A methodology for elicitng, modelling, and evaluating expert knowledge for an adaptive work-integrated learning system

Tobias Ley\textsuperscript{a,c,*}, Barbara Kump\textsuperscript{b}, Dietrich Albert\textsuperscript{c}

\textsuperscript{a}Know-Center Graz, Inffeldgasse 21a, A-8010 Graz, Austria
\textsuperscript{b}Graz University of Technology, Knowledge Management Institute, Inffeldgasse 21a, A-8010 Graz, Austria
\textsuperscript{c}University of Graz, Cognitive Science Section, Universitätsplatz 2, A-8010 Graz, Austria

Received 7 April 2008; received in revised form 25 November 2009; accepted 8 December 2009
Communicated by p. mutholland
Available online 16 December 2009

Abstract

We present a methodology for constructing and evaluating models for adaptive informal technology-enhanced workplace learning. It is designed for knowledge-intensive work domains which are not pre structured according to a fixed curriculum. We extend research on Competence-based Knowledge Space Theory which has been mainly applied in educational settings. Our approach employs systematic knowledge elicitation and practically feasible evaluation techniques performed as part of the modelling process for iterative refinement of the models. A case study was performed in the Requirements Engineering domain to apply and test the developed methodology. We discuss lessons learned and several implications for knowledge engineering for adaptive workplace learning.

Keywords: Competence-based knowledge space theory; Work-integrated learning; Domain modelling; Knowledge elicitation; Model evaluation

1. Introduction

Recent advances in information and communication technologies have had a significant impact on learning at work by facilitating integration of learning into working processes directly at the workplace. Work-integrated learning as it is seen here is a form of informal workplace learning as conceptualized by Eraut (2004) or Garavan et al. (2002) and it shares with this the informal, interpersonal and experiential character and draws out the place and process of work as the main setting of learning activities. Learning activities are not determined by a course structure but by the working context, i.e. by individual characteristics of the learner, the task at hand, the business process context, the organisational environment, or several of those (Farmer et al., 2004; Lindstaedt et al., 2008; Schmidt, 2004). Work-integrated learning systems\textsuperscript{1} can support a learner in her search by exploiting available context information and pre-selecting and structuring available content without restricting her self-regulation. Such approaches seek to make use of available knowledge artefacts in an organisation and reuse them for learning purposes, thereby creating learning opportunities “on the fly”.

Imagine, for example, a Requirements Engineer who develops a context model for a flight control system. Further imagine, the Requirements Engineer is a recent graduate and has never developed a context model for a system in a practical setting. Hence, in this task, the person may benefit greatly from available documents, like context model templates or previous applications of these templates in similar projects, or from advice from knowledgeable colleagues. In the specific situation, our Requirements Engineer may not be aware that all these resources are available, or that they could be helpful to her. A learning

\textsuperscript{1}We use the term “work-integrated learning system” to refer to the technical system which supports the learning process of an individual in the context and in the process of his or her daily work.
system could provide guidance on what content is available, and could use information about the Requirements Engineer, in order to select the most relevant content for her.

The issue of using context information for selecting appropriate learning resources has been addressed in research focussing on the development of adaptive eLearning systems (e.g., Albert et al., 2002; Brusilovsky and Nijhavan, 2002; De Bra et al., 2004; Eklund and Brusilovsky, 1999; Rich, 1999). Here, adaptivity means that learners are provided with information that is neither redundant (i.e. already known), nor irrelevant (i.e. not useful for performing the task) nor incomprehensible. While a human teacher easily, intuitively, and in a flexible manner is able to respond to the learner’s (work related) needs, establishing adaptivity constitutes a complex and demanding matter within the construction of an eLearning system.

In order to realise adaptivity with a learning system, different types of formal models need to be constructed. For this purpose, traditional intelligent tutoring systems take as a starting point either a well structured task domain with algorithmic tasks, such as performing tasks with a computer software (e.g., Rich, 1999), or a well structured curriculum (e.g., Billett, 2006). However, this is rare in the case of work-integrated learning. Very often, neither the tasks are algorithmic, nor is the learning domain sufficiently well structured in advance. Therefore, modelling is especially difficult in the case of work-integrated learning.

Even though a variety of adaptive learning systems have been presented in the literature, the process of how the necessary formal models were built is hardly ever described in a way that it can be replicated. In most cases, formal models are created by a domain expert and then considered valid. We would argue that this assumption does not necessarily hold. Valid formal models, however, constitute a necessary pre-condition for the success of an adaptive learning system. Obviously, if the models are not valid, this impairs the quality of the adaptive learning support.

With this article, we suggest a methodology for constructing and evaluating formal models for an adaptive work-integrated learning system. The modelling methodology is designed for application domains which are knowledge-intensive (i.e. dealing predominantly with the acquisition, application and transfer of knowledge) and less structured than those of traditional tutoring systems (i.e. allowing for considerable amount of leeway in the execution of tasks). It employs systematic elicitation and modelling of expert knowledge and a rigorous evaluation technique. Consequently, the methodology we present improves the process of building models in two ways. First, it constitutes a theoretically founded guideline for creating different types of formal models. Second, it includes as an inherent part of the modelling methodology several validation checks (i.e. checking whether the underlying assumptions are fulfilled) from which it is possible to judge the current quality of the models and improve them in an iterative way. The latter point is especially important. Because of the complexities and irregularities encountered in practical settings, we feel that it is impossible to devise a methodology which in itself is guaranteed for valid and reliable models. Instead, reliability and validity can only be guaranteed by making quality checks and refining accordingly.

The remainder of this paper is organized as follows: in Section 2, we propose a number of models for adaptive work-integrated learning. In Section 3, we then present a variation of Doignon and Falmagne’s Theory of Knowledge Spaces as the theoretical basis. We then present our modelling methodology in Section 4. At each step of the methodology, we report lessons learned we obtained in a case study, where the methodology was applied for the requirements engineering domain. We discuss the applicability of the methodology and several critical issues we came across during the case study in Section 5 and conclude with an outlook on future work in Section 6.

2. Eliciting and evaluating models for adaptive technology-enhanced work-integrated learning

The first step in building a knowledge model for adaptive technology-enhanced work-integrated learning is to collect and structure the expert knowledge that is available in the domain of interest (e.g., the learning domain of Requirements Engineering). Subsequently, and in line with e.g., Benyon and Murray (1993), we refer to the structured expert knowledge of a domain as the Domain Model. In our conception, the Domain Model for a work-integrated learning system consists of two components: the tasks that need to be performed in the domain, such as creating a context model in the domain of Requirements Engineering, and the knowledge and skills needed to perform these tasks, such as an understanding of the concept of system boundaries which constitutes one important element of a context model in Requirements Engineering. For this second component we adopt the term competency, which relates to a collection of knowledge and skills that is used to perform the tasks in question.

In line with the knowledge-intensive domains we are targeting, the use of competencies as more holistic latent constructs should enable the system to go beyond a mere performance support system where information about exact task executions or defined problem solving procedures is the only input variable. Taking into account human competencies and their role in the production of performance should allow learners to develop capacities to perform in a broad range of situations. Competencies as related to learning and work performance have also received great attention within skill based content personalisation (Cardinali, 2006), or in organisational eLearning (Sicilia, 2007). In brief, these approaches seek to tailor content presentation for learning purposes to the learner’s current learning needs which can be derived from (a)
demands of the current task and (b) available knowledge, skills and prior experience of the learner. This kind of analysis—the comparison of task demands with a learner’s abilities—has also been termed competency gap analysis (see Sicilia, 2005).

In order to perform a competency gap analysis, a model of the learner’s available knowledge (competency state) is required. In line with e.g., Benyon and Murray (1993) or Jameson (2003), we refer to that kind of model as the User Model. As a component of the learning system, the User Model depicts the learner’s abilities either in terms of (observable) tasks he or she is able to achieve, or in terms of (latent) competencies (e.g., by inferring skills and knowledge from the performance of observable tasks), or in terms of both.

The User Model constitutes the rationale for presenting individualised learning opportunities and has to be updated according to the learning progress. Ideally, this update happens to a large extent automatically, as the learner’s use of the system is monitored. Such automatic updating certainly presents a major challenge. However, in the case of work-integrated learning, where learning happens directly in the task context, there is a potential for updating the User Model according to past task executions (Ley et al., 2006). It is for this reason that the Domain Model in our case reflects the connections between tasks and competencies needed to perform those tasks, and we refer to the updating of the User Model as a task-based competency assessment.

For building both, the Domain Model and the User Model as components of a learning system, eliciting knowledge from experts is necessary. Knowledge acquisition and elicitation methods have a long tradition in the area of constructing expert systems. For example Cooke (1994), or Shadbolt and Burton (1989) present and discuss diverse knowledge elicitation techniques for knowledge based systems. With the growing demand for adaptive systems to support workplace learning and development, the topic has recently received more attention once again.

Knowledge elicitation constitutes a critical phase in building the Domain Model and the User Model of a learning system and the application of appropriate knowledge elicitation techniques is crucial to the success of work-integrated learning systems. The main difficulty associated with knowledge elicitation has been termed the knowledge specification and acquisition gap, denoting the gap that exists between the actual knowledge of experts and the knowledge they might be able to articulate (Castro-Schez et al., 2004, see also Studer et al., 1998). This holds especially for situations where learning prerequisites need to be modelled: there is still a lack of methods to support domain experts in deciding which elements are prerequisite for learning some other elements.

With these considerations in mind, it becomes clear that before a competency gap analysis can be conducted to devise individualised learning opportunities, it has to be assured that Domain and User Model are empirically valid, i.e. that they correspond to the true state of affairs. This question of evaluating and validating models has been discussed both in psychological conceptions of modelling (e.g., Brewer, 2000; Sohn and Doane, 2002), in the domain of eLearning (e.g., Millán and Pérez-De-La-Cruz, 2002) as well as in knowledge engineering (e.g., Cooke, 1994; Studer et al., 1998; Sure et al., 2004), and human–computer interaction (e.g., Chin, 2001; Fischer, 2001).

In practise, the evaluation of formal models, especially for adaptive work-integrated learning systems, is very often neglected for several reasons. On the one hand, the activity of building formal models is costly in itself (Cooke, 1994), and expert resources required for building, evaluating and revising the models are scarce. On the other hand, most modelling approaches are inherently difficult to validate.

In the next section, we present a theoretical approach, Competence-based Knowledge Space Theory, which can serve as a reference framework for building formal models, and which clearly formalizes all assumptions so that they can be checked with different validation techniques.

3. Competence-based Knowledge Space Theory for modelling work-integrated learning

Competence-based Knowledge Space Theory (based on the Competence–Performance Approach by Korossy, 1997, 1999) is a mathematical psychological theory that originated from research into the Theory of Knowledge Spaces (see e.g., Doignon and Falmagne, 1985; Falmagne et al., 1990). Knowledge Space Theory structures knowledge elements in terms of their learning prerequisites which makes it especially suited for adaptive learning.

The practical convenience of Knowledge Space Theory and its extensions has been demonstrated in several areas. Among others, the approaches were integrated in a large scale and commercially successful environment for adaptive testing and teaching in high-school mathematics (ALEKS; see ALEKS Corp., 2003) and in a demonstration system to learn and teach elementary probability theory (RATH; see e.g., Hockemeyer and Albert, 1999). The application of Knowledge Space Theory was also suggested for modelling less formalised fields such as workflow-processes (Stefanutti and Albert, 2002), educational and

---

2In line with the general convention, we refer to the task of gathering information from any source as knowledge acquisition, and to the sub-task of gathering knowledge from the expert as knowledge elicitation (see e.g., Shadbolt and Burton, 1989).

3For similar conceptions see Düntsch and Gediga (1995), or Doignon (1994).

4Our presentation of Competence-based Knowledge Space Theory constitutes a simplification, as the approach originally takes into account the case of more than one possible “learning history” (Doignon and Falmagne, 1999, p. 62) for one task. Within the present work, only one such learning history is considered.
cross-cultural values (Albert et al., 2003) or child philosophy (Pilgerstorfer et al., 2006). Ley and Albert (2003a) proposed the use of Competence-based Knowledge Space Theory in knowledge management to establish worker profiles and derive individual educational needs. Ley (2006) examined the applicability of the approach in four case studies and obtained promising results in terms of construct validity.

3.1. Basic assumptions of Competence-based Knowledge Space Theory

The Theory of Knowledge Spaces by Doignon and Falmagne (1985, 1999) presupposes that a field of knowledge can be parsed into a set \( \mathcal{A} \) of problems (or tasks) \( x \in \mathcal{A} \) each of which has a correct response. Instances of tasks from the Requirements Engineering domain are Plan and Prepare Acquisition Sessions, Identify a Basic List of Stakeholders, or Build a First-Cut Context Model.

According to Competence-based Knowledge Space Theory, every person is able to accomplish a particular subset \( \mathcal{Z} \) of tasks of \( \mathcal{A} \), and accordingly he or she is in a particular performance state \( \mathcal{Z} \subseteq \mathcal{A} \). For instance, the Requirements Engineer introduced in Section 1 may be able to identify stakeholders and to build a first-cut Context Model but may not be able to develop an extended Context Model. The pattern of mastery and failure of the Requirements Engineer in all tasks of the domain constitutes her solution pattern, and the set of tasks she is able to accomplish represents her performance state.

One of the central ideas of Knowledge Space Theory is that solution dependencies exist among problems of a domain. These solution dependencies are formally denoted by a prerequisite relation \( \prec \subseteq \mathcal{A} \times \mathcal{A} \) that is interpreted as follows: when \( a \prec b \) holds for two tasks \( a, b \in \mathcal{A} \), one can say that \( a \) is a prerequisite for \( b \). For example, a prerequisite relationship might be given between the two tasks (a) Build a First-Cut Context Model and (b) Develop an Extended Context Model. The relationship \( a \prec b \) means that being able to develop a first-cut context model (a) is a prerequisite for being able to build an extended Context Model (b). Accordingly, every performance state in the domain that contains the task (b) would also contain the task (a). For our example, this means that because of the prerequisite relation, there is no Requirements Engineer who is able to build an extended Context Model but is not able to build a first-cut Context Model. Because of these dependencies among tasks, not every conceivable subset of the set of tasks \( \mathcal{Z} \) constitutes a feasible performance state. The set of all feasible performance states in a set \( \mathcal{A} \) is called performance structure \( \mathcal{P} \) on \( \mathcal{A} \), denoted by the pair \( (\mathcal{A}, \mathcal{P}) \). If \( \emptyset, \mathcal{A} \in \mathcal{P} \) and \( \mathcal{P} \) is stable under union, the pair \( (\mathcal{A}, \mathcal{P}) \) is called performance space on a domain.\(^5\)

This closure under union property is sensible from the learning and teaching perspective: it is reasonable to assume that any union of two performance states can be reached by learning, and, hence, this union of the two sets should also be a performance state and be part of the performance space. The closure under union property also leads to a computational simplification as it allows for storing the space as a base. The base of a performance space is the set of performance states which cannot be generated by taking unions of other performance states. Thus each performance state in \( \mathcal{P} \) can be written as union of base elements.

By extending Knowledge Space Theory, Korossy and others describe a domain not only by a set \( \mathcal{A} \) of tasks \( x \in \mathcal{A} \) but, in addition, the domain is conceived by a set \( \mathcal{E} \) of elementary competencies \( e \in \mathcal{E} \) for each of which it can be assumed whether a person has this competency or not.

Competencies are knowledge and skills which a person has available to perform a task, and which can be acquired through learning. Examples for Requirements Engineering competencies are data-gathering skills, or knowledge about adjacent systems. Analogous to the performance structure, a prerequisite relation is assumed on the set of elementary competencies. The collection of all knowledge and skills of a person form his or her competence state \( K \). The set of all feasible competence states can be modelled through a finite, non-empty family of competence states, the competence structure \( (\mathcal{E}, K) \). The pair \( (\mathcal{E}, K) \) is a competence space if \( \emptyset, \mathcal{E} \in K \) and \( K \) is stable under union.

The formal linkage between competencies and tasks happens by defining a relationship between tasks and competence states. For each task \( x \in \mathcal{A} \) and each competence state \( K \in K \) it is uniquely determined whether or not \( x \) can be successfully performed in \( K \). Formally spoken, the set of tasks \( \mathcal{A} \) is interpreted in \( K \) by the mapping \( k(A) \), which is called interpretation function. The interpretation function assigns to each task \( x \in \mathcal{A} \) a set \( k \) that contains all elementary competencies \( e \in \mathcal{E} \) which are necessary and sufficient for successfully performing the task.\(^6\) For example the task Build a First-Cut Context Model may require two competencies: Knowledge about Context Models (i.e. knowledge about the elements and relations of these models) and Ability to produce a Context Model (i.e. knowledge about the notation and rules and how to apply them). The task can therefore be solved by every person whose competence state contains at least these two competencies, i.e. obviously also by persons whose competence states include additional competencies.

Another mapping, the representation function \( p(K) \) assigns to each competence state \( K \in K \) (and each person respectively) a unique (possibly empty) set of tasks which can be solved in that competence state (and by that person). Note that the representation function is fully determined by the interpretation function.

\(^5\)The mathematical properties of performance structures and the relationship to corresponding prerequisite relations are formulated in Doignon and Falmagne (1999).

\(^6\)In Korossy (1997), the set of competencies \( K \in K \) that are necessary and sufficient for solving the problem was referred to as minimal interpretation, \( M \subseteq k \), of a task \( x \in \mathcal{A} \).
Within the present work, a model consisting of a competence space, an interpretation function and a representation function and a resulting performance structure is referred to as competence–performance structure.

3.2. Constructing a competence–performance structure for work-integrated learning

As mentioned previously, a central aspect of work-integrated learning is the close connection between task performance and learning which makes it a natural area of application for Competence-based Knowledge Space Theory. At this stage, we will derive an approach for constructing a competence–performance structure for a work-integrated learning application. In the next section, we will then present an illustrative example.

