Resolving Variations in Learning Spaces for Experiential Learning

Eric Ras
Fraunhofer IESE, Fraunhofer-Platz 1, 67663 Kaiserslautern, Germany
eric.ras@iese.fraunhofer.de

Abstract. Today, systems should react based on explicit demands from the learner or even proactively react based on changes in the working environment. The success of this type of systems depends on their ability to adapt and personalize the learning environment to the learner's needs. This paper presents an approach using a decision model that allows resolving variations in a so-called learning goal structure template by using different types of context information. These adapted templates are then used to create so-called Learning Spaces, which are developed during the process of reusing explicit experience packages in software engineering. The Learning Spaces are delivered in an adapted Wiki called *Software Organization Platform* (SOP), which integrates knowledge management and e-learning.

Keywords: adaptive educational hypermedia system, context, experiential learning, learning space, software engineering, experience management.

1 Introduction

Software engineering is a very knowledge intensive activity and strongly relies on an individual's competencies. The short innovation cycles in software engineering lead to many learning situations where new knowledge is required to solve new challenges during daily work. In the last thirty years, the fields of software reuse and *experience management* (EM) have increasingly gained importance. EM supports the collection, pre-processing, analysis, and dissemination of experiences.

However, different problems occur when experience documented by experts is reused by novices. Experience is often documented by domain experts. Expert knowledge is somehow 'routine'. This makes it challenging for experts to document experiences appropriately and to make them reusable for others. Novices lack software engineering background knowledge and are not able to connect the experience to their knowledge base. Hence, they often misinterpret or even fail to understand other people's documented experience. A more detailed summary of problems related to understanding and learning from documented experience can be found, for example, in [1] and [2]. Most of our daily learning is, in fact, experience-based. Most of the research done in the area of experiential learning is based on the work of Kolb and Fry [3]. They investigate the on-going learning processes when people learn from their experiences. Ideally, people could learn effectively from experiences when all four phases of Kolb's Experiential Learning Circle [4] are passed. To address the problems, an adaptive educational hypermedia system has been developed to produce so-called Learning Spaces for enhancing the understanding and application of experience packages by using the experiential learning cycle as a basis.

Section 2 lists the different adaptation techniques of Adaptive Educational Hypermedia Systems. Section 3 explains the process for generating Learning Spaces by using decision models. Section 4 provides a conclusion and gives information about upcoming evaluation activities.

2 Adaptive Educational Hypermedia Systems

Adaptive Educational Hypermedia Systems allow learning to be adapted to specific user needs and requirements. Brusilovsky, for example, distinguishes between adaptive navigation and adaptive presentation [5]: *Adaptive navigation* alters the structure presented to the learner according to the individual learner characteristics. Adaptive navigation is used to guide the learner through the learning space. *Adaptive presentation* refers to content adaptation and alters the way content is visually displayed to the learner based on a learner model

The success of adaptation techniques depends on how good an AHS separates the content from its structure and its presentation. For example, the so-called closed corpus problem in adaptive hypermedia states that the systems are working with a closed set of artifacts (e.g., fine-grained learning objects or documents) [5] [6], and that the alterations or modifications are defined in between the documents (e.g., by using the relation 'required prerequisites'). This makes it difficult to reuse the adaptive functionality of the system, and does not allow extending the document space or even work in an open environment like the Web (open corpus). Now, ontologies based on semantic web technologies are increasingly used for modeling knowledge in adaptive web systems.

3 Learning Space Generation

A Learning Space follows a specific global learning goal and is created based on context information about the current situation and the context description of an experience package. The Learning Space is presented by means of linked Wiki pages within the Software Organization Platform (SOP). SOP intends to support specific software engineering activities such as requirements engineering [7], experience management, and project management. Hence, by integrating the Learning Space generation and presentation functionality into SOP, knowledge management and e-learning have been merged into one system [8].

Dey defines context as ”any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves [9].” This approach uses the following context categories (more details about the derivation of these categories can be found in [10]): individual context (e.g., role, skill and competence profiles, learning preference); group context (e.g., team size, team members,); process context (e.g., activity); product context (e.g., type of product, complexity, quality); project context (e.g., size, effort, resources, customer); and organization context (e.g., competence development strategies). This context information can be used for adaptive Learning Space generation where variabilities are resolved by means of queries to the context ontology.

3.1 Basic Concepts and Generation Process

The generation process starts with the adaptation of a so-called generic *LearningSpaceStructureTemplate* (this step is elaborated in more detail in subsequent sections). This template reflects the high-level structure of a Learning Space. Each *LearningSpaceStructureTemplate* is refined by a set of *LearningGoalTemplates*. These templates reflect a concrete learning activity structure and refer to a learning goal by using the taxonomy of educational objectives of Anderson and Krathwohl [11], i.e., a *LearningGoalTemplate* refers to a concrete cognitive process (i.e., (remember, understand, apply, analyze, evaluate, create) and a knowledge dimension (i.e., factual, conceptual, procedural, or meta-cognitive knowledge). For example, a *remember_project* template is from the type *remember conceptual knowledge* because the objective is to *recall* a specific project with all related factual concepts such as individuals, groups, used processes, and customer. Each of these templates is implemented by a *LearningPage* (i.e., this corresponds physically to a Wiki page). Such a page contains several *LearningComponents* consisting of *LearningElements* (see [12] for the details).

Four activities are necessary to produce a context-specific Learning Space (the first one will be described in more detail): Template Resolving: instantiates a generic *LearningSpaceStructureTemplate* by resolving variabilities by means of a decision model and context information; Template Completion: instantiates entries of the *LearningGoalStructureTemplate* by entering concrete topic keywords; Content Search: uses topic keywords and the relations in the *LearningGoalTemplates* to search for concrete learning resources; Content Presentation: entries in the *LearningGoalTemplates* are replaced by *LearningElements* and the templates are transferred to a presentation format (i.e., Wiki), which results in the *LearningSpace*.

3.2 Decision Models

Decision models come from the domain of product line engineering, which is part of software engineering. Product line engineering aims at the systematic development of a set of similar software systems by understanding and controlling their common and distinguishing characteristics [13]. In order to control these so-called variabilities, they need to be identified, their interrelationships have to be defined, and alternatives have to be modeled. Going back to Learning Spaces, variabilities could depend on all context characteristics described previously. For example, project, product, and process characteristics could have an impact on the navigation and presentation adaptivity of the Learning Space. The variabilities are defined by means of so-called optional and alternative variation points, which represent variabilities in the Learning Space. *Optional* variation points refer to two choices, with one choice having to be selected. More than one choice can be selected in case of an *alternative* variation point. An example is illustrated in Fig. 1. A *LearningSpaceStructureTemplate* could contain variable elements in terms of the used *LearningGoalTemplates* and/or the *Links* between them. A decision model contains a set of decisions (i.e., question/choice(s) pairs) that describe and document these variation points. After answering the decisions, the answers are used to resolve the variation points. If a decision refers to one variation point, the decision is called a *simple* decision. *Complex* decisions refer to more than one variation point.

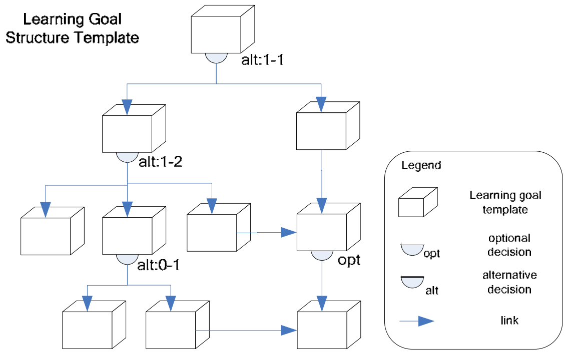

Fig. 1. Example of Learning Space Structure and Decisions

4.3 Resolving Process

Only one *LearningSpaceStructureTemplate* and one related decision model exist for creating a Learning Space that enriches a selected experience package. The adaptation within a Learning Space is done by resolving variabilities in the template by using the context information and the selected global learning goal level (according to [14]), which is chosen by the software engineer (see. Fig. 2). After retrieving the *LearningSpaceStructureTemplate* and the *DecisionModel*, the next step resolves the decisions of the decision model by using information about the *CurrentContext*, the *ContextOntology*, and the *GlobalLearningGoalLevel*. The latter has been selected by the developer when he or she decided to use a Learning Space before reusing the experience package. The context vector refers to concepts of an ontology, which is available in the OWL-DL format. This resolving step resolves on the higher abstraction level of adaptivity. The variabilities on the lower content level (i.e., *LearningComponent* and *LearningElement*) are resolved in the step *Template Completion*. For each question in the decision model, a SPARQL query is forwarded to the *ContextOntology* in order to answer the different decisions. Queries are built based on the context information stored in the description of the experience package and on *CurrentContext*.

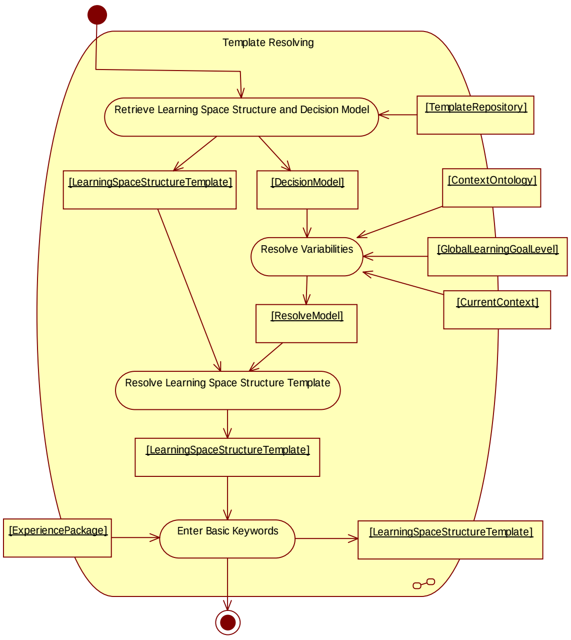

Fig. 2. Template Resolving Activity Diagram

Example question: "Did *John* work in *project xyz*, where the experience package was documented?": A query will be created that retrieves whether the instance "John" of the ontology class *individual* has a relation *working_in* to the instance "project xyz" of the class *project*. If the answer is "no", then the first *LearingGoalTemplate* "Remember_Project" will be deleted. Other decisions on this level are related to the *GlobalLearningGoalLevel* and to the competence level of the individual regarding the product and process addressed by the experience package. The answers, respectively the decisions, are stored in the *ResolveModel*. Each choice of a decision (i.e., alternatives answers of the query) is related to a set of operations that resolve the variation points in the *LearningGoalStructureTemplate*: e.g., delete *LearningGoalTemplate*, add *Link* between specific types of templates, etc. They are executed in the next step *Resolve Learning Goal Structure Template*. After this step, the learning goal templates and the links between them are adapted to the current context and the experience package. The last step enters a first set of basic keyword into the templates, which are used for later retrieval of *LearningElements*. The *LearningGoalStructureTemplate*, the *LearningGoalTemplates*, and the *DecisionModel* are stored in XML.

5 Conclusion and Future Work

Decision models promise a better possibility to separate the variabilities from the structure, content, and its representation. They allow capturing variable characteristics of the Learning Space and allow attaching operations that perform the adaptation to the current working context and experience package. In this approach, resolving is done on two levels. The first one focuses on the level of learning goals and the related templates with links (presented in this paper). The second step refers to the content level and its presentation. The usage of decision models addresses the problem of the closed corpus [5] [6] because the adaptation is not coupled to a fixed set of learning resources, but to types of Learning Space concepts. A one-factor within-subject experiment in experience reuse will be conducted in August 2007 with 24 students at the University of Kaiserslautern. The results will provide a baseline for future investigations regarding the impact of context-aware Learning Space generation on knowledge gain and task performance in experience reuse.

References

[1] E. Ras and S. Weibelzahl, "Embedding Experiences in Micro-didactical Arrangements," presented at SEKE 2004, Banff, Canada, 2004.

[2] J. Rech, E. Ras, and B. Decker, "Riki: A System for Knowledge Transfer and Reuse in Software Engineering Projects: Strategies beyond Tools," in *Open Source for Knowledge and Learning Management*, M. Lytras and A. Naeve, Eds.: Idea Group, Inc., 2006.

[3] D. A. Kolb and R. Fry, "Toward an applied theory of experiential learning," in *Theories of Group Process*, C. Cooper, Ed. London: John Wiley, 1975.

[4] D. A. Kolb, *Experiential learning: experience as the source of learning and development*. Englewood Cliffs, N.J.: Prentice-Hall, 1984.

[5] P. Brusilovsky, "Adaptive Hypermedia," *User Modeling and User-Adapted Interaction*, vol. 11, pp. 87-110, 2001.

[6] N. Henze and W. Nejdl, "Extendible adaptive hypermedia courseware: Integrating different courses and web material," in *Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (AH 2000)*. Trento, Italy, 2000.

[7] B. Decker, E. Ras, J. Rech, P. Jaubert, and M. Rieth, "Wiki-Based Stakeholder Participation in Requirements Engineering," *Software, IEEE*, vol. 24, pp. 28-35, 2007.

[8] E. Ras, J. Rech, and B. Decker, "Lernräume für erfahrungsbasiertes Lernen mit Wiki-Systemen im Software Engineering," *Zeitschrift für E-Learning*, 2007.

[9] A. K. Dey, "Understanding and Using Context," *Personal and Ubiquitous Computing*, vol. 5, pp. 4-7, 2001.

[10] E. Ras, J. Rech, and B. Decker, "Workplace Learning in Software Engineering Reuse," presented at I-KNOW ‘06, Special track on “Integrating Working and Learning”, Graz, Austria, 2006.

[11] L. W. Anderson and D. R. Krathwohl, *A taxonomy for learning, teaching, and assessing: a revision of Bloom's taxonomy of educational objectives*, Complete ed. New York: Longman, 2001.

[12] J. Rech, E. Ras, and B. Decker, "Riki: A System for Knowledge Transfer and Reuse in Software Engineering Projects: Strategies beyond Tools," in *Open Source for Knowledge and Learning Management*, M. Lytras and A. Naeve, Eds.: Idea Group, Inc., 2007.

[13] D. Muthig, "A Lightweight Approach Facilitating an Evolutionary Transition Towards Software Product Lines," in *Series of in Experimental Software Engineering*. Kaiserslautern: PhD Thesis University of Kaiserslautern, 2002.

[14] L. W. Anderson and D. R. Krathwohl, *A taxonomy for learning, teaching, and assessing: a revision of Bloom's taxonomy of educational objectives*, Complete ed. New York: Longman, 2001.