In: Claus Pahl (ed.): Architecture Solutions for E-Learning Systems, Idea-Group Publishing, 2007

IMPACT OF CONTEXT-AWARENESS ON THE ARCHITECTURE OF LEARNING SUPPORT SYSTEMS

Andreas Schmidt

FZI Research Center for Information Technologies

Information Process Engineering

Haid-und-Neu-Straße 10-14

76131 Karlsruhe

Germany

Fon +49 721 9654-732

Fax +49 721 9654-733

Andreas.Schmidt@fzi.de

IMPACT OF CONTEXT-AWARENESS ON THE ARCHITECTURE OF LEARNING SUPPORT SYSTEMS

Abstract

Recently, the situatedness of learning has come to the center of attention in both research and practice, also a result of the insight that traditional learning methods in the form of large decontextualized courses lead to inert knowledge; i.e., knowledge that can be reproduced, but not applied to real-world problem solving. In order to avoid the inertness, pedagogy tries to set up authentic learning settings, an approach increasingly shared in e-learning domain. If we consider professional training, it is the immediacy of purpose and context that makes it largely different to learning in schools or academic education. This immediacy has the benefit that we actually have an authentic context that we need to preserve. The majority of current e-learning approaches, however, ignores this context and provides decontextualized forms of learning as a multimedia copy of traditional presence seminars. We show how making learning solutions aware of the context actually affects their architecture and present a showcase solution in the form of the Learning in Process service-oriented architecture.

Keywords

User context, context-aware, adaptation, ontologies

INTRODUCTION

In the wake of constructivism dominating pedagogy research during the last years, the situatedness of learning has come to the center of attention, also a result of the insight that traditional learning methods in the form of large decontextualized courses lead to inert knowledge; i.e., knowledge that can be reproduced, but not applied to real-world problem solving (Bereiter & Scardamalia 1985, Renkel et al. 1996). In order to avoid the inertness, pedagogy tries to set up authentic learning settings, an approach increasingly shared in e-learning domain. If we consider professional training, it is the immediacy of purpose and context that makes it largely different to learning in schools or academic education. This immediacy has the benefit that we actually have an authentic context that we need to preserve. The majority of current e-learning approaches, however, ignores this context and provides decontextualized forms of learning as a multimedia copy of traditional presence seminars.

Context-aware system behaviour can foster learning processes in several areas:

- If we consider the delivery of appropriate learning content, we can support employees in embedding learning activities into their work processes. We can recommend fine-grained learning resources, and make the recommendations aware of derived learning needs *(what)*, but also aware of interruptibility and stress level *(when and how).*
- Within learning objects, we can adapt the instructional strategy to the learner's current situation, such as modifying the difficulty level or the playfulness in response to (a) personal characteristics, but also (2) whether it is the end of a long and hard day of meetings, after a period of boring paperwork, or early in the morning.

- Finally, we can also foster informal learning activities by bringing together learners that are dealing with the same topic areas or the same business process activities as soon as we know what others were doing recently.

In this chapter, we want to present a systematic service-oriented approach of extending current learning support services (which comprise learning management systems, learning content management systems, communication and collaboration services) with context-aware functionality. This approach covers all aspects of dealing with context information, i.e., context acquisition, context management, context augmentation, and context-aware adaptation of system behavior.

BACKGROUND

What is context?

Although it seems to have become common sense to acknowledge that "context" is important to consider for state-of-the-art system development in general and learning support in particular, there is no shared understanding of what "context" is. Bazire & Brézillon 2005 have analyzed the scientific literature of several fields in order to find out the commonalities and came up with a vague notion of a set of constraints that can influence the behavior of a system in a given task. The most generally accepted definition in the community of ubiquitous computing is given by (Dey 2001): "*Context is 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.*" This still does not define what is "the situation". In (Schmidt 2006) the situation of a user is defined as a relevant subset of the state of the world at a given point in time (including the respective knowledge of history and expectations for the future at that point in time).

These definitions leave the most important question open: what is actually relevant? From a theoretical point of view, this question cannot be answered exhaustively. From a practical point of view, we can approach this problem by considering the two aspects of context-awareness (Schmidt 2005a): (1) knowing about the user of the user and (2) adapting system behavior to this context. Context acquisition methods determine the supply side and context-aware (learning) support methods determine the demand side. Typical learning support methods (Schmidt 2005) are about retrieving or recommending resources; an important guiding concept for adaptation of such system behavior is the notion of “subjective relevance” (Swanson 1986). While traditional information retrieval assumes that relevance of a result with respect to a query can be objectively assessed (i.e., relevance is a function of the query and the resource), the notion of subjective relevance postulates that relevance also depends on the user (and her context).

Context and Learning Processes

Although the situatedness of learning has been investigated extensivelsy, the notion of an operationalized context with distinct features has only rarely been approached. Prior work is rather scattered among various communities, each with its only limited notion of context A brief summary of the most important approaches shall be summarized in the following (cp. Schmidt & Braun 2006):

– **Business-process-oriented knowledge management** (BPOKM, e.g. Abecker 2004) has realized the importance of the process context for context-aware delivery and storage of knowledge assets. Recently, the approach was further developed towards informal learning techniques, e.g. in (Farmer 2003). While it is true that business processes are an important element of the work context, they definitely are too narrow, although there are some approaches extending it like (Hädrich & Priebe 2005).