Korossy, 1997 (see also Korossy and Held, 2001) describes the rough steps in the process of developing a competence–performance structure as a knowledge acquisition and modeling procedure with experts. Typical tasks have to be identified and a set of competencies has to be established which are necessary for performing the tasks. The set of competencies has to be structured with regard to “diagnostically relevant” prerequisite relationships (Korossy, 1999). As suggested above, establishing a prerequisite relation on the set of elementary competencies constitutes the hardest challenge for ill-structured learning domains which we are typically facing in the case of work-integrated learning: one can imagine that experts may be overstrained when they are asked to decide whether, e.g., analytical skills are prerequisites for the ability of abstraction, whether it is right the other way round or whether the two are independent from each other.

In order to overcome the difficulties of directly establishing a prerequisite relation on the set of elementary competencies, we have introduced the simplified method of a task–competency matrix (Ley and Albert, 2003a, 2003b; see also Ley, 2006). With this matrix, experts are asked to assign to each task the set of competencies necessary for performing the task (the minimal interpretation of that task). The idea is to identify feasible competence states by means of the interpretation function: if a person is able to perform a certain task, he or she should have all required competencies. Hence, determining the minimal interpretation of a task leads to a feasible competence state. By closing the set of minimal interpretations for a set of tasks under union, the set of all feasible competence states in a domain, i.e. the whole competence space, is obtained. The corresponding performance structure is then obtained by assigning to each competence state the set of tasks that are solvable in the respective state.

3.3. An illustrative example

To illustrate the mechanics of our approach and the potential for application, we will give a small illustrative example taken from the case study to be presented later in more detail (in Section 4).

Assume the set of ten tasks and the set of six competencies from the Requirements Engineering domain given in Table 1.

Also assume the interpretation function given as a task–competency matrix in Table 2. In each line of the table, crosses indicate that a task requires the set of competencies as a minimum. The rightmost column of the table gives the minimal interpretation for each task. In case the minimal interpretation cannot be generated by the union of two or more other elements, it is an element of the base. For this reason, the minimal interpretation of task 5_3, which is \{13, 15, 20\}, is not part of the base. In total, the base of the competence space in the example comprises seven competence states.

From the minimal interpretation given in Table 2, a prerequisite relation on the set of competencies can be derived, as there is a formal relationship between the set of competence states and the prerequisite relation on the set of competencies (as described in Section 2.1). As all minimal interpretations constitute competence states, we can approximate the prerequisite relation on the set of competencies using the following rule: a prerequisite relationship is always assumed between two competencies when only one is present when the other is also present. More formally, this means, a competency a is prerequisite for a competency b (a < b) if whenever b is assigned to a

<p>| Table 1 |</p>
<table>
<thead>
<tr>
<th>Set of tasks and set of elementary competencies (Example).</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Tasks</strong></td>
</tr>
<tr>
<td>3_1</td>
</tr>
<tr>
<td>4_2</td>
</tr>
<tr>
<td>4_3</td>
</tr>
<tr>
<td>4_5</td>
</tr>
<tr>
<td>4_6</td>
</tr>
<tr>
<td>5_1</td>
</tr>
<tr>
<td>5_2</td>
</tr>
<tr>
<td>5_3</td>
</tr>
<tr>
<td>5_4</td>
</tr>
<tr>
<td>5_5</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Competencies</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>Knowledge about the Activity Model and the activity descriptions</td>
</tr>
<tr>
<td>12</td>
<td>Knowledge about the Context Model</td>
</tr>
<tr>
<td>13</td>
<td>Knowledge about the Strategic Dependency Model (SD-Model)</td>
</tr>
<tr>
<td>15</td>
<td>Knowledge about the Strategic Rationale Model (SR-Model)</td>
</tr>
<tr>
<td>16</td>
<td>Knowledge of validating the SR Model</td>
</tr>
<tr>
<td>20</td>
<td>Ability to produce an SR Model</td>
</tr>
</tbody>
</table>
task then a is also assigned to that task for all tasks of the domain. For example, we arrive at the prerequisite relation given in Fig. 1. For instance, a person who has Competency 16 is assumed to also have Competencies 13 and 15. In other words, Competencies 13 and 15 are prerequisites for Competency 16. This is because there is no base state (and hence no competence state) that includes 16 without at the same time including 13 and 15.

In continuing with the example, we now show how to assess a learner’s competence state from the tasks performed in the past (task-based competency assessment), and how to compute a competency gap for a task at hand. For task-based competency assessment, the learner’s task performance is reviewed. Imagine a Requirements Engineer, who is able to perform all tasks except for task 5_4_Check that each individual SD Model is complete and correct with stakeholder goals, soft goals, tasks and resources and task 5_5 Validate the SR Model against the SD Model (cross-check), in other words the set \{3_1, 4_2, 4_3, 4_5, 4_6, 5_1, 5_2, 5_3\} constitutes the performance state of that person according to the example introduced previously. The competence state of the person would then be inferred by taking the union of the well performed tasks’ minimal interpretations. Consequently, the competence state of the Requirements Engineer (\{3, 12, 13, 15, 20\}) would comprise all competencies except for competency 16 Knowledge of validating the SR Model.

Conducting a competency gap analysis happens by building the difference of a learner’s competence state and the minimal interpretation of a task or a set of tasks the learner should perform. For example, if a person only had the competency 15 Knowledge about the Strategic Rationale Model (SR-Model), and should perform task 4_2 Model the system’s hard and soft goals, the person would need to acquire the competency 20 Ability to produce an i* Model. In other words, the set \{20\} constitutes the competency gap of this person.

This deterministic approach to competency assessment is of course limited in the face of uncertainty. Stochastic knowledge assessment routines, originally proposed by Falmagne and Doignon (1988), take into account uncertainties in updating the user model. Falmagne et al. (2006)
show how such knowledge assessment happens in a large scale adaptive learning system that makes use of Knowledge Spaces by endowing the knowledge structure with a likelihood distribution and updating this distribution using Bayesian inference rules. Because in Knowledge Space Theory these stochastic methods do not have to be built into the domain model (see e.g., Villano, 1992), they will not play a major role in the following presentation of our modelling methodology. We will extend this discussion in Section 5.5 when we talk about the application of the domain model for user modelling.

4. A methodology for building and evaluating competence–performance structures for work-integrated learning

After we have established Competence-based Knowledge Space Theory as a promising approach for adaptive work-integrated learning, we will now present a methodology for building a competence–performance structure for this purpose.

Because the application of the methodology has to be feasible in practical settings, the following constraints have to be taken into account. First, in order to minimise the involved domain experts’ efforts, we are relying mainly on document analyses, and on reusing existing sources (e.g., competency catalogues) during knowledge acquisition. Only at critical stages of the modelling process (e.g., the review of task lists, or the task–competency assignment), we are involving experts. Second, we are attempting to embed model evaluation into the process of model development, and thereby again minimizing experts’ efforts. Finally, the methodology does not require spatial proximity of the modelling team, and hence it takes into account limited opportunities for face to face meetings. As can be seen from Fig. 3, the methodology we are suggesting consists of five steps initially being performed in sequence. As we have indicated previously, it is an important feature of the methodology that it is iterative: with the information obtained during model evaluation, further elicitation of tasks and competencies or a change in the model needs to take place (see the broken arrows in the figure).

A case study was performed to apply and test the developed methodology for establishing and evaluating...
competence–performance structures in the Requirements Engineering work domain. With the case study it was to be examined whether the methodology was appropriate in practical settings. For this purpose, we decided to involve two independent experts and to establish the models in two separate sub-domains in order to be able to make comparisons later on.

Table 3 gives an overview of each step of the modelling methodology, its purpose, the suggested methods to be employed and the roles responsible for carrying out the activities. In the following sections, we will describe each step in more detail together with lessons learned from our case study.

4.1. Defining the scope and purpose of the models

In our view, defining the scope and purpose is an important first step in knowledge engineering, while surprisingly it is often neglected in other methodologies (see Schreiber et al., 1999; Uschold et al., 1998 for notable exceptions). It is generally employed to generate requirements on the models which will be heavily influenced by their intended use in the adaptive system. The better the intended purpose is known, the easier it will be to formulate these requirements in advance. The intended purpose also determines many of the subsequent model design decisions including granularity and the degree of model precision and validity that is needed.

A second important question in this step is the learning domain that is the intended target, and its intended scope. At this stage, the decision has to be made who is the target group, and what knowledge shall be conveyed. Partly, scoping of the models will also depend on the amount of documented information sources about the domain and the availability of experts.

4.1.1. Procedure and outcomes

In our case study, the purpose of the models had been specified in advance by an intensive requirements analysis with stakeholders in the application domain and by describing the intended use cases of the adaptive learning system. Use cases, such as the accessibility of documents, learning material and experts in a work-integrated fashion, and the tailoring of these materials to the current user’s task context and prior knowledge, gave us a good basis for specifying the intended use of the models. Also it became apparent that competency assessment would have to exploit information on task executions, since a test-like assessment scenario was not felt to be adequate for the intended user groups.

