Collaborative Learning: From Practice To Theory And Back.* Proceedings of ED-MEDIA 2004: World Conference on Educational Multimedia, Hypermedia & Telecommunications, June 21-26, 2004, Lugano, Switzerland.

Bangert, A. W. (2004). The Seven Principles of Good Practice: A framework for evaluating on-line teaching. *The Internet and Higher Education, 7* (3), 217-232.

Barrett-Lennard, G. T. (1998). *Carl Rogers' Helping System - Journey and Substance*. London: Sage Publications.

Baskerville, R. L. (1999). Investigating Information Systems with Action Research. *Communications of the Association for Information Systems, 2*, http://cais.isworld.org/articles/2-19/.

Baskerville, R. L., & Wood-Harper, A. T. (1996). A Critical Perspective on Action Research as a Method for Information System Research. *Journal of Information Technology, 11* (235-246).

Bauer, C. (2003). *Go blended! Enhancing learning by using technology in cooperative person centred learning* (Exposé for Dissertation): University of Vienna, Department of Computer Science and Business Informatics.

Baumgartner, P. (2003). *Ontological stratification of virtual learning activities - Developing a new categorization scheme.* Proceedings of International Workshop on Interactive Computer-Aided Learning (ICL'03), Villach, Austria.

Baumgartner, P., Häfele, H., & Maier-Häfele, K. (2003). *E-Learning Praxishandbuch - Auswahl von Lernplattformen.* Innsbruck, Austria: StudienVerlag.

Baumgartner, P., Häfele, K., & Häfele, H. (2002). *CD Austria: e-Learning*. Retrieved Sep 29, 2003, from http://www.bildung.at/statisch/bmbwk/e-learning.pdf

Baxter, S., & Gray, C. (2001). The application of student-centred learning approaches to clinical education. *International Journal of Language and Communication Disorders, 36* (Supplement), 396-400.

Baylor, A. L., & Ritchie, D. (2002). What factors facilitate teacher skill, teacher morale, and perceived student learning in technology-using classrooms. *Computers & Education, 39* (4), 395-414.

Beck, K., & Johnson, R. E. (1994). *Patterns Generate Architectures.* Proceedings of European Conference on Object-Oriented Programming (ECOOP) '94, Bologna, Italy, pp. 139-149.

Bergin, J. (2002). *Fourteen Pedagogical Patterns* [Online]. Retrieved Sept 27, 2002, from http://csis.pace.edu/~bergin/PedPat1.3.html

Bergin, J., Eckstein, J., Manns, M. L., & Wallingford, E. (2001). *Patterns for Gaining Different Perspectives* [Online]. Retrieved Jan 16, 2003, from www.pedagogicalpatterns.org/current/gaindiffperspective.pdf

Boehm, B. W. (1988). A Spiral Model of Software Development and Enhancement. *IEEE Computer, 21* (5), 61-72.

Boeree, G. C. (1998). *Personality Theories -- Carl Rogers* [Online]. Retrieved June 18, 2002, from http://www.ship.edu/~cgboeree/rogers.html

Booch, G. (1994). *Object-Oriented Analysis and Design with Applications* (2nd ed.). Redwood City, CA: Benjamin/Cummings.

Booch, G., Rumbaugh, J., & Jacobson, I. (1999). *The Unified Modeling Language User Guide*. Reading, MA: Addison-Wesley.

Borchers, J. (2001). A Pattern Approach to Interaction Design. *AI & Society Journal of Human-Centred Systems and Machine Intelligence, 15* (4), 359-376.

Botturi, L. (2003). *E2ML - Educational Environment Modeling Language*. Proceedings of ED-MEDIA 2003: World Conference on Educational Multimedia, Hypermedia & Telecommunications, June 23-28, 2003, Honolulu, Hawaii, USA, pp. 304-311.

Brown, J. S., Collins, A., & Duguid, S. (1989). Situated cognition and the culture of learning. *Educational Researcher, 18* (1), 32-42.

Brown, W. J., McCormick III, H. W., & Thomas, S. W. (1999). *AntiPatterns and Patterns in Software Configuration Management*. New York: John Wiley & Sons.

Brunner, J., & Polin, L. (1999). *Definition of Project-Based Learning*. Retrieved Oct 25, 2003, from http://hale.pepperdine.edu/~jabrunne/wpgrid.html

Buck Institute for Education. (2002). *Project Based Learning Handbook: Introduction to Project Based Learning*. Retrieved Jan 9, 2004, from http://www.bie.org/pbl/pblhandbook/intro.php

Budinsky, F., Finnie, M., Yu, P., & Vlissides, J. (1996). Automatic Code Generation from Design Patterns. *IBM Systems Journal, 35* (2), 151-171.

Bühler, K. (1907). Tatsachen und Probleme zu einer Psychologie der Denkvorgänge: I. Über Gedanken. *Archiv für die gesamte Psychologie, 9*, 279-305.

Buschmann, F. (1993). Rational Architectures for Object-Oriented Software Systems. *Journal of Object-Oriented Programming, 6* (5), 30-41.

Buschmann, F., Meunier, R., Rohnert, H., Sommerlad, P., & Stal, M. (1996). *Pattern-Oriented Software Architecture: A System of Patterns*. Chichester, U.K.: John Wiley & Sons.

Caeiro, M., Llamas-Nistal, M., & Anido, L. (2004). *E-learning Patterns: an Approach to Facilitate the Design of E-learning Materials.* Proceedings of VII Congresso Iberoamericano de Informatica Educativa, Oct 13-15, Monterrey, Mexico.

Chase, C. C., & Geldenhuys, K. M. (2001). Student-centred teaching in a large heterogeneous class. *Medical Education, 35* (11), 1071.

Chen, P. (1976). The Entity-Relationship Model: Toward a Unified View of Data. *ACM Transactions on Database Systems, 1* (1), 1-36.

Chidamber, S. R., & Kemerer, C. F. (1994). A Metrics Suite for Object Oriented Design. *IEEE Transactions on Software Engineering, 14* (9), 476-493.

Cho, K., & Schunn, C. D. (2003). *Validity and Reliability of Peer Assessments with a Missing Data Estimation Technique.* Proceedings of ED-MEDIA 2003: World Conference on Educational Multimedia, Hypermedia & Telecommunications, June 23-28, 2003, Honolulu, HI, USA, pp. 1511-1514.

Chou, C. C. (2004). A Model of Learner-Centered Computer-Mediated Interaction for Collaborative Distance Learning. *International Journal on E-Learning, 3* (1), 11-18.

Clark, D. (2000). *Brainstorming*. Retrieved Dec 14, 2003, from http://www.nwlink.com/~donclark/perform/brainstorm.html

Coad, P. (1992). Object-oriented patterns. *Communications of the ACM, 35* (9), 152-159.

Cognition and Technology Group at Vanderbuilt. (1993). Anchored instruction and situated cognition revisited. *Educational Technology, 33* (3), 52-70.

Coplien, J. O. (1992). *Advanced C++ Programming Styles and Idioms*. Reading, MA: Addison Wesley.

Coplien, J. O. (1994). Generative pattern languages: An emerging direction of software design. *C++ Report, 6*.

Coplien, J. O. (1998). Software Design Patterns: Common Questions and Answers. In L. Rising (Ed.), *The Patterns Handbook: Techniques, Strategies, and Applications* (pp. 311-320). New York: Cambridge University Press.

Coplien, J. O., & Schmidt, D. (Eds.). (1995). *Pattern Languages of Program Design*. Reading, MA: Addison Wesley.

Cornelius-White, J. (2003). *Meta-Analysis of Student-Centered Education* (Research Proposal): University of Texas, Department of Psychology.

Cornelius-White, J. H. D., Hoey, A., Cornelius-White, C., Motschnig-Pitrik, R., & Figl, K. (2004). Person-Centered Education: A Meta-Analysis of Care in Progress. *Journal of Border Educational Research, 3* (1), 81-87.

Cotton, J. (1995). *The Theory of Learning*. London: Kogan Page.

Cristea, A., & Garzotto, F. (2004). *Designing Patterns for Adaptive or Adaptable Educational Hypermedia: a Taxonomy*. Proceedings of ED-MEDIA 2004: World Conference on Educational Multimedia, Hypermedia & Telecommunications, June 21-26, 2004, Lugano, Switzerland.

Cronje, J. C. (2001). Metaphors and models in Internet-based learning. *Computers & Education, 37*, 241-256.

Curbera, F., Nagy, W. A., & Weerawarana, S. (2001). *Web Services: Why and How?* Proceedings of OOPSLA'2001 (Workshop on Object-Oriented Web Services), Tampa, Florida. Available online at http://www.research.ibm.com/people/b/bth/OOWS2001/nagy.pdf.

Damasio, A. (1999). *The Feeling of What Happens: Body and Emotion in the Making of Consciousness*. New York, NY: Harcourt Brace & Company.

Derntl, M. (2004). *The Person-Centered e-Learning Pattern Repository: Design for Reuse and Extensibility*. Proceedings of ED-MEDIA 2004: World Conference on Educational Multimedia, Hypermedia & Telecommunications, June 21-26, 2004, Lugano, Switzerland, pp. 3856-3861.

Derntl, M., & Hummel, K. A. (2005). *Modeling Context-Aware e-Learning Scenarios*. Proceedings of Third IEEE Conference on Pervasive Computing and Communication Workshops, Mar 8-12, 2005, Kauai Island, Hawaii, USA, pp. 337-342.

Derntl, M., & Mangler, J. (2004). *Web Services for Blended Learning Patterns.* Proceedings of 4th IEEE International Conference on Advanced Learning Technologies, Aug 30-Sept 1, Joensuu, Finland, pp. 614-618.

Derntl, M., & Motschnig-Pitrik, R. (2003a). *Employing Patterns for Web-Based, Person-Centered Learning: Concept and First Experiences.* Proceedings of ED-MEDIA 2003: World Conference on Educational Multimedia, Hypermedia & Telecommunications, June 23-28, 2003, Honolulu, Hawaii, pp. 1690-1697. Available online at http://www.pri.univie.ac.at/~derntl/papers/EDMEDIA03Patterns.pdf.

Derntl, M., & Motschnig-Pitrik, R. (2003b). *Towards a Pattern Language for Person-Centered e-Learning.* Proceedings of Society for Information Technology & Teacher Education International Conference (SITE 2003), Albuquerque, New Mexico, USA. Available online at http://www.pri.univie.ac.at/~derntl/papers/SITE03Patterns.pdf.

Derntl, M., & Motschnig-Pitrik, R. (2004a). *BLESS - A Layered Blended Learning Systems Structure.* Proceedings of 4th International Conference on Knowledge Management (I-KNOW'04), June 30 - July 2, Graz, Austria, pp. 592-599.

Derntl, M., & Motschnig-Pitrik, R. (2004b). *A Pattern Approach to Person-Centered e-Learning based on Theory-Guided Action Research.* Proceedings of 4th International Conference on Networked Learning (NLC), Lancaster, UK.

Derntl, M., & Motschnig-Pitrik, R. (2004c). *Patterns for Blended, Person-Centered Learning: Strategy, Concepts, Experiences, and Evaluation.* Proceedings of 2004 ACM Symposium on Applied Computing (SAC), Nicosia, Cyprus, pp. 916-923.

Derntl, M., & Motschnig-Pitrik, R. (2005). The Role of Structure, Patterns, and People in Blended Learning. *The Internet and Higher Education, 8* (2), 111-130.

Dietz-Uhler, B., & Bishop-Clark, C. (2001). The use of computer-mediated communication to enhance subsequent face-to-face discussions. *Computers in Human Behavior, 17* (2001), 269-283.

Dillenbourg, P. (2002). Over-Scripting CSCL: The risks of blending collaborative learning with instructional design. In P. A. Kirschner (Ed.), *Three worlds of CSCL. Can we support CSCL* (pp. 61-91). Heerlen: Open Universiteit Nederland.

Dix, A., Finlay, J., Abowd, G., & Beale, R. (1998). *Human Computer Interaction* (2nd ed.): Prentice Hall.

Eckstein, J. (2000). *Learning to Teach and Learning to Learn* [Online]. Retrieved Jan 16, 2003, from http://www.pedagogicalpatterns.org/examples/LearningAndTeaching.pdf

Eckstein, J., Bergin, J., & Sharp, H. (2002a). *Feedback Patterns* [Online]. Retrieved Jan 4, 2003, from http://www.pedagogicalpatterns.org/current/feedback.pdf

Eckstein, J., Bergin, J., & Sharp, H. (2002b). *Patterns for Active Learning* [Online]. Retrieved Jan 4, 2003, from http://www.pedagogicalpatterns.org/current/activelearning.pdf

Eckstein, J., Manns, M. L., Sharp, H., & Sipos, M. (2003). *Teaching from different Perspectives* [Online]. Retrieved Aug 28, 2003, from http://www.pedagogicalpatterns.org/current/FromDifferentPerspectives.pdf

Eckstein, J., Marquardt, K., Manns, M. L., & Wallingford, E. (2001). *Patterns for Experiential Learning* [Online]. Retrieved Nov 25, 2002, from http://www.pedagogicalpatterns.org/current/experientiallearning.pdf

Eden, A. H., Hirshfeld, Y., & Yehudai, A. (1998). *LePUS - A Declarative Pattern Specification Language* (Technical Report No. 326/98). Tel Aviv: Department of Computer Science, Tel Aviv University.

Eden, A. H., Yehudai, A., & Gil, J. (1997). *Precise Specification and Automatic Application of Design Patterns.* Proceedings of 12th IEEE International Conference on Automated Software Engineering (ASE) '97, Incline Village, Nevada, USA, pp. 143-152.

e-Learning Centre. (2004). *Mobile learning* [Online]. Retrieved Feb 2, 2004, from http://www.e-learningcentre.co.uk/guide2elearning/mobile.htm

E-LEN Project. (2003). *E-LEN project homepage* [Online]. Retrieved Jan 14, 2004, from http://www.tisip.no/E-LEN/

Enkenberg, J. (2001). Instructional design and emerging teaching models in higher education. *Computers in Human Behavior, 17*, 495-506.

Ensher, E. A., Heun, C., & Blanchard, A. (2003). Online mentoring and computer-mediated communication: New directions in research. *Journal of Vocational Behavior, 63* (2003), 264-288.

Eriksson, H.-E., & Penker, M. (1998). *UML Toolkit*. New York: John Wiley & Sons.

Ewing, J., & Miller, D. (2002). A framework for evaluating computer supported collaborative learning. *Educational Technology & Society, 5* (1), 112-118.

Firesmith, D. G. (1993). *Object-Oriented Requirements Analysis and Logical Design: A Software Engineering Approach*. New York: John Wiley & Sons.

Forrester, D., & Jantzie, N. (1998). *Learning Theories* [Online]. Retrieved Aug 29, 2003, from http://www.ucalgary.ca/~gnjantzi/learning_theories.htm

Fowler, M. (1998). *Analysis Patters: Reusable Object Models*. Menlo Park, CA: Addison Wesley.

Fricke, A., & Völter, M. (2000). *Seminars - A Pedagogical Pattern Language about teaching seminars effectively* [Online]. Retrieved Dec 12, 2002, from http://www.voelter.de/data/pub/tp/tp.pdf

Gagnè, R. M. (1977). *The Conditions of Learning* (3rd ed.). New York: Holt, Rinehart & Winston.

Gagnè, R. M., & Briggs, L. J. (1979). *Principles of Instructional Design* (2nd ed.). New York: Holt, Rinehart & Winston.

Gamboa, F., Perez-Silva, J. L., Lara-Rosano, F., Cabiedes, F., & Miranda, A. I. (2001). A Student Centered Methodology for the Development of a Physics Video Based Laborator. *Interacting with Computers, 13* (5), 527-548.

Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1995). *Design Patterns - Elements of Reusable Object-Oriented Software*. Reading, MA: Addison-Wesley.

Gamma, E., Helm, R., Johnson, R. E., & Vlissides, J. (1993). *Design Patterns: Abstraction and Reuse of Object-Oriented Design.* Proceedings of European Conference on Object-Oriented Programming (ECOOP) '93, Kaiserslautern, Germany, pp. 406-431.

Gardner, K. M., Rush, A. R., Crist, M., Konitzer, R., Odell, J. J., & Teegarden, B. (1998). *Cognitive Patterns: Problem-Solving Frameworks for Object Technology*. New York: Cambridge University Press.

Garrison, D. R., & Kanuka, H. (2004). Blended learning: Uncovering its transformative potential in higher education. *The Internet and Higher Education, 7* (2), 95-105.

Garzotto, F., Retalis, S., Tzanavari, A., & Cantoni, I. (2004). *From Pedagogical Paradigm to Hypermedia Design Patterns: Where to start?* Proceedings of ED-MEDIA 2004: World Conference on Educational Multimedia, Hypermedia & Telecommunications, June 21-26, 2004, Lugano, Switzerland.

Geyer-Schulz, A., & Hahsler, M. (2001). *Software Engineering with Analysis Patterns* (Technical Report No. 01/2001). Vienna: Institut für Informationsverarbeitung und -wirtschaft, Wirschaftsuniversität Wien.

Goodyear, P., Avgeriou, P., Baggetun, R., Bartoluzzi, S., Retalis, S., Ronteltap, F., et al. (2004). *Towards a Pattern Lanugage for Networked Learning*. Proceedings of 4th International Conference on Networked Learning (NLC'04), Lancaster, England, pp. 449-455.

Govindasamy, T. (2002). Successful implementation of e-Learning - Pedagogical considerations. *The Internet and Higher Education, 4* (3-4), 287-299.

Graham, I. (1995). *Migrating to Object Technology*. Workingham, England: Addison-Wesley.

Gruber, H., Mandl, H., & Renkl, A. (1995). Situiertes Lernen in multimedialen Lernumgebungen. In L. J. Issing & P. Klimsa (Eds.), *Information und Lernen mit Multimedia* (pp. 167-178). Weinheim: Beltz Psychologie Verlags Union.

Hailey, D. E. J., Grant-Davie, K., & Hult, C. A. (2001). Online education horror stories worthy of a Halloween: A short list of problems and solutions in online instruction. *Computers and Composition, 18*, 387-397.

Hamid, A. A. (2002). e-Learning - Is it the "e" or the learning that matters? *The Internet and Higher Education, 4* (3-4), 311-316.

Harrison, N., Foote, B., & Rohnert, H. (Eds.). (1999). *Pattern Languages of Program Design 4*. Reading, MA: Addison Wesley.

Henderson-Sellers, B. (1995). *Object-Oriented Metrics: Measures of Complexity*. Upper Saddle River, NJ: Prentice-Hall.

Hiltz, S. R., & Turoff, M. (2002). What makes learning networks effective? *Communications of the ACM, 45* (4), 56-59.

Hitz, M., & Kappel, G. (2003). *UML@Work* (2nd ed.). Heidelberg, Germany: d.punkt Verlag.

Holzinger, A. (2001). *Definition of Multimedia*. Retrieved April 8, 2004, from http://www-ang.kfunigraz.ac.at/~holzinge/mml/mml-multimedia-definition.html

Holzinger, A. (2005). Usability Engineering Methods for Software Developers. *Communications of the ACM, 48* (1), 71-74.

Hummel, K. A. (2003). *Kontextsensitives Mobiles Lernen*. Proceedings of Fachtagung Mobiles Lernen und Forschen [in German], Kassel, Germany.

IEEE LTSC. (2004a). *IEEE LTSC*. Retrieved Feb 1, 2004, from http://ltsc.ieee.org/

IEEE LTSC. (2004b). *IEEE LTSC WG1: Architecture & Reference Model*. Retrieved Feb 1, 2004, from http://ltsc.ieee.org/wg1/

IEEE LTSC. (2004c). *IEEE LTSC WG12: Learning Object Metadata*. Retrieved Feb 1, 2004, from http://ltsc.ieee.org/wg12/

IMS Global Learning Consortium. (2003a). *About IMS*. Retrieved Feb 1, 2004, from http://www.imsproject.org/aboutims.cfm

IMS Global Learning Consortium. (2003b). *IMS Learning Design Best Practice and Implementation Guide*. Retrieved Nov 21, 2003, from http://www.imsglobal.org/learningdesign/ldv1p0/imsld_bestv1p0.html

IMS Global Learning Consortium. (2003c). *IMS Learning Design Information Model*. Retrieved Aug 10, 2004, from http://www.imsglobal.org/learningdesign/ldv1p0/imsld_infov1p0.html

IMS Global Learning Consortium. (2003d). *IMS Learning Design Specification*. Retrieved Oct 4, 2003, from http://www.imsglobal.org/learningdesign/index.cfm

IMS Global Learning Consortium. (2004). *IMS Specifications*. Retrieved Feb 1, 2004, from http://www.imsglobal.org/specifications.cfm

Infinite Innovations Ltd. (2001). *Definitions of Brainstorming* [Online]. Retrieved Dec 14, 2003, from http://www.brainstorming.co.uk/tutorials/definitions.html#brainstorming

Ingram, A. L., Hathorn, L. G., & Evans, A. (2000). Beyond chat on the internet. *Computers & Education, 35*, 21-35.

Ismail, J. (2002). The design of an e-learning system ~ Beyond the hype. *The Internet and Higher Education, 4* (4), 329-336.

ISO. (no date). *JTC 1/SC 36*. Retrieved Feb 1, 2004, from http://www.iso.org/iso/en/stdsdevelopment/tc/tclist/TechnicalCommitteeDetailPage.TechnicalCommitteeDetail?COMMID=4997

Issing, L. J., & Klimsa, P. (Eds.). (1997). *Information und Lernen mit Multimedia* (2nd ed.). Weinheim: Psychology Verlags Union.

Jacobson, I., Booch, G., & Rumbaugh, J. (1999). *The Unified Software Development Process*. Reading, MA: Addison-Wesley.

Jacobson, I., Christerson, M., Jonsson, P., & Övergaard, G. (1992). *Object-Oriented Software Engineering - A Use-Case Driven Approach*. Reading, MA: Addison-Wesley.

Jacobson, I., Ericsson, M., & Jacobson, A. (1995). *The Object Advantage - Business Process Reengineering with Object Technology*. Workingham, England: Addison-Wesley.

Johnson, R. E. (1992). *Documenting Frameworks using Patterns*. Proceedings of Object-Oriented Programming, Languages, and Applications (OOPSLA) '92, Vancouver, Canada, pp. 63-76.

Karagiannis, D., Junginger, S., & Kühn, H. (1999). *Foliensammlung zu "Geschäftsprozessmodellierung" [slide collection "Business Process Modeling"]*. Vienna: Deparment of Knowledge Engineering, University of Vienna.

Kendall, E. A. (1998). Utilizing Patterns and Pattern Languages in Software Engineering Education. *Annals of Software Engineering, 6*, 281-294.

Kirschenbaum, H., & Henderson, V. L. (Eds.). (1989). *The Carl Rogers Reader*. London: Constable.

Klose, M. (2002). *Design Patterns für digitale Produkte im digitalen Wirtschaftsraum.* Unpublished PhD Thesis, University of St. Gallen, St. Gallen, Switzerland.

Kock, N. (2003). The three threats of action research: a discussion of methodological antidotes in the context of an information systems study. *Decision Support Systems, 1062*, 1-22.

Kolas, L., & Staupe, A. (2004). *Implementing delivery methods by using pedagogical design patterns.* Proceedings of ED-MEDIA 2004: World Conference on Educational Multimedia, Hypermedia & Telecommunications, June 21-26, 2004, Lugano, Switzerland.

Koper, R. (2001). *Modeling units of study from a pedagogical perspective: the pedagogical meta-model behind EML*, from http://eml.ou.nl/introduction/docs/ped-metamodel.pdf

Kraft, N. (2003). *Criteria for Authentic Project-Based Learning*. Retrieved Dec 22, 2003, from http://www.rmcdenver.com/useguide/pbl.htm

Kruse, K. (2002). *e-Learning Alphabet Soup - A Guide to Terms* [Online]. Retrieved Aug 25, 2003, from http://www.e-learningguru.com/articles/art1_1.htm

Laurillard, D. (2001). *Rethinking University Teaching: A Conversational Framework for the Effective Use of Learning Technologies* (2nd ed.). London: Routledge Farmer.

Lave, J., & Wenger, E. (1991). *Situated Learning: Legitimate Peripheral Participation*. Cambridge, UK: Cambridge University Press.

Lea, D. (1994). Christopher Alexander: An Introduction for Object-Oriented Designers. *ACM Software Engineering Notes, 19* (1), 39-46.

Levy, P. (2002). *Researching Networked Learning and Teaching: a Case Study in Practitioner Knowledge Construction.* Proceedings of 3rd International Conference on Networked Learning (NLC), Shefflield, UK, pp. 100-108.

Macdonald, J. (2003). Assessing online collaborative learning: process and product. *Computers & Education, 40* (2003), 377-391.

Mallich, K. (2003). *Auswertungen zur Lehrveranstaltung Web Engineering - SS 2003* (Technical Report): Department of Computer Science and Business Informatics, University of Vienna.

Mangler, J. (2005). *CEWebS - Cooperative Environment Web Services* (Master thesis): Faculty of Computer Science, Univ. of Vienna.

Mangler, J., & Derntl, M. (2004). *CEWebS - Cooperative Environment Web Services.* Proceedings of 4th International Conference on Knowledge Management (I-KNOW '04), June 30-July 2, Graz, Austria, pp. 617-624.

Martin, R. C., Riehle, D., & Buschmann, F. (Eds.). (1998). *Pattern Languages of Program Design 3.* Reading, MA: Addison-Wesley.

McCormack, C., & Jones, J. D. (1997). *Building a Web-Based Education System.* New York, NY: Wiley.

McGorry, S. Y. (2003). Measuring quality in online programs. *The Internet and Higher Education, 6* (2), 159-177.

McNeil, S. G., Robin, B. R., & Miller, R. M. (2000). Facilitating interaction, communication and collaboration in online courses. *Computers & Geosciences, 26* (2000), 699-708.

Mergel, B. (1998). *Instructional Design & Learning Theory* [Online]. Retrieved Sept 1, 2003, from http://www.usask.ca/education/coursework/802papers/mergel/brenda.htm

Meszaros, G., & Doble, J. (1998). A Pattern Language for Pattern Writing. In R. C. Martin, D. Riehle & F. Buschmann (Eds.), *Patter Languages of Program Design 3* (pp. 529-574). Reading, MA: Addison-Wesley.

Mockford, C. D. (1994). The use of peer group review in the assessment of project work in higher education. *Mentoring and Tutoring, 2* (2), 45-52.

