Lindstaedt, S. N., Scheir, P., Lokaiczyk, R., Kump, B., Beham, G., Pammer, V.
Knowledge Services for Work-integrated Learning
in: Proceedings of the European Conference on Technology Enhanced Learning (ECTEL) 2008, Maastricht, The Netherlands, September 16-19, pp 234-244

# Knowledge Services for Work-integrated Learning

Stefanie N. Lindstaedt[1, 2], Peter Scheir[1, 2], Robert Lokaiczyk[3], Barbara Kump[2], Günter Beham[2], Viktoria Pammer[1, 2]

[1] Know-Center Graz, Austria
[2] Graz University of Technology, Austria
[3] SAP Research, Germany

**Abstract.** In order to support work-integrated learning scenarios task- and competency-aware knowledge services are needed. In this paper we introduce three key knowledge services of the APOSDLE system and illustrate how they interact. The context determination daemon observes user interactions and infers the current work task of the user. The user profile service uses the identified work tasks to determine the competences of the user. And finally, the associative retrieval service utilizes both the current work task and the inferred competences to identify relevant (learning) content. All of these knowledge services improve through user feedback.

**Keywords:** technology enhanced learning, task detection, contextualized retrieval

## 1 Introduction

*Work-integrated learning* (Lindstaedt & Farmer, 2004) (Smith, 2003) happens very frequently (often without being noticed) during social interaction while knowledge workers collaboratively create or refine digital artefacts, communicate aspects of their work, or examine existing documents. The role of a knowledge worker, embodied in social interaction, is subject to continuous change: at one point in time, a knowledge worker acts as a learner, at another point in time she acts as an expert (teacher), and then again she is simply getting her work done – all depending on her expertise with regard to the subject matter at hand (Lave & Wenger, 1991).

The design of computational support for work-integrated learning is at the core of the EU-funded integrated project APOSDLE (Advanced Process-Oriented Self-Directed Learning Environments, www.aposdle.org). The APOSDLE approach (Lindstaedt et al., 2007) is to support learning and teaching episodes tightly integrated into the work processes by taking into account the work context, such as the task at hand, and the prior knowledge of the knowledge worker. Workers are provided with knowledge artefacts relevant to their work context, thus raising their own awareness of learning situations, content, and people that may be useful for learning at that point in time. This context-aware knowledge delivery takes place within the usual computational work environment of the user, raising her awareness about relevant resources without having to switch to dedicated learning or knowledge management systems.

One major challenge within the project is to not rely on specifically created (e)Learning content, but to reuse existing (organizational) content which was not necessarily created with teaching in mind. We tap into all the resources of an organizational memory which might encompass project reports, studies, notes, intermediate results, plans, graphics, etc. as well as dedicated learning resources (if available) such as course descriptions, handouts and (e)Learning modules. The challenge we are addressing is: How can we make this confusing mix of information accessible to the knowledge worker in a way that she can advance her competencies with it? (Lindstaedt & Mayer, 2006).

While we have addressed APOSDLE's support for the role of the expert in (Lokaiczyk et al., 2007) and the support for the role of the learner in (Bonestroo et al., 2007) in the following we specifically focus on the support for the role of the worker, specifically the work context-aware identification of relevant documents and people.

Within the field of technology enhanced learning (TEL) user context awareness has been extensively studied within mobile learning applications. In these applications user context is determined to a large extent by the physical location of the user (Naismith et al., 2004). However, there are many more learning situations which can benefit from contextualized support. Examples of such situations include but are not limited to situated learning (Lave & Wenger, 1991) within the context of communities of practice, self-directed learning and informal learning (Eraut, 2004). Within the case of work-integrated learning the relevant user context is constituted specifically by the current work task the user is attempting to complete.

User context determination plays a crucial role in the overall approach. In order to be able to provide the user with information, learning material and links to people relevant to her task at hand, the system needs to identify the work task reliably. The goal is to identify a user's current work task based on the user's interactions with her work environment (key strokes, mouse movements, applications used, etc.) and the metadata and content of the resources accessed (mail messages, documents, links to people, etc.). The identified task then updates the user profile and causes a number of activities to be started pro-actively: re-computation of a user's learning goal and learning goal histories, search of knowledge artefacts relevant to the learning goal, search of people relevant to the learning goal, and the dynamic creation of learning events. The results of these searches are displayed in the form of resource and people lists unobtrusively to the user.