The domain selected for testing the developed modelling methodology was Requirements Engineering, more specifically the Requirements Engineering with Scenarios in User-centred Environments process (RESCUE, Maiden and Jones, 2004). We selected the RESCUE process for our case study, because it is a well-tried Requirements Engineering method that has been successfully applied in practice. RESCUE was developed to standardise the process of requirements elicitation and requirements management. The designated outcome of the RESCUE process is a catalogue of consistent and valid requirements for a future system. We think of RESCUE as a typical domain for work-integrated learning as it is on the one hand clearly structured into separable tasks and processes. On the other hand, it offers considerable leeway for performing the tasks and requires significant amounts of skills and knowledge to be acquired.

Throughout the RESCUE process, different modelling and analysis activities run in parallel. These sub-processes strongly inform each other and are aligned at various synchronisation points. We chose the first two subprocesses of RESCUE, namely Activity Modelling and System Goal Modelling, as the domains for the case study.

Activity Modelling is done to provide an understanding of how people work in order to baseline possible changes to it. Data on human activities are gathered using observations or interviews, structured into activity descriptions and built into the Activity Model. The aim of the second stream, System Goal Modelling, is to model the future system boundaries and to identify dependencies between actors for goals to be achieved. This is done by building and iteratively refining a graphical Context Model using the representation that shows what actors can achieve on their own, and what they depend on others for.

The reason we chose these two sub-domains was twofold. First, it was found during the scoping phase that...
Table 3
Overview of the steps in the methodology: purposes, methods and responsibilities.

(1) Defining the scope and purpose of the model

Purpose of this step:
- Generating requirements on the resulting models
- Scoping of the domain

Suggested methods and activities:
- Requirements analysis techniques, e.g., use case methods, stakeholder analysis

Roles and responsibilities:
- Facilitating process (RE), providing input (St)

(2) Eliciting knowledge

Purpose of this step:
- Generating lists and descriptions of tasks and competencies

Suggested methods and activities:
- Gather experts and documents with performance and learning focus, existing task and competency catalogues
- Perform content analysis techniques
- Generate personas and concrete instances of task performance with experts
- Complement knowledge elicitation with formal techniques, like repertory grid or critical incidents technique
- Validate lists with experts informally in structured interviews

Roles and responsibilities:
- Performing analysis (KE), providing input (DEP)
- Performing analysis (KE), providing input and validation (DEL)

(3) Linking tasks and competencies

Purpose of this step:
- Generating the interpretation function, i.e. sets of competencies needed to perform in tasks

Suggested methods and activities:
- Gather expert ratings using the task–competency matrix

Roles and responsibilities:
- Facilitating process (KE), providing input (DEP, DEL)

(4) Building competence–performance structures

Purpose of this step:
- Generating the knowledge base from elicited knowledge

Suggested methods and activities:
- Employ computational procedures from knowledge space theory, i.e. deriving base, competence space and prerequisite relation on competencies

Roles and responsibilities:
- Performing analysis (KE)

(5) Evaluating competence–performance structures

Purpose of this step:
- Evaluating the knowledge base in terms of reliability and validity
- Suggesting ways to improve the knowledge base

Suggested methods and activities:
- Check reliability using inter-rater or test–retest consistency measures
  - Check for validity using actual or notional solution patterns or assessments (e.g., by employing the
    Leave One Out method)
  - Generate suggestions for revisiting the knowledge base where reliability or validity is low

Roles and responsibilities:
- Performing analysis (KE), validating and providing input for model adaptations (DEP, DEL)

*aRoles: Stakeholders (St), Requirements Engineer (RE), Knowledge Engineer (KE), Domain Experts with Performance Focus (DEP), Domain Experts with Learning and Teaching Focus (DEL).
sufficient documented material and expertise was available for the two, both for modelling purposes and for learning with the realized system. A more detailed description of the available materials and the experts is given in Section 4.2.

The second reason for choosing these two sub-processes was that they were sufficiently distinct in terms of some characteristics that we deemed to be important for the applicability of our methodology: the amount of practical expertise with the two streams differed, as experts had more practical experience with System Goal Modelling. The amount of structuring in terms of a clearer separation and sequencing of tasks was also more pronounced in System Goal Modelling than in Activity Modelling. This would enable us to compare the impact of different domain characteristics for the applicability of our methodology. After having presented the results of the case study, we will return to this discussion of domain applicability in Section 5.1. We will also discuss the effects of this phase for granularity of the models (in Section 5.3) and for the amount of model precision and validity needed (in Section 5.5).

4.2. Eliciting knowledge: tasks and elementary competencies

For constructing a competence–performance structure, tasks and competencies need to be collected that cover the selected domain (see Section 3.2). For doing so, suitable experts as well as existing documentations have to be identified. Complying with the requirements of work-integrated learning, the chosen documents and experts need to cover two types of foci. First, a performance focus requires that experts have performed the tasks that are being modelled, and the material needs to come from actual task executions. Second, a learning focus has to be present in which experts have to make inferences about knowledge needed for a task and the documents ideally have some relevance for learning in the tasks.

This means that experts ideally have both, practical experience in the domain as well as experience in supervising others or in teaching. Documents need to be both learning content (e.g., a description of how to perform a context analysis) but also results from prior executions of tasks (e.g., context models from prior projects). The documents can later be used as a seed for the actual learning material used by the system.

Next, sets of tasks and competencies have to be formulated. In our experience, it is a good idea to start with the tasks as experts find it more natural to describe a domain in terms of what they do. Also, this way the model of learning is more directly related to actual task performance. The selected tasks form the operationalisation of the domain. Hence, it is important to ensure that they cover the whole domain to be modelled. For this it is helpful to sequence tasks, as it may help experts to come up with a consistent list. Task granularity is important, as it will determine the number of assigned competencies and the granularity of the whole model. At the outset, we decided to rely mainly on experts’ intuition and on the use cases from the scoping phase when deciding about the granularity of tasks.

In a next step, a set of elementary competencies has to be identified. Clearly, this is the more difficult task as it involves making inferences and is therefore more error prone (Lievens et al., 2004). Formal techniques which we have employed to support this process include the Critical Incidents Technique (Ley and Albert, 2003a) and the Repertory Grid Technique (Ley and Albert, 2003b, see also Gaines and Shaw, 1993). Especially the latter helps to draw out underlying assumption that experts have and is also perceived to be helpful by the experts themselves.

As in the present case, our experts were experienced teachers, we decided against these formal techniques. Instead, we relied on extensive existing documentation on the RESCUE process, as well as on existing competency catalogues. We also made an attempt to ground the elicitation process by referring the experts to concrete situations and persons they had worked within the past.

4.2.1. Procedure and results of the case study

First of all, one set of tasks to be achieved in the Activity Modelling stream and one set of tasks for System Goal Modelling were derived. To enable a later comparison of the resulting models, the two streams were modelled separately. Since a number of detailed descriptions and tutorials for the RESCUE process were available, a content analytical approach was chosen to derive candidate lists of tasks and elementary competencies. These were later validated with two requirements engineering experts with considerable expertise in the RESCUE methodology.

4.2.1.1. Deriving tasks and initial competency phrases: summarising qualitative content analysis. The main part of tasks were elicited from the extensive RESCUE process documentation (Maiden and Jones, 2004), a very detailed description of the four streams that includes theory, worked examples (e.g., from a requirements elicitation project for a transportation management system) and practical guidance.

We employed a summarising qualitative content analysis (e.g., Mayring, 2003) to derive initial task and competency descriptions. For the tasks, the whole document was walked through phrase by phrase. All phrases in the document that described activity (i.e. no explanations, no theory) were collected and paraphrased, i.e. reworded in a simple grammatical form without elaborations. To give an example, the text passage “the analysis of the activity should identify constraints of the domain in order for the design to support practitioners in complying with them” was transformed into “identify constraints of the domain”.

The list of activities was further reduced by removing activities that did not fit the intended granularity for tasks, i.e. too specific and too general activities. Similar tasks were accumulated into one task. For example, the phrases “identify external resources” and “identify resources available to achieve goals (observable)” were aggregated
into “identify external resources that are available to achieve goals”. In contrast, the phrases “identify the system’s goals and high level functional goals” and “identify local goals” were not merged, since they were assumed to be performed in a different manner (and thus to require different sets of competencies). In doing so, a first list of 85 tasks was identified, 60 for Activity Modelling and 25 for System Goal Modelling.

To derive competency descriptions, the RESCUE documentation was analyzed in a similar manner. Phrases that potentially indicated knowledge or skills required for Activity Modelling and System Goal Modelling were extracted and summarized. Moreover, existing competency catalogues, e.g., from van den Berg (1998) or the Occupational Information Network (“O*NET”, National O*NET Consortium, 2005), were used to complement the list of competency phrases. The latter, “O*NET”, is a database of worker attributes that was created to support, e.g., the development of job descriptions. From all these sources, a list of 98 phrases describing competencies was produced.

