View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by D-Scholarship@Pitt

Domain Modeling for Personalized Guidance

Peter Brusilovsky
University of Pittsburgh

INTRODUCTION: DEFINING THE PROBLEM OF PERSONALIZED GUIDANCE

This chapter attempts to untangle the relationships between personalized guidance and domain modeling, as well as to explain how domain modeling could be used to provide personalized guidance. The problem of personalized guidance has a long history in the area of adaptive educational systems (AES). In fact, the very first recognized AES SCHOLAR (Carbonell, 1970) focused on guiding students to the most relevant facts and questions about the geography of South America. The SCHOLAR functionality was based on a domain model in the form of a semantic network and an overlay student model. Since that time, a considerable share of research in the field of AES has focused on different kinds of personalized guidance, and the majority of this work relied heavily on domain modeling—which makes these two research directions heavily interconnected.

In this chapter, personalized guidance is defined as any approach used to guide learners to the most appropriate learning content[1] that accounts for any unique features of the learner, including current knowledge level, interest, and motivation, among others. Selecting the learning activity to work on is known as the *outer loop* problem in the field of AES (VanLehn, 2006) (with the inner loop referring to assisting students *inside* the selected activity). The goal of personalized guidance in the outer loop is to provide the best educational experience for every learner by guiding the learner through the optimal sequence of learning activities that lead to the best educational outcome, such as faster learning, more reliable knowledge, or a more enjoyable process.

Early attempts to create an adaptive learning process were made in the field of classic computer-adaptive instruction (CAI) where different answers on a multiple-choice test could lead to transferring the learner to different content elements. However, this approach has not been considered to be truly adaptive, since decisions were made on the basis of the last answer, rather than on the full picture of student knowledge. To do better—namely, to make decisions on the whole state of a student's knowledge—a personalized system needs more advanced infrastructure. It needs to understand how student knowledge can be represented, how student interactions with learning content could be used to update this knowledge, how missing knowledge could be identified, and how it would be possible to learn content that can help with filling a particular knowledge gap. As demonstrated by many AES starting from SCHOLAR,, a domain model can assist with performing these functions.

The scope of domain modeling is broader than the needs of personalized guidance. A domain model (DM) in the AES field is traditionally understood to be any formal representation of knowledge about the domain. This model can serve various needs, not just personalized guidance. However, due to the special needs of personalized guidance, DMs used for personalized guidance have some common features, which enable different types of personalization, some of which will be reviewed in this paper. To examine the specific needs of personalized guidance, we need to understand how educational content is organized and which decisions have to be made to guide students to the most appropriate elements of this content.

Early AES studies focused on a relatively narrow domain (usually a small part of a course) and have considerably low volumes of content (Brusilovsky, 1992; McArthur, Stasz, Hotta, Peter, & Burdorf, 1988; Wescourt, Beard, & Gould, 1977). In these AES studies, content has no additional structure; in other words, it was an unstructured pool (usually a pool or problems), and the goal of adaptive sequencing, which is the oldest kind of personal guidance, was to select the best item from this pool to offer to the

[1] In a more general sense, we should talk about learning as a sequence of learning activities or tasks, not just content; however, in modern e-learning, these activities are triggered by interacting with specific types of content—problems, examples, animations, collaborative tasks, and so forth.

student. Nowadays, however, many educational systems cover broader domains (such as whole-semester or year-long courses) and have much larger volumes of content. To better manage this content, the domain is usually subdivided into several *topics* (sometimes also called units), and every learning content item is assigned to one of these topics. The topics could be relatively independent from each other as in an *inq-its.org* learning environment (Sao Pedro, Baker, & Gobert, 2013); form a fixed pedagogical sequence, as in Deep Tutor (Rus et al., 2013); or form a partial order with some topics depending on some other earlier topics, as in QuizPACK (Sosnovsky & Brusilovsky, 2015). In addition, modern systems routinely operate with different kinds of learning content, such as book-style presentations, video fragments, animated examples, simple questions, and problems, among others (Figure 1). Some good examples of this two-level "topic-content" organization are college courses with lectures structured by topics, books organized into chapters, and folder structures in learning management systems (LMSs). Since this organization is most typical, we use it to discuss decisions that have to be made by personalized guidance in this general context. While content organization might be more complex in some cases (such as a hierarchy of topics, book subsections, or LMS subfolders), a two-level organization of "topics-content" is still sufficient to analyze key decisions. On the other hand, it is easy to see that earlier simpler organization with a single pool and one type of learning content is just a special simple case of the two-level organization that includes only a single topic.

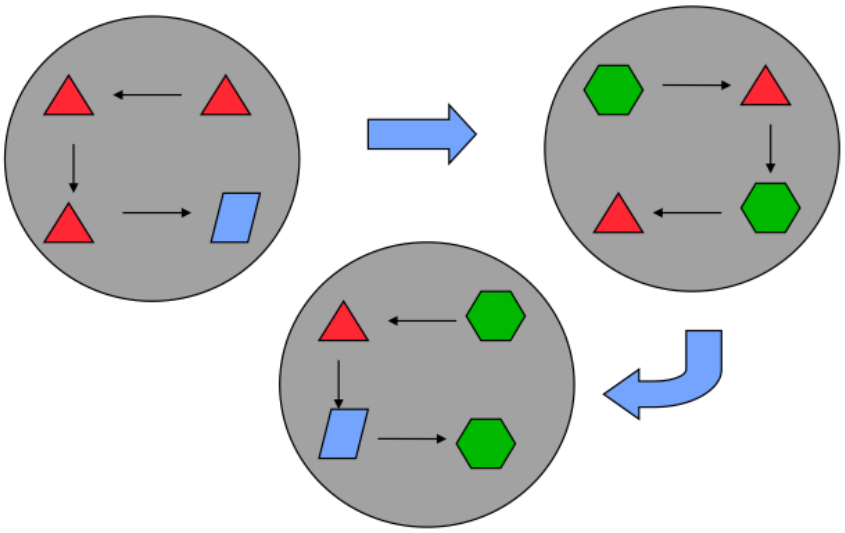


Figure 1: Two level "topic-content" organization of learning content in modern educational systems

Within this two-level organization, the decisions about continuing the educational process must usually be made on two levels. On each level, someone should decide (1) what to do next, and (2) when to stop. On the course level, it means selecting the next best topic to study and deciding when the learning process is over (everything has been learned). On the topic level, it means selecting the next content item and deciding when the topic has been mastered. Each of these decisions could be made adaptively with an understanding of the whole picture of student knowledge, but this is usually not the case even in AES. In general, there are four distinctive alternatives for making each of the four decisions (two decisions on each of two levels):

- The decisions could be made in advance by the system developer or instructor. A classic example of it is programmed learning. A more current example of this model is the developer-fixed order of topics and tasks in many intelligent tutoring systems (ITSs).
- The decisions could be left for students to make, usually by selecting a link to the desired topic or content. Hypertext-based learning introduced as an alternative to programmed learning

(Hammond, 1989) provides a classic example of this approach, while modern LMSs and learning content repositories (Cafolla, 2006) that provide unrestricted access to all learning content offer a modern example.

- An adaptive algorithm could make the decision for the student. This approach was introduced in the very first AES SCHOLAR (Carbonell, 1970) and has become one of the signature technologies for AES and ITS in the form of content or topic sequencing, as mentioned above (Brusilovsky, 1992; McArthur et al., 1988; Wescourt et al., 1977), mastery learning (Corbett & Anderson, 1995), and adaptive course generation (Diessel, Lehmann, & Vassileva, 1994).
- An adaptive system can *assist* the student in making the decision. The classic example of this combined approach is adaptive navigation support (Brusilovsky, 2007) in educational adaptive hypermedia, which used intelligent techniques to annotate, re-order, or remove links to content and topics, while ultimately leaving the final choice to the student. A more recent example is provided by educational recommender systems (Manouselis, Drachsler, Verbert, & Duval, 2013) that generate a ranked list of the most useful learning resources for the student to choose from.