Moore, M. G., & Kearsley, G. (1996). *Distance Education: A Systems View*. Belmont, CA: Wadsworth.

Motschnig-Pitrik, R. (2001). *Using the Internet with the Student-Centred Approach to Teaching - Method and Case Study.* Proceedings of International Workshop on Interactive Computer-Aided Learning (ICL) 2001, Villach, Austria.

Motschnig-Pitrik, R. (2002a). getProfile: Anforderungsanalyse an Wirschaftsinformatiker(innen) aus der Sicht der Wirtschaft. *OCG Journal, 1*, 8-11.

Motschnig-Pitrik, R. (2002b). *Supporting Student-Centred Teaching with New Media: Case Study and Experience Report.* Proceedings of 3rd International Conference on Networked Learning (NLC) 2002, Sheffield, U.K, pp. 456-464.

Motschnig-Pitrik, R. (2004a). Blended Learning in einer grossen Informatik-Lehrveranstaltung: Personenzentriert oder Handlungsorientiert? In H. O. Mayer & D. Treichel (Eds.), *eLearning und Handlungsorientiertes Lernen - Grundlagen und Praxisbeispiele* (pp. 219-248). Munich, Germany: Oldenbourg.

Motschnig-Pitrik, R. (2004b). *Person Centered e-Learning in a major academic course: What are the results and what can we learn from them?* Proceedings of 4th International Conference on Networked Learning (NLC), Lancaster, UK.

Motschnig-Pitrik, R., & Derntl, M. (2002). *Student-Centered e-Learning (SCeL): Concept and application in a students' project on supporting learning.* Proceedings of International Workshop on Interactive Computer-Aided Learning (ICL) 2002, Villach, Austria.

Motschnig-Pitrik, R., & Derntl, M. (2003a). *Can the Web Improve the Effectiveness of Person-Centered Learning? Case Study on Teaching and Living Web-Engineering.* Proceedings of International Conference WWW/Internet (ICWI'03), Algarve, Portugal.

Motschnig-Pitrik, R., & Derntl, M. (2003b). *Cooperative Person-Centered e-Learning: Moving from Exams to a Web-Engineering Knowledge Base.* Proceedings of International Workshop for Interactive Computer-Aided Learning (ICL'03), Villach, Austria.

Motschnig-Pitrik, R., Derntl, M., & Mangler, J. (2003). *Web-Support for Learning Contracts: Concept and Experiences.* Proceedings of Second International Conference on Multimedia and Information & Communication Technologies in Education (m-ICTE'03), Badajoz, Spain.

Motschnig-Pitrik, R., & Holzinger, A. (2002). Student-Centered Teaching Meets New Media: Concept and Case Study. *Journal of Educational Technology & Society, 5* (4), 160-172.

Motschnig-Pitrik, R., & Mallich, K. (2002). *Learning at three levels: Web-support and experiential learning in project management.* Proceedings of International Workshop for Interactive Computer-Aided Learning (ICL'02), Villach, Austria.

Motschnig-Pitrik, R., & Mylopoulos, J. (1992). Classes and Instances. *International Journal of Intelligent and Cooperative Information Systems, 1* (1), 61-92.

Motschnig-Pitrik, R., Randa, P., & Vinek, G. (2002). *Specifying and Analysing Static and Dynamic Patterns of Administrative Processes.* Proceedings of Xth European Conference on Information Systems (ECIS) 2002, Gdansk, Poland, pp. 862-871.

Mylopoulos, J., & Easterbrook, S. (2003). *What is Conceptual Modeling?* [Presentation slides]. Retrieved Aug 11, 2003, from http://www.cs.toronto.edu/~jm/340S/PDF2/CM2.pdf

Naidu, S., Cunnington, D., & Jasen, C. (2002). The Experience of Practitioners with Technology-Enhanced Teaching and Learning. *Journal of Educational Technology & Society, 5* (1), 23-34.

Neumann, H. A. (1998). *Objektorientierte Softwareentwicklung mit der Unified Modeling Language (UML).* Munich, Germany; Vienna, Austria: Hanser.

Nichols, M. (2003). A theory for eLearning. *Journal of Educational Technology & Society, 6* (2), 1-10.

Nielsen, J. (1995). *Multimedia and Hypertext - The Internet and Beyond*. Boston, MA: AP Professional.

Nierstrasz, O. (1989). A Survey of Object-Oriented Concepts. In W. Kim & F. H. Lochovsky (Eds.), *Object-Oriented Concepts, Databases, and Applications* (pp. 3-21). Reading, MA: Addison-Wesley.

Nykl, L., & Motschnig-Pitrik, R. (2002). Eine Theorie zur Vereinigung der Persönlichkeitstheorie von C. R. Rogers mit dem Gedankengut von L. S. Vygotskij. *Psychologie in Österreich, 5/2002*, 252-261.

Object Management Group. (2003). *OMG Unified Modeling Language Specification, Version 1.5, March 2003* [Online]. Retrieved May 12, 2003, from http://www.omg.org/cgi-bin/doc?formal/03-03-01

O'Donohue, W., & Kitchener, R. (Eds.). (1999). *Handbook of Behaviorism*. San Diego, California: Academic Press.

OMG. (2003). *OMG Unified Modeling Language Specification, Version 1.5, March 2003* [Online]. Retrieved May 12, 2003, from http://www.omg.org/cgi-bin/doc?formal/03-03-01

Open Learning Technology Corporation. (1996). *Anchored Instruction* [Online]. Retrieved Aug 29, 2003, from http://www.educationau.edu.au/archives/cp/04a.htm

Orvis, K. L., Wisher, R. A., Bonk, C. J., & Olson, T. M. (2002). Communication patterns during synchronous Web-based military training in problem solving. *Computers in Human Behavior, 18*, 783-795.

Ottosson, S. (2003). Participation action research - A key to improved knowledge of management. *Technovation, 23*, 87-94.

Passerini, K., & Granger, M. J. (2000). A developmental model for distance learning using the Internet. *Computers & Education, 34*, 1-15.

Pedagogical Patterns Project. (2000). *The Pedagogical Patterns Project* [Online]. Retrieved Jan 16, 2003, from http://sol.info.unlp.edu.ar/ppp

Pedagogical Patterns Project. (2002). *Pedagogical Patterns Project homepage* [Online]. Retrieved Nov 5, 2002, from http://www.pedagogicalpatterns.org

Pena-Mora, F., & Vadhavkar, S. (1996). Augmenting design patterns with design rationale. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing, 11*, 93-108.

Pree, W. (1994). *Design Patterns for Object-Oriented Software Development*. Reading, MA: Addison-Wesley.

Reeves, T. (1997). *Evaluating What Really Matters in Computer-Based Education* [Online]. Retrieved Aug 29, 2003, from http://www.educationau.edu.au/archives/CP/REFS/reeves.htm

Riehle, D., & Züllighoven, H. (1996). Understandung and Using Patterns in Software Development. *Theory and Practice of Object Systems, 2* (1), 3-13.

Rogers, C. R. (1959). A Theory of Therapy, Personality, and Interpersonal Relationships, as Developed in the Client-Centered Framework. In S. Koch (Ed.), *Psychology: A Study of a Science* (Vol. 3): McGraw-Hill.

Rogers, C. R. (1961). *On Becoming a Person - A Psychotherapists View of Psychotherapy*. London: Constable.

Rogers, C. R. (1983). *Freedom to Learn for the 80's*. Columbus, Ohio: Charles E. Merrill Publishing Company.

Rogers, C. R., & Freiberg, H. J. (1994). *Freedom to Learn* (3rd ed.). Columbus, Ohio: Charles E. Merrill Publishing Co.

Rosenberg, L. H. (1998). *Applying and Interpreting Object Oriented Metrics*. Proceedings of 10th Annual Software Technology Conference (STC'98), Salt Lake City, Utah. Available online at http://satc.gsfc.nasa.gov/support/STC_APR98/apply_oo/apply_oo.html.

Royce, W. W. (1970). *Managing the Development of Large Software Systems: Concepts and Techniques*. Proceedings of IEEE Western Electronic Show and Convention (WESCON), Los Angeles, CA, pp. 1-9.

RSS-DEV Working Group. (2001). *RDF Site Summary (RSS) 1.0 Specification* [Online]. Retrieved Mar 29, 2004, from http://web.resource.org/rss/1.0/spec

Rumbaugh, J., Blaha, M., Premerlani, W., Eddy, F., & Lorenson, W. (1991). *Object-Oriented Modeling and Design*. Englewood Cliffs, NJ: Prentice Hall.

Ryback, D. (1998). *Putting Emotional Intelligence to Work - Successful Leadership is More Than IQ*. Boston, Massachusetts: Butterworth-Heinemann.

Salmon, G. (2000). *E-Moderating - The Key to Teaching and Learning Online*. London: Kogan Page.

Sampson, D. G., & Kastradas, K. (2004). Guest Editorial: Special Theme on Learning Communities & Web Service Technologies. *IEEE Learning Technology, 6* (1), 5-6.

San Mateo County Office of Education. (2001). *Why Do Project-Based Learning?* Retrieved Dec 22, 2003, from http://pblmm.k12.ca.us/PBLGuide/WhyPBL.html

Schmidt, D., Stal, M., Rohnert, H., & Buschmann, F. (2000). *Pattern-Oriented Software Architecture. Volume 2: Patterns for Concurrent and Networked Objects*. Chichester, U.K.: John Wiley & Sons.

Schreurs, J., Moreau, R., & Picart, I. (2003). *Blended learning and live sessions in livelong learning - A case study.* Proceedings of International Workshop on Interactive Computer-Aided Learning (ICL), Villach, Austria.

Seng, L., & Mohamad, F. S. (2002). Online learning: Is it meant for science courses? *The Internet and Higher Education, 5* (2), 109-118.

Shaw, M. (1989). *Heterogeneous design idioms for software architecture.* Proceedings of Fifth International Workshop on Software Specification and Design, Pittsburgh, Pennsylvania, USA, pp. 143-146.

SIL International. (1999a). *What is a workshop?* [Online]. Retrieved Oct 27, 2002, from http://www.sil.org/linguistics/GlossaryOfLinguisticTerms/WhatIsAWorkshop.htm

SIL International. (1999b). *What is an idiom?* [Online]. Retrieved Dec 2, 2002, from http://www.sil.org/linguistics/GlossaryOfLinguisticTerms/WhatIsAnIdiom1.htm

Simons, G. F. (1994). *Conceptual modeling versus visual modeling: a technological key to building consensus* [Online paper]. Retrieved Aug 11, 2003, from http://www.sil.org/cellar/ach94/ach94.html

Sluijsmans, D. M. A., Moerkerke, G., van Merrienboer, J. J. G., & Dochy, F. J. R. C. (2001). Peer Assessment in Problem Based Learning. *Studies in Educational Evaluation, 27*, 153-173.

Sommerville, I. (1992). *Software Engineering* (4th ed.). Workingham, England: Addison-Wesley.

Sowe, S. K., Karoulis, A., Stamelos, I., & Bleris, G. L. (2004). Free Open-Source Learning Community and Web-Based Technologies. *IEEE Learning Technology, 6* (1), 27-29.

Steege, S. (1999). *What is Brainstorming?* Retrieved Dec 14, 2003, from http://www.buffalostate.edu/centers/creativity/Resources/Reading_Room/Steege-99.html

Steeples, C., & Zenios, M. (2004). *Organisational Patterns for E-Leraning Centres.* Proceedings of ED-MEDIA 2004: World Conference on Educational Multimedia, Hypermedia & Telecommunications, June 21-26, 2004, Lugano, Switzerland.

Stewart, B. L. (2004). Online learning: a strategy for social responsibility in educational access. *The Internet and Higher Education, 7* (4), 299-310.

Stroustroup, B. (1986). *The C++ Programming Language* (1st ed.). Reading, MA: Addison-Wesley.

Stumptner, M., & Schrefl, M. (2000). *Behavior Consistent Inheritance in UML.* Proceedings of 19th International Conference on Conceptual Modeling (ER2000), October, 2000, Salt Lake City, Utah, pp. 527-542.

Susman, G. I., & Evered, R. D. (1978). An assessment of the scientific merits of action research. *Administrative Science Quarterly, 23* (4), 582-603.

Tausch, R., & Tausch, A.-M. (1998). *Erziehungs-Psychologie*. Göttingen, Germany: Hogrefe.

Thomas, J. W. (2000). *A Review of Research on Project-Based Learning*. Retrieved Jan 9, 2004, from http://www.bobpearlman.org/BestPractices/PBL_Research.pdf

Tidwell, J. (1999). *Common Ground* [Online]. Retrieved Nov 19, 2002, from http://www.mit.edu/~jtidwell/interaction_patterns.html

Tidwell, J. (2002). *UI Patterns and Techniques* [Online]. Retrieved Nov 19, 2002, from http://time-tripper.com/uipatterns

Trifonova, A., & Ronchetti, M. (2003). *Context Dependent Services in an M-Learning Environment: The Printing Case.* Proceedings of International Conference WWW/Internet 2003, Algarve, Portugal, pp. 635-638.

Trochim, W. M. K. (2000). *Research Methods Knowledge Base: Levels of Measurement*. Retrieved Jan 18, 2004, from http://trochim.human.cornell.edu/kb/measlevl.htm

Tsai, S., & Machado, P. (2002). E-learning, Online Learning, Web-based Learning, or Distance Learning: Unveiling the Ambiguity in Current Terminology. *eLearn Magazine, 2002* (7), 3.

van Welie, M., & van der Veer, G. C. (2003). *Pattern Languages in Interaction Design: Structure and Organization.* Proceedings of 9th IFIP TC13 International Conference on Human-Computer Interaction (Interact 2003), Sept 1-3, 2003, Zürich, Switzerland.

Vlissides, J. (1998). *Pattern Hatching: Design Patterns Applied*. Reading, MA: Addison-Wesley.

Vlissides, J. M., Coplien, J. O., & Kerth, N. L. (Eds.). (1996). *Pattern Languages of Program Design 2*. Reading, MA: Addison-Wesley.

Vygotsky, L. S. (1978). *Mind in Society*. Cambridge, MA: University Press.

Vygotsky, L. S. (1992). *Geschichte der höheren psychischen Funktionen*. Münster: Lit Verlag.

W3C. (2001). *Web Service Description Language (WSDL) 1.1*. Retrieved Mar25, 2004, from http://www.w3.org/TR/wsdl

W3C. (2003a). *HTTP - Hypertext Transfer Protocol*. Retrieved Feb 8, 2004, from http://www.w3.org/Protocols/

W3C. (2003b). *Web Services Architecture*. Retrieved Jan 12, 2004, from http://www.w3.org/TR/ws-arch/

W3C. (2004a). *HyperText Markup Language (HTML)*. Retrieved Feb 8, 2004, from http://www.w3.org/MarkUp/

W3C. (2004b). *SOAP Specifications*. Retrieved Oct 8, 2003, from http://www.w3.org/TR/soap

Wallace, F. L., & Wallace, S. R. (2001). Electronic office hours: a component of distance learning. *Computers & Education, 37* (2001), 195-209.

Watson, J. B. (1997). *Behaviorismus* (4th ed.). Eschborn, Germany: Verlag Dietmar Klotz.

Wen, M. L., & Tsai, C.-C. (2003). *College Students' Perceptions of and Attitudes Toward Online Peer Assessment.* Proceedings of ED-MEDIA 2003: World Conference on Educational Multimedia, Hypermedia & Telecommunications, June 23-28, 2003, Honolulu, Hawaii, pp. 1883-1888.

Wenger, E. (1998). *Communities of practice - Learning, meaning, and identity*. Cambridge: Cambridge University Press.

Weston, C., Gandell, T., McAlpine, L., & Finkelstein, A. (1999). Designing Instruction for the Context of Online Learning. *The Internet and Higher Education, 2* (1), 35-44.

Wideman, M. R. (2002). *Project Brainstorming - What is brainstorming?* Retrieved Dec 14, 2003, from http://www.maxwideman.com/issacons4/iac1446/tsld002.htm

Wittern, J.-O., & Tausch, A.-M. (1983). Personenzentrierte Haltungen und Aktivitäten von Lehrern und seelische Lebensqualität ihrer Schüler im Unterricht. *Psychologie in Erziehung und Unterricht, 30*, 128-134.

Wolf, K., & Liu, C. (1995). New Clients with Old Servers: A Pattern Language for Client / Server Frameworks. In J. O. Coplien & D. Schmidt (Eds.), *Pattern Languages of Program Design* (pp. 51-64). Reading, MA: Addison-Wesley.

Wolfinger, R. (2004). *LOCC Lines of Code Counter homepage*. Retrieved May 19, 2005, from http://centcom.sitefactory.at/locc.htm

Young, C., Foulkes, A., & Thomas, P. (2004). *Web-Based Learning - A Guide for Postgraduate Tutors and Lecturers* [Online]. Retrieved Jan 3, 2005, from http://www.glow.ac.uk/resources/part1/soar1.pdf

Zimmer, W. (1995). Relationships between Design Patterns. In J. O. Coplien & D. Schmidt (Eds.), *Pattern Languages of Program Design* (pp. 345-364). Reading, MA: Addison-Wesley.

Appendix: Pattern Intents

🗀 Assessment

- Assessment Phases (169): Use Generic Evaluation to assess participants, and Collect Feedback on the learning activity from participants.

🗀 Course Types

- Interactive Lecture (172): In courses or scenarios where transmission of information is the main goal use Interactive Elements to minimize pure lecturing.
- Lab Course (175): Describes a course type where application-oriented lab practice, with concurrent Project-Based Learning is used throughout the course.
- Project-Based Learning Course (178): Use Project-Based Learning as the primary method of the learning process, and Blended Evaluation of projects for evaluation of participants.
- Seminar (181): Increase active participation in an otherwise presentation-centric seminar by Exchange of Contributions, short presentations with longer discussions in the Presentation Phases, and Blended Evaluation of contributions in the Assessment Phases.

🗀 Evaluation

- Blended Evaluation (186): Use a mix of Self-, Peer- and Instructor-Evaluation to actively involve participants in the Assessment Phases and to take into account as many views on participants' contributions as possible.
- Evaluation (189): Evaluation is used to produce valuing assessment of a participant's learning performance. It generically characterizes scenarios that may be used collateral to learning activities as well as in the Assessment Phases of a course.
- Examination (195): Evaluate participants' learning progress in a structured way by doing oral or written examinations using predefined questions.
- Generic Evaluation (198): Use Instructor-evaluation plus any mix of other Evaluation scenarios in Assessment Phases. This allows instructors to involve participants in the assessment process and to collect multiple views on contributions.
- Instructor-Evaluation (201): The instructor evaluates participants' achievements, contributions, and/or performances in Courses and learning activities. Instructor-evaluation is a necessity in almost any educational scenario.

- INSTRUCTOR-EXAMINATION (204): Evaluate participants using a structured set of questions.
- PEER-EVALUATION (206): Peer-evaluation fosters active participation and engagement of participants, as they take on the roles of peers as evaluators in EVALUATION scenarios.
- SELF-EVALUATION (215): Self-evaluation fosters critical reflection on a participant's own contributions and learning progress, as the participant is both evaluator and evaluation target at the same time.
- SELF-EXAMINATION (219): Provide participants with the option of evaluating themselves in a uniform, structured way by providing questions and expected answers.

🗀 Feedback

- COLLECT FEEDBACK (221): Solicit feedback from participants to enable qualitative analysis and subsequent improvement of the employed learning scenarios.
- FEEDBACK FORUM (224): COLLECT FEEDBACK in a semi-structured way by soliciting postings to instructor-initiated ONLINE DISCUSSION threads. This additionally allows for open discussion of feedback postings.
- QUESTIONNAIRE (229): A questionnaire is a form of COLLECTING FEEDBACK in a structured way by specifying and providing a set of items/questions along with scaled, possible responses.
- REACTION SHEETS (238): Solicit reactions sheets on specific aspects of learning scenarios and activities to COLLECT FEEDBACK in an open, unstructured way.

🗀 General

- ACHIEVEMENT AWARD (245): Reward originators of outstanding contributions as determined in the ASSESSMENT PHASES with achievement award certificates.
- ALTERNATING PHASES (249): Presence phases alternate with online phases. This embodies the essence of blended learning scenarios.
- COLLECT (252): Collect an information item by issuing a collect request so as to make the holder/owner of the item PUBLISH it to the collector.
- COURSE (256): Courses are arranged primarily in three consecutive phases: PRELIMINARY PHASES in the beginning, followed by the main course phases, and concluded by ASSESSMENT PHASES.
- DIARY (272): Make participants' efforts transparent by making them keep track of their work in diaries, especially in collaborative and/or iterative learning processes.

- INITIAL MEETING (276): Outline course style and objectives in an initial MEETING and CONSIDER CONVENTIONAL STYLE.
- PRELIMINARY PHASES (280): PUBLISH relevant content and resources as well as information on course style, activities, and objectives prior to an INITIAL MEETING where these issues are discussed.
- PRESENTATION PHASES (284): Let participants prepare themselves for presentation MEETINGS by EXCHANGE OF CONTRIBUTIONS prior to the presentations. Prepared this way, the traditional long-presentation-and-short-discussion-scenario can be replaced by active discussions following short, concise presentations.
- PUBLISH (287): Disclose an information item (i.e., text, file, or completed form) to a certain target location, person, role, or group of roles and/or persons.
- STAFF MEETING (290): If more than one staff member is involved in the organization and/or execution of a COURSE or learning activity, staff members meet periodically to synchronize processes and discuss evolved issues and problems.
- TEAM WORKSPACES (292): Provide teams with private workspaces, which they can use to create, store, work on, and share their contributions and other documents.

Interactive Elements

- APPROVAL (302): The instructor reviews and approves PROPOSALS according to guidelines PUBLISHED in a proposal request.
- BRAINSTORMING (305): COLLECT and subsequently PUBLISH ideas gathered in brainstorming sessions, either online or present.
- CHAT (310): Provide facilities for synchronous communication among participants, instructors, tutors, and/or guests.
- COMPUTER-MEDIATED COMMUNICATION (313): Asynchronous and synchronous means of online communication allow for online interaction and exchange among participants independent of time and location.
- CONSIDER CONVENTIONAL STYLE (316): Offer participants who dislike self-initiated scenarios the option to switch to a more conventional, directive course style.
- CONSULTATION (319): Provide options for participants to seek synchronous (CHAT) or asynchronous (ONLINE DISCUSSION) consultation from teaching staff or experts regarding specific questions, topics or problems.
- ELABORATE GOALS AND EXPECTATIONS (322): Elaborate and PUBLISH participants' goals and expectations (and also fears) for the course or for specific learning activities.

- EXCHANGE OF CONTRIBUTIONS (326): Let participants exchange and discuss their contributions and ideas online.
- INFORMATION GATHERING (329): Participants and instructors interact with the primary target to collect information which shall be gathered collaboratively and shared among all participants.
- INTERACTIVE ELEMENT (335): Set of learning activities involving active participation and interaction among participants, instructors, and/or tutors.
- MARKET (337): Provide facilities for participants and instructor to offer and exchange any kind of useful documents or resources.
- MEETING (344): Use meetings for face-to-face interaction and for preparation and conclusion of online phases.
- ONLINE DISCUSSION (347): Provide facilities for asynchronous online communication among participants, instructors, tutors, and/or guests.
- PROBLEM PROPOSALS (356): Let participants choose and solve problems of personal or particular professional interest to make learning processes more self-initiated, more authentic, and learning effects more persisting.
- PROPOSAL (359): Use PROPOSAL and subsequent APPROVAL scenarios in contexts where participants are encouraged to freely choose or to propose, for example in PROBLEM PROPOSALS or TEAM BUILDING.
- TEAM BUILDING (363): In teamwork scenarios, let participants choose their team partners. Restrict only team size to about 2 – 5 members, as appropriate for the current learning activity.
- THEORY ELABORATION (369): Certain (aspects of) topics or subject areas are elaborated and subsequently PUBLISHED and/or presented by participants.
- TUTORIAL (373): For complex technical or application-oriented scenarios involving new or sophisticated tools and methods, let tutors do introductory technical tutorials.
- WORKSHOP (376): Use workshops as application-oriented, particularly interactive MEETINGS, where the focus is on collaboration and/or sharing among participants.

🗀 Project-Based Learning

- KNOWLEDGE BASE CONSTRUCTION (378): Use LEARNING CONTRACTS in a way to advance the construction of a knowledge base in a specific subject area from single contributions and knowledge fragments.

- Learning Contracts (381): Let teams/participants propose topics they want to elaborate and sign contracts defining learning targets and expected contributions for each team/participant.
- Project-Based Learning (387): Participants elaborate projects iteratively and incrementally in several successive Project Milestones. Participants may work out individual projects, may be organized in teams, or may collaborate collectively on a single group/course project.
- Project Milestone (397): Project work is accomplished in a series of well-defined phases, each producing a certain set of artifacts. These milestone solutions are Published and presented by the project owners.

LEBENSLAUF

Name:	**Michael Derntl**
Adresse:	Lacknergasse 23/11 1170 Wien, Österreich
Geboren:	17. März 1978 in Linz, Oberösterreich
Nationalität:	Österreich

Ausbildung

1984–1988	Volksschule 45 in Linz, Oberösterreich
1988–1996	Bundesrealgymnasium Fadingerstrasse 4 in Linz, Österreich Abschluss: Matura im Juni 1996
1996–2001	Studium der Wirtschaftsinformatik an der Universität Wien, Österreich Abschluss: Mag.rer.soc.oec im November 2001 Diplomarbeitsthema: "*Geschäftsprozessorientierte Anwendungsentwicklung – Ein Ansatz zur Kopplung von ADONIS und Rational Rose*", betreut von Prof. Dimitris Karagiannis
seit 03/2002	Doktoratsstudium an der Fakultät für Informatik der Universität Wien Dissertationsfach: Wirtschaftsinformatik Dissertationsthema: "*Patterns for Person-Centered e-Learning*", betreut von Prof. Renate Motschnig

Beschäftigung

1999–2002	Selbständiger Softwareentwickler
seit 04/2002	Wissenschaftlicher Mitarbeiter am Institut für Knowledge and Business Engineering (vormals Institut für Informatik und Wirtschaftsinformatik), sowie am Research Lab for Educational Technologies, Fakultät für Informatik, Universität Wien
seit 03/2005	Stellvertretender Leiter des Research Lab for Educational Technologies an der Fakultät für Informatik, Universität Wien