4.2.1.2. Validation of tasks and competencies

The first, Expert 1, was professor of Systems Engineering who had been a principal investigator on numerous projects in the fields of Requirements Engineering and socio-technical systems design. The other, Expert 2, a research assistant of Expert 1, had done significant work in the areas of human–computer interaction and Requirements Engineering. She also had taught various courses and published a significant amount of papers in related areas. According to their own judgement, both domain experts had considerable practical experience in System Goal Modelling, and to a lesser degree in Activity Modelling.

As the experts and the modelling team were spatially separated (the Experts were in England, the modelling team in Austria), the review of tasks and elementary competencies happened via email and telephone. Prior to the phone call, the lists of tasks were e-mailed to the RESCUE experts and they were asked to run through the task lists and to make suggestions for modifications. Also, the experts were asked to jointly think of five Requirements Engineers they had both worked within the past and describe them in a short paragraph. We will refer to these as personas. The personas were used in the current phase for drawing on expert’s practical experience, as well as in the validation phase (see Section 4.5.1.2).

During the phone call, the meanings of ambiguous tasks were clarified or removed. After this review of the task lists, in total 47 tasks remained, 29 of which belonged to Activity Modelling, and 18 belonged to the System Goal Modelling stream. This list was later accepted as the final version (see Appendix A).

After that, the RESCUE experts were asked to think of the personas and some concrete situations in which tasks were performed and to brainstorm competencies necessary for performing the tasks. During the interview, the experts came up with 36 additional phrases describing competencies, which were added to the list of 98 phrases that had been derived from document analyses. When considering the intended granularity of competencies as defined in the scoping phase, it became clear that the list of competency phrases would have to be reduced. As a more specific guideline and in line with the purpose of the system, it was useful to think of learning content that could be consumed in 1–2 h working time. Also, if two separate elements of knowledge or skill would always be used together in all of the tasks, and additionally would most likely appear together in any learning material that was available, this constituted a good argument that they could be combined.

Hence, competency phrases referring to such low-level competencies (e.g., Knowledge about strengths and weaknesses of every acquisition technique) were categorised into more general competencies (e.g., Knowledge about acquisition techniques available). A catalogue of competencies was derived (see Appendix B) in which the original phrases were preserved under the merged competencies so as to retain the meaning of the merged competencies. After validation with the RESCUE experts, the final competency catalogue consisted of 33 competencies (20 knowledge, 13 skills).

4.2.2. Lessons learned

The method of a Qualitative Content Analysis led to useful first lists of activity descriptions and competency phrases, and thereby significantly facilitated subsequent interactions with the experts. As to the experts involved, we certainly were fortunate to have available experts both with practical experience and with experience of supervising and teaching others. The latter clearly helped to identify competencies which turned out to be the more difficult part. We used several hints when interacting with the experts: we asked them to think about concrete persons they had supervised in the past and why these persons did a task well or needed support. We asked them to think of how people had developed expertise in a certain task, or about what they were teaching. Although we did not use any formal methods in this stage, we would recommend them in a case where only experts with less of a learning focus are available.

One of the main issues in this phase and a regular topic of discussion was the granularity of the model. We found that in general, experts had a good intuition about the granularity as they could draw on their own experience. We often reminded the experts of how the models would be used in a work-integrated learning fashion. The experts found it helpful to think of the possible future learning interventions and the available content to decide whether the granularity was in line with the overall purpose. We will
be giving some more thought to granularity issues as well as some formal considerations in Section 5.3.

4.3. Linking tasks and competencies

To build the competence–performance structure, it is necessary to link tasks and competencies obtained in the previous step to establish the interpretation function. We applied the method of a task–competency matrix as described in Section 3.2 which asks experts to assign to each task the set of all competencies necessary for an adequate performance. We decided to let experts make these judgements in increments of ‘‘absolutely necessary’’ (2), ‘‘somewhat necessary’’ (1), and ‘‘not necessary’’ (0). First, our experience had shown that raters feel more comfortable when they can rate in increments, and second we were intending to use only the extreme value of (2) in order to get more stable results for the structures. We also decided to obtain these assignments independently from both experts in order to check for inter-rater reliability later on and, hence, to obtain an impression of the agreement of experts’ understanding of the domain.

4.3.1. Procedure and results of the case study

In the case study, the task–competency matrix consisted of an Excel spreadsheet with all 47 tasks in the rows and all 33 competencies in the columns. This spreadsheet was sent to the RESCUE experts via e-mail before we called each expert separately. During the calls, the experts were instructed to start with the first Activity Modelling task, and to assign the value 2 to every competency in the catalogue that they regarded ‘‘indispensable’’ for performing the task well. Every competency they regarded as ‘‘somehow important, nice to have’’, should be given the value 1. Once the procedure was clear, the experts were asked to finish the assignments on their own and to return the results via e-mail (see Appendix C for an example of such an assignment). Table 4 reports the descriptive results in terms of the number of assignments for each expert and for each stream.

4.3.2. Lessons learned

In both streams there was a large discrepancy of the amount of times value 1 was assigned by each of the two experts. Most likely, this was due to the fact that the description of ‘‘somewhat necessary’’ was not interpreted in the same way by the two experts. Using a scale for rating has proven to be a good strategy, as this helped to uncover these rating tendencies. As we later dichotomized the scale for building the models (see Section 4.4), these different rating tendencies did not cause any problems.

We also informally asked the experts about their experiences with assigning competencies in the matrix. Both experts were in agreement that the task had been extremely demanding requiring them to make over one thousand judgements. Also, they criticised the presentation format (Excel spreadsheet), and considered this to be very error prone. We are taking this feedback into consideration when we discuss the task–competency matrix and its application in Section 5.4.

4.4. Building competence–performance structures

The task–competency matrices derived in the previous step form the basis for spanning competence–performance structures. The procedure for doing so has been presented in Section 3.3. It is important to note that this procedure does not involve any more modelling effort, but it is merely a computational procedure.

In our case study, we had derived independent assignments from the two experts. A natural next step would be to find ways of merging the two to reduce errors. This can be done by the two experts agreeing on common assignments, the drawback being that it is time consuming and that the experts may not be available. So we decided to check for the effects of computational merging and apply the validation procedures to both the separate structures as
As well as a computationally merged structure. One method for merging two (or more) performance spaces computationally was suggested by Dowling (1994) who built a common space by taking the union of the different spaces. This procedure leads to a large number of performance states, and in turn to a reduction of prerequisite relationships (and hence reduction of the adaptive potential of the structures). Alternatively, one can unite the two minimal interpretations for each task, and this was what we did in the case study.

4.4.1. Procedure and results of the case study

Four competence–performance structures were built on the basis of the task–competency assignments obtained from the two RESCUE experts in the previous modelling step: from each expert’s task–competency assignment, a separate structure was derived for Activity Modelling and System Goal Modelling applying the procedure described in Section 3.3. The rating scales were dichotomized, that is only competencies with an assigned ‘2 (mandatory)’ for a task were used to build the structures. Additionally, a merged structure was produced for System Goal Modelling by uniting each task’s minimal interpretation. In other words, each competency at least one of the experts had considered ‘mandatory’ for the accomplishment of a certain task was also considered ‘mandatory’ in the merged assignment. A competence–performance structure was then derived for the merged assignment. The assignments for Activity Modelling were not merged because of low inter-rater agreement (see Section 4.5.1.1).

Table 5 summarises the most important characteristics of the five structures derived from the RESCUE experts’ task–competency assignments. The second column shows the numbers of competencies considered mandatory in the respective assignment, i.e. the cardinalities of the sets of elementary competencies. Furthermore, the numbers of all possible combinations of competencies, i.e. the cardinalities of the power sets on the sets of elementary competencies, are given, as well as the cardinalities of the bases and the competence spaces.

Table 5

<table>
<thead>
<tr>
<th>Assignment</th>
<th>( N_{\text{comp}} )</th>
<th>Power set( ^a )</th>
<th>Base</th>
<th>C. space</th>
</tr>
</thead>
<tbody>
<tr>
<td>Activity Modelling</td>
<td>Expert 1 22</td>
<td>4194 304</td>
<td>25</td>
<td>3338</td>
</tr>
<tr>
<td>Expert 2 21</td>
<td>2097 152</td>
<td>23</td>
<td>849</td>
<td></td>
</tr>
<tr>
<td>System Goal Modelling</td>
<td>Expert 1 19</td>
<td>524288</td>
<td>15</td>
<td>756</td>
</tr>
<tr>
<td>Expert 2 15</td>
<td>32768</td>
<td>12</td>
<td>166</td>
<td></td>
</tr>
<tr>
<td>Merged structure 22</td>
<td>4194 304</td>
<td>14</td>
<td>420</td>
<td></td>
</tr>
</tbody>
</table>

\( ^a \)Cardinality of the power set was calculated as \( 2^n \) where \( n \) is the number of competencies.

Fig. 4 gives a picture of the prerequisite relation on the set of elementary competencies (merged competence–performance structure for System Goal Modelling) separated into knowledge and skills. Numbers refer to competencies given in Appendix B. Edges indicate transitive prerequisite relationships among competencies, with lower vertices being prerequisites for upper vertices.

Fig. 4. Hasse-diagram of the prerequisite relation on the set of elementary competencies (merged competence–performance structure for System Goal Modelling) separated into knowledge and skills. Numbers refer to competencies given in Appendix B. Edges indicate transitive prerequisite relationships among competencies, with lower vertices being prerequisites for upper vertices.
performing validity checks which will be described in the next section.

4.5. Evaluating competence–performance structures: checking reliability and validity

Before the structures are employed in an adaptive learning system, it needs to be shown that the resulting structures correspond to the true state of affairs. Our approach here is to employ standard measures of reliability and validity which can be established in the modelling process. The purpose is to give experts and modellers early feedback about the quality of the structures for an iterative refinement.

Evaluating the reliability in establishing a competence–performance structure means to examine the degree to which different modellers (or experts) give consistent estimates of the same phenomenon, i.e. examining inter-rater reliability (e.g., Kline, 2005). For the present work, nonparametric measures of contingency were considered the most appropriate for comparing the two RESCUE experts’ task–competency matrices. In cases where only one expert is involved, consistency can be checked by a variation of test–retest reliability, e.g., by having the expert repeat some of the assignments at a later point. We have employed the latter method to compare assignments from a repertory grid interview to those obtained from task–competency matrices and found close to 80% agreement (Ley et al., 2006).

The validity of a competence–performance structure concerns the model’s usefulness for accurately assessing competencies and predicting performance in the context of the model’s application. For competence–performance structures, empirical validity can be demonstrated by examining whether observed solution patterns coincide with predicted performance states. This validation method has been carried out for several domains, e.g., chess, word problems and mathematics (e.g., Albert and Lukas, 1999; Doignon and Falmagne, 1999; Falmagne et al., 2006).

In knowledge-intensive work domain such as Requirements Engineering, a direct empirical validation by comparing observed solution patterns with predicted performance states is difficult or even impossible for the following reasons. A sufficient number of workers is required, who are neither able to achieve every task nor fail at every task. And, in order to obtain accurate solution patterns, every person under consideration must have tried to perform every task of the domain. Both these conditions may be hard to meet. Alternative validation methods focussing on construct validity have been employed for competence–performance structures in knowledge management (e.g., Ley, 2006; Ley and Albert, 2003b). Another possibility to overcome the problem of a small validation sample is suggested by the Leave One Out cross-validation method (e.g., Ley et al., 2006; Witten and Frank, 2005). Instead of a large empirical sample, the application of Leave One Out for validating a competence–performance structure only requires a small number of complete solution patterns (at least one), where complete means the assessment of one worker’s performance in every task of the domain under consideration.

The Leave One Out method considers the solution pattern of one person for \(n - 1\) tasks, that is one of the tasks is “left-out”. His or her competence state is inferred from the solution pattern in the remaining \(n - 1\) tasks by taking the union of all mastered tasks’ minimal interpretations. From the obtained competence state, the person’s performance in the \(n\)th task, the left-out task, is predicted: if the minimal interpretation of the \(n\)th task is a subset of the competence state, the person should perform the task well. Otherwise, the person should not be able to perform the task. This procedure is repeated for each task and each person under consideration. A contingency coefficient can be computed in order to determine whether a solution pattern of a person coincides with the performance (mastery or failure) in a task as predicted by the Leave One Out method.

It should be noted that this procedure corresponds to a task-based competency assessment procedure in which a person’s competency state is predicted from an analysis of her previous task performance. We therefore consider this validation procedure to be especially applicable in our case.

4.5.1. Procedure and outcomes of the case study

4.5.1.1. Checking reliability. To assess reliability in terms of inter-rater agreement between two experts’ task–competency assignments, contingency-tables were produced and a contingency coefficient \((\tau_b)\) was computed. Inter-rater agreement was examined across both streams as well as for each stream separately. Significance was tested two-tailed on a 5%-level.

The contingency coefficient of the raters’ assignments was moderately positive \((\tau_b = 0.32, z = 8.60, p < 0.01)\) across both streams. Looking at the two streams separately, the coefficient was 0.25 for Activity Modelling \((z = 5.90, p < 0.01)\) and 0.45 for System Goal Modelling \((z = 6.13, p < 0.01)\). Obviously, especially for Activity Modelling, the two RESCUE experts had a rather different understanding of which competencies were required for performing the Activity Modelling tasks.

4.5.1.2. Checking validity. To employ the Leave One Out method, estimations for complete solution patterns of persons performing the tasks are needed. For this, we used the personas the RESCUE experts had come up within the elicitation phase (see Section 4.2.1.2). Notional solution patterns for these five personas were collected by means of a validation questionnaire that was filled out by the two RESCUE experts.

The validation questionnaire was an Excel spreadsheet with the descriptions of all five personas and the lists of tasks for Activity Modelling and System Goal Modelling. Experts were requested to go through the task lists for each persona and decide for each task, whether the person would be able to accomplish the task without assistance (value 2) with assistance (value 1), or not at all (value 0).
conclude that the assessments were done reliably. Overall, the agreement between the experts’ assessment of personas was of satisfactory magnitude from which we expected, the solution patterns generated by both experts for Activity Modelling were higher correlated with the predictions of their own model than with the model of the other expert. This was not the case for System Goal Modelling derived from her own assignment, and $\tau_b = 0.26$ ($z = 3.01, p < 0.01$) for the predictions of Expert 2 for Activity Modelling and the assessment by Expert 1. In 3 of 4 cases, predictions of the model of Expert 2 were more consistent with both experts’ appraisals than the predictions of Expert 1 and of the merged structure. As expected, the solution patterns generated by both experts for Activity Modelling were higher correlated with the predictions of their own model than with the model of the other expert. This was not the case for System Goal Modelling derived from her own assignment, and $\tau_b = 0.26$ ($z = 3.01, p < 0.01$) for the predictions of Expert 2 for Activity Modelling and the assessment by Expert 1. In 3 of 4 cases, predictions of the model of Expert 2 were more consistent with both experts’ appraisals than the predictions of Expert 1 and of the merged structure.

4.5.2. Lessons learned

The results pertaining to reliability are in line with Ley (2006), who found rather high agreement within (stability), but low agreement between raters for task–competency matrices in knowledge-intensive work domains. Obviously, the two RESCUE experts were having different views about what competencies were required for performing certain tasks. A reason for this may have been that our task descriptions had been rather short so that some tasks were possibly interpreted differently by the two experts. More comprehensive documentation of results would have been advantageous.

Despite the rather low inter-rater reliability, the validity coefficients we obtained were moderate to high. We take this result as a first indication for the structures being robust for the purpose they were conceived for, namely to diagnose a competence state from task-based information. Obviously, merging competence–performance structures by uniting each task’s minimal interpretation did not enhance validity. Consequently, other ways to improve
validity by integrating competence–performance models should be sought. We will return to a more thorough discussion of reliability and validity in the overall discussion in Section 4.5.

Summarizing from the above, the procedures employed here (inter-rater reliability and the Leave One Out method) would suggest the following: in the case of the Activity Modelling stream, it would be necessary that the experts come to agree about some of the modelled tasks. A starting point could be those tasks that have especially low degrees of agreement and in which performance is poorly predicted by the Leave One Out method. For System Goal Modelling, we would recommend that the structure of Expert 2 is taken as a basis and subsequently refined by taking into account inconsistencies obtained from the analyses. For example, when looking at Fig. 5 which shows reliability coefficients for all tasks modelled in System Goal Modelling, it appears that task 4_1, 4_2 and 4_4 would be candidates for such refinement, as they obtained especially low scores.

5. Overall discussion and future work

The modelling methodology presented in this paper is based on a formal model of the relationships between tasks and competencies needed to perform the tasks. Hence, the methodology follows a theory-based and systematic approach for eliciting expert knowledge. In doing so, it complies with several demands that have been brought forward for establishing explicit linkages between the manifest behavioural level (task performance) and the latent level (competencies needed) in competency modelling (Lievens et al., 2004; Schmitt and Chan, 1998).

The developed modelling methodology was found to be generally feasible in organisational settings. Modelling the Domain Model and the User Model of a domain by starting from the tasks seems to be well suited for a work-integrated approach as it focuses the competencies very much on the actual tasks that need to be performed. Nonetheless, several improvements are conceivable in different modelling stages, and some of them have been identified in the previous sections. At this stage, we have identified five critical issues pertaining to the application of the methodology which we will now discuss in more detail. In each section, we will also present ongoing and future work to address the research challenges.

5.1. Applicability across domains

As stated in the introduction, applicability of our approach in different domains is still a matter of ongoing research. In this paper, we have gathered evidence about differences in two subdomains of the Requirements Engineering domain, Activity Modelling and System Goal Modelling. It turned out that inter-rater reliability was especially low for the first subdomain. This is also the one that is less formally structured in terms of a clear separation into tasks. This may have resulted in a poorer understanding between the two experts of what a particular task entails. For these less structured domains, we feel it is therefore important to establish higher agreement between the experts. This can be accomplished by more interactive methods, such as workshops or collaborative modelling environments (see also next section).

Practical feasibility is another important concern. Work in four other knowledge-intensive domains is currently being undertaken to check for more generalisable findings. Furthermore, we are continuing to seek ways for exploiting existing data sources (e.g., existing document collections or competency catalogues) to reduce manual modelling effort. For example, Pammer et al. (2007) show how term extraction from a collection of documents can support domain modelling.

5.2. Distributed and iterative modelling

When modelling in practical settings, one is faced with a number of challenges that threaten the applicability of rigorous and systematic procedures prescribed in theory. As a case in point, shortly after we had performed modelling of the two subdomains, one of the two experts was assigned to a different project and was no longer available to us. Besides availability, there are also issues around the distributed nature of modelling teams where it is often impractical or impossible to gather the team for extended periods (e.g., in a workshop setting). It is for these pragmatic reasons that our methodology seeks to support development of valid structures in a distributed setting, to allow for fine-tuning the applicability of the model to its intended purpose, and for repetitive refinement in those parts of the models where there are major discrepancies. We have gathered promising evidence that in spite of having abstained from using more interactive methods, the validity of the models has been rather high. Also, we have identified techniques to overcome a model’s deficits by looking at single tasks with low inter-rater agreement. Information about which tasks are problematic and need further improvement may be used as an input for more interactive and costly methods like modelling workshops to make them more effective.

Work is currently being undertaken to provide modelling tool support for the methodology we have presented here. In particular, we are currently designing a web-based modelling and collaboration tool. An extension of a Semantic Media Wiki will allow for communicating about tasks and competency descriptions and collaborative modelling more easily (Rospocher et al., 2008).

5.3. Granularity of the models

A critical issue, and one of frequent debate in our case study, was the question of the right level of granularity for
the models. In general, this is not a one-off decision. Clearly, to answer the question of the right level of granularity the focus should be on the intended use of the models. In our case, we did well with introducing some granularity the focus should be on the intended use of the models. In general, this is not a one-off decision.

More formally, the method of the task–competency matrix allows one to generate indicators that check the coverage of the models. For example, an index showing that several tasks have a very low number of competencies assigned (e.g., 0–2), or a very high number (e.g., more than 15) may indicate a mismatch in the granularity of the task and competency models.

While the issue of granularity will always require some sort of trading off in the process of modelling, there are conceptualizations that more formally describe tasks and competencies and which give hints for an appropriate level of granularity. For tasks that describe knowledge-intensive activities, one can rely on the CommonKADS methodology (Schreiber et al., 1999) which has suggested a typology of knowledge-intensive tasks. In a similar manner Anderson and Krathwohl (2001) have introduced an extension of Bloom’s taxonomy of learning goals which may be taken as the basis for a competency model (this idea has been briefly sketched in Ley et al., 2008).

5.4. Use of the task–competency matrix

The informal feedback we gathered about the use of task–competency matrix led us to consider several alternative strategies. The alternative suggested by research into Knowledge Space Theory would be to directly model the prerequisite relation by an expert query procedure. The drawback from this would be that this is usually perceived to be even more demanding because it is difficult to decide on the prerequisite relation without having a curriculum in mind.

A different alternative would be to assign metadata to available learning material which describes the kinds of learning goals the material teaches without considering the task demands (e.g., Conlan et al., 2002). Here, the drawback would be that no automated competency assessment would be possible, and that the learning goals would not be focused on actual demands (such as is the case when a competency gap analysis is employed), but rather on available learning material.

One decisive advantage in using task–competency matrices is the possibility to simply add or remove tasks or competencies from the structures without the need for a complete revision of the structures. Albert and Kaluscha (1997) have shown how this procedure affects the underlying structures.

We are currently designing a mapping tool that supports the use of task–competency mappings in the modelling process. It is intended that this tool will also integrate some basic measures of model coverage, reliability and validity such as those presented here. And, finally, our plan is also to integrate some graphical representation (e.g., of the prerequisite structures) into this tool (e.g., Nussbaumer et al., 2008), so that experts can be shown the resulting structures in the modelling process and make changes as they see fit.\footnote{We thank one of the anonymous reviewers of an earlier draft for pointing this out.}

5.5. Validity and use of the models

We have presented an approach and several methods to embed model evaluation into the modelling process for iterative refinement. The application of these evaluation methods is dependent on a formal model of the correspondence between the competence and the performance level. To the best of our knowledge, this is a unique feature of our methodology within the field of knowledge engineering for adaptive learning systems.

The question of the validity of a model in general comes back to the question of how the model will be used. In our view, iterative validation should be checked against and tailored towards measures that relate to the effectiveness of model use. This is exemplified by our use of the Leave One Out validation method which simulates use of the models for task-based competency assessment. As this method did make rather robust and valid judgements, we take this as an indication that the models are “fit for purpose”.

Actual use of our models in work-integrated learning systems follows a “scruffy” approach (Lindstaedt et al., 2008), meaning that a set of deliberately simple models are complemented by the use of probabilistic methods during runtime. In Section 3.2, we have indicated such stochastic procedures which could be used to deal with uncertainties in the user model. Intelligent Tutoring approaches have often employed Bayesian Nets in modelling uncertainties between evidence and hypotheses about latent knowledge states (e.g., Conati et al., 2002; Jameson, 1996). These approaches usually employ a more fine-grained modelling of problem solving processes, and therefore allow for more exact diagnoses. Also, instead of prerequisites relations, they model causal inferences. Nevertheless, Desmarais and colleagues have shown how procedures that operate on Bayesian Net structures (directed acyclic graphs) can be readily applied in the context of Knowledge Spaces for building item to item structures (Desmarais and Gagnon, 2006), and for user modelling (Desmarais et al., 2005). A promising direction of research appears to be a variation of the multiplicative updating rule originally proposed by Falmagne and Doignon (1988) which does not operate on the Knowledge States, but rather exploits the prerequisites structures of competencies.\footnote{Thomas Augustin, Personal Communication, 22 April 2009.}
to revisit parts of the models to improve them until they are valid to a degree that is consistent with the intended purpose of the models. This approach coincides with more iterative modelling methodologies (Angele et al., 1998) and with the fact that in our experience models need constant adaptation anyway.

6. Conclusions and outlook

With this work, we presented a psychological theory, the Competence-based Knowledge Space Theory, and an iterative methodology for modelling and validating the Domain Model and the User Model of an adaptive learning system. A valid competence–performance structure integrates both, the Domain and the User Model for a given domain and hence allows for directly deriving the most appropriate learning opportunity in a theory-driven manner. Moreover, it incorporates hypotheses (e.g., in order to perform a task, a certain set of skills is required) which can be directly exposed to different validation techniques.

The next step for our research will be to devise methods and measures to empirically evaluate a model once established in a work setting, i.e. to explore its fit to real data instead of notional solution patterns. This may be achieved with analysing documents produced by workers or by questioning supervisors about their employees’ actual performance in the modelled tasks. We are currently designing evaluation measures and strategies that may be employed in such a workplace setting when the models are operational.

Acknowledgements

APOSDLE (www.aposdle.org) is partially funded under the FP 6 of the European Community within the IST Work programme (Project no. 027023). The Know-Center is funded within the Austrian COMET Programme (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. We thank the editor and three anonymous reviewers for the constructive remarks on an earlier draft of this manuscript.

Appendix A. Reviewed list of tasks for Activity Modelling and System Goal Modelling

Activity Modelling

Gather data on human activity
1_1 Plan and prepare acquisition sessions, decide on acquisition methods

Analyse the functional structure on the work domain
1_2 Analyse the scope/validity domain of the new tool
1_3 Identify properties of the domain and the environment that constrain current and future performance
1_4 Identify properties of the system that constrain performance
1_5 Identify degrees of freedom allowing for that variability that makes the system flexible
1_6 Identify the system’s goals and high level functional goals
1_7 Identify external resources that are available for practitioners to achieve their goals

Identify tasks/goals to be achieved (independently of how they are to be achieved or by whom they are to be achieved)
1_8 Identify the workers’ prescribed goals as defined by norms and regulations
1_9 Identify the prescribed tasks to be executed to achieve the prescribed goals

Identify strategies how tasks can be achieved independently of who is executing them
1_10 Identify non-prescribed goals set up by the practitioners to achieve prescribed goals
1_11 Identify non-prescribed tasks that operators put in place to achieve goals while coping with internal and external constraints
1_12 Identify different strategies of the workers to acquire and process information
1_13 Identify information needs for the new tool
1_14 Identify inefficiencies of the existing system experienced by the users
1_15 Identify resources available to achieve goals
1_16 Analyse the different action sequences envisaged in certain situations
1_17 Identify and record inter- and intra-subjective variability to highlight work processes
1_18 Analyse contextual features that are sources of variability
1_19 Analyse the relevance of an aiding tool in relation to certain actions

Identify social organisation and co-operation between actors (both human and artificial)
1_20 Identify prescribed goals that are to be achieved by a team of workers
1_21 Identify prescribed tasks that are to be achieved by a team of workers
1_22 Identify different forms of communication and the exchange of information among workers
1_23 Analyse the mutual knowledge of teams or team members

Identify worker competencies such as knowledge, rules or skills that workers need to effectively accomplish their tasks
1_24 Identify semantic knowledge the workers need for effectively accomplishing the task