The last two cases present two different approaches to *personalized* guidance. To be considered as adaptive, an educational system should use personalized guidance to make at least one of the four decisions; however, the remaining decisions could still be left to the instructor or students. Historically, researchers have explored almost the whole range of possible combinations of fixed, free, and guided choices on two levels. For example, the well-known *mastery learning* approach (Corbett & Anderson, 1995) popular in ITS usually has a fixed order of topics, as well as a fixed order of problems within a topic; however, a decision on when to stop working on a topic is adaptively made on the basis of student knowledge modeling. Topic-based adaptive navigation support approaches (Sosnovsky & Brusilovsky, 2015) provide personalized guidance to help students in selecting a topic to work on, while leaving the choice of content items within topic to the students. In contrast, a system like Problets (Kumar, 2006) can leave the choice of topic to work on to the student, but once the topic is selected, can provide an adaptive sequence of exercises. An adaptive testing system like SIETTE (Conejo, Guzman, & Millán, 2004) can leave the choice of topics to test to the instructor while simultaneously generating an adaptive sequence of questions within these topics. Finally, even if an adaptive system offers personalized guidance on both levels, it might use different approaches on different levels. For example, the INSPIRE system (Papanikolaou, Grigoriadou, Kornilakis, & Magoulas, 2003) uses knowledge-based personalization to provide personalized guidance on the topic level while learning style-based personalization for content selection within a topic. In most of the cases mentioned above, the use of domain models is critical for providing personalized guidance; however, depending on the level and the kind of decisions to be made, AES could use considerably different domain models and personalization approaches. The next section reviews several types of domain models that are used for personalized guidance, and explains how student models and content models should be built on the top of these models to support different kinds of guidance.

RELATED RESEARCH: DOMAIN, STUDENT, AND CONTENT MODELING

Domain Modeling

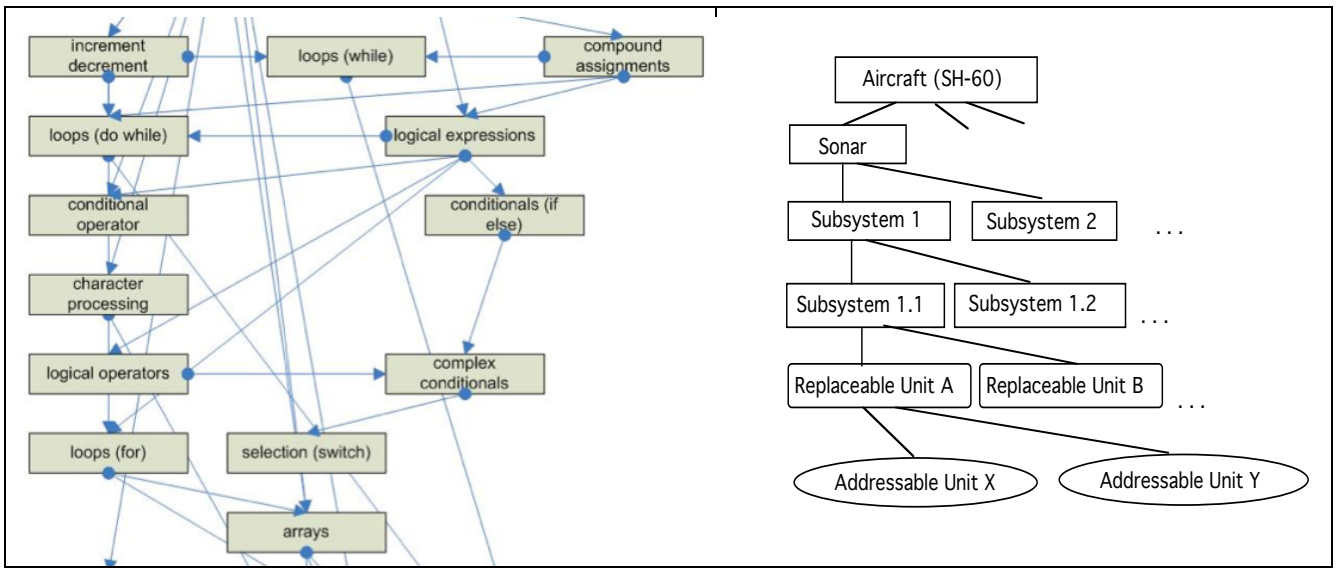
Domain models can be classified in several different ways. In this chapter, we follow Sleeman (1985), who suggested classifying models by the nature and form of information contained in the model, as well as the methods of working with it. Following this suggestion, we analyze domain models along three dimensions: what is being modeled (nature); how this information is represented (structure); and how different kinds of models are used for personalized guidance (usage).

The *nature* of domain models is primarily defined by the kind of domain knowledge (which is sometimes also called expert knowledge) represented in the model, which in turn, might depend on the type of personalization that these models have to support. The majority of AESs has focused on representing two

types of domain knowledge: conceptual knowledge (or *about* knowledge) and procedural knowledge (or *how* knowledge). Both kinds of knowledge have been used since the early days of AESs. For example, SCHOLAR (Carbonell, 1970) represented conceptual knowledge about South America in the form of a network of concepts, while Lisp-Tutor (Anderson, Farrell, & Sauers, 1984) represented procedural problem-solving knowledge of LISP as a set of problem-solving rules. More recently, Stellan Ohlsson has suggested modeling a different kind of procedural knowledge: not the knowledge that allows the user to solve the problem, but the knowledge that allows the user to evaluate the correctness of the solution (Ohlsson, 1992). This knowledge is typically represented as a set of constraints as in SQL-Tutor (Mitrovic, 2003).

The nature of knowledge represented in a specific AES is usually determined by the kind of personalized support that it provides. AESs that focus on helping users to solve educational problems usually rely on procedural problem solving knowledge. Systems that focus on assessing student problem solutions prefer to use constraints. AESs that focus on presenting declarative educational content (such as textbook or manual pages) and assessing knowledge of them usually rely on conceptual knowledge about the domain.

In addition, the nature of represented knowledge depends on the *granularity* of representation. The majority of domain models represent domain knowledge in a structural form. A structured domain model is composed of a set of domain knowledge components (KCs). Each KC represents a fragment of knowledge for the given domain. KCs could represent knowledge fragments of different types and granularity; see Koedinger, Corbett, and Perfetti (2012) for an excellent review of various KC types. The type of knowledge represented by a KC is defined by the nature of the domain model (conceptual, problem-solving, evaluation), while the granularity is typically defined by domain experts who decide the level of detail that a domain knowledge representation should have. In different systems, the same kind of knowledge could be represented at different levels of granularity. For example, the KC of procedural knowledge could represent fine-grained rules or coarse-grained competencies, while the KC of conceptual knowledge can represent fine-grained concepts or coarse-grained topics. As with many other design decisions, the granularity of knowledge representation depends on the intended use for such domain models. In particular, advanced problem-solving feedback that is usually critical to support problem-solving usually requires finer-grained representation; in contrast, many kinds of personalized guidance can work successfully with relatively coarse-grained models.



(a) (b)

Figure 2: Network domain models: (a) A fragment of domain model for C programming with prerequisite links (Sosnovsky & Brusilovsky, 2015); (b) Domain model for SH-60 IETM with part-of links (Brusilovsky & Cooper, 2002).

By their *structure*, domain models may be divided into *set models* and *network models*. Set models, also called *vector models* (Brusilovsky, 2003), are formed by sets of independent KCs that have no internal structure. In more advanced network domain models, KCs are connected to each other by different kinds of relationships that form a semantic network. This network represents the structure of the domain that is covered by an AES. The most popular kinds of links in AESs are *prerequisite links* between the KCs. A prerequisite link represents the fact that one of the related KCs has to be learned before another. Prerequisite links can support several personalized guidance approaches. In many AES systems, prerequisite links are the only kind of links between KCs (Davidovic, Warren, & Trichina, 2003; Farrell et al., 2003; Henze & Nejdl, 2001; Papanikolaou et al., 2003). For example, the QuizGuide domain model is a network of Java programming topics that are connected by prerequisite links (Figure 2a). Other types of popular links are the classic ontological links "is-a" and "part-of" (Brusilovsky & Cooper, 2002; De Bra, Aerts, & Rousseau, 2002; Hoog et al., 2002; Sangineto, Capuano, Gaeta, & Micarelli, 2008; Steinacker et al., 2001; Trella, Conejo, & Bueno, 2002; Vassileva, 1998). An example of the ontological *part-of* concept hierarchy for the SH-60 helicopter IETM is shown in Figure 2b, and an example of *is-a* taxonomy of educational objectives and concepts for a Java *problet* (Kumar, 2006) is shown in Figure 3. The popularity of these links have gradually increased, following the progress of Semantic Web research and the increased use of more formal ontologies as domain models (Dagger, Wade, & Conlan, 2004; Mitrovic & Devedzic, 2004; Sangineto et al., 2008; Sosnovsky & Dicheva, 2010; Trausan-Matu, Maraschi, & Cerri, 2002).

Domain Models for Student and Content Modeling

Structured domain models are critical for personalized guidance, because they provide a bridge between individual student models that represent the current state of knowledge of a particular learner and content that could be recommended for that learner. In other words, both the knowledge (or interest) of each learner and the units of learning content are described in terms of the domain model. The presence of this “common language” allows for tracing a specific lack or knowledge or presence of misconceptions to the learning content that could directly address this problem.

One of the most important functions of the domain model is to provide a framework for long-term modeling and representation of the learner's domain knowledge. Long-term modeling is performed by following the student in the learning process, observing her actions and learning progress, and using these observations to maintain a dynamic model of student knowledge about the domain.

The majority of AES use an overlay model of student domain knowledge, which is also known as an overlay *student model* (Brusilovsky & Millán, 2007). The key principle of the overlay model is that for each domain model KC, an individual user knowledge model stores some data that is an estimation of the user knowledge level on this KC. In its simplest form, it is a binary value (known—not known) that enables the model to represent a user's knowledge as an overlay of domain knowledge. This form was popular in early ITSs; for example, it was the type of model used by the SCHOLAR system to store student knowledge about South America. Today, almost all adaptive systems use a *weighted* overlay model that can distinguish several levels of knowledge for each KC by using a qualitative value (Grigoriadou, Papanikolaou, Kornilakis, & Magoulas, 2001) (i.e., good-average-poor), an integer quantitative value (for example, from 0 to 100) (Brusilovsky, Eklund, & Schwarz, 1998; De Bra & Calvi, 1998), or a probability that the user knows the concept (Corbett & Anderson, 1995; Conati, Gertner, & Vanlehn,, 2002; Henze & Nejdl, 2001). A weighted overlay model of user knowledge can be represented as a set of "concept–value" pairs, one pair for each KC. The overlay model is powerful and flexible because it can independently measure the learner's knowledge of different KCs.

The domain model also provides a basis for describing the learning content in terms of domain knowledge. To enable personalized guidance, every fragment of learning content should be connected with one or more domain KCs that are presented or assessed by that fragment. These connections define the organization of learning content and its relationships to the domain knowledge. From the prospect of

personalization, there are two principal approaches to structure content in respect to the domain model, which are usually called topic-based and concept-based organization.

Topic-based organization assumes that each fragment of educational content is related to one and only one domain model KC. In most cases, this is a relatively coarse-grained KC that corresponds to a relatively large course topic (which is why it is usually called topic-based). Topic-based organization is simpler and more intuitive for both authors and users. It is also easier from the prospect of student modeling: both successes and failures when working with content can be clearly attributed to a single connected topic. At the same time, this model is relatively weak from the prospect of personalization, as will be explained below. A simple example of topic-based organization is the adaptive testing system SIETTE (Conejo et al., 2004), where the domain model is represented as a hierarchy of topics and the content space is formed by a large set of questions. Each SIETTE question here is associated with exactly one of the topics (Figure 4). Problets (Kumar, 2006) offer a similar example of topic-based organization: its KCs form a hierarchy (Figure 3) and each problem offered by the system is associated with exactly one lower-level educational objective. QuizGuide (Sosnovsky & Brusilovsky, 2015) with its prerequisite-based network of topics (Figure 2a) also uses topic-based organization, where each domain topic has several quizzes. Due to its transparency, topic-based content organization has been used as an organizing principle in several authoring frameworks for personalized systems (Specht & Oppermann, 1998; Vassileva & Deters, 1998; Weber, Kuhl, & Weibelzahl, 2002).

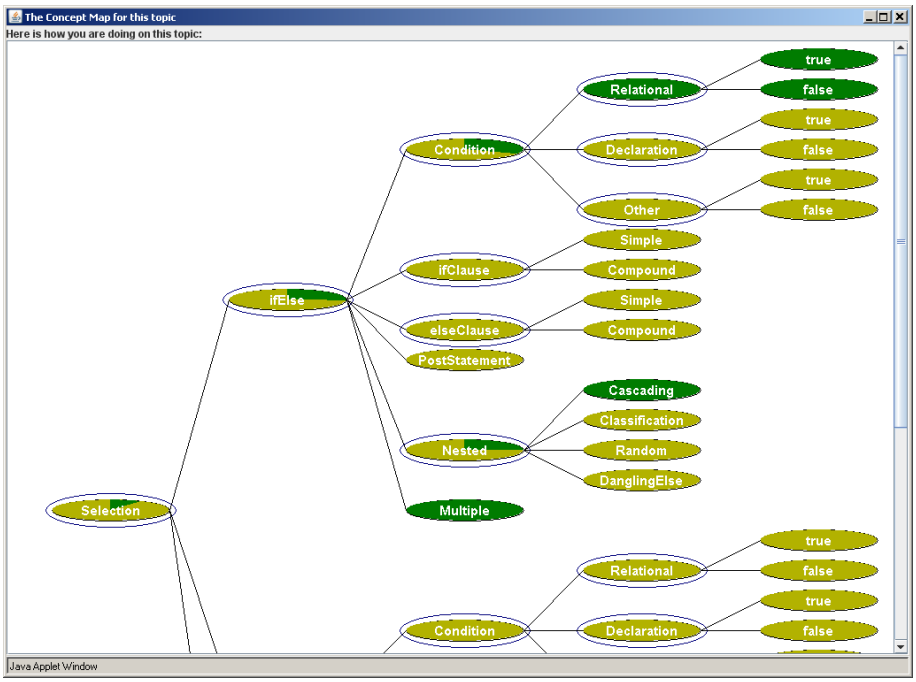


Figure 3**:** A part of the domain hierarchy for practicing *if/if-else* statements in Java (Kumar, 2006). The hierarchy is formed by learning objectives and concepts, connected by is-a relationships.

Concept-based organization is a more general case, and is used when each fragment of content is related to several KCs. In most cases, these are finer-grained KCs that are frequently referred as "concepts". Concept-based organization is more powerful from the prospect of personalization, but it is more difficult to visualize. It also makes the process of content connection to the domain model more complex and requires a more experienced authoring team. In many cases, however, the nature of the domain demands concept-based content organization. For example, in programming and mathematics, elementary constructs and operators are often selected as domain model concepts or rules. As a result, almost any meaningful problem or example is associated with several KCs. Even in a relatively small domain, such as the Simplex

algorithm (Millán, Loboda, & Pérez de la Cruz, 2010), this organization might create a complex domain structure, such as the network of skills and problems shown in Figure 5.

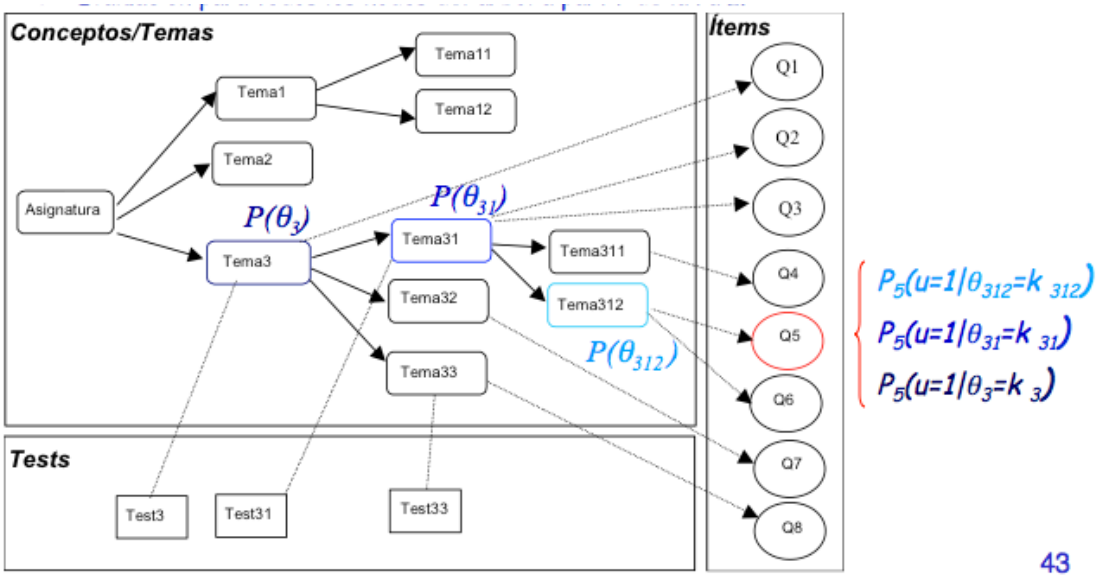


Figure 4: Content organization in SIETTE. Each question is associated to one of the topics in the hierarchy of topics.

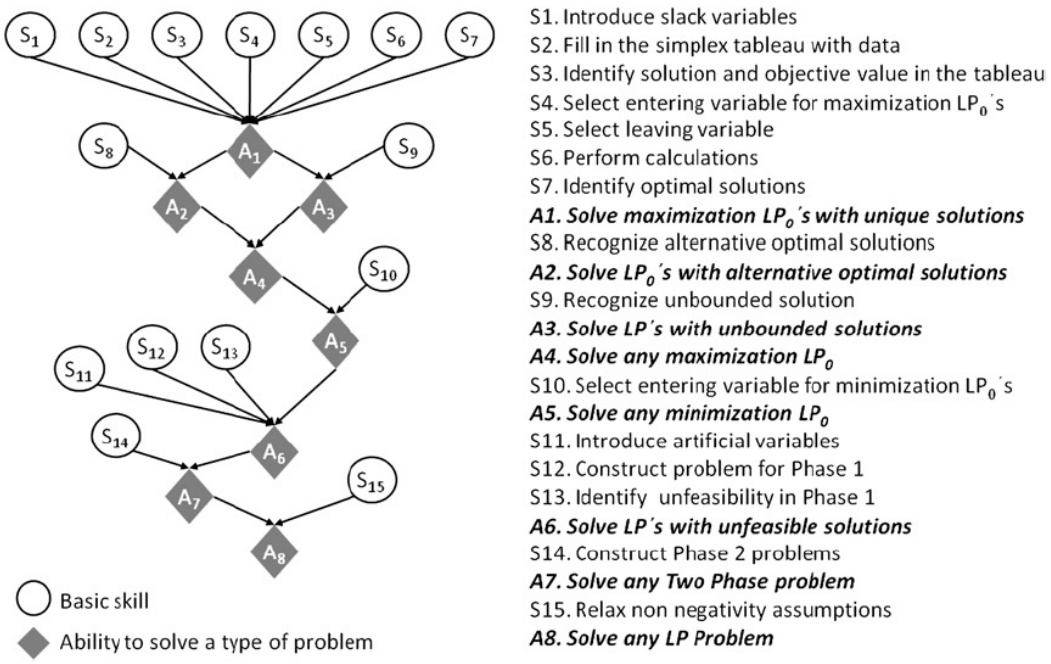


Figure 5: Domain model and content organization in an ITS for the Simplex algorithm (Millán et al., 2010)

From the perspective of student modeling, a concept-based approach presents the problem of *blame allocation*: in case of student failure to solve a problem that is connected to several KCs, as in Figure 5, a student modeling approach needs to decide how to attribute this problem to the lack of knowledge in different associated KCs. To resolve this problem, ITSs use step-by-step problem solving support with

model tracing (Anderson, Boyle, Corbett, & Lewis, 1990), where student performance on each step is attributed to a single domain model KC. If step-by-step tracing is not possible, reliable modeling might require complex Bayesian networks (Millán et al., 2010) for blame allocation, or apply advanced solution analysis that can reliably recognize correctly and incorrectly used KCs in student solutions (Weber & Brusilovsky, 2001). An example of a problem-based system with concept-based indexing is SQL-Tutor (Mitrovic, 2003), where each SQL problem is associated with multiple domain *constraints*. Advanced diagnostic capabilities allow this system to recognize which constraints were satisfied and which were not satisfied in each student solution.

Some systems with concept-based content organization and a large content space use different types of links to connect a fragment of content with domain concepts. For example, the adaptive hypermedia system InterBook (Brusilovsky et al., 1998) distinguishes *outcome* concepts for a content page (those explained by this page) and *background* concepts (those not presented on a page, but that are required for understanding it). This organization supports both complex hyperspace and several types of personalized guidance (Figure 6).

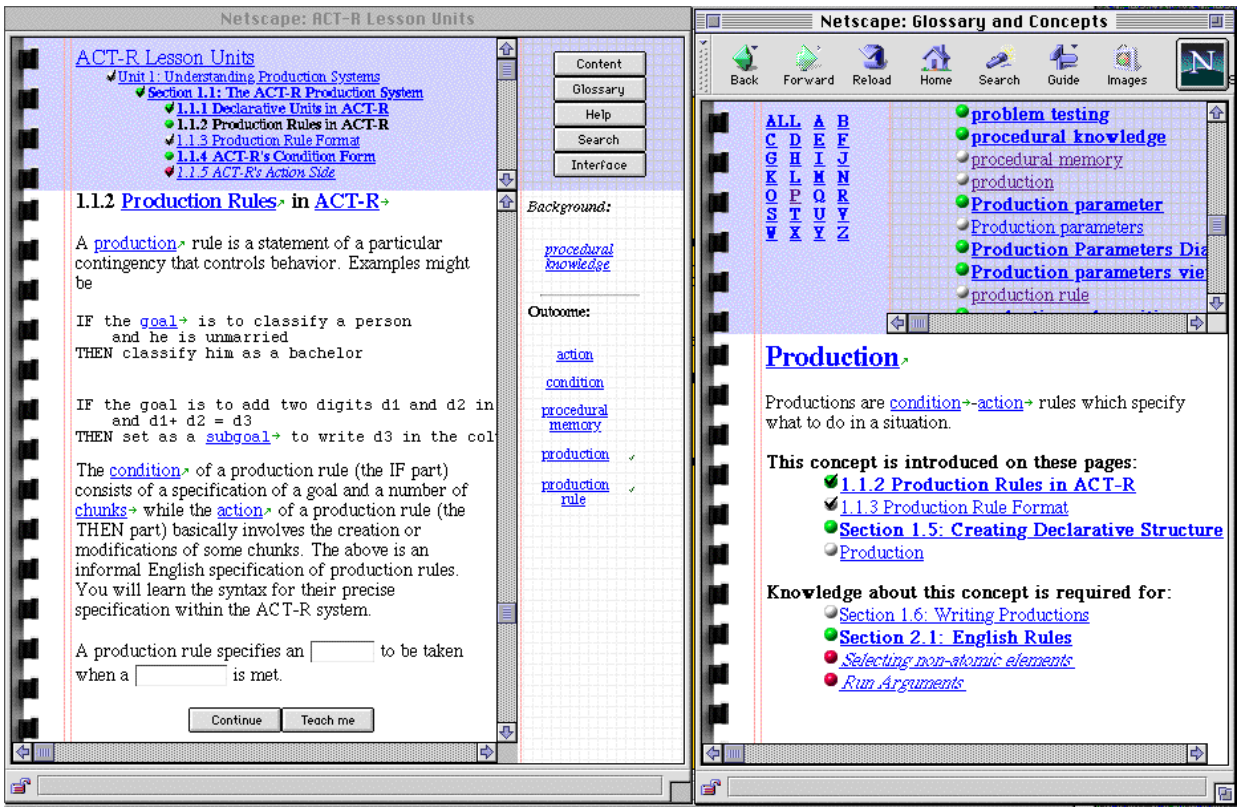


Figure 6: InterBook content organization: Each content page (left) is connected to several domain concepts that could be either background or outcome concepts. Each concept (right) is connected to multiple content pages as outcome or background.

PERSONALIZED GUIDANCE BASED ON DOMAIN MODELS

As explained in the in the introduction, personalized guidance in a general case can be provided on two levels: helping students to select the most appropriate course topic and helping to select next educational content fragment that will engage the user into the most appropriate activity. It has been also mentioned that on each level the decision could be prescribed by the instructor or course author, left to the student, or

offered by the personalized guidance mechanism with or without student input. The ability to support personalization on each level and the kind of personalization support that the system can provide, to a large extent is determined by the nature of the domain model and content organization in the system. This section briefly reviews most typical cases of personalized guidance and explains how the domain models could support it.

Topic-based Personalized Guidance

Topic-based organization of a course is frequently used as a basis for domain modeling. In other words, a topic in the domain model corresponds to a course topic. This could be considered to be a relatively simple and coarse grain domain model; however, it could be still used to guide users to the most appropriate topic. An example of topic-based guidance is provided by the QuizGuide system (Sosnovsky & Brusilovsky, 2015). The QuizGuide domain model is a network of course topics that are connected by prerequisite relationships (Figure 2a). For each topic, the overlay student model stores the achieved level of student knowledge. C programming quizzes form the content, with each quiz belonging to one of the topics. QuizGuide provides personalized topic-level guidance by annotation-based adaptive navigation support. Each topic is annotated with a "target" icon that expresses two important aspects. The color of the target shows how timely the work on this topic is in the current course context (Figure 7). Recommended topics are bright blue, their immediate prerequisites are light blue, older topics are grey, and forthcoming topics for which the student might not yet be ready are crossed out. The number of arrows in the target shows the estimated level of knowledge: no arrows mean little to no knowledge, while three arrows indicate good knowledge. These icons could be easily generated using the current state of the student model and the topic-based domain model. This approach does not explicitly tell the student what to do next, but provides personalized guidance for selecting the right topic, depending on student intentions. For example, a student who is interested in advancing through the course should focus on current, but not yet learned topics, while a student who wants to prepare for an exam could work with content from current and immediate past topics that are not yet fully mastered. While this approach is relatively easy to organize, it is surprisingly efficient (Sosnovsky & Brusilovsky, 2015). Simpler versions of topic-based guidance are now used in several practical systems, such as Khan Academy (https://www.khanacademy.org/).

The problem of coarse-grain topic-based guidance is its weakness in guiding users *within* a topic. The first reason is that the quality of student modeling provided by traditional modeling approaches, such as Bayesian knowledge tracing (Corbett & Anderson, 1995) over coarse-grained topics is relatively low, as shown in Sosnovsky and Brusilovsky(2015). It is not critical for navigation support when students are engaged in the decision of what to do next, but is important for all cases where a decision is made for the student. In particular, mastery learning over coarse topics (deciding when the topic is sufficiently learned in order to stop working with in-topic content) is technically feasible, given that knowledge modeling is done on the topic level, but might not be very reliable. The second reason is the inability to apply a topic-level model of student knowledge to distinguish and recommend a specific content within a topic. Indeed, from the prospects of topic-based content organization, all content items that belong to the same topic are equal: they allow users to practice the same knowledge and are equally ready to be applied. As a result, systems with coarse-grain topic-based models have to use additional information to guide users to a specific content within the selected topic. On the modeling side, it calls for more advanced approaches, such as FAST, that could use various additional features (González-Brenes, Huang, & Brusilovsky, 2014). On the navigation support and sequencing side, these models demand richer content-based or usage-based knowledge about every piece of content. For example, INSPIRE (Papanikolaou et al., 2003) uses coarse-grain topic-based navigation support to guide users to the most appropriate topics. The support is provided in the form of adaptive annotation in the form of an empty, full, or partially filled glass next to the topic link, and reflects the current level of knowledge on the topic (Figure 8). In addition, it classifies fragments that belong to the same topic into different content types and levels and uses information about the current level of knowledge on the topic and the student's learning style to offer the most relevant content for the selected topic.

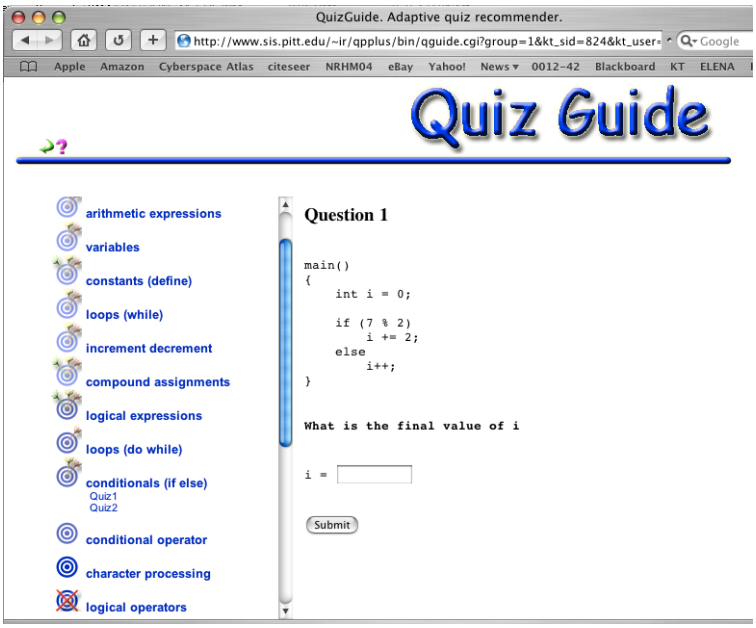


Figure 7: Topic-based adaptive navigation support in QuizGuide with dynamic "target" icons

Systems with finer-grain topic-based guidance usually have their topics organized in a topic-subtopic hierarchy, just like domain models in Problets (Figure 3) and SIETTE (Figure 4). In this organization, the upper levels of this hierarchy typically correspond to the main course topics, while lower levels define a tree of subtopics within topics in a course. This finer-grain organization supports some reasonable approaches to content sequencing within topics. Since student knowledge is now independently tracked for each low-level topic, sequencing algorithms can distinguish content items associated with different subtopics and may have a reason to prefer one or another, depending on the state of the student model. Even reasonably simple sequencing algorithms based on ensuring subtopic coverage (Kumar, 2006) can now provide useful personalized guidance. More sophisticated IRT-based approaches, like the one used in SIETTE (Conejo et al., 2004) can produce highly personalized sequences of content items. Similarly, a finer-grain model could serve as a basis for more precise mastery learning. The weak side of finer-grain topic-based approaches is the complexity of the domain model, which makes it harder to visualize it as a whole to offer adaptive navigation support (in topic-based organization, each content fragment is related to exactly one topic; as a result, navigation support has to be organized on the level of the topics). While coarse-grained flat domain models, such as those used in INSPIRE (Papanikolaou et al., 2003), QuizGuide (Sosnovsky & Brusilovsky, 2015), and MasteryGrids (Loboda, Guerra, Hosseini, & Brusilovsky, 2014) have straightforward linear visualization, a deep fine-grain hierarchy is usually too large to visualize. One solution to this problem is to visualize only a fragment of the whole model for each top-level topic, as shown in Figure 3, or use a zoomable hierarchical domain model to provide navigation support, such as a zoomable treemap used in the KnowledgeZoom system (Brusilovsky, Baishya, Hosseini, Guerra, & Liang, 2013).

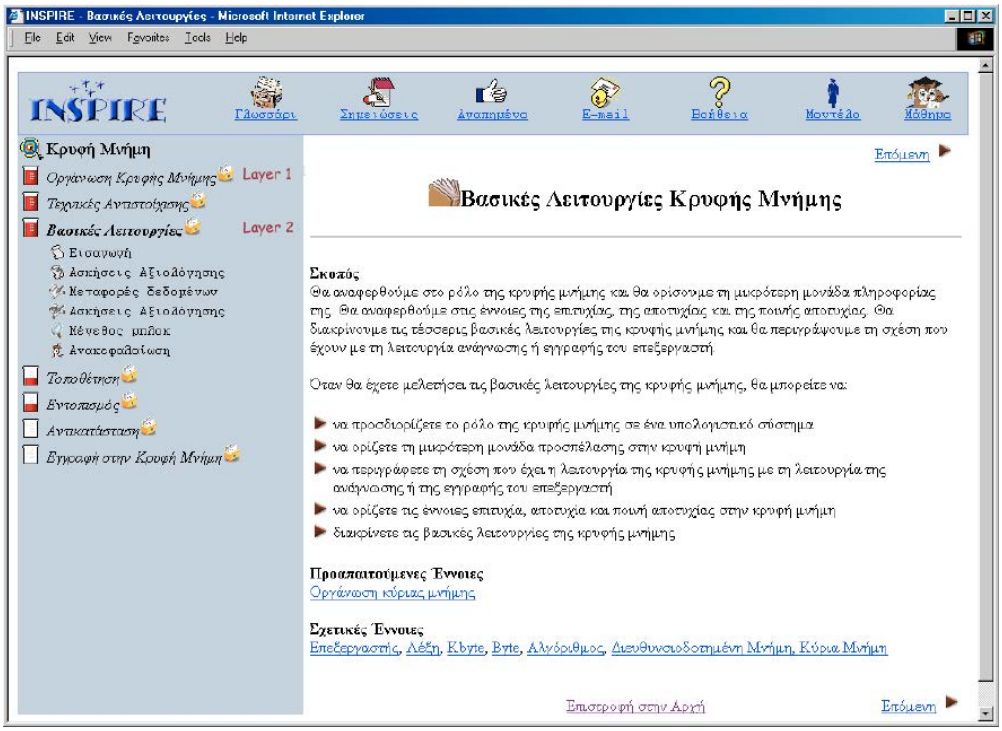


Figure 8: Topic-based content organization in INSPIRE (Papanikolaou et al., 2003). Each concept (annotated with a glass icon on the right) includes a number of content fragments of different types that are selected for the user on the basis of current topic knowledge and learning style.

Concept-based Personalized Guidance

Concept-based personalized guidance must work on the top of a more sophisticated content organization, where every content item could be related to many domain model KCs (such as concepts or rules, among others). This context introduces opportunities for sophisticated guidance approaches. When deciding whether to recommend a problem or another content item to a student, a guidance algorithm usually needs to balance two aspects: whether the item is useful (namely, does it introduce missing or insufficiently learned concepts) and whether its difficulty is appropriate (namely, to what extent the student is familiar with concepts that are required to understand the content or solve the problem). In the case of topic-based guidance, these decisions are simply made by examining the current knowledge level of the topic to which the content item belongs, as well as prerequisites for this topic. In the more advanced case of concept-based guidance, knowledge of all concepts related to the item should be considered when introducing a large number of possible situations. The situation becomes even more complicated if a content item has separate sets of *prerequisite* concepts (which are required to work with the item) and *outcome* concepts, which are practiced while working with the item (Bieliková et al., 2014; Brusilovsky et al., 1998; De Bra et al., 2003; De Bra et al., 2013) as shown in Figure 6. In this case, a personalized guidance algorithm should separately assess the knowledge of prerequisite and outcome concepts to balance both the readiness and the usefulness of each item.

The classic approach to provide personalized guidance in domains with concept-based content organization is sequencing; namely, the selection of the generation of the single “next best” activity. A range of sequencing approaches were suggested and explored relatively early in domains with complex problems, such as programming and mathematics (Barr, Beard, & Atkinson, 1976; Brecht, McCalla, & Greer, 1989; Brusilovsky, 1992; McArthur et al., 1988; Wescourt et al., 1977). These approaches usually apply some type of scoring function to determine the “goodness” of each candidate item, and then select

the item with the best score. However, due to the large number of decisions to be made using relatively noisy student models and imperfect knowledge engineering, the quality of concept-based sequencing has never been sufficiently good. As a result, sequencing has gradually been overshadowed by safer approaches. In problem-oriented ITS, it was replaced by more reliable mastery learning, where all content is carefully sequenced in advance and personalized guidance is only used to make a more reliable decision that a topic is mastered. In AES with more diverse content, sequencing was overtaken by adaptive navigation support and recommendation. As mentioned above, adaptive navigation support modifies existing links in a hyperspace of learning content using link annotation, ranking, or removal. For example, the popular "traffic light" link annotation approach (De Bra & Calvi, 1998; Eklund & Brusilovsky, 1998; Henze & Nejdl, 2001; Weber & Brusilovsky, 2001) marks links to content that is not yet ready to be learned with red bullets, content that is not useful anymore with white bullets, and content that is both ready and useful with green bullets (Figure 6). Content recommendation creates a new ranked list of recommended content items (Figure 9). In both cases, annotation or ranking decisions are usually based on content scoring functions that are similar to those used in classic sequencing. However, both, navigation support and recommendation approaches allow the student to choose from several opportunities, while explaining which of them are considered as good and why. This introduces "a human in the loop," correcting possible mistakes that could be made when a sequencing approach bets on one presumably best option. It is important to acknowledge, however, that over the last few years, a new generation of data-driven sequencing algorithms (Doroudi, Holstein, Aleven, & Brunskill, 2015; Rowe & Lester, 2015; Tang, Gogel, McBride, & Pardos, 2015) brought the ideas of sequencing back into the focus of the research community. These algorithms are based on a large volume of learning data rather than on imperfect content engineering produced by domain experts, and can potentially deliver more reliable content suggestions.

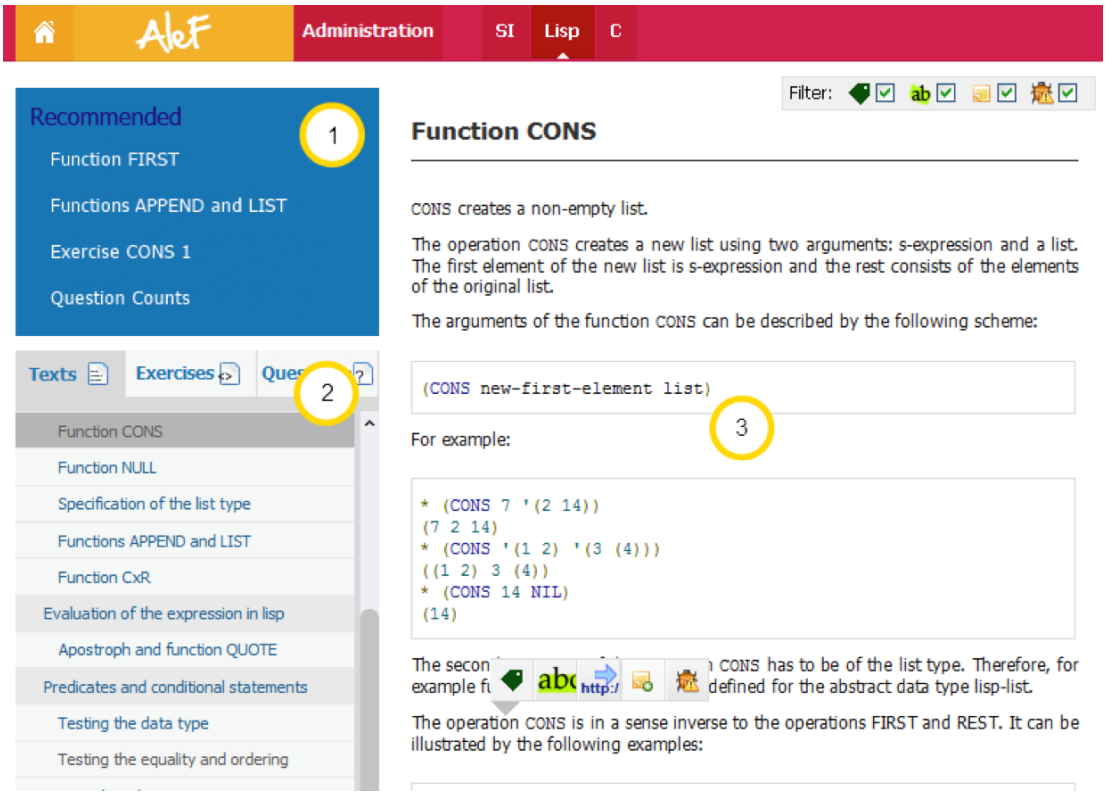


Figure 9: Learning content recommendations in ALEF (Bieliková et al., 2014): (1) list of recommended learning objects; (2) full list of learning objects organized by topic; (3) selected learning object.

Just as in the case of finer-grain topic-based guidance, the positive sides of concept-based content organization are somewhat balanced by the difficulty of presenting the concept-based content space to the students. As explained above, in coarse-grained topic organization, all content could be presented as a simple hierarchy organized along the sequence of topics (Figure 7, Figure 8), however in concept-based organization it is not feasible: the number of KCs is usually quite large and each content item is connected to many KCs. While in very small domains, the resulting network of concepts and content items could be simply presented to the user (Figure 5), in regular cases, it is not feasible.

A recommended approach for this situation is combining the straightforward presentation of topic-based organization and the personalization power of concept-based organization. The combination could be produced by grouping fine-level domain concepts into coarse-level topics, preferably corresponding to the sequence of course lectures or textbook chapters. Once this grouping is done, course content could also be grouped into topics according to its outcome concepts and presented within an easy-to-understand topic sequence. The SQL QuizGuide system (Brusilovsky et al., 2010) shows an example of this approach (Figure 10). While the whole organization looks similar to a topic-based course (Figure 7), the finer-grained concept layer allows for independently assessing the usefulness of each problem for the target students and generating link annotations that show the estimated difficulty and fraction of already mastered concepts for every problem. As a result, a single type of content organization is used to provide adaptive navigation support on both the topic level and the concept level.

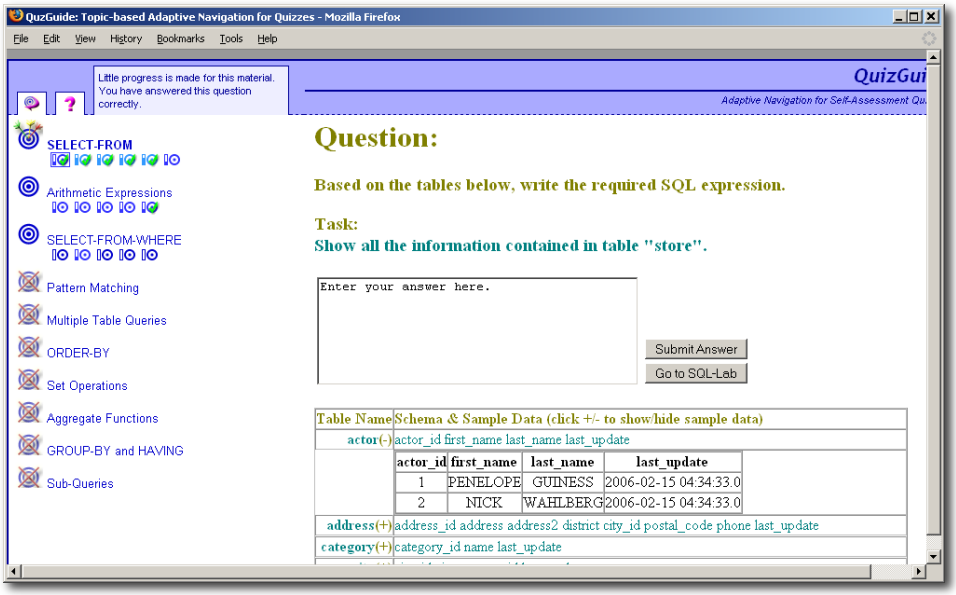


Figure 10: Content space in SQL QuizGuide was organized as a list of topics with several problems associated with each topic

A more complex case of presenting several kinds of personalized guidance on the basis of the transparent topic-based organization is provided by the Mastery Grids system (Loboda et al., 2014) and is shown in Figure 11. Mastery Grids uses traditional topic-level navigation support that displays current user knowledge for course topics (top row). To complement that, it uses social navigation support to help students in selecting both the topic and the content items within a topic. Social navigation support integrates the progress of all students in class for each content item and for each topic, and displays class progress in comparison with the student's own progress. Progress is shown using color density: darker blue levels show topics and content extensively used by students in class, while a lighter color marks less explored content and topics. By comparing personal progress with the class progress, a student can determine the most appropriate topics, problems, or examples to study. In addition, MasteryGrids uses a content recommendation algorithm; however, instead of presenting recommended content as a ranked list, it marks recommended topics and content items with start icons of different size. This allows for the

combination of several levels of suggestions in helping the student to select the most appropriate content item to work with.

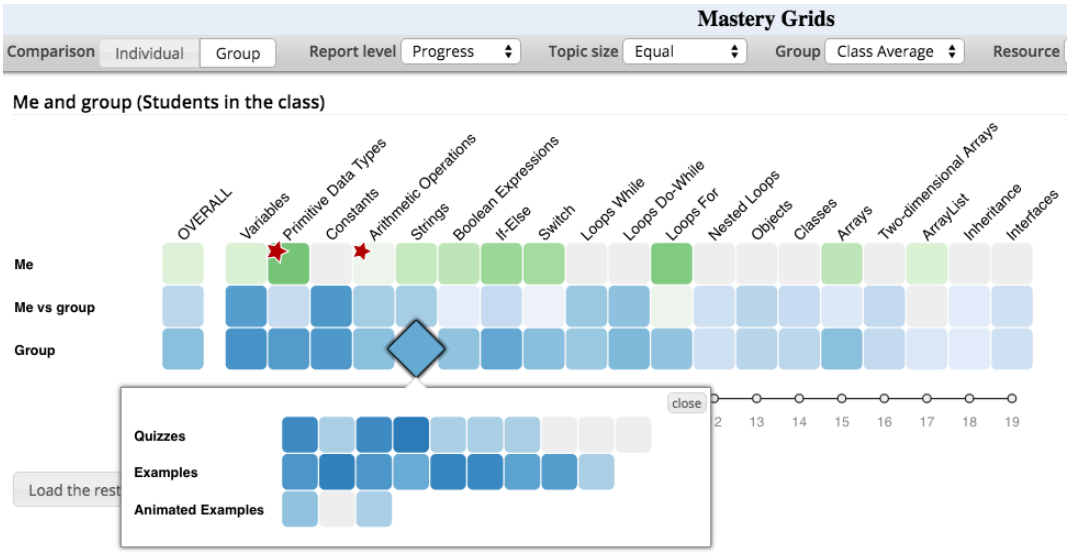


Figure 11: A combination of knowledge-based navigation support (green color), social navigation support, (blue color) and content recommendations (red stars) in the Mastery Grids system

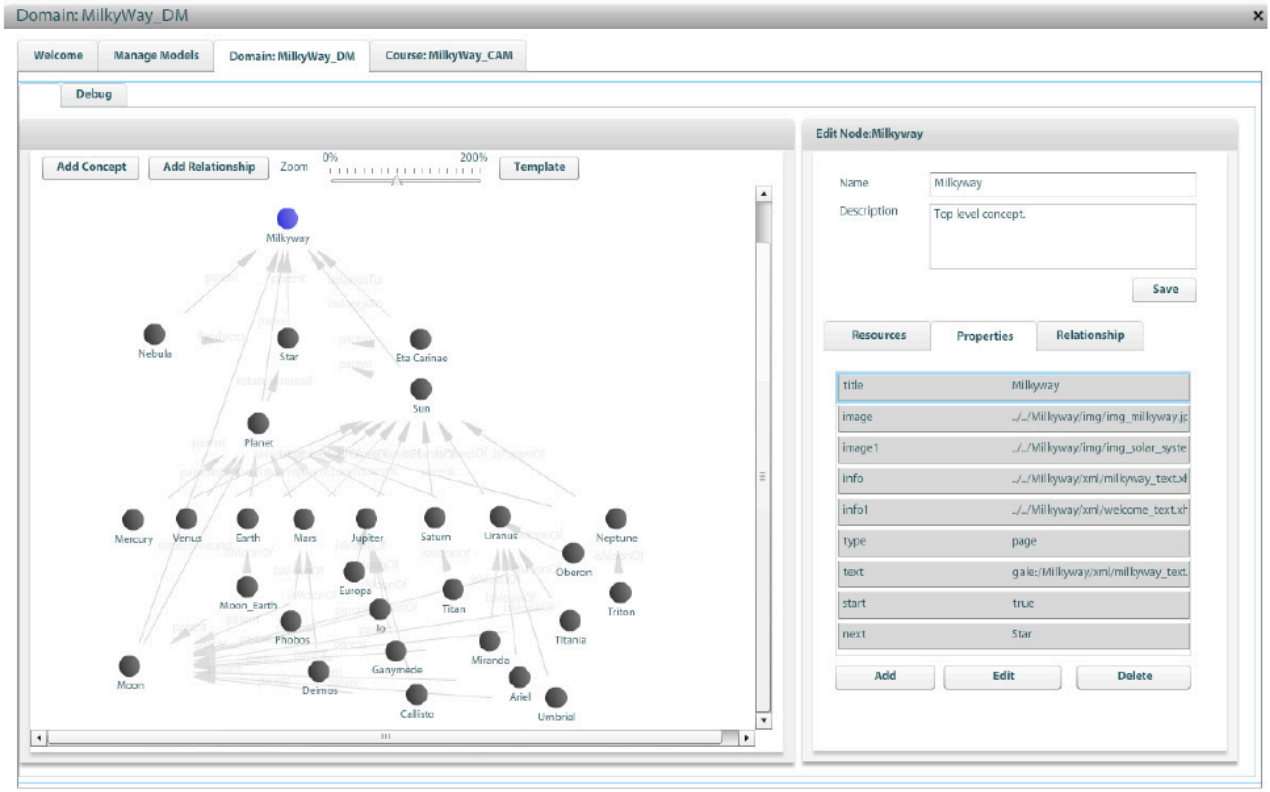


Figure 12: Extensive opportunities for defining domain models in GRAPPLE (De Bra et al., 2013).

RECOMMENDATIONS AND FUTURE RESEARCH

This chapter examined the use of domain models for providing personalized guidance in AES. We reviewed approaches for building domain models, structuring student models and learning content on the basis of domain models, and providing several kinds of personalized guidance within this content. This review offers several recommendations to the authors of authoring systems and frameworks for building AES. Most importantly, it shows that to meet the needs of the possible diversity of the domains and the related diversity of learning content, an authoring system should support a range of domain model organizations, including different level of KC granularity and diverse links between KCs. It should also support several kinds of learning content organization, as reviewed in the chapter. More specifically, an authoring system should offer have rich tools to define a set of KC for the target domain and structure then into a rich network a range of links. It should allow adding a variety of learning content types and connecting each content fragment with domain model KCs. Domain model should serve as a basis for the overlay student model, i.e., for each domain model KC, a system should maintain a variable that represent current level of knowledge.

On the top of this infrastructure, the authoring system should offer a selection of computational student modeling approaches such as BKT. Depending on a specific approach, the system should also allow the author to specific student model parameters (such as transfer, guess, and slip probability for BKT).

Finally, it should also also allow authors to choose among several ways of guiding the user through the learning content, from predefined order and free hypertext to mastery learning, sequencing, navigation support, and recommendations. It should also allow for different mechanisms and algorithms to fuel personalized guidance approaches. Without supporting this diversity, an authoring framework will be limited to a subset of domains and application contexts.

The necessity to provide rich tools and flexibility for autjors of AES has been apparently learned by the developers of authoring systems and frameworks. While early authoring frameworks, such as InterBook (Brusilovsky et al., 1998), ACE (Specht & Oppermann, 1998), MetaLinks (Murray, 2003), ECSAIWeb (Sanrach & Grandbastien, 2000), NetCoach (Weber et al., 2002), and SUGUE (Carmona, Bueno, Guzmán, & Conejo, 2002) usually supported one specific approach to domain modeling, content organization, and personalized guidance, more recent systems, such as Diogene (Sangineto et al., 2008), GRAPPLE (De Bra et al., 2013), and ALEF (Bieliková et al., 2014) gradually integrated many successful design features of earlier systems. These systems offer opportunities for rich domain modeling with multiple kinds of domain links, flexible content organization, and pluggable personalized guidance (Figure 12).

REFERENCES

Anderson, J. R., Boyle, C. F., Corbett, A. T., and Lewis, M. W. (1990). Cognitive modelling and intelligent tutoring. *Artificial Intelligence 42* (1), 7-49.

Anderson, J. R., Farrell, R., and Sauers, R. (1984). Learning to program in LISP. *Cognitive Science 8*, 87-129.

Barr, A., Beard, M., and Atkinson, R. C. (1976). The computer as tutorial laboratory: the Stanford BIP project. *International Journal on the Man-Machine Studies 8* (5), 567-596.

Bieliková, M., Šimko, M., Barla, M., Tvarožek, J., Labaj, M., Móro, R., Srba, I., and Ševcech, J. (2014). ALEF: From Application to Platform for Adaptive Collaborative Learning. In: N. Manouselis, H. Drachsler, K. Verbert and O. Santos (eds.): *Recommender Systems for Technology Enhanced Learning.* Springer New York, pp. 195-225.

Brecht, B. J., McCalla, G., and Greer, J. (1989). Planning the content of instruction. In: D. Bierman, J. Breuker and J. Sandberg (eds.) *Proceedings of 4th International Conference on AI and Education,* Amsterdam, 24-26 May 1989, Amsterdam, IOS, pp. 32-41.

Brusilovsky, P. (1992). A framework for intelligent knowledge sequencing and task sequencing. In: C. Frasson, G. Gauthier and G. McCalla (eds.) *Proceedings of Second International Conference on Intelligent Tutoring Systems*, ITS'92, Montreal, Canada, June 10-12, 1992, Springer-Verlag, pp. 499-506.

Brusilovsky, P. (2003). Developing Adaptive Educational Hypermedia Systems: From Design Models to Authoring Tools. In: T. Murray, S. Blessing and S. Ainsworth (eds.): *Authoring Tools for Advanced Technology Learning*

Environments: Toward cost-effective adaptive, interactive, and intelligent educational software. Kluwer: Dordrecht, pp. 377-409.

Brusilovsky, P. (2007). Adaptive navigation support. In: P. Brusilovsky, A. Kobsa and W. Neidl (eds.): The Adaptive Web: Methods and Strategies of Web Personalization. *Lecture Notes in Computer Science*, Vol. 4321, Berlin Heidelberg New York: Springer-Verlag, pp. 263-290.

Brusilovsky, P., Baishya, D., Hosseini, R., Guerra, J., and Liang, M. (2013). KnowledgeZoom for Java: A Concept-Based Exam Study Tool with a Zoomable Open Student Model. In: *Proceedings of 2013 IEEE 13th International Conference on Advanced Learning Technologies*, Beijing, China, July 15-18, 2013, pp. 275-279.

Brusilovsky, P. and Cooper, D. W. (2002). Domain, Task, and User Models for an Adaptive Hypermedia Performance Support System. In: Y. Gil and D. B. Leake (eds.) *Proceedings of 2002 International Conference on Intelligent User Interfaces*, San Francisco, CA, January 13-16, 2002, ACM Press, pp. 23-30.

Brusilovsky, P., Eklund, J., and Schwarz, E. (1998). Web-based education for all: A tool for developing adaptive courseware. In: H. Ashman and P. Thistewaite (eds.) *Proceedings of Seventh International World Wide Web Conference*, Brisbane, Australia, 14-18 April 1998, Elsevier Science B. V., pp. 291-300.

Brusilovsky, P. and Millán, E. (2007). User models for adaptive hypermedia and adaptive educational systems. In: P. Brusilovsky, A. Kobsa and W. Neidl (eds.): The Adaptive Web: Methods and Strategies of Web Personalization. *Lecture Notes in Computer Science*, Vol. 4321, Berlin Heidelberg New York: Springer-Verlag, pp. 3-53.

Brusilovsky, P., Sosnovsky, S., Lee, D., Yudelson, M., Zadorozhny, V., and Zhou, X. (2010). Learning SQL programming with interactive tools: from integration to personalization. *ACM Transactions on Computing Education, 9* (4), Article No. 19, pp. 1-15.

Cafolla, R. (2006). Project MERLOT: Bringing Peer Review to Web-Based Educational Resources. *Journal of Technology and Teacher Education, 14* (2), 313-323.

Carbonell, J. R. (1970). AI in CAI: An artificial intelligence approach to computer aided instruction. *IEEE Transactions on Man-Machine Systems MMS, 11* (4), 190-202.

Carmona, C., Bueno, D., Guzmán, E., and Conejo, R. (2002). SIGUE: Making Web Courses Adaptive. In: P. De Bra, P. Brusilovsky and R. Conejo (eds.) *Proceedings of Second International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (AH'2002)*, Málaga, Spain, May 29-31, 2002, Springer-Verlag, pp. 376-379.

Conati, C., Gertner, A., and Vanlehn, K. (2002). Using Bayesian Networks to Manage Uncertainty in Student Modeling. *User Modeling and User-Adapted Interaction, 12* (4), 371-417.

Conejo, R., Guzman, E., and Millán, E. (2004). SIETTE: A Web-based tool for adaptive teaching. *International Journal of Artificial Intelligence in Education, 14* (1), 29-61.

Corbett, A. T. and Anderson, J. R. (1995). Knowledge tracing: Modelling the acquisition of procedural knowledge. *User Modeling and User-Adapted Interaction, 4* (4), 253-278.

Dagger, D., Wade, V., and Conlan, O. (2004). A Framework for Developing Adaptive Personalized eLearning. In: J. Nall and R. Robson (eds.) *Proceedings of World Conference on E-Learning, E-Learn 2004*, Washington, DC, USA, November 1-5, 2004, AACE, pp. 2579-2587.

Davidovic, A., Warren, J., and Trichina, E. (2003). Learning benefits of structural example-based adaptive tutoring systems. *IEEE Transactions on Education, 46* (2), 241-251.

De Bra, P., Aerts, A., Berden, B., de Lange, B., Rousseau, B., Santic, T., Smits, D., and Stash, N. (2003). AHA! The Adaptive Hypermedia Architecture. In: *Proceedings of the Fourteenth ACM Conference on Hypertext and Hypermedia*, Nottingham, UK, ACM, pp. 81-84.

De Bra, P., Aerts, A., and Rousseau, B. (2002). Concept Relationship Types for AHA! 2.0. In: M. Driscoll and T. C. Reeves (eds.) *Proceedings of World Conference on E-Learning, E-Learn 2002*, Montreal, Canada, October 15-19, 2002, AACE, pp. 1386-1389.

De Bra, P. and Calvi, L. (1998). AHA! An open Adaptive Hypermedia Architecture. *The New Review of Hypermedia and Multimedia, 4*, 115-139.

De Bra, P., Smits, D., van der Sluijs, K., Cristea, A., Foss, J., Glahn, C., and Steiner, C. M. (2013). GRAPPLE: Learning Management Systems Meet Adaptive Learning Environments. In: A. Peña-Ayala (ed.) *Intelligent and Adaptive Educational Learning Systems: Achievements and Trends*.

Diessel, T., Lehmann, A., and Vassileva, J. (1994). Individualised course generation: A marriage between CAL and

ICAL. *Computers and Education, 22* (1/2), 57-64.

Doroudi, S., Holstein, K., Aleven, V., and Brunskill, E. (2015). Towards Understanding How to Leverage Sense-making, Induction/Refinement and Fluency to Improve Robust Learning. In: *Proceedings of the 8th International Conference on Educational Data Mining (EDM 2015)*, Madrid, Spain, June 26-29, 2015.

Eklund, J. and Brusilovsky, P. (1998). Individualising Interaction in Web-based Instructional Systems in Higher Education. In: *Proceedings of The Apple University Consortium's Academic Conference*, Melbourne, Australia, September 27-30, 1998, pp. 27-30.

Farrell, R., Thomas, J. C., Dooley, S., Rubin, W., Levy, S., O'Donnell, R., and Fuller, E. (2003). Learner-driven assembly of Web-based courseware. In: A. Rossett (ed.) *Proceedings of World Conference on E-Learning, E-Learn 2003*, Phoenix, AZ, USA, November 7-11, 2003, AACE, pp. 1052-1059.

González-Brenes, J. P., Huang, Y., and Brusilovsky, P. (2014). General Features in Knowledge Tracing to Model Multiple Subskills, Temporal Item Response Theory, and Expert Knowledge. In: *Proceedings of the 7th International Conference on Educational Data Mining (EDM 2014)*, London, UK, July 4-7, 2014, pp. 84-91.

Grigoriadou, M., Papanikolaou, K., Kornