Getting to Know Your User – Unobtrusive User Model Maintenance within Work-Integrated Learning Environments

Stefanie N. Lindstaedt[1, 2], Günter Beham[1, 2], Barbara Kump[1, 2] and Tobias Ley[2, 3]

[1]Knowledge Management Institute, TU Graz
Inffeldgasse 21a, 8010 Graz
{slind, gbeham, bkump}@tugraz.at

[2]Know Center
Inffeldgasse 21a, 8010 Graz
{slind, gbeham, tley}@know-center.at

[3]Cognitive Science Section, University of Graz
Universitätsplatz 2, 8010 Graz, Austria
tobias.ley@uni-graz.at

Abstract. Work-integrated learning (WIL) poses unique challenges for user model design: on the one hand users' knowledge levels need to be determined based on their work activities – testing is not a viable option; on the other hand users do interact with a multitude of different work applications – there is no central learning system. This contribution introduces a user model and corresponding services (based on SOA) geared to enable unobtrusive adaptability within WIL environments. Our hybrid user model services interpret usage data in the context of enterprise models (semantic approaches) and utilize heuristics (scruffy approaches) in order to determine knowledge levels, identify subject matter experts, etc. We give an overview of different types of user model services (logging, production, inference, control), provide a reference implementation within the APOSDLE project, and discuss early evaluation results.

Keywords: user model, service-oriented architecture, work-integrated learning, adaptivity

1 Work-Integrated Learning

The goal of most approaches to support *work-integrated learning* (WIL) is on assisting knowledge workers in advancing their knowledge and skills directly within their 'real' work tasks instead of in dedicated (artificial) learning situations. Work-integrated learning is focusing on seamlessly integrating working and learning [1]. It is relatively brief and unstructured (in terms of learning objectives, learning time, or learning support), and its main aim mostly is to enhance task performance. From the learner's perspective, WIL is spontaneous and often unintentional. Learning in this case is a by-product of the time spent at the workplace. This especially requires that WIL support is embedded within the 'real' computational work environment of the user and not provided in dedicated eLearning systems. Moreover, it is essential that WIL support utilizes 'real' content from the organizational memory (such as project reports, calculations, presentations) and repurpose it for learning. WIL is a learning concept developed for supporting continuous competence development at the workplace. It assumes that its users have basic knowledge of the learning domain in question and the ability to guide their own learning processes [2]. The WIL concept has been applied to the domains of requirements engineering, software simulations, innovation management, and intellectual property rights management [3].

Consider a scenario within the learning domain of requirements engineering. Laura is a software engineer busy creating *human activity models* (work task) based on *user interviews* she has performed in the past (previous work task). She uses a variety of different unrelated applications such as MS Word, Viso and Requisite Pro to accomplish the task (computational work environment). Even though she is unsure on how to best approach human activity modelling, she is not interested in switching to a dedicated eLearning system. Instead, she relies on different WIL services which sit silently in the background and offer their help only when needed. In her current situation, a WIL service identifies her missing knowledge in human activity modelling and recommends learning content about human activity modelling as well as examples of human activity models previously created within the organization. Another WIL service identifies people within the organization who are more experienced in human activity modelling than herself and recommends them for collaboration.

We propose an adaptive approach to supporting WIL. Adaptive systems have been developed for various purposes such as helping users to find information, supporting learning, and supporting collaboration [4]. Therefore, we consider adaptivity as a highly promising means to support a variety of different learning practices [2] at the workplace, e.g. searching for relevant knowledge, or trying to find knowledgeable colleagues. In order to make an environment adaptive to the user it needs to 'know' what the user is able to do and what she is not able to do. For this purpose the environment contains a user model, which can be understood as a representation of "the knowledge about the user, either explicitly or implicitly encoded, that is used by the system to improve the interaction" ([5], p.6).

The goal of this contribution is to present our conceptual approach to user model design and maintenance which addresses the specific challenges of WIL (Section 2). In Section 3, we show its relevance by introducing a reference implementation which was developed as part of the EU-funded, integrated project APOSDLE (www.aposdle.org, Advanced Process-Oriented Self-Directed Learning Environment). After a brief discussion of related work (Section 4), we will report on a pilot evaluation in Section 5.

APOSDLE is now at the end of its third year (of four year total). Within the first two years two prototypes have been developed. The third (and last) prototype is currently being implemented. We report here mainly about lessons learned from the evaluations of the first two prototypes and give an outlook on the third prototype.

2 WIL User Models and User Model Services

A number of design and usability challenges have to be tackled in order to not outweigh the benefits of the adaptation to the individual user. [4] has identified *predictability & transparency*, *controllability*, *unobtrusiveness*, *privacy*, and *breadth of experience* as critical challenges for adaptive systems. For WIL environment design, *unobtrusiveness* and *privacy* constitute the hardest challenges.

In line with [4], we will use the term *obtrusiveness* to refer to the extent to which the system places demands on the user's attention which reduce the user's ability to concentrate on his or her primary tasks. We argue that the main source of obtrusiveness related to the user model is based on the ways in which information about individual users is acquired and maintained. In systems that support learning it is often natural to administer tests of knowledge or skill. The main advantages of testing are that it can be used in many domains and it is easy to implement. However, testing is highly obtrusive and cannot be applied to WIL for many reasons including the absence of the one correct solution for most work tasks (consider Laura's scenario above). One of the major challenges for WIL therefore is finding ways for updating the user model in an unobtrusive manner.

The second challenge for an adaptive system which is seen to be tougher for WIL than for other application cases is the issue of user *privacy*. To enforce user privacy in a WIL system which maintains a user model, appropriate organisational and technical measures have to be applied. Enhancing privacy in adaptive systems is a quite complex task as it depends on the organisational environment, data collected, privacy regulations etc. Additionally there is no standard approach to enhance privacy in adaptive systems [6]. Here we only want to point out the importance of privacy considerations in the WIL context but do not report on possible approaches.

A third challenge for adaptivity within WIL situations is the need for *seamless integration* of the adaptive learning support into existing work environments of the user. Typically, an adaptive system adapts its own functionality to the user based on her interactions with this very same system. Within WIL the challenge is to utilize the interactions of a user with potentially all her applications (e.g. MS Word and Requisite Pro in the above scenario) in order to adapt learning support functionality. Thus, we face a situation in which a user interacts with a large variety of applications and expects support for learning within the current work situation.

2.1. Designing a WIL User Model

There are several ways to diagnose user skills and maintain a user model. The knowledge represented in the user profile can be elicited explicitly from the user but it can also be acquired implicitly from inferences made about the user [7].

In our view, implicit acquisition means tracking *naturally occurring actions* [8]. Naturally occurring actions include all of the actions that the user performs with the system that do not have the express purpose of revealing information about the user. These actions may range from major actions like contacting a knowledgeable person about a certain topic to minor ones like scrolling down a page.

A number of interesting approaches have been suggested in other adaptive systems. For instance, researchers interested in adaptive hypertext navigation support have developed a variety of ways of analyzing the user's navigation actions to infer his or her interests or to propose navigation shortcuts (see, e.g., [9]). [10] came up with an unobtrusive approach for user learning interest profiles implicitly from user observations only. The problem is that such approaches cannot easily be re-used for adaptive work-integrated learning.

We therefore suggest tackling the challenge of user model maintenance by observing naturally occurring actions of the user [4] which we interpret as *knowledge indicating events* (KIE). KIE denote user activities which indicate that the user has knowledge about a certain topic. In the context of Laura's scenario the repeated execution of a task *user interviews* can be seen as a KIE for domain concepts such as *structured interviews* and *card sorting*. Another KIE for *card sortin*g could be that Laura has been contacted repeatedly about this topic in the role of an expert. KIE thus are based on usage data.

Our approach goes into a similar direction as [11], who suggest using attention metadata for knowledge management and learning management approaches. It is also related to the approach of evidence-bearing events (e.g. [12]). So far both approaches have been discussed from a rather technical point of view, e.g. the technological infrastructure necessary to identify and collect attention metadata. In addition, it has been speculated on how they could be applied in different settings. Our work provides a holistic framework for the use of KIE for the maintenance of WIL user models: starting with the identification of relevant KIE, their use for updating a WIL user model, and its technical realization through WIL user model services (see below).

In order to interpret KIE an underlying model is needed in the WIL user model which allows relating user actions to knowledge and skills and drawing conclusions on the user's knowledge level. Research into organizational structures identified that many companies create and maintain different types of formal models, so called *enterprise models* of their work domain [13]. The three most popular models are work domain models (typically represented as an ontology), process or task models (typically represented as a workflow or process model), and competency (or skill) structures (typically represented as a simple list or matrix). Such models provide a comprehensive representation of the whole domain. Based on these insights, we propose to structure the WIL user model as an overlay (for a definition see [5]) of existing enterprise models of the application domain in question.

2.2 Designing WIL User Model Services

Integrating learning support into work practices does not only mean running a WIL system and applications already deployed in organizations side by side, but also the possibility to extend and enrich existing applications. In order to meet this requirement, we propose a service oriented architecture (SOA) approach to WIL user model design and maintenance based on the OASIS reference model[1]. Firstly, the paradigm of SOAs allows us to split a kaleidoscope of adaptive functionality into different sub-

[1] http://docs.oasis-open.org/soa-rm/v1.0/soa-rm.pdf

groups (services) that can be used independently from each other. Secondly, services can easily be integrated in existing applications which make them especially attractive for the WIL situation. In Laura's scenario their functionality could include: Predicting Laura's performance in the (for her novel) task *stakeholder analysis* which involves previously mastered domain concepts such as structured interviews; detecting Laura's current learning need based on her missing experience in human activity modeling; and recommending an learning path for how to acquire skills in human activity modeling which is optimized based on her prior experiences. The latter has been seen as a crucial function in the context of work-integrated learning [14]. Thirdly, services are formally described which provides an overview of service functionality, protocols, etc. Eventually, existing services can be used for implementing new services (*service mesh-up*). With the term *WIL user model services*, we refer to all kinds of WIL functionality that maintains and utilizes the data stored within the WIL user model.

Despite their advantages, the main limitation of KIE is that they are imprecise and hard to interpret [4]. In order to draw meaningful conclusions based on KIE we propose to use a hybrid approach – utilizing available semantic structures (such as enterprise models) as well as scruffy methods (e.g. heuristics) to interpret the user's actions. These challenges are met with the design of *hybrid WIL user model services* [15] which maintain and interpret the WIL user model. We have identified four core types of services, covering the the basic needs of an WIL environment.

Logging services are responsible for updating the WIL user model with new observed KIE, and thus provide the basis for all other services. Sensors within the WIL environment (possibly from many different applications) send detected user activities (such as task executions, collaboration events) to logging services to be added to the user model. Pre-processing of incoming user activities are handled here. This could involve the transformation of user activities into a format required by the user model, or enriching incoming data with timestamps and other system related information. *Production services* make the stored KIE available to other (client) services within the WIL environment. Based on the specific requirements of the client, production services filter or aggregate KIE – they provide specialized views on the KIE. For example, one such service could produce a list of all tasks executed by one user. The receiving client could then provide visualizations of task executions over time. Views also offer a way to retrieve usage data associated with a specific enterprise model. Besides providing predefined views filtering usage data, production services could also allow to query the user model with individual parameters

Inference services process and interpret KIE to draw conclusions about different aspects of users, such as levels of knowledge. Inferences are then utilised to adapt the functionality of the service itself, or by providing the outcome to other services. A WIL user model allows generating inferences in different ways. Heuristics could be directly applied on KIE to generate aggregated information about users. Exploitation of KIE with regard to enterprise models, or a hybrid approach by combining heuristics with organisational models, could also lead to inferences.

Control services provide ways to control KIE stored in the user model. Controlling usage data is important for handling privacy issues and imprecise KIE collected in the user model. Privacy issues could be addressed by applying certain privacy policies of organizations to KIE. An example would be a policy about data retention, demanding the deletion of KIE after a certain period of time. The aspect of imprecise data can be addressed by presenting users with an overview of KIE associated with them. Based on this overview users could then use a control service to manually delete or modify the collected data.

3 The Second APOSDLE Prototype

The aim of the adaptive WIL system *APOSDLE* is to improve knowledge worker productivity by supporting learning situations within everyday work tasks. The understanding of a user's knowledge level and her learning goals is a central part of the APOSDLE environment. A comprehensive overview of APOSDLE and its functionality has been given in [16]. In this contribution, we only describe the mechanisms that are related to the user model and user model services.

3.1 APOSDLE Enterprise Models

As mentioned above we suggest designing a WIL user model as an overlay of existing organizational models. Within APOSDLE we have chosen to implement three organizational models for one and the same application domain, the domain model, task model and learning goal model. In order to build these models, we have developed a Modeling Methodology [17] which supports the creation of integrated models (instead of separate ones). All three models and the meta-schema are represented in OWL and are stored within a component referred to as the *knowledge base*.

The purpose of the *domain model* is to provide a semantic and logic description of the work domain (e.g. requirements engineering) which also constitutes the learning domain of an APOSDLE deployment environment. The domain is described in terms of concepts (e.g. requirements) and relations (e.g. is part of) that are relevant for this domain. Technically speaking the domain model is an ontology that defines a set of meaningful terms which are relevant for the domain and, which are used to classify and retrieve knowledge artifacts.

The objective of the *task model* is to provide a formal description of the tasks (e.g. human activity modeling) the knowledge worker can perform in a particular domain. The task model identifies and groups tasks and their interdependencies and determines a formalization of patterns and procedures occurring in a business domain. The very core of a process model is a control flow. For the sake of consistency with the domain model, we have also translated the control flow into an OWL ontology.

The *learning goal model* establishes a relation between the domain model and the task model. It maps tasks of the task model to concepts of the domain model. A learning goal describes knowledge and skills needed to perform a task, with respect to a certain topic in the domain model. For example, the learning goal 'apply card sorting to task user interviews' means that in order to perform user interviews Laura needs to know how to apply the card sorting method. In other words, each learning goal refers to one topic in the domain model. This formalism is necessary for a number of functionalities provided by the APOSDLE user model services. For example, it enables the determination of user skills from past task executions (*task-based knowledge assessment*, see *people recommender service* below), or the identification of a user's learning need within a certain task (see *learning need service* below). Within APOSDLE, the formalisms employed for achieving these functionalities are based on competence-based knowledge space theory (e.g. [18]) which is based on Doignon & Falmagne's knowledge space theory [19]. Competence-based knowledge space theory is a framework that formalizes the relationship between overt behavior (e.g. task performance) and latent variables (knowledge and skills needed for performance) and that has several advantages for WIL environments. One such advantage is that the mappings afford the computation of *prerequisite relationships* between learning goals (see e.g. [20]). This allows us to identify learning goals which should be mastered by the user on the way to reaching a higher level learning goal.