- **Macroadaptive e-learning approaches** like (Woelk & Agarwal 2002) or (Davis et al. 2005) mainly adapt to the learner in terms of delivery. They filter the learning content based on the learner's competencies and the knowledge requirements of the current position or business process context. While this is an important step into the direction of context-aware learning support, they only consider rather static elements of the context, which does not allow for deeper integration of working and learning processes. Interesting developments are in the direction of context-aware recommendations like in (Lemire et al. 2005), but they have still a notion of context too limited for holistic workplace learning support.
- **Microadaptive e-learning approaches** and adaptive hypermedia approaches are probably the area of research with the longest history and highest activity (Park & Lee 2002). They focus primarily on the learning object behavior itself and how to adapt it to the learner and her characteristics. The main problem of current adaptive e-learning approaches is that they do not consider learning in a work context, but rather set up artificial contexts in learning labs. They allow for a deep contextualization on the personal level, but neglect the organizational context completely.

As one of the first approaches to bring together the views of the different communities into a holistic ontology of context in a workplace setting was the ontology developed with the Learning in Process project (Schmidt 2005), which will be presented in the case study section.

Existing Architectural Approaches

Existing architectures for e-learning systems like the LTSA architecture (IEEE 2003), or the Learning Services Stack (Blackmon & Rehak 2003) do not consider context at all. It is not the case that these architectures do not allow for context awareness, but they do not provide clear separations (resulting later in clear interfaces) that are essential for manageable context aware behavior.

Although so far context-awareness of learning solutions has not been approached from an architectural point of view, there are two areas where we can identify relevant architectural considerations: context-aware applications in general and adaptive educational hypermedia systems as a very specialized form of context-aware learning support system.

For the area of context-aware applications, one of the most recent and most comprehensive architectural approaches is that of (Henricksen & Indulska 2006). They divide the functionality into six layers. The lower-most layer is the *context gathering layer* consisting of sensors and interpreters/aggregators for deriving more abstract context information from low-level sensor data. The next layer is the context reception layer that provides an interface between the gathering layer and the *context management layer*. This layer is responsible for storing context data in a context repository and keeping it consistent. The next layer adds the *query* functionality to the repository, providing expressive descriptive possibilities. Ontop of this layer is the *adaptation layer* for encapsulating the adaptation logic for the top-most layer *(application layer).*

The reference architecture for adaptive educational hypermedia systems from (Karampiperis, & Sampson 2005) divided the functionality into two layers: (1) the runtime layer and the (2) the storage layer. The storage layer divides the models needed into the user model, the adaptation model, the domain model and the media space. On the runtime layer, three components are identified: a presenter, a adaptation rule parser, and a behaviour tracker (for updating the user model).

CHALLENGES OF CONTEXT-AWARENESS AND ARCHITECTURAL PROBLEMS

At first sight, context-awareness simply translates into making existing learning management systems taking into account the characteristics of the context of the user, e.g., by showing only learning offers that are relevant to the current context. But this naïve view does not hold when investigating the problem more closely (and it does also show why commercial personalization falls short):

User context is hard to model. It seems so easy to speak about context as a kind of synonym for everything that was so far not considered, but modelling it (i.e., making it computationally accessible) is a challenge: how to identify features with which to describe it? How to distinguish features from irrelevant ones? How to find the right level of abstraction? Understanding context is by magnitudes harder than understanding a domain for which we model explicit interaction between a user and the system. Usually modelling context involves investigation of contextual influence factors to subjective relevance and other issues.

User context is hard to make use of. Even if we have a fair context schema, it remains still a challenge to actually operationalize context-awareness. Between a statement like "this should be considered when interacting with the user" and an algorithm that answers how to adapt system behaviour is a huge difference. It also involves a lot of trial and error so that flexibility to implement adaptivity is needed. But what is even more important is that context-awareness changes the paradigm of learning support. If we want to respond to contextual needs of a user, it means embedding it into the work environment, and embedding is only possible if we break up monolithic learning management systems into individual, loosely coupled services and applications. The system does no longer prestructure separate learning times, but rather provide functionality to initiate learning processes and to suggest learning opportunities: (reactive) learning management becomes (proactive) learning support.

User context is hard to acquire. Even if we know how to model and use context information, we still have to deal with the problem of acquiring it. For many contextual features there are no "sensors" and never will be. So we have to rely on heuristics and indirect methods that need to consider a wide range of sources like desktop activity, existing data in

ERP systems, communication history etc. And: we need to realize that the heuristics yield imperfect results: incomplete, uncertain, outdated and contradictory information.