In this contribution we give a detailed account on the technological approaches involved in completing this complex context-aware retrieval process from context identification to information delivery. Section 2 gives an overview of this process and introduces the three key knowledge services which are combined to implement it. In Section 3 we shortly give an insight into the role of semantic models within the system. The introduced key knowledge services are then detailed as follows: Section 4 elaborates on how tasks are detected, Section 5 details the computation of learning goal gaps, and in Section 6 the retrieval approach itself is described. We close with an outlook on evaluations currently under way and some ideas for future work.

## 2 Key APOSDLE Knowledge Services

Figure 1 below illustrates the information flow through the system and the key knowledge services provided by APOSDLE. We distinguish two alternative paths: the ‘Learning Tool Path’ (steps 6a and 7a in Figure 1) uses the available user information to dynamically create *learning events* (for more details on this path please refer to (Bonestroo et al., 2007)). The ‘Related Resources Path’ (step 6b of Figure 1) uses associative retrieval to find knowledge artefacts that are related to the knowledge worker’s current situation. This contribution explains in detail how this related resource path is supported.

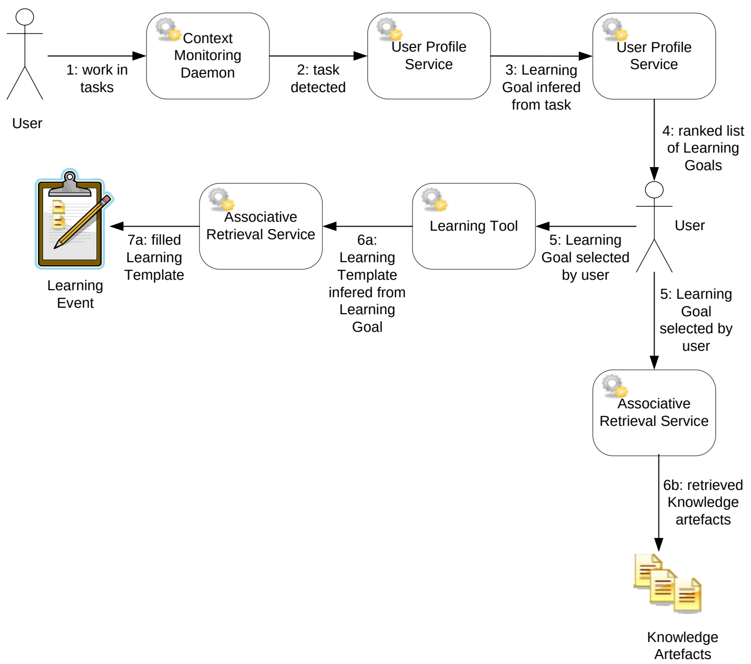


**Figure 1:** Information flow and key knowledge services within APOSDLE.

Based on low level system events the *Context Monitoring Daemon* (CMD, see Section 4) detects the current task a knowledge worker operates in (steps 1 and 2 in Figure 1). The CMD is a background service installed on the client computer. The CMD gathers low-level system events and maps them to tasks from the task model using machine learning algorithms. The goal of the CMD is to automatically identify the current user’s work task.

The detected current task of the user is then stored in the User Profile of the user which is accessed using the *User Profile Service* (UPS) (steps 3 and 4 in Figure 1). The UPS is part of the APOSDLE Platform running on the server side. It serves as a repository for user-related information and as an engine enabling the system to infer information about the user. The UPS utilizes a history-based user profile representation where activities of users are stored together with a timestamp. Based on this history of user activities the UPS infers information about the user as described in more detail in Section 5. The user is able to overrule the *learning goal* suggested by the UPS (step 5 in Figure 1).

Based on the selected learning goal a search is triggered engaging the *Associative Retrieval Service* (ARS, see section 6) (step 6b in Figure 1). The ARS is used for context-based retrieval of resources for work-integrated learning. It incorporates semantic similarity between concepts in the domain model, content based similarity between knowledge artefacts and semantic annotations of knowledge artefacts.

## 3 The Role of Semantic Models within APODLE

This section explains the three model structures and their relationships which provide the basis for reasoning within APOSDLE:

- Domain model – provides a representation of the learning domain in OWL (Web Ontology Language) format
- Task model – provides a representation of the work tasks to be supported in YAWL (Yet Another Workflow Language) format
- Learning goal model – provides a mapping between domain concepts, tasks and general learning goal types in OWL (Web Ontology Language) format

### Domain Model

The objective of the domain model is to provide a semantic description of the learn domain of an APOSDLE environment. The domain is described in terms of concepts, relations, and objects that are relevant for this domain. Technically speaking the domain model is an ontology that defines a set of concepts which are relevant for the domain. These concepts are also used for semantic annotation of documents (or parts thereof). In the following we will refer to the combination of a document (or part thereof) together with one or more domain concepts as a knowledge artefact. This annotation process can either happen automatically (using text based classification algorithms) or manually. These semantic annotations are used later within the Associative Network to retrieve relevant knowledge artefacts.

### Task Model

The objective of the task model is to provide a formal description of the tasks the knowledge worker can perform in a particular domain. The YAWL workflow system (van der Aalst & ter Hofstede, 2005) is used as conceptual basis for the task modelling. This formal description is used in various ways within APOSDLE. One aspect is the task prediction, which needs a set of predefined tasks. Another important aspect is the dependant task-competence mapping forming the *learning goal model* (below).

### Learning Goal Model

Learning goals constitute a mapping between tasks and domain concepts in order to realize an adaptive system. A learning goal describes knowledge and skills needed to perform a task. It is defined as a discrete element of a cognitive activity (learning goal type) connected with a domain model element. The formalisms employed are based on Competence-based Knowledge Space Theory (Korossy, 1997) and they provide several advantages for APOSDLE. One important advantage is that it allows the computation of learning goals through a learning need analysis by comparing knowledge needed to execute a task and the knowledge state of the user. Another one is the possibility to infer a user's learning history by examining the work task she has engaged with in the past (task-based learning history). A final advantage of utilizing Knowledge Space Theory within APOSDLE is that the mappings afford the computation of prerequisite relationships between learning goals. This allows APOSDLE to identify learning goals which should be mastered by the user on the way to reaching a higher level learning goal.

### APOSDLE Knowledge Base

The APOSDLE Knowledge Base (AKB) contains the three different models briefly discussed above, and the meta-model schema interrelating them. Within this AKB each model is stored in its original format (OWL, YAWL) and also within the meta-model schema. The meta-model schema is an ontology represented in OWL. The advantage of keeping the models in both representations is that within the original representation (for example, YAWL for tasks) reasoning about processes is possible while within the OWL representation reasoning about the overall model is supported.

## 4 Task Detection - from low level user events to work task

Our approach to context awareness in based on a common architecture by Baldauf et al. (2007) which includes the use of agents in three layers separating the detection of context, reasoning and actions based on context.

Layer 1, the monitoring of user events, is implemented using the CMD. This knowledge service observes a set of system sensors to gather operating system events, which reflect the user’s interaction with the computer desktop during the working process. These sensors include e.g. information about the foreground and background applications, the user’s input to these applications, the files opened during the work process and the textual features of the desktop, such as window titles and website content. A taxonomy of the implemented sensors is shown in Figure 2. These system level events, which are collected in an unobtrusive way, are then logged with a time-stamp and correlated to the current work task of the user.

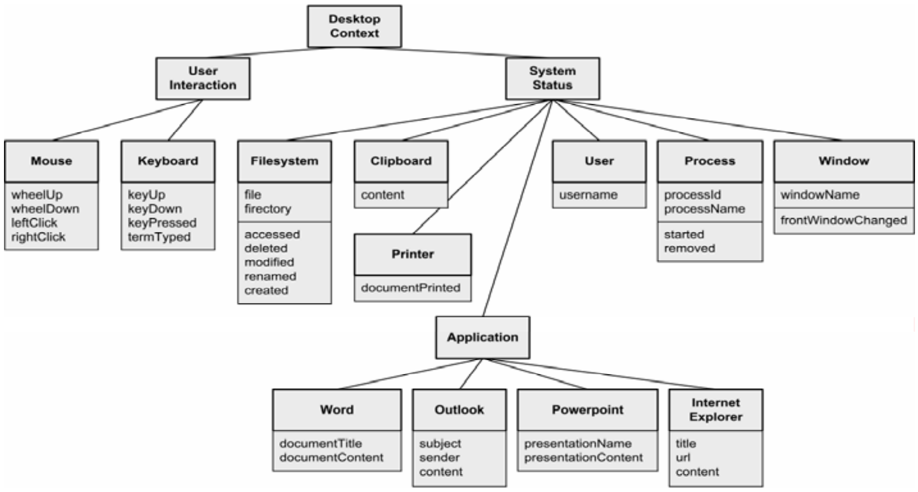


**Figure 2:** Taxonomy of work context sensors.

Layer 2, the context reasoning layer, is realized as a machine learning classification. During the training phase task executions by a number of different users are captured and labelled. These captured execution logs are then utilized to train a classifier, which is able to distinguish between the different tasks assuming that they significantly differ in their execution log (see Figure 3). This assumption depends of course on the definition of the work process and the task granularity but has a plausible foundation. Knowledge workers often use different applications or files for different process steps in the working process. Consequently, the usage of a particular file, application or term during the work process can be exploited as an indication for the execution of a particular work task from the task model.

During run-time the CMD tries to automatically classify the work task of the user. A prediction is made, if the following conditions apply:

1. the user changes its role from worker to learner (switches from a Non-APOSDLE application to APOSDLE)
2. the detected task is included in the adjusted work process
3. the self-assessment of the classification is above a defined threshold
4. the predicted task is not already manually set by the user
5. the time span to the last prediction is above a certain threshold

The user will be presented then the currently detected task. Here the user can decide how to proceed. If the user chooses ‘yes’ or ‘no’ the internal training process will be enforced with this decision. If the user chooses ‘set always’ she fully trusts the automatic task detection and this task will be always automatically set. This information is fed back to the Automatic Task Detection module.

Beside the automatic task detection the user still has the ability to manually set her current work task and optionally a sub-task as a fallback or in case she wants to learn independently from the current work task.

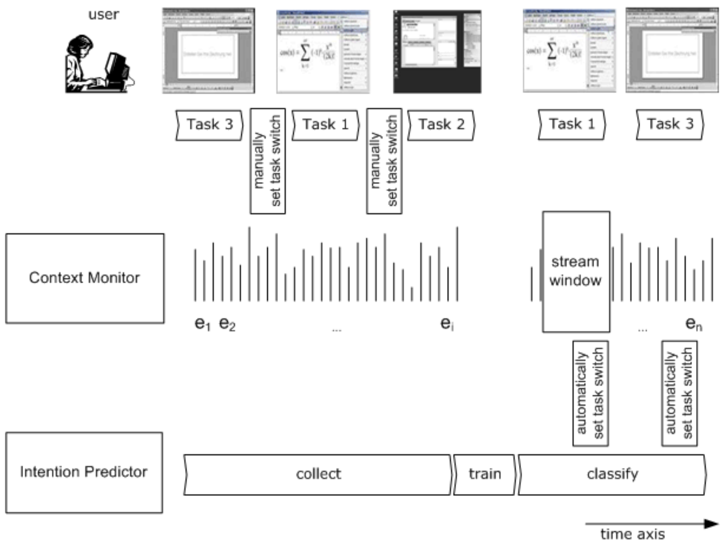


**Figure 3**: Task detection as classification challenge.

The whole approach already shows a good precision in an a posteriori analysis, but a low recall. Therefore, we decided for the current version to rely on the one hand on the a posteriori prediction model and on the other hand to incrementally extend this model with new training instances produced by the user during her APOSDLE usage. This makes the task prediction an online learner with a self-improving classification model.

The idea could be further extended by taking into account other implicit or explicit user feedback mechanisms, which were taken care of at design time but are still in their infancy. They will be taken into account for the design of the third prototype. In addition, an extensive real-world user study is necessary to show the real benefit of automated task prediction in terms of performance or quality increase during the work process.

## 5 Competency Gap Calculation - from task to learning goals

Once a task is set in the APOSDLE sidebar, the inference mechanisms of the User Profile Service (UPS) provide a ranked list of learning goals. By default, the highest ranking learning goal is used for further computations. However, the list of learning goals is displayed in the sidebar, where the user has the possibility to over-rule the ranking, and to select a lower ranking learning goal that she wants to tackle. Once a learning goal is set, the UPS performs another inference, namely the computation of experts with respect to a given learning goal. As for learning goals, a ranked list of experts is computed by the UPS.

The calculations of the UPS are based upon on the models for tasks, learning goals and a mapping of tasks and learning goals, which assigns to each task all learning goals that are required for performing the task. A learning goal is formally represented by a pair composed of a domain model element and a learning goal type. The task-learning goal mapping as been defined by Knowledge Engineers is stored in the Structure Repository. Algorithms for realizing the inference task (as recommending experts, generating a ranked list of learning goals, etc.) are all implemented within the User Profile Service. Note, that especially the algorithms for computing the learning need of a user and detecting experts strongly rely on a proper and clear concept of what a learning need or an expert actually are. Briefly they can be written down informally as follows: A 'learning need' is the discrepancy between the learning goals that are needed by a user to execute a task at hand, and the learning goals that a user has 'demonstrated' in the past, according to her user profile.

Technically, the learning need is a list of learning goals. The UPS applies a ranking algorithm for sorting the learning goals according to theoretical assumptions, from the 'most urgent' to the 'least urgent' for the user to acquire. This ranked list of learning goals is presented to the user. Within the conceptualization of the second APOSDLE prototype, an expert is a person who has 'demonstrated' a learning goal more often than the learner herself in a task execution.

The inferences in the UPS are all based on the task-learning goal mapping. In any situation, the context of the user is determined by the task at hand, and by the task history of a user, i.e. by all tasks that the user has executed in the past. Each time a user executes a task, all learning goals that are assigned to the task are added to the learning history of the user. This means, the learning history of a user is inferred from her task history by counting the number of how many times the user has "applied" a learning goal in a task execution. The computations of the learning need, the ranking of learning goals, and the ranking of experts that are suggested to the user in a concrete situation are based on the learning history of the users. The algorithm for ranking the learning goals takes into account the basic assumptions of the Competence-based Knowledge Space Theory (Korossy, 1997), and extension of Doingon & Falmagne's Knowledge Space Theory (Doignon & Falmagne, 1999). Moreover, the ranked list of learning goals has the following characteristics: Learning goals that never have been applied, or that have been applied less frequently are ranked higher than learning goals that have been applied more frequently. Learning goals that are "more important" in the learning domain, i.e. learning goals that are assigned to more tasks than others, are ranked higher.

In addition to the aforementioned inference functionality the UPS also provides a partial representation of the user's 'current context' to the Associative Retrieval Service to be used for context-based retrieval. The current context consists of domain model elements, which have either been decided by the UPS to be highly relevant for the user's current work task and learning need, or which have actively been selected by the user. To provide the Associative Retrieval Service with relevant domain model elements, the UPS generates a ranked list of learning goals. The sidebar displays them to a user who then can select one learning goal out of this list. All further inferences are based on this selection. Thus the most important domain model for a user's current context is the domain model connected to the selected learning goal. All other domain models connected to learning goals also presented in the sidebar will be provided as additional context information allowing the Associative Retrieval Service to broaden its query when needed.

## 6 Retrieval - from concepts to knowledge artefacts

The Associative Retrieval Service (ARS) creates a network representation, based on textual similarities, of a collection of knowledge artefacts for retrieval. This well explored approach is enhanced with techniques from knowledge representation and reasoning by analogy, resulting in a network model for finding similarities in a collection of knowledge artefacts by both, content based and knowledge based similarities. Within the APOSDLE project this network representation is referred to as the *Associative Network* (see Figure 5). In this Associative Network every concept from the domain model and every knowledge artefact present in the system are represented by a node. These nodes are associated by means of their similarly and by the semantic annotations of knowledge artefacts with concepts. The use of a network representation as underlying structure for the retrieval process allows for both, exact search for filling learning templates (step 6a in Figure 1) and associative search for resource retrieval (step 6b in Figure 1).

For processing the information in the network representation we employ a processing technique called spreading activation. Spreading activation originates from cognitive psychology (cf. (Anderson, 1983)) where it serves as mechanism for explaining how knowledge is represented and processed in the human brain. The human mind is modelled as a network of nodes, which represent concepts and are connected by edges. Starting from a set of initially activated nodes in the net, the activation spreads over the network (Sharifian & Samani, 1997). During search, energy flows from a set of initially activated information items over the edges to their neighbours. The information items with the highest level of energy are seen to be the most similar to the set of nodes activated initially. A detailed introduction to spreading activation in information retrieval can be found in (Crestani, 1997). A description of our studies on the topic can be found in (Scheir & Lindstaedt, 2006), (Scheir et al., 2007a), (Scheir et al., 2007b).

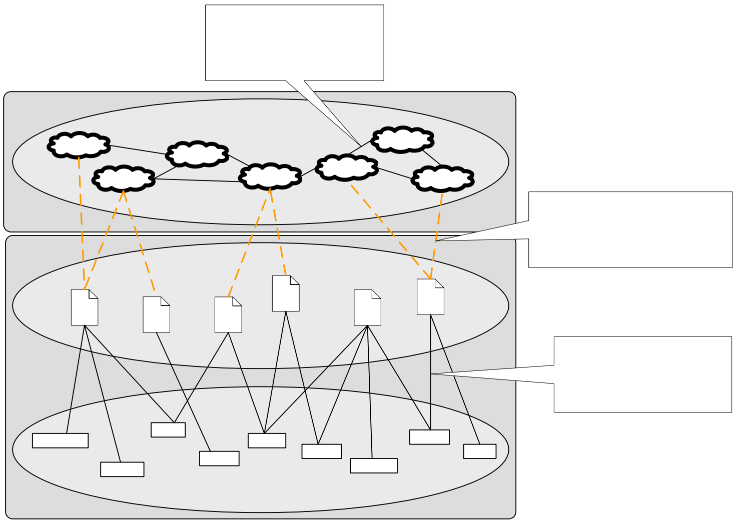


**Figure 5:** Semantic and content based similarity within the associative network. Concept layer and document/term layers are connected by semantic annotations.

## 7 Conclusion and Outlook

The second APOSDLE prototype was completed in Spring 2008. Currently the installations of the four domain specific environments at our four application partners are under way. EADS (Paris, France) is employing APOSDLE for the support of their software based flight simulation department. The Darmstadt Chamber of Commerce and Industry (Germany) applies it to the support of the REACH guidelines for chemical industries. ComNetMedia (Dortmund, Germany) aims at introducing new requirements engineering processes into their organization. And ISN (Graz, Austria) will be employing the system for the support of their consulting processes in innovation management. In each organization the end users receive training for the use of the APOSDLE system.

The evaluation will be conducted in two phases: First, a workplace evaluation takes place in which we observe and interview users of the application partners during system use. Secondly, the individual key knowledge services (as described in this paper and a few additional ones) will be evaluated in detail in controlled environments.

This paper introduced three knowledge services which were implemented within the second prototype of the APOSDLE environment. These services aim at (1) identifying a user's work context (task), (2) infer the competences she has displayed during the use of the system, and (3) use both the task as well as the competences to identify content within the organizational memory of the organization which could help advance the competences of the user. Together they build a complex retrieval process to support work-integrated learning. Evaluations are currently under way and first results can be reported at the conference.

**Acknowledgments.** This work has been partially funded under grant 027023 in the IST work programme of the European Community. The Know-Center is funded within the Austrian COMET Program - Competence Centers for Excellent Technologies - under the auspices of the Austrian Ministry of Transport, Innovation and Technology, the Austrian Ministry of Economics and Labor and by the State of Styria.

## References

Anderson, J. R. (1983). A spreading activation theory of memory. Journal of Verbal Leaning and Verbal Behaviour, *22*, 261--295.

Baldauf, M. & Dustdar, S. (2004). A Survey on Context-Aware Systems. TUV-1841-2004-24, Technical University of Vienna.

Bonestroo, W., Ley, T., Kump, B. & Lindstaedt, S.N. (2007). Learn@Work: Competency Advancement with Learning Templates. In: Martin Memmel, Eric Ras, Martin Wolpers, and Frans Van Assche (Eds.), Proceedings of the 3rd Workshop on Learner-Oriented Knowledge Management, 9-16, RWTH, Aachen.

Cristani, M. & Cuel, R. (2005). A survey on ontology creation methodologies. International Journal Semantic Web Information Systems, 1, 2, 49–69.

Doignon, J. & Falmagne, J. (1999). Knowledge Spaces, Springer, Heidelberg.

Eraut, M. (2004). Informal learning in the Workplace. In: Studies in Continuing Education, 26, 2 (2004) 247-243.

Korossy, K. (1997). Extending the theory of knowledge spaces: A competence-performance approach, Zeitschrift für Psychologie, Vol. 205, pp. 53-82.

Lave, J. & Wenger, E. (1991). Situated learning. Legitimate peripheral participation. Cambridge: Cambridge University Press.

Lindstaedt, S. N., Ley, T., Mayer, H. (2007). APOSDLE - New Ways to Work, Learn and Collaborate, in Proceedings of the 4th Conference on Professional Knowledge Management WM2007, 28. - 30. März 2007, Potsdam, Germany, 381-382, GITO-Verlag, Berlin.

Lindstaedt, S.N. & Mayer, H. (2006). A Storyboard of the APOSDLE Vision, in W. Nejdl and K. Tochtermann (Eds.), Innovative Approaches for Learning and Knowledge Sharing (LNCS 4227), 628-633, Springer, Berlin.

Lindstaedt, S.N. & Farmer, J. (2004). Kooperatives Lernen in Organisationen, in CSCL-Kompendium - Lehr- und Handbuch für das computerunterstützte kooperative Lernen Oldenbourg Wissenschaftsverlag, München, Germany.

Lokaiczyk, R., Godehardt, E., Faatz, A., Görtz, M., Kienle, K., Wessner, M. & Ulbrich, A. (2007). Exploiting Context Information for Identification of Relevant Experts in Collaborative Workplace-Embedded E-Learning Environments. EC-TEL 2007: 217-231.

Naismith, L., Lonsdale, P., Vavoula, G., & Sharples, M. (2004). Literature Review in Mobile Technologies and Learning (Literature Review No. 11): University of Birmingham.

Scheir, P. & Lindstaedt, S. N. (2006). A network model approach to document retrieval taking into account domain knowledge. In: Martin Schaaf & Klaus-Dieter Althoff, (ed.). *LWA 2006. Lernen - Wissensentdeckung - Adaptivitдt, 9.-11.10.2006 in Hildesheim*. Universitдt Hildesheim, , pp. 154-158.

Scheir, P., Pammer, V. & Lindstaedt, S. N. (2007). Information Retrieval on the Semantic Web - Does it exist? *In 'LWA 2007, Lernen - Wissensentdeckung - Adaptivitðt, 24.-26.9. 2007* in Halle/Saale (accepted for publication)'.

Scheir, P., Ghidini, C., Lindstaedt, S.N. (2007). Improving Search on the Semantic Desktop using Associative Retrieval Techniques, Proceedings of I-MEDIA 2007 and I-SEMANTICS 2007, Graz, Austria, September 5-7, 2007, 221-228

Sharifian, F. & Samani, R. (1997), Hierarchical spreading of activation, in Farzad Sharifian, ed., Proc. of the Conference on Language, Cognition, and Interpretation, IAU Press, , pp. 1--10.

Smith, P. J. (2003). Workplace Learning and Flexible Delivery. In: Review of Educational Research, 73, 1 (2003) 53-88.

Van der Aalst, W.M.P. & ter Hofstede, A.H.M. (2005). YAWL: Yet another workflow language. Information Systems, 30, 4, 245-275.