3.2 APOSDLE Workflow

In the second APOSDLE prototype, recommendations of learning goals, (learning) material, and knowledgeable colleagues are always provided depending on a user's current work task. Since here we do not have a learning system as the central application but integrated into the work environment, we need a way of observing what users are doing in order to identify their current task (and potentially other KIE). In APOSDLE, this task detection is realised by a specialized agent [21]. This agent observes the user interactions (e.g. key strokes, mouse movements, applications specific actions) with typical MS Office and Internet applications and compares them to previously learned task specific interaction patterns of the organization. Whenever a new task execution is detected the APOSDLE logging service is invoked.

The role of the user's current task in APOSDLE's second prototype is twofold. On one side, the tasks serves as a trigger for learning, as it determines the knowledge and skills that a user needs to have in order to perform the task successfully. The knowledge and skills required for performing the task are compared to the knowledge and skills of the user (*learning need analysis*), and a *learning need* is identified. The *learning need* of a user is a (possibly empty) set of learning goals (based on domain concepts), about which the user needs to learn. In order to facilitate the learning process, the learning goals are presented to the user as a *learning path*, i.e. an optimized sequence in which the learning goals should be tackled in order to maximize learning transfer. On the other side, the task constitutes the (currently only) KIE in the second APOSDLE prototype. In line with competence-based knowledge space theory [18] the underlying heuristics is the following: If a user is able to perform a task, she has all knowledge and skills required for this task. If the user selects a learning goal, APOSDLE triggers a search of knowledge artifacts relevant to the learning goal, and a search of people relevant to the learning goal. The results of these searches are displayed in the form of resource and people lists unobtrusively to the user. The people list contains a number of *knowledgeable persons* with respect to the learning goal. The decision of who is a knowledgeable person obviously is based on the data in the APOSDLE user model and is made by the *people recommender service*. At any time, the user has the possibility to view her task executions logged by APOSDLE, and she has the possibility to delete them. Additionally, the user can choose between three different pre-defined privacy levels (*public*, *private*, *anonymous*), which define the visibility of usage data presented to other users. For instance, if the privacy level *public* is selected, other users have access to the task history of the user.

3.3 The APOSDLE User Model

The APOSDLE user model is an overlay of the topics in the domain model. Whenever a user executes a task (e.g. user interviews) within the APOSDLE environment the counter of that task within her user model is incremented. The APOSDLE user model counts how often the user has executed the task in question. It therefore constitutes a simple numeric model of the tasks (KIEs) which are related to one or several topics in the domain. Based on the learning goal model we can infer that the user has knowledge about all the topics related to that task (e.g. structured interviews and card sorting). Therefore, by means of an inference service (see below), information is propagated along the relationships defined by the learning goal model, and the counter of all topics related to the task are also incremented. Consequently, the APOSDLE user model contains a value for each user and each topic at any time during system usage. As mentioned above, within the first two APOSDLE prototypes we have been focusing on task executions as the only KIE. On the one hand, this was for reasons of simplification, on the other hand the task execution was seen as most relevant KIE. We report on our lessons learned in Section 6.

3.4 APOSDLE User Model Services

The APOSDLE system provides service implementations for all types of WIL user model services proposed previously. Figure 1 presents an overview of APOSDLE user model services and how data is exchanged with the user model and corresponding APOSDLE client applications.

The APOSDLE system implements two different **logging services**. The *work context logging service* is dedicated to collect executions of tasks corresponding to the task model (delivered from the task detection agent). Logging information consists of a user identifier, a task identifier and an optional timestamp (depends on privacy settings). The second logging service, *resource activity logging service* collects all activities related to resources presented to users. Such actions are reading documents, engaging in learning events, or contacting another user. Both services receive their data from work context observation components running on the APOSDLE client Applications (see Figure 1). Incoming data is transferred into the required format of the APOSDLE user model and stored in a database backend. Taking the scenario introduced in Section 1, the *work context logging service* would update the user model with Laura's current task (*human activity models*). Other actions she might do while performing the task are logged by the *resource activity logging service*.

In order to allow users to examine the information WIL services have gathered, APOSDLE offers two **production services**. The *usage data history service* delivers a history of task executions and all resource-based actions. The output of this service is basically a history of all events including all KIEs. Another feature is that relations between events are also preserved. It provides a way to visualize which steps users

have undertaken when executing a certain task. It also features a more in depth view into data generated by different services. The usage data history service can for example provide an overview how learning goals evolved over time based on tasks users have executed. The *usage data history service* in our scenario provides Laura with an overview about the knowledge the user model assumes she has acquired about her current task. It could also help her to get to know people who have been recommended to her by an inference service (by showing her a top ten list of performed tasks for a recommended user).
The *evaluation service* is another kind of production service. It is specially designed to export different aspects of usage data for evaluation outside the APOSDLE system. In APOSDLE this service generates files containing detailed information about task executions, system usage, and information from inference services.

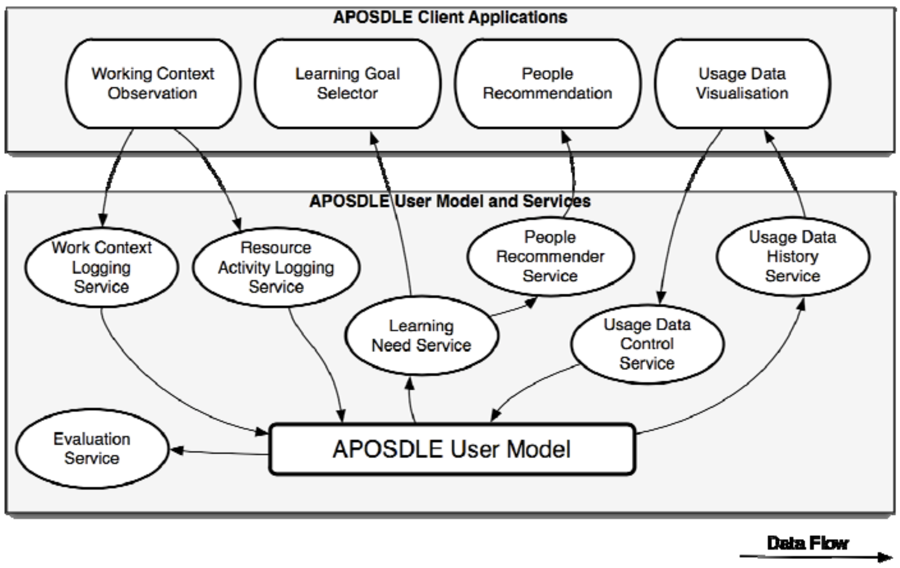

Fig. 1. Interaction of APOSDLE user model and user model services with APOSDLE client Applications

As one of the most important **inference services** the *learning need service* allows computing a learning need for a user. Its design is driven by the goal to support knowledge workers based on their knowledge level. A user's learning need is inferred in three steps.

Starting with the user's current task, the user model is queried to retrieve the required learning goal vector *r* for this task. The vector *r* represents for all learning goals of the domain whether or not they are required to perform certain task. The user model is again queried with the *required learning goal vector* as parameter to retrieve the current knowledge levels vector *k* for the user. The vector *k* consists of numeric counters for all learning goals of the domain. Each counter represents the knowledge gained by a user for a learning goal. The higher the count the more knowledge has been gained.

Step two calculates the knowledge gap a user might have for a certain task. A knowledge gap vector *g* is obtained by normalizing the current knowledge levels vector *k* and subtracting it from the required learning goal vector *r*. The resulting vector *g* provides knowledge levels ranging from 0 (learning goal might have been reached, great experience) to 1 (learning goal was not addressed until now, less to no experience).

The third step generates the learning need based on the knowledge gap calculated in step two. The less experience a user has acquired for a learning goal (low value in *g*), the higher the rank of the learning goal. The 'most required' learning goal is therefore listed on the top of the learning need. The learning need is used by the APOSDLE system in two ways. An application running in the working environment of the user visualizes the result as a ranked list. The first learning goal is automatically pre-selected, which invokes an information retrieval service to find resources relevant for the learning need. In the case of Laura the *learning need service* recommends learning goals helping her to accomplish the task human activity modelling. Based on these learning goals an information retrieval system provides her with resources previously created within the organization.The Learning Need Service also provides other services with current knowledge levels of users. This feature is utilized for example by the *people recommender service* described below as basis for its inferences.

The *people recommender service* aims at finding people within the organization which have expertise related to the current learning goal of the user. This service provides similar functionality as the expert finding systems described in [22]. Users specialised in certain topics are represented in the user model with high knowledge levels for these topics. Other users can now individually be provided with colleagues having equal or higher experience. Compared to the MetaDoc system [23] this service uses a more dynamic way of identifying experts. Knowledgeable users are identified by comparing the current knowledge levels vectors of all users with the knowledge level vector of the user who will receive the recommendation. To infer knowledgeable users, the people recommender service utilises the Learning Need Service to retrieve current knowledge levels vectors for all users. The next step removes all users with lower knowledge levels compared to the user receiving the recommendations. The remaining users are then ranked according to their knowledge levels in the current knowledge level vectors. The most knowledgeable user will be ranked highest. The service can be configured to use the availability status of users as ranking criteria. This setting allows recommending only users currently available. Moving back to the scenario the *people recommender service* recommends a list of people within Laura's organization who are more experienced in human activity modelling than the service assumes Laura currently is.

APOSDLE implements two **control services**. The *usage data control service* allows users to modify and delete any usage data. APOSDLE clients present users with a task history provided by *usage data history service*, and invoke the *usage data control service* to delete task executions selected by users. A dedicated privacy component (part of the APOSDLE server) also accesses this service to enforce certain privacy policies on usage data.

The APOSDLE environment is implemented as a Java client-server architecture applying the SOA paradigm to structure the server functionality into services. A dedicated component on the server exposes all services as web services. Client applications [24] can connect to one or more services depending on the features needed. Within in the server, all services are connected to one or more components implementing the actual functionality. All user model services run independently and communicate with the user model or other services using their exposed interfaces. In the second APOSDLE prototype orchestration is done by manually specifying for each service, where other services are located.

4 Related Systems

Throughout the previous sections we have discussed related research with respect to our research approach. In this section we provide a short overview of other research projects dealing with related user modelling architectures. As we are not aware of any adaptive learning systems specifically dedicated to WIL, we shortly discuss similar approaches in the area of adaptive e-learning and compare them to APOSDLE.

KnowledgeTree [12] is a distributed architecture for adaptive e-learning separating its functionality into different servers and services. As APOSDLE, KnowledgeTree utilizes a centralised, event-based user model to track student activities. Adaptations of functionality and content are separated from the user model into servers (similar to the APOSDLE user model services). A main difference exists in the way how user events are collected. KnowledgeTree collects usage data from users interacting with web sites (portals) providing, e.g. learning courses about programming. APOSDLE does not provide dedicated sites to record data, but focuses on collecting usage data from the users' working environments. Following this approach APOSDLE is open to a large set of data sources (applications) providing input to refine user models.

ELENA [25] is another architecture providing personalized support for learners by following the paradigm of a service-oriented architecture. ELENA describes its services very detailed in an ontology and uses several interoperability standards. ELENA integrates all services into a central top-level service communicating with client applications. Services are also adapted based on a learner profile, and offered as web services using WSDL descriptions. We see the approach of ELENA as complementary to the APOSDLE approach as it focuses more on interoperability and integration of existing learning repositories rather than trying to use document repositories available in companies as sources for learning material.

5 Pilot Evaluations

For APOSDLE's second prototype, our aim was to formatively evaluate the user model and its maintenance mechanisms. The outcomes of these evaluations now inform the re-design of the user model for APOSDLE's third prototype.

Evaluation of a user model and its maintenance is difficult for several reasons [26] (for example, in most cases a control group using a version of the system without a user model or with a different user model is not available) and a number of preconditions have to be fulfilled: A heterogeneous group of persons needs to be available with different levels of knowledge of the learning domain in question. Typically, in the case of evaluations within organizations such a group of persons is hard to find. In order to compare the user model with the 'true' knowledge of the users the knowledge of all the participants in the evaluation study must be known. This again is a hard challenge in the context of WIL. For the sake of external validity of the evaluation study, i.e. in order to generalize the outcomes to real-world WIL situations, the participants should execute genuine work tasks and interpret the usefulness of the offered learning content to their specific situation.

With these challenges for evaluation in mind, three pilot evaluations were designed and conducted to assess the WIL user model and WIL user model services for APOSDLE's second prototype: (i) a paper-based lab study with students of requirements engineering, (ii) an analysis of user log data from the application of APOSDLE's second prototype in use in different application domains and (iii) an observation of students in the domain of *statistical data analysis* trying to learn with the prototype. The three pilot evaluations are briefly described below. Due to the low validity of external criteria in study (i), the non-authentic behaviour of users in study (ii), the low number of participants in study (iii) and because the participants in studies (i) and (iii) were students in a lab and not workers at their workplaces, the outcomes of these studies cannot be generalized. Still they serve as valuable input for further development of the APOSDLE user model and the KIE approach in general.

A paper-based lab study (i) was conducted with 17 students of the learning domain *requirements engineering* in order to compare different algorithms for updating the user model based on the students' task performance. Self appraisal, appraisal of a supervisor and personal learning need assessment were used as criteria against which the outcomes of different updating algorithms were tested. Self-appraisal was included in the study in order to investigate if it would be a useful criterion for comparison in realistic WIL settings (where performance tests of the workers cannot be applied). The results caution towards the use of self-appraisal information in WIL. A low correlation was found between predicted task performance (based on the algorithms) and self-assessed task performance which however might be due to the low validity of the criterion variable (self-assessed task performance). Therefore, no definite conclusions could be drawn on the usefulness of the algorithms to be applied in the APOSDLE user model.

The second APOSDLE prototype suggests (based on the current work task) a ranked list of learning goals the user should tackle in order to improve her performance. For each learning goal the user can select between different *learning activities* (*reading a text*, *performing a learning event* or *contacting a person*). Each learning activity of a user is logged by the *evaluation service* together with the rank position of the learning goal the learning activity is related to. In study (ii), this information was analysed. Our hypothesis was that the rank position of a learning goal should be correlated to the frequency with which a learning activity was performed. That is, the closer the rank position to the top of the list, the more often a learning activity should be performed. Limited log data is available for 35 users in four application domains, with at least 8 different users in each domain. The results of our log data analysis show low correlations between the rank position of a learning goal and the frequency with which a learning activity was executed. However, since the users did not use the APOSDLE prototype regularly during their work but instead used it to explore the possibilities of WIL the log data represents non-authentic user behaviour. Therefore, we are not ready to reject our ranking algorithm. Instead further examination is needed. Despite the study's (ii) severe limitations we learned that the approach of task-based maintenance of the user model is extremely sensitive with respect to misuse. If a user does not use the system 'seriously' but just 'plays around with it' (or if the user just clicks on several tasks unintentionally) the task-based approach quickly leads to inappropriate user models and inappropriate rankings of learning goals. For further development of the KIE approach, this points to the necessity of basing the inferences on more reliable input data than just the information that the user "clicked on a task". This could be realized by adding a condition, e.g. regarding a task execution only as KIE if the user did not click on any other task within a certain time span (e.g. 10 sec). A further possibility for improving the reliability of the KIE would be looking at additional KIE (e.g. the user carries out a learning activity about a topic).

Finally, in study (iii), 5 students were observed and interviewed while they were trying to learn with APOSDLE in the learning domain of *statistical data analysis*. Our aim was to investigate the effects of the learning goal ranking on the actual performance of users in realistic tasks which they were not able to solve before the study (pretest). In the pilot study, control groups were used to compare three different versions of the ranking algorithm: (a) the ranking algorithm as it was designed for APOSDLE (taking into account both the requirements of the task and the knowledge of the user), (b) a shuffled list of learning goals required for the task at hand (taking into account the requirements of the task but not the knowledge state of the user), and (c) a set of learning goals randomly selected (neither taking into account requirements of the task, nor the knowledge state of the user). Each of the participants had to solve three different tasks, one for each version of the algorithm. With versions (a) and (b), the previously unknown tasks could be solved by all participants, whereas the task could be solved by none of them when algorithm (c) was applied. Additionally, a slight difference in the users' behavior was found between versions (a) and (b): In case of version (a), users by tendency selected less learning goals and more frequently carried out learning activities for learning goals on the top of the list in comparison with version (b) of the ranking algorithm. This serves as a first indicator, that the ranking algorithm is useful. Of course, further experimentation with larger samples is needed.

6 Conclusion and Outlook

This contribution presents our approach to user model design based on KIE for WIL environments in which unobtrusive assessment of user's knowledge levels is essential. A variety of hybrid user model services operate on this user model in order to add observed KIE, to provide its information (possibly in a filtered and aggregated manner) to other WIL applications, to infer knowledge levels and learning needs, and to allow users to examine and adapt their user model data. The APOSDLE environment serves as a reference implementation of the concepts proposed.

In APOSDLE's first and second prototypes, the maintenance of the user profile was solely based on past tasks performed. While there is some evidence that in fact most learning at the workplace is connected to performing a task, and that task performance is a good indicator for available knowledge in the workplace, this restriction to tasks performed certainly limits the types and number of assessment situations that are taken into account. It is evident that a user's knowledge and skills do manifest themselves through other types of user interactions with the WIL system. For example, a user who seeks help while performing a task might be in a different knowledge state than a user who provides help to others. Additionally, the tasks a user performs may be driven by organizational constraints or simply by task or job assignments and they may therefore only draw a partial picture of the knowledge and skills a user has available. Moreover, in study (ii) the approach of using task as the only basis for user model maintenance has turned out to be extremely error-prone and vulnerable to fallacious user behavior, such as accidentally clicking on tasks, or 'playing around with the system'. In order to improve the knowledge level assessment of the APOSDLE environment we are currently working on including a variety of different KIE such as collaboration events and document creation. In addition, we also plan to incorporate negative KIE, such as unsuccessful task executions. In doing so, instead of inferring the minimum competency state, i.e., competencies a worker has available at the minimum, the 'real' competency state of a worker could be approximated.

Acknowledgements

APOSDLE is partially funded under the FP6 of the European Commission within the IST Workprogramme (project number 027023). The Know-Center is funded within the Austrian COMET Program - Competence Centers for Excellent Technologies - under the auspices of the Austrian Federal Ministry of Transport, Innovation and Technology, the Austrian Federal Ministry of Economy, Family and Youth and by the State of Styria. COMET is managed by the Austrian Research Promotion Agency FFG.

References

1. Lindstaedt, S. N., Ley, T., Mayer, H.: Integrating Working and Learning in APOSDLE. In: Proceedings of the 11th Business Meeting of the Forum Neue Medien, November 10-11, 2005, University of Vienna, Austria (2005).
2. Eraut, M., Hirsh, W.: The Significance of Workplace Learning for Individuals, Groups and Organisations. SKOPE, Oxford & Cardiff Universities (2007).
3. Christl, C., Ghidini, C., Guss, J., Lindstaedt, S., Pammer, V., Scheir, P., Serafini L,: Deploying semantic web technologies for work integrated learning in industry. A comparison: SME vs. large sized company. In: A. Sheth et al. (eds.): Proceedings of the ISWC 2008, 7th International Semantic Web Conference, Karlsruhe, Germany, Oct 26-30, 2008, (2008).
4. Jameson, A.: Adaptive interfaces and agents. In: J. A. Jacko, &. A. S. (eds.): Human-computer interaction handbook, pp. 305-330, Erlbaum, Mahwah, NJ (2003).
5. Kass, R., Finin, T.: Modeling the User in Natural Language Systems. Computational Linguistics, 14 (3), 5-22 (1988).
6. Wang, Y., Kobsa, A.: Respecting User's Individual Privacy Constraints in Web Personalization. In: C. Conati, K. M. a. G. P.(eds.): Proceedings of the UM07, 11th International Confe-

rence on User Modeling, Corfu, Greece., pp. 157-166, Springer, Berlin, Heidelberg, New York (2007).
7. Benyon, D. R., Murray, D. M.: Adaptive systems; from intelligent tutoring to autonomous agents. Knowledge-Based Systems, 6 (4), 197-219 (1993).
8. Jameson, A.: Adaptive interfaces and agents. In A. Sears, J. A. Jacko (eds.): Human-computer interaction handbook. Erlbaum, Mahwah, NJ, 305-330 (2003)
9. Goecks, J., Shavlik, J.: Learning users' interests by unobtrusively observing their normal behavior. In: IUI 2000: International Conference on Intelligent User Interfaces (2000) 129-132.
10. Schwab, I., Kobsa, A.: Adaptivity through Unobstrusive Learning. KI Special Issue on Adaptivity and User Modeling, 3, 5-9 (2002).
11. Wolpers, M., Martin, G., Najjar, J., Duval, E.: Attention Metadata in Knowledge and Learning Management. In: Proceedings of the I-Know 2006 (2006).
12. Brusilovsky, P.: KnowledgeTree: A Distributed Architecture for Adaptive E-Learning. In: WWW 2004, May 17–22, 2004, New York, USA (2004) 104-113.
13. Fox, M., Grueninger, M.: Enterprise modeling. AI Magazine, 19 (3), 109-121 (1998).
14. Billett, S.: Constituting the Workplace Curriculum. Journal of Curriculum Studies, 38 (1), 31-48 (2006).
15. Lindstaedt, S. N., Ley, T., Scheir, P., Ulbrich, A.: Applying Scruffy Methods to Enable Work-integrated Learning. Upgrade: The European Journal of the Informatics Professional, 9 (3), 44-50 (2008).
16. 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 (2008) 234-244.
17. Ghidini, C., Rospocher, M., Serafini, L., Kump, B., Pammer, V., Faatz, A., Zinnen, A., Guss, J., Lindstaedt, S.: Collaborative Knowledge Engineering via Semantic MediaWiki. In: Proceedings of the I-Semantics 2008, Graz, Austria, Sep. 3-5 2008 (2008) 134-141.
18. Korossy, K.: Extending the theory of knowledge spaces: A competence-performance approach. Zeitschrift für Psychologie, 205, 53-82 (1997).
19. Doignon, J., Falmagne, J.: Spaces for the assessment of knowledge. International Journal of Man-Machine Studies, 23, 175-196 (1985).
20. Ley, T., Ulbrich, A., Scheir, P., Lindstaedt, S. N., Kump, B., Albert, D.: Modelling Competencies for supporting Work-integrated Learning in Knowledge Work. Journal of Knowledge Management, 12 (6), 31-47 (2008).
21. Lokaiczyk, R., Godehardt, E., Faatz, A., Goertz, M., Kienle A.; Wessner, Wessner, M., Ulbrich, A.: Exploiting Context Information for Identification of Relevant Experts in Collaborative Workplace-Embedded E-Learning Environments. In: Proceedings of the EC-TEL, EC-TEL 2007, Crete, Grece, 15-20 Septmber 2007 (2007) 217-231.
22. Yimam-Seid, D., Kobsa, A.: Expert finding systems for organizations: Problem and domain analysis and the demoir approach. Journal of Organizational Computing and Electronic Commerce, 13 (1), 1-24 (2003).
23. Boyle, C.: An adaptive hypertext reading system. User Modeling and User-Adapted Interaction, 4 (1), 1-19 (1994).
24. APOSDLE Consortium: Second Prototype APOSDLE. (2008). http://www.aposdle.tugraz.at/media/multimedia/files/second_prototype_aposdle.
25. Dolog, P., Hentze, N., Nejdl, W., Sintek, M.: Personalization in distributed e-learning environments. In: Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters (2004), 170 - 179.
26. Chin, D. N.: Empirical Evaluation of User Models and User-Adapted Systems. User Modeling and User-Adapted Interaction, 11, 181-194 (2001).