How do these challenges affect the design of an architecture of learning support systems:

- **Specialized context middleware.** The sheer complexity of dealing with context information demands for factoring it out of other functional components: it is not just an add-on, but a task of its own. Especially the integration and aggregation of various context sources and the imperfection of the context information itself should be hidden from services concerned with learning.
- **Service-oriented Architecture.** The paradigm of learning support instead of learning management demands for a new type of flexibility. This can only be provided with a state-of-the-art services-oriented approaches.
- **Shared context manager.** While at first sight, the services require different aspects of the context (often at different abstraction levels), a deeper investigation reveals that there is sufficient overlap so that have synergies from a shared user context manager.
- **Separation of adaptation and context.** The naïve view may view a direct mapping of context features to adaptation parameters. But something like "preferred conciseness" (adaptation parameter) is not in the context (because it is way too application-specific), but rather a result of features like skill level, general learning characteristics etc. So we need a clear separation and a calculation of adaptation parameters.
- **Separation of pedagogical strategies and enabling infrastructure.** On a fine-grained level, there is no sound body of knowledge about how contextual factors affect learning processes and especially the appropriateness of guiding interventions.
- **Background domain knowledge.** Both for the aggregation of low-level context information and for the translation of context features into adaptation parameters, domain-specific background "knowledge" is needed

- **Asynchronous interaction patterns.** Context-aware applications do not only respond to actions of user, but also to context changes, e.g., for recommending proactively. But this means that we cannot solely rely on request-response interactions, but need a publish-subscribe pattern. But as we do not want every context change to be reflected at the user interface, we need a coordinating instance mediating context changes.

A REFERENCE ARCHITECTURE FOR CONTEXT-AWARE LEARNING SUPPORT

Based on the previous sections, a reference architecture for context-aware learning support systems can be derived, which is summarized in fig. 1.

The lower-most layer consists of **external information sources**. This could be sources for eliciting user context information, but also sources for providing learning material. These raw external sources are wrapped so that they provide context information in line with a system-wide context ontology and provide a well-defined behavior, either a push sources (data is materialized in the context manager) or as pull sources (queries on demand, need to support a declarative query language). On top of this layer is the **infrastructure layer** that provides the following basic services

- A *User context management service* stores collected or inferred user context information and provides a consistent view (Schmidt 2006) on it, abstracting from contradictions, uncertainty, and outdated information. This service needs to provide both a declarative query language for query-response interaction and a subscription facility for asynchronous notification.
- An *artifact repository* provides access to all types of artifacts that could play a role for learning processes. These artifacts could be learning objects, documents, or

communication artifacts. In many cases, this artifact repository will store just the metadata and a link to a physical resource.

- An *ontology service* provides open and declarative access to (a) a shared conceptual model forming the basis of the services' interaction and (b) background knowledge needed for adaptive behaviour. Via a centralized ontology service, a loosely-coupled architecture becomes possible without losing a high degree of semantic coherence.

On top of this infrastructure come the **learning support services**. These provide reusable functionality that is needed by applications that try to support learning processes. Typical services on this layer are: competency assessment, competence gap analysis, or on-demand learning program compilation.

These services are all reactive in the sense that they are invoked in the request-response paradigm. But context-awareness usually requires also proactive behaviour that responds to changes of the context. In order to encapsulate strategies for proactive learning support, a **learning coordination layer** is introduced. This layer subscribes to the user context management service, makes use of the learning support services and initiates activities in the two remaining layers. The strategies on this layer basically represent the pedagogical approaches which translate knowledge of the context of the user into interventions into learning processes.

The **adaptation layer** is responsible for (1) translating interventions by the learning coordination layer into application-specific *actions* and (2) user context information into *adaptation parameters* that accompany the actions. The main task of this layer is to provide context-aware interfaces to end user applications that can be context-aware, but usually are not. From an architectural point of view, it is desirable that service types are defined (like LMS, instant messaging application, etc.).

The top-most **(end user applications)** layer is represented by those application end users (i.e., learners) actually interact with. This includes classical learning runtime environments (like they form part of learning management solutions), but also communication and collaboration services for more informal forms of learning. Some of these applications will explicitly know about the user's context, but many of them do not. These are supposed to provide interfaces for adaptation. These interfaces are used by the underlying adaptation layer.

The architecture provides a flexible framework for various pedagogical approaches that make use of knowledge of the context of the learner:

- Highly automated approaches can recommend learning opportunities to the learner, e.g. suggesting learning objects designed to deliver certain competencies, or suggesting colleagues that can provide help in problem solving, based on organizational models of processes, tasks, roles and their respective competency requirements.
- On the micro-level, learning objects can adapt their presentation to context-dependent personal preferences, e.g., affecting the conciseness or playfulness of learning objects.
- Learner-empowering approaches (e.g., based on Personal Learning Environments) can be less intervening and can provide more relevant navigational options and can take into account personal goals and how they fit into the situation.

Which of these approaches is actually taken, depends on the strategies implemented in the learning coordination layer which can flexibly orchestrate both the learning services and the end user applications.