Model human activity
2_1 Identify the major types of activity in the current system the automation will affect
Model activities in a functional matter with respect to a purpose they serve
Break down actions in the normal course into their physical, cognitive and communicative components
Structure the particular activities into activity descriptions by using the Activity Modelling templates
Review the Activity Model

System Goal Modelling

Determine system boundaries
Use the findings of the Activity Model to identify system boundaries
Build a first-cut Context Model to identify system boundaries
Identify a basic list of stakeholders
Carry out an initial stakeholder analysis to determine the major categories of system stakeholder
Consider for each stakeholder whether he corresponds to an adjacent actor
Develop an extended Context Model to refine the first-cut model of system boundaries

Determine system dependencies, goals and rationale
Allocate functions between actors according to boundaries
Model the system's hard and soft goals
Interpret the Activity Model and integrate the identified actors and goals into the SD-Model
Identify the intentional strategic actors
Model dependencies between strategic actors for goals to be achieved and tasks to be performed
Model dependencies between strategic actors for availability or entity of resources
Write different forms of dependency descriptions

Refine system dependencies, goals and rationale
Refine the Strategic Dependency Model
Refine the Strategic Rationale Models
Produce a single, integrated SR model using dependencies in the SD model
Check that each individual SD model is complete and correct with stakeholder goals, soft-goals, tasks and resources
Validate the i-th SR model against the SD model by cross-checking the two models

Appendix B. Reviewed competence catalogue. knowledge and skills

Knowledge

1. Knowledge about acquisition techniques available
Knowledge about strengths and weaknesses of every technique
Knowledge about limits on use of each method
Knowledge about minimum conditions for method use
Knowledge about method interdependencies

2. Knowledge about how to organise an acquisition programme
Knowledge about guidelines for selecting from a broad range of different methods
Knowledge about practical constraints on each session

3. Knowledge about the Activity Model and the activity descriptions
Knowledge about the content of activity descriptions
Knowledge about the properties and the content of an Activity Model

4. Knowledge about actors, tasks, goals and resources
Knowledge about what is an actor, what is an 'intentional strategic actor'
Knowledge about different kinds of goals (prescribed, non-prescribed, local, soft-goal, ...)
Knowledge about what is a task
Knowledge about resources (means available to achieve goals and sub-goals)
Knowledge about the connection between actors, tasks and goals
Knowledge about the differentiation between these concepts

5. Knowledge about possible physical/environmental factors influencing human activity
Knowledge about preconditions of actions,
Knowledge about contextual features (distinctive features of the working context)
Knowledge about constraints (properties of the environment that need to be taken into account when deciding about an action)
Knowledge about the differentiation between these concepts

6. Basic understanding of human cognition, background in cognitive psychology
Knowledge about different types of knowledge (tacit, semi-tacit, non-tacit)
Knowledge about properties of non-tacit, semi-tacit and tacit knowledge
Basic understanding of selection and evaluation of information
Knowledge about individual and collective information processing strategies
Knowledge about individual and collective problem solving strategies
7. **Basic understanding of social and organisational behaviour**
   - Basic knowledge of social psychology or sociology
   - Knowledge about the appreciation of social networks
   - Knowledge about different forms of team cooperation

8. **Knowledge about the domain and the environment of the system**
   - Domain knowledge
   - Knowledge of the environment

9. **Knowledge of different types of system stakeholders**
   - Knowledge about the responsibilities and roles of different types of system stakeholders

10. **Basic technical understanding**
    - Knowledge about automation available
    - Knowledge about possible technical solutions
    - Knowledge about technical possibilities

11. **Knowledge about different types of adjacent systems**
    - Knowledge about the taxonomy of adjacent system types to consider
    - Knowledge about the properties of different types of adjacent systems

12. **Knowledge about the Context Model**
    - Ability to read and understand a Context Model
    - Knowledge about the properties and the content of a Context model
    - Knowledge about guidelines to develop an extended Context model

13. **Knowledge about the Strategic Dependency Model (SD-Model)**
    - Ability to read and understand an SD model
    - Knowledge about the properties and the content of an SD-Model
    - Knowledge of different types of the dependency link
    - Knowledge about guidelines for the different possibilities of wording for links between elements of the SD-Model

14. **Removed: knowledge about the different possibilities of notation of the elements of the SD-Model**

15. **Knowledge about the Strategic Rationale Model (SR-Model)**
    - Ability to read and understand an SR model
    - Knowledge about the properties and the content of an SR-Model

16. **Knowledge of validating the SR Model**
    - Knowledge of guidelines and techniques to cross-check the SD- and SR-Model

17. **Knowledge of brainstorming techniques**

18. **Knowledge of creativity techniques**

19. **Ability to produce a Context Model**

20. **Ability to produce an i* Model**

**Skills**

21. **Standard project management skills**
    - Ability of creating and maintaining an environment that guides a project to its successful completion
    - Understanding the procedures and methods that define a project while confronting and overcoming the problems encountered over the project lifespan.

22. **Listening skills**
    - Ability to carefully listen in a way that you do not prejudge information
    - Ability to clearly understand spoken, partly expressed, and unspoken messages from others (workers, stakeholders)
    - Understand the message being conveyed and identify the key ideas the speaker is sending by paying close attention to what is being communicated both verbally and non-verbally
    - Ability of giving full attention to what other people are saying, taking time to understand the points being made, asking questions as appropriate, and not interrupting at inappropriate times

23. **Communication skills**
    - Ability of sending clear and convincing messages
    - Ability to clearly articulate your ideas
24. Collaboration skills and empathy
   Ability to balance a focus on tasks with attention to relationships
   Ability of sensing others’ feelings and perspective and taking an active interest in their concerns

25. Writing skills
   Ability to write well
   Ability of non-biased writing
   Ability to relay a message or represent one’s thoughts in written format with clarity and accuracy

26. Learning skills
   Ability to acquire and integrate new information by gathering data
   Ability to understand the implications of new information for both current and future problem solving and decision making

27. On-line data gathering skills
   Skills of collecting data while practitioners are working with the system
   Ability to make non-biased observations
   Knowledge of on-line data gathering techniques such as protocol analysis, ethnography, …

28. Off-line data-gathering skills
   Skills of eliciting information from practitioners while they are not involved in any activity structured and unstructured interviewing skills, non-prejudging interviewing, asking the right questions
   Knowledge of direct elicitation techniques, off-line data gathering techniques such as card sorting, laddering, …

29. Judgement and decision making
   Ability to consider the relative costs and benefits of potential actions to choose the most appropriate one

---

<table>
<thead>
<tr>
<th>Task</th>
<th>Competency</th>
<th>Rating</th>
</tr>
</thead>
<tbody>
<tr>
<td>24.5</td>
<td>Ability to balance focus on tasks</td>
<td>1.0</td>
</tr>
<tr>
<td>24.6</td>
<td>Ability of sensing others’ feelings</td>
<td>1.0</td>
</tr>
<tr>
<td>25.1</td>
<td>Ability to write well</td>
<td>1.0</td>
</tr>
<tr>
<td>25.2</td>
<td>Ability of non-biased writing</td>
<td>1.0</td>
</tr>
<tr>
<td>26.1</td>
<td>Ability to acquire and integrate new information</td>
<td>1.0</td>
</tr>
<tr>
<td>26.2</td>
<td>Ability to understand implications of new information</td>
<td>1.0</td>
</tr>
</tbody>
</table>

Fig. 6. Task–competency matrix of Expert 1 (Extract).
Ability of using logic and reasoning to identify the strengths and weaknesses of alternative solutions, conclusions or approaches to problems
Ability to make effective decisions quickly, based on a careful and balanced consideration of all available facts
Ability to identify complex problems and review related information to develop and evaluate options and implement solutions

30. Ability of thinking in processes and analysing processes
Ability to regard a particular course of action as intended to achieve a goal or result
Ability of analysing actors and concepts with regard to their mutual dependencies

31. Ability of structuring data and organising information
Ability to arrange things or actions in a certain order or pattern according to a specific rule or set of rules
Ability to organise the material you collect

32. Ability of functional decomposition
Ability to break down concepts (goals, tasks, ...) into sub-elements
Ability to decompose processes into sub-elements

33. Analytical skills
Ability to provide a logical, in-depth analysis of a problem or situation
Ability to decompose and structure the problem space
Ability to examine the situation in a systematic way
Ability of filtering out the important information
Ability to extract the most important thing of what somebody says
Ability to select and understand the most important facets of some peace of work

34. Ability of abstraction
Ability to abstract from literal objects or instances and to extract concepts such as actors, goals, etc. from them
Ability to understand a general rule of examples people are telling you


APOSDELE Consortium, 2006. Use Cases and Application Requirements 1 (First Prototype), Project Deliverable D6.2, Consulted March 4, 2008 ⟨http://www.aapodele.org/media/multimedia/files/use_cases_applica tion_requirements_1_first_prototype⟩.


Desmarais, M., Gagnon, M., 2006. Bayesian Student Models Based on Item to Item Knowledge Structures. In: Neidl, W., Tochtermann, K., (Eds.), Proceedings of the 1st European Conference on Technology-