User Context Aware Delivery of E-Learning Material: Approach and Architecture

Andreas Schmidt
(Forschungszentrum Informatik (FZI), Karlsruhe, Germany
Andreas.Schmidt@fzi.de)

Claudia Winterhalter
(Forschungszentrum Informatik (FZI), Karlsruhe, Germany
Claudia.Winterhalter@fzi.de)

Abstract: Current E-Learning solutions are not sufficiently aware of the context of the learner, i.e. the individual characteristics, the organization and the work processes and tasks. This can be achieved by modular learning objects and semantical metadata for their contextualization. This allows to deliver learning material that is relevant to the current situation of the learner. This paper presents the general approach and the architecture.

Key Words: e-learning, learning objects, user context-aware retrieval, ontologies

Categories: K.3.1, H.3.3

1 Introduction

During the last decade, there has been a major shift towards the constructivist learning theory and its descendants. Especially in e-learning, it has become an important insight that learning basically is the construction and refinement of knowledge structures in the learner's mind. This construction process depends mainly on the engagement of the learner. Knowledge cannot simply be transferred or trained, it has to be built in each individual. Learning, thus, in the constructivists' perspective, should be self-determined and situated in real-life situations.

Corporate e-learning has begun to enable workers for self-paced learning. It is however still too little interwoven with the actual work so that there is always a mismatch between the current knowledge demand and applicability on the one side and the delivered learning units on the other side. E-learning so far is not aware of the work processes and situations. In most practical cases, it is not even aware of the specific company it takes place in – due to the fact that the production of learning material tailored to internal business processes is too expensive to produce. Likewise, e-learning is not fully aware of the individual characteristics of the learner (e.g. which pedagogical process or approach fits best).

But how can we make e-learning aware of the work processes, the company characteristics and the individual learner – i.e. the *context* of the learner? This can be achieved by seven major steps that will be explained later in more detail:

- break down courses into modular units (learning objects)
- make learning objects adaptable
- make semantical and didactical relationships between learning objects explicit
- model the learner's context along several dimensions (personal, organizational, topical, …) and their knowledge requirements
- contextualize the learning objects according to this model
- acquire knowledge about the user's situation and its knowledge requirements
- find matching learning objects and adapt their internal structure to the learner

Most of these measures have already been discussed in different areas of e-learning and knowledge management research. We want to present here an integrative, ontology-based approach that allows for putting context-aware e-learning into practice. This approach currently flows into the conception and implementation of Learning in Process[1], an EU co-funded project aiming at contextualized learning object delivery and the integration of e-learning and knowledge management.

2 General Architecture and Framework

2.1 Ontologies for organizational and context modelling

Our approach heavily relies on semantic modelling of the learner's environment. For this purpose, we make use of ontologies and Semantic Web technologies (like e.g. [Stojanovic 01]). In order to keep the modelling task manageable, we divide the ontology into several sub-ontologies in an approach similar to the KnowMore project ([Abecker et al. 00], [Elst et al. 01]):

- Organizational ontology (roles, departments)
- Process ontology (workflow representations)
- Task ontology
- Knowledge area ontology

For each of these ontologies, a vocabulary of base concepts is defined so that for a specific company, it only requires the specification of instances (like actual departments, actual roles and a knowledge tree).

In order to limit the amount of effort required to build an ontology, we have taken a layered approach, i.e. each ontology part is organized in layers so that the upper layers can be shared with other entities and the lower layers can still be extended in a company-specific way.

The approach is also scalable:

- It is not required to model all of these different aspects, e.g. it could be sufficient to model only knowledge areas and organizational structures.

---

[1] http://www.learninginprocess.com

- The level of detail of the models can be chosen according to the needs of the company and later be readjusted.

The most important aspect of ontology-based modelling is the specification of knowledge requirements. With each context dimension like role, organizational unit, process/task etc. we associate a required competency. A *competency* is defined as a knowledge area plus a competency level (e.g. newbie, beginner, skilful, competent, expert). Then we can specify the required competencies for a certain situation.

2.2 Learning Objects and their metadata

The system deals with Learning Objects (LO), which are the minimal unit of pedagogically reasonable learning content. This learning object consists of arbitrary content (like text, pictures, animations, video sequences etc.) and metadata that describes the learning objects. We use here the metadata defined by LOM plus some extensions that are required by our platform.

The key to providing context-aware learning object delivery is to describe the (1) objectives and (2) prerequisites in the form of competencies. Objectives are the competencies that successful completion of the Learning Object will deliver, prerequisites are competencies that are required to understand the content of the Learning Object.

Additionally, a Learning Object can have semantic and didactical relationships with other learning objects (like *references*, *is-example-for* etc.). These relationships are defined in a didactical ontology. These allow to capture semantic connections between content units that were split apart due to the modularization into Learning Objects.

It is to note that that the learner normally is not delivered a single Learning Object so that also his learning is atomized. The primary unit of delivery to the learner is the Learning Program which basically represents a package of Learning Objects that are already personalized to the individual learner. So modularization into Learning Objects is basically something the system needs internally to flexibly combine the same content units for different purposes.

2.3 Architecture of the system

In LIP we have foreseen four main user roles: Coordinator (for managing the ontologies and global properties), Author (for creating Learning Objects), Learners and Administrators (for system administration).

The different types of users interact with the three main parts of the system:

- **Ontology Editor.** A learning coordinator or author can model the different aspects of a company and of the required knowledge. This ontology editor is based on the KAON OIModeler[2].
- **Authoring tool.** This tool is not designed to create content (here we leverage on existing content creation tools), but for creating learning objects from

[2] http://kaon.semanticweb.org

existing content and contextualize them with references to the ontology. The objectives and prerequisites of a Learning Object have to be specified.

- **Delivery platform.** This is the actual learner's environment for (context-aware or assigned) delivery of learning objects and managing learning progress.
- **User Monitoring.** As an optional and user application-specific component, we also have user monitoring facilities. These allow to capture information about what the user is currently doing.

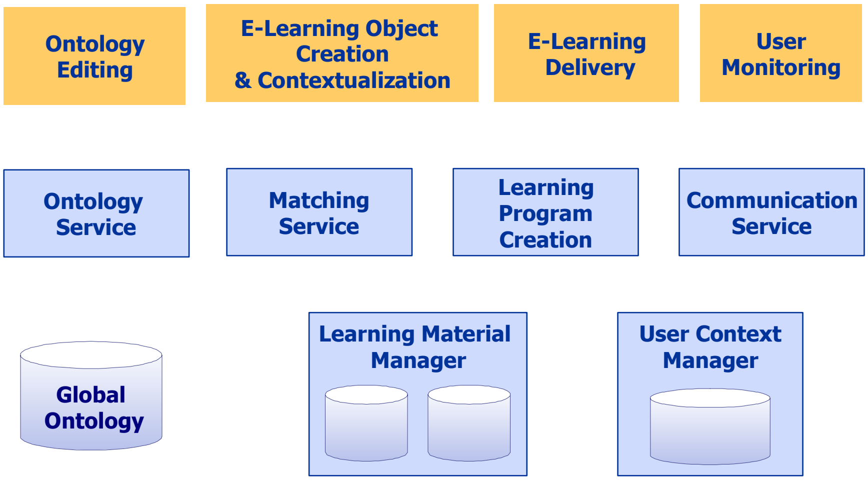

Figure 1: High-level architecture of the system

Below we have a layer of services that contain the application logic.

- **Ontology Service.** This service provides persistence and query capabilities for the ontological data. It also offers possibilities to define views and register triggers that are to be executed when certain changes occur.
- **Matching Service.** The task of this service is to bring together the situation of the user and the available Learning Objects. For that, it calculates the knowledge requirements of the current context and checks for Learning Objects that can deliver missing competencies.
- **Learning Program Creation.** This component is responsible for compiling complete Learning Programs, containing everything required to understand successfully complete a set of Learning Objects.
- **Communication Service.** As learning has been recognized to be highly dependent on social interactions, communication is a very important feature in the system. For the actual implementation, we rely on messaging infrastructures like groupware and instant messaging system, but because we

know what users do, what they learn etc. we can facilitate the search for communication partners.

Two of these services, which constitute the key innovations of LIP, are the matching service and the learning program creation service, which will be described below in more detail.

The lowermost layer consists of the following services:

- **Ontology Repository.** This is the actual persistent storage of the ontological data (based on database technologies).
- **Learning Object Manager.** This component stores Learning Objects including their metadata and their content and makes them accessible for the upper-layer services.
- **User Context Manager.** The user context manager is a very important part of the system. Only the knowledge about the user enables the system to suggest appropriate learning material to the user. The user context manager collects all information supplied by different context sources and provides an aggregated view, taking into account the imperfection of the information.

3 Delivery Process

3.1 Context acquisition

Apart from modelling the user context, the most crucial point in context-aware applications is the acquisition of context information. How do we know what the user currently does, or what he intends to do? Of course, there is no single way of determining a user's context. This heavily depends on the work environment. Therefore, we have foreseen an open approach that allows for plugging in different context sources. These context sources only have to describe their view of the context of the user according to the company's ontology. These context perspectives are then integrated into a single context abstraction. Context sources could be:

- **Workflow systems**. If a company has a workflow management system already in place, this will be a major source of information about the current task ([Maus01]).
- **Human Resources systems.** For interoperability with Human Resources systems, we provide mapping functionality between our internal context (or user) model and HR-XML, which is an emerging standard in that field.
- **Web browsers.** We are currently implementing browser plugins (as Browser Helper Object for Internet Explorer and a plugin for Mozilla) that allows to capture relevant interactions from web-based user interactions.
- **Office applications.** We can use standardized customization facilities of office applications to generate semantic events, e.g. based on templates or documents opened. A Microsoft Office plugin is currently under development that captures specific interactions of the user with the system.

In the future, we can also make use of Text Mining algorithms that acquire ontological information.

- **Custom applications.** In order to support not only standard off-the-shelf software solutions, but also custom enterprise applications, a framework is being developed that allows to define application-specific semantic events and their effects on user context information.

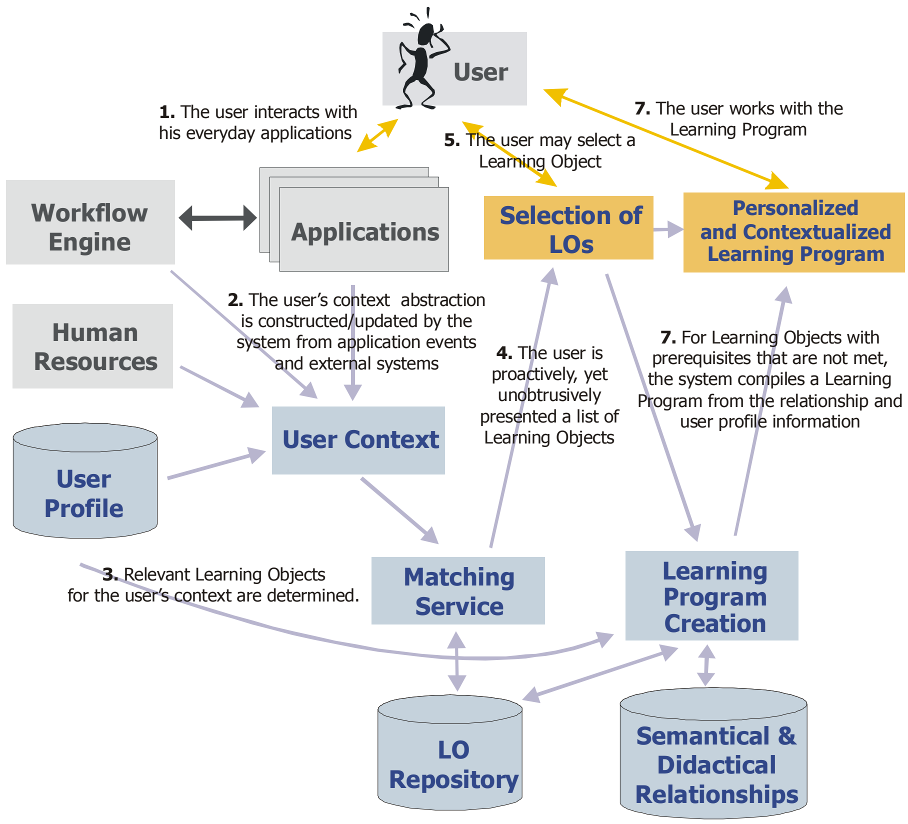

Figure 2: Overview of the delivery process

Apart from collecting the information, it is equally important to realize the nature of this collected information. It is always incomplete, mostly uncertain or inaccurate. And it becomes outdated at varying rates (e.g. birth date never gets outdated, but the current task does). The context management has to take these properties of user context information into account. Therefore, our context management in LIP is based on techniques for dealing with imperfect information (e.g. probabilistic models).

3.2 Matching strategy

The matching service tries to find relevant learning objects for a given user context. It computes a similarity measure between the current user context abstraction and the ontological metadata of each learning object and then can present a ranked list of relevant learning objects. This matching service has to take into account:

- learning history of the learner, i.e. learning objects already worked through and possible changes in successive versions
- current task (e.g. workflow activity)
- role of the learner (certain objects are only relevant for certain roles)
- long-term desired skills
- …

The algorithm for computing relevant learning objects can be sketched roughly as follows:

- **Determine the knowledge gap.** For each of the context dimension instances in the user's current context (i.e. for his role, for the department he works in, for the process and task he is occupied with), the required competencies are determined and compared to the user's competencies. The difference is the knowledge gap.
- **Retrieve learning objects delivering those competencies.** Since learning objects have always an objective associated with them, it is not difficult to retrieve appropriate learning objects. Additionally, certain preferences of the learner can be taken into consideration.

3.3 Learning Program Compilation

After the user selecting an object from the list of recommended learning objects, the system compiles a learning program for the user. In this step, pre-requisites and other didactical dependencies are taken into consideration (like available tests, exercises etc.). If a learning object provides the possibility of adaptation to individual learners (this could range from presenting a view that shows the changes to an updated learning objects up to complex learning objects that support different learning paths for different learning styles), this step also configures the learning objects. The user is then presented a structured collection of learning objects, which we call learning program.

This process can be decomposed into the following steps:

- **Transitive closure.** The first step is to determine if the user meets all prerequisites of the learning object. If not, additional learning objects have to be determined that deliver the required competencies. The step has to be repeated transitively until all prerequisites are met (or the system has no learning objects for that competency available). In this step, it is possible that redundant learning objects are included.
- **Topological sort of the learning objects.** The next step is compute a partial order on the set of learning objects determined in the previous step. This

order is computed from the prerequisites/objectives information, and from didactical recommendations.

- **Optimal learning programs.** By making use of this partial order, we can compute now learning programs. Usually, there will not only be a single learning program possible, so we have to select the "optimal" ones by employing user defined criteria (e.g. estimated time, preferred learning style) etc.

4 Conclusion and outlook

We have presented the general idea of modularizing and contextualizing learning objects in order to be able to deliver learning objects that are relevant to the learner's current situation. This enables e-learning to fit more into the environment it takes place in. Still some issues remain that are sketched in the following sections

4.1 Implementation in SMEs

One of the most critical issues for e-learning solutions especially in smaller companies are the costs for creating high quality learning content. Our approach primarily focuses on the delivery, but it can also help to address this issue in two ways:

- It is not expected that the companies have to build all their learning content from scratch. Instead, learning content can be bought from specialized training companies. Then the company can still make it fit to their employees by contextualizing it, i.e. adding company-specific metadata by relating it to job roles, projects, tasks, or simply by assigning it to specific employees. This way, learning content can be reused without being unanware of the specific environment.
- But this system can also be used to empower employees to document process-specific knowledge that is normally not made explicit. Thus, such an e-learning platform can be a simple entry (and an important part of) into knowledge management. Especially, the introduction of new employees in medium-skilled jobs can be improved a lot by having explicit competencies and explicit content trying to deliver these competencies in a structured way.

During the project, we will explore both issues further. On the one hand, we will illustrate a possible application service provider scenario in which not only the content is bought from specialized training companies, but also the system itself is operated by a specialized provider. On the other hand, we will investigate how to integrate this solution with knowledge management systems.

4.2 Evaluation

This approach is currently being implemented within the LIP project and will be evaluated at two organizations. This includes a first stage of scenario-based evaluation techniques that flows back into the development process and a second hands-on evaluation with the operational system.

4.3 Further work

As identified in the system architecture, the most crucial point is how the system acquires and manages information about the user's context. Here methods have to be developed that cope with the dynamics of this information and its imperfection. A first probabilistic model has already been developed and has to be put into practice.

Acknowledgements

We wish to thank the other partners of the LIP consortium and especially Christian Abeln and Brian Egan from CAS Software AG, Karlsruhe, and Piotr Jachowicz from Neurosoft Poland, for fruitful discussions on this topic. LIP has been co-funded by the European Union under contract IST-2001-32518 within the 5th Framework Programme of IST since September 2002.

References

[Abecker et a. 00] Andreas Abecker, Ansgar Bernardi, Knut Hinkelmann, Otto Kühn und Michael Sintek: „Context-Aware, Proactive Delivery of Task-Specific Information: The KnowMore Project". DFKI GmbH International Journal on Information Systems Frontiers (ISF) 2 (314); Special Issue on Knowledge Management and Organizational Memory. Kluwer 2000. S.139-162.

[Budzik, Hammond 00] Jay Budzik und Kristian J. Hammond: „User Interactions with Everyday Applications as Context for Just-in-time Information Access". In: Proceedings of Intelligent User Interfaces 2000. ACM Press. 2000. Seiten 44-51

[Elst et al. 01] Ludger van Elst, Andreas Abecker und Heiko Maus: „Exploiting User and Process Context for Knowledge Management Systems". German Research Center for Artificial Intelligence (DFKI). Workshop on User Modeling for Context-Aware Applications at the 8th International Conference on User Modeling, Sonthofen, Germany. July 13-16, 2001. Internet: [http://orgwis.gmd.de/~gros/um2001ws/papers/elst.pdf]

[Fischer, Ye 01] Gerhard Fischer und Yunwen Ye: „Exploiting Context to Make Delivered Information Relevant To Tasks and Users" Technischer Bericht, Center for Lifelong Learning and Design Department of Computer Science and Institute of Cognitive Science, University of Colorado, Boulder, CO, USA, 2001 Internet: [http://orgwis.gmd.de/~gross/um2001ws/papers/fischer.pdf]

[Maus 01] Heiko Maus: „Workflow Context as a Means for Intelligent Information Support". Modeling and Using Context. 3rd. International and Interdisciplinary Conference, CONTEXT. Dundee, Scotland. 2001. Internet: [http://www.dfki.uni-kl.de/frodo/dok/Context01.pdf]

[Maedche et al. 03] Alexander Maedche, Boris Motik, Ljiljana Stojanovic, Rudi Studer, Raphael Volz An Infrastructure for Searching, Reusing and Evolving Distributed Ontologies In: Proceedings of WWW 2003.

[Stojanovic et al. 01] L. Stojanovic, S. Staab and R. Studer : „eLearning based on the Semantic Web". WebNet2001 - World Conference on the WWW and Internet, Orlando, Florida, USA, Oktober 2001. [Internet: http://wim.fzi.de/wim/publications/entries/1011598970.pdf]