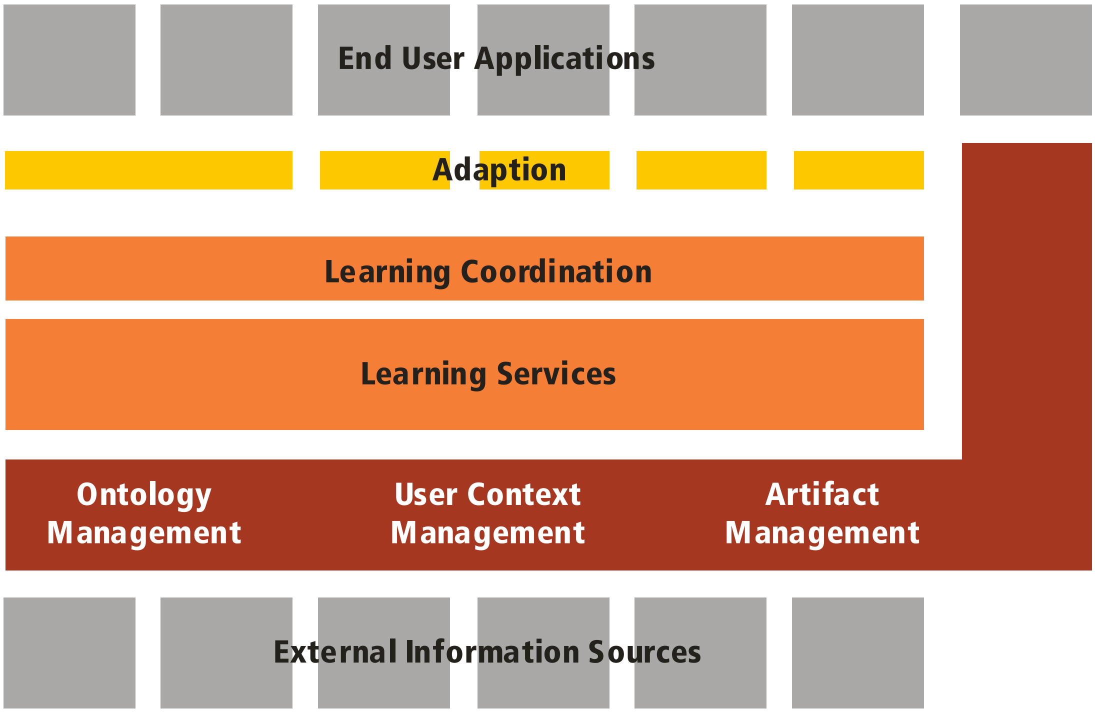

Figure 1: Reference Architecture for Context-Aware Learning Support Systems

In the spirit of other e-learning standards, it is clear that the roll-out of context-aware systems in the large at least a small set of context ontologies in order to be able to reuse (1) wrappers for context sources, (2) learning coordination strategies responding to context changes, and (3) developing reusable context-aware learning content. But this does not affect the architecture as such, which assumes that within the system there is a shared ontology. If this shared ontology is achieved through individual adaptation at each installation or already in the production phase, is a different topic.

CASE STUDY: LEARNING IN PROCESS

This reference architecture has been implemented in two demonstrators within the EU project *Learning in Process* (Schmidt 2005). Its primary goals have been the integration of working and learning on a process level and learning management, knowledge management, human capital management and collaboration solutions on a technical level. One of the early insights was that isolated and monolithic learning management systems are not suitable for integrating learning into personal and organizational activities. So the project concentrated on providing fine-grained learning on demand embedded into work processes based on a methodological framework (context-steered learning, Schmidt & Braun 2006).

Context-steered learning seems to be a natural transition from e-learning and knowledge management approaches. It is based on the assumption that there are small learning units that can be used on demand. Context-steered learning can be visualized as a process cycle, which appears as an on-demand ’detour’ of the working processes and can be broken down into the following system primitives (see fig. 2):

- *Initiate.* In the first phase, the system detects based on observations of the work context and background knowledge (which competencies are required in which context) if there is a learning opportunity. This functionality refers to the timing (when) and modality (how) of interventions.
- *Select.* Appropriate learning resources that help to satisfy the learner’s knowledge need and that fit to the learner requirements are selected. This could be learning objects, casual documents, but also colleagues or external “experts” for informal communication.
- *Deliver*. It may seem that recommending learning objects (or other documents) already imply that we have determined what to recommend. But this is only partially true. Certain resources cannot be understood by the learner because she does not meet the prerequisites. So it is often necessary to compile longer learning programs that incorporate the prerequisites.
- *Adapt.* This is the domain of classical micro adaptivity in e-learning. This incorporates the adaptation of presentation (e.g., adapting the level of detail to be presented, or the difficulty of exercise) and behaviour of (active) learning content (e.g., simulations that can reference cases from the current working environment).

– *Record.* One often neglected aspect in the business context of classical formal training are certificates that can be obtained after successfully attending training activities. As a replacement in more informal context where no certificates exist, electronic portfolios can take the role.

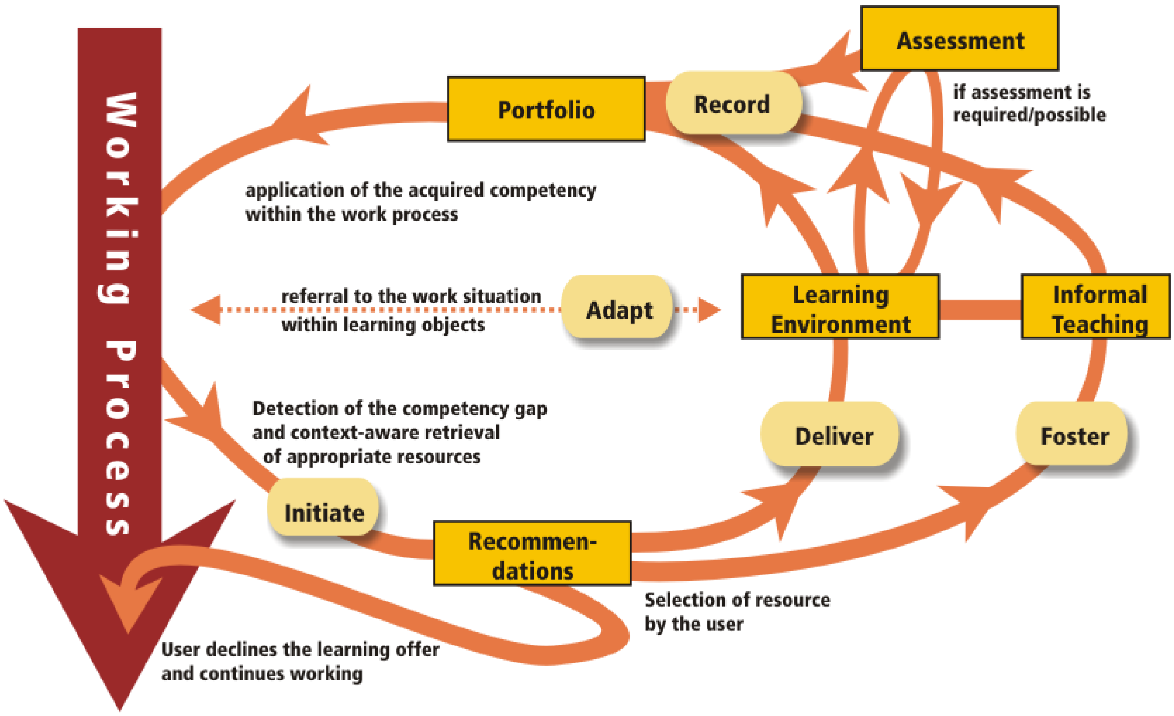

Figure 2: Context Steered Learning (based on Schmidt & Braun 2006)

After completion of this micro learning process, the learner returns to his working process and has the possibility to apply the newly acquired competencies—and to return to the learning process if it has turned out that learning was not as successful as expected. This could involve communicating to a recommended colleague.

The architecture of the developed system for supporting context-steered learning (see fig. 3) was basically organized into three layers: infrastructure, learning services and end-user applications, where the middle layer already had the separation into a proactive learning coordinator and reactive services. In the following section, we want to briefly present some of the key components.

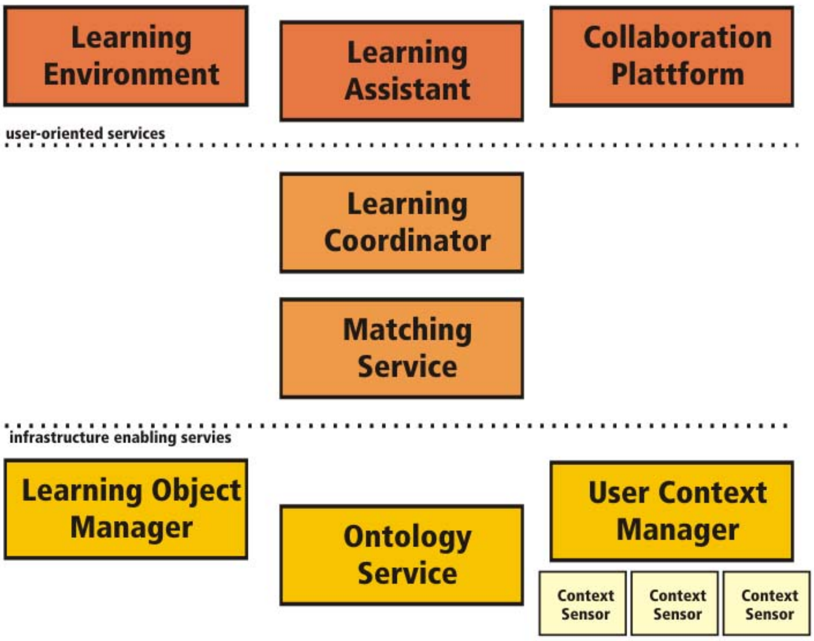

Figure 3: Architecture of LIP

User Context Manager

Identification of what context actually is one of the basic challenges. We need to operationalize the different situational factors that affect the appropriateness system behaviour (like recommending learning opportunities, adapting presentation etc.). Within LIP we have elaborated the following features (see also Fig. 4):

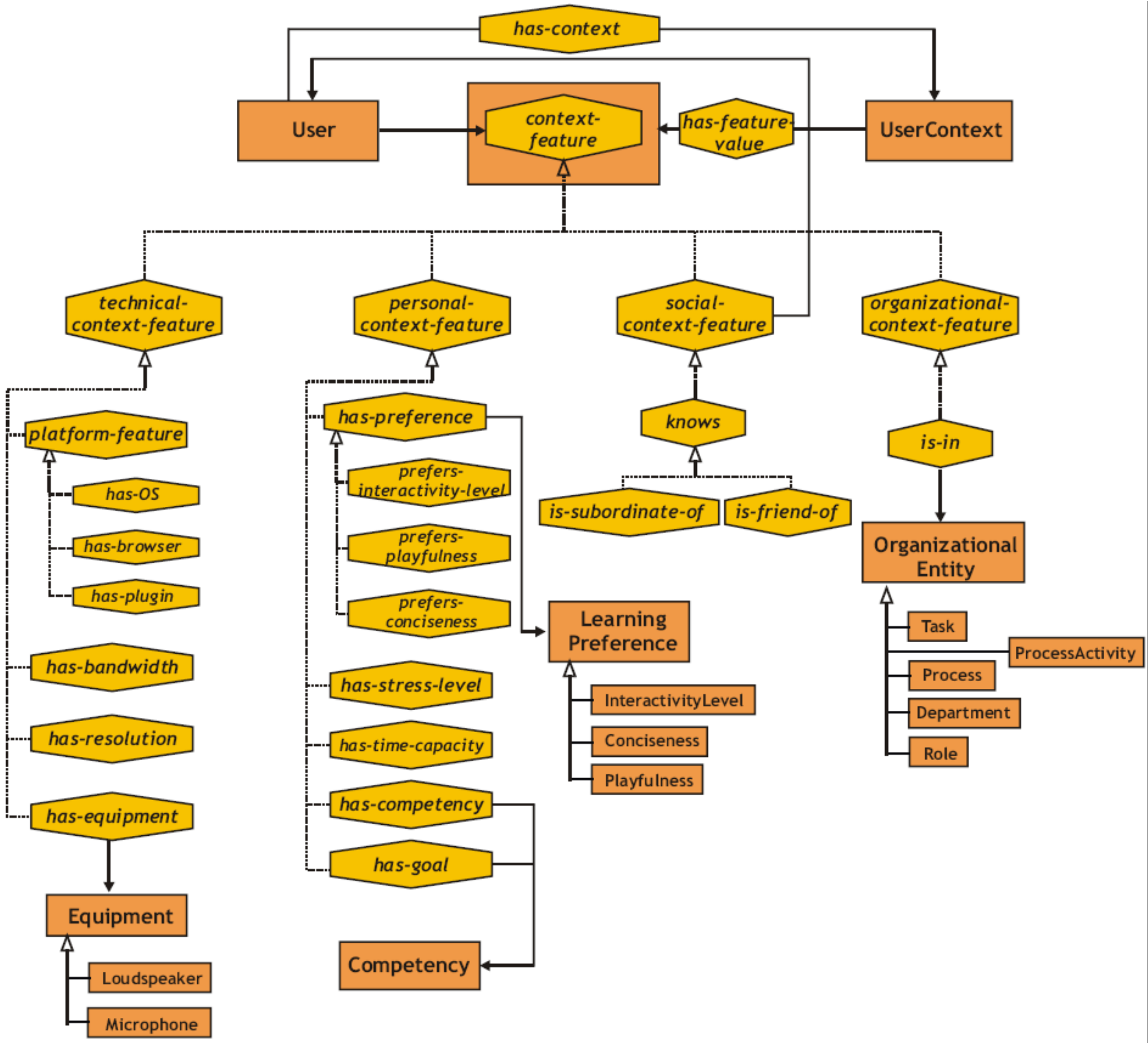

Figure 4: LIP Context Ontology

- *Personal*. This encompasses previously acquired knowledge or competencies, goals (divided into short-term and long-term), preferred interactivity level and semantic density (for learning content), preferred communication channel (synchronous/asynchronous, voice/written), and current time capacity/time pressure.
- *Social*. This refers to qualified relationship information towards other users, which especially affects the informal learning part by communicating with other learners.
- *Organizational*. This encompasses organizational unit, role(s), current business process (or process step) and current task (as an activity that cannot be easily mapped to a business process).

- *Technical.* This encompasses user agent (operating system, browser, plugins etc.), bandwidth, and available audio devices.

However, modelling (i.e., identifying the relevant context features) is not enough. The acquisition is the much harder part. A key enabler for that was an appropriate context management infrastructure that is geared towards the special requirements of high-level context information, which are the aspects of imperfection and dynamics. As it is typically not possible to acquire context information on a high abstraction level directly, the system has to use indirect methods with limited certainty and precision of its results. The combination of different methods (sometimes even one method on its own) yields contradictory results. These problems are aggravated by the dynamic nature of user context information where different elements of the context change at different pace. Here we have used a probabilistic representation, feature-specific aging mechanisms, and conflict resolution methods (described in more detail in Schmidt 2006).

As context sensors, we have relied on a wide range of different sources): browser plugins for Internet Explorer and Mozilla for browsing activities (and Windows explorer actions on the file system), a Microsoft Office plugin for information about active documents, Microsoft Outlook plugin for access to calendar and contacts. These low-level application events were aggregated into context changes using heuristics (which were often specific to the company environment), e.g., using information about document templates (which are good indicators in administrative environments), the location on the network drive, the structure of intranet applications, or simple keyword extraction techniques.

Matching Service

A Matching Service can compile personalized learning programs from the available learning material (*Learning Object Manager*), the user's current context (User Context Manager) and the context's knowledge requirements (provided by the *Ontology Service*). The matching procedure allows for compiling on demand personalized learning programs based on the current competency gap. This matching procedure can be divided into the following three parts:

- **Competency gap analysis.** In this analysis, the system retrieves the user's current context from the user context manager. The current knowledge gap is the set of current competency requirements (from the ontology service) minus the set of current competencies of the user. For this knowledge gap, the system can retrieve appropriate learning objects from the learning object manager. In order to fill the knowledge gap, we retrieve all learning objects that deliver one of the competencies in the knowledge gap.
- **Learning program compilation.** Usually a single learning object will not be enough to bridge the gap, because the gap is too big, and because learning objects themselves can have prerequisites that the user does not meet yet. Therefore, we need to provide the user with a complete learning program. This is accomplished by recursively adding learning objects for unsatisfied prerequisites and other didactical dependencies (which are part of learning object metadata) and pruning based on features in the user's context.
- **Preference-based ranking.** After compiling several possible learning programs,the system ranks the alternatives according to the user or organizational preferences (soft criteria). As a result of this process, the user can be presented with the ranked list, from which he can select the desired learning program.

End-User Applications

For the interaction with the user, LIP has realized or made use of three types of applications:

- The **Learning Assistant** represents the component that displays recommendations to the user and captures context changes from the user's interactions with her applications. This component typically resided on the user's machine, although some server-side processing is involved. Within LIP, we have implemented a tray application that can display a sidebar on demand, and an embedded learner assistant that forms part of an intranet application.

- Learning can be organized by the learner in the **Learning Environment**, which allows for finding, scheduling and executing learning programs. As sketched above, the possibility of simulation the application of newly learnt knowledge is a promising functionality. In order to enable learning management systems for such type of learning objects, LIP has extended the standardized SCORM API available to learning objects at execution time with direct access to context information. This is achieved through mapping the context features to the CMI data model of SCORM. This technically enables the creation of truly adaptive learning objects.
- A **Collaboration Platform** was "contextualized" with the help of this service by providing contextualized expert finder functionality, group formation and interaction spaces, where learners can themselves create "knowledge assets" which can be made available (e.g. by recommendation or in self-steered learning processes) to other learners based on the context in which they were created.

Evaluation results

The Learning in Process system was developed with a high degree of end-user involvement through formative evaluation with instruments like scenario-based evaluation techniques (cp. Cook et al. 2004). Involved as end users were two medium-sized companies in the IT industry; their tasks ranged from administrative up to programmer and consultant. The summative evaluation was based on a set of around 100 learning objects (and less pedagogically designed documents), several of them from external sources like Microsoft training material. The evaluation was carried out on site of the companies with around 50 employees in total with the final prototype system as described above.

Results of both the formative and summative evaluation have shown a high degree of acceptance for the context-steered learning method, although usability issues with GUI-related components turned out to be very critical. During the evaluation procedure, the corporate culture was discovered to be crucial for the success of such a learning method. Learning of employees within working processes must be highly appreciated and must be understood as a shared responsibility of both the individual and the organization, and the collected data must not be used for any other purpose than facilitating their learning. Otherwise, such a system will not be used in an appropriate way.

The architecture proved to be able to accommodate to different company IT environments, one of them mainly based on Microsoft technologies, the other one mainly on open source software. The service-oriented approach allowed for an easy adaptation.

FUTURE TRENDS

With the first experiences with context-awareness in learning support systems, it has become possible to develop architectural patterns embodying best practices in this area. Such a reference architecture should be complemented by shared reference ontology connecting the different actors and entities conceptually. Interfaces of services do not provide enough semantic glue to ensure smooth interoperation of different services in such a setting. Such a reference ontology must also provide the conceptual integration with other corporate systems like competence management, knowledge management, or business process management. A first step towards this has been presented in (Schmidt & Kunzmann 2006).

Another issue is to be considered are the implications of *e-learning 2.0*: the importance of content and learning material generated by learners themselves. Currently, the architecture reflects the separation of roles between content creators and learners, but if we move to more democratic system paradigms, we have to consider personal learning environments and semantic desktop environments as possible technical enablers. These bottom-up approaches will have a deep impact on the architecture of learning support systems. But as the notion of a personal learning environment is still emerging, it is too early to see a reference architecture for e-learning 2.0 developing.

CONCLUSION

Reference architectures are an important step towards rolling out new technologies on the market on a larger scale. With the described architecture, an important step has been made for context-aware learning support, but there is still a long way to go before context-awareness (apart from very simple adaptive behaviour) becomes a standard feature of learning environments. This can be traced back to the fact that dealing with context poses several hard challenges (Schmidt 2005a): how to model, how to acquire and how to make reasonably use of it. This is not primarily a technical issue, but also a methodical issue. In pedagogy, we have no model for contextualized computer-based learning support. But as soon as we can roll out first products, training experts can experiment with the new possibilities.

REFERENCES

Abecker, A. (2004). Business-Process Oriented Knowledge Management: Concepts, Methods and Tools, PhD thesis at the University of Karlsruhe

Bazire, M.; Brézillon, P. (2005). Understanding Context Before Using It. In: Modeling and Using Context: 5thInternational and Interdisciplinary Conference CONTEXT 2005, Paris, France, July 5-8, 2005, Springer, 35-54

Bereiter, C., Scardamalia, M. (1985). Cognitive Coping Strategies and the Problem of 'Inert Knowledge'. In Chipman, S., Seagal, J., Glaser, R., eds.: Thinking and Learning Skills. Volume 2. LEA, Hillside, NJ, USA

Blackmon,W., Rehak, D. (2003). Customized Learning: A Web Services Approach. In: World Conference on Educational Multimedia, Hypermedia and Telecommunications (EDMEDIA 03), 2003

Cook, J., Bradley, C., Franzolini, P. (2004). Gathering Users' Requirements Using the Artifact-Claims-Implication Technique. In: Proceedings of Ed-Media 2004, Lugano, Switzerland, June 21-26

Davis, J., Kay, J., Kummerfeld, B., Poon, J., Quigley, A., Saunders, G., Yacef, K.: Using workflow, user modeling and tutoring strategies for just-in-time document delivery. *Journal of Interactive Learning* 4 (2005) 131–148

Dey, A.K. (2001). Understanding and Using Context. *Personal and Ubiquitous Computing Journal* 1(5), 4–7.

Farmer, J. (2003). Ad hoc Coach System: Supporting Task-oriented Teaching and Learning under Time Pressure. In Rauterberg, M., Menozzi, M., Wesson, J. (eds.): Human-Computer Interaction INTERACT '03: IFIP TC13 International Conference on Human-Computer Interaction, September 1-5, 2003, Zurich, Switzerland.

Hädrich, T., Priebe, T. (2005). Supporting knowledge work with knowledge stance-oriented integrative portals. In: Proceedings of the Thirteenth European Conference on Information Systems

Henricksen, K., Indulska, J. (2006) Developing context-aware pervasive computing applications: Models and approach. *Journal of Pervasive and Mobile Computing* 2 (1), 37–64

IEEE (2003). IEEE 1484.1 IEEE Standard for Learning Technology Systems Architecture (LTSA)

Karampiperis, P., Sampson, O. (2005) Adaptive Learning Resources Sequencing in Educational Hypermedia Systems. *Educational Technology & Society* 8 (4), 128–147

Park, O.C., Lee, J. (2004). Adaptive instructional systems. In Jonassen, D.H., ed.: Handbook of research for educational communications and technology. 2nd edition edn. Lawrence Erlbaum, Mahwah, N.J., 651–684

Lemire, D., Boley, H., McGrath, S., Ball, M. (2005). Collaborative filtering and inference rules for context-aware learning object recommendation. *International Journal of Interactive Technology and Smart Education* 2

Renkl, A., Mandl, H., Gruber, H. (1996). Inert knowledge: Analyses and remedies. *Educational Psychologist* 31(2), 115-121.

Schmidt, A. (2005) Bridging the Gap Between E-Learning and Knowledge Management with Context-Aware Corporate Learning Solutions. In: Althoff et al. (eds.): Professional Knowledge Management. Third Biennial Conference, WM 2005, Kaiserlautern, Germany, April 2005. Revised Selected Papers, Lecture Notes in Artificial Intelligence (LNAI), volume 3782, Springer, 203-213

Schmidt A. (2005a) Potentials and Challenges of Context-Awareness for Learning Solutions. In: Proceedings of the 13th Annual Workshop of the SIG Adaptivity and User Modeling in Interactive Systems, Saarbrücken, October 2005

Schmidt, A. (2006) Ontology-based User Context Management: The Challenges of Dynamics and Imperfection. In: Proceedings of the International Conference on Ontologies, Databases and Applications of SEmantics (ODBASE 2006), On the Move Federated Conferences (OTM), Springer, Lecture Notes in Computer Science (LNCS), vol. 4275, 995-1011

Schmidt, A., Braun, S. (2006). Context-Aware Workplace Learning Support: Concepts, Experiences and Visions. In: Nejdl, W., Tochermann, K. (eds.): Innovative Approaches for Learning and Knowledge Sharing. Proceedings of the First European Conference on Technology-Enhanced Learning (EC-TEL 06), Heraklion, Crete, Lecture Notes in Computer Science (LNCS), vol. 4227, pp. 518-524

Schmidt. A., Kunzmann, C. (2006). Towards a Reference Ontology for Combining Competence Management and Technology-Enhanced Workplace Learning. In: Proceedings of OntoContent ’06, On the Move Federated Conferences (OTM), Springer, Lecture Notes in Computer Science (LNCS), vol. 4278, 1078-1087

Swanson, D.R. (1986). Subjective versus Objective Relevance in Bibliographic Retrieval Systems.: *Library Quarterly* 56(4), 389–398

Woelk, D., Agarwal, S. (2002). Integration of e-learning and knowledge management. In: World Conference on E-Learning in Corporate, Government, Health Institutions, and Higher Eduction. Volume 1, pp. 1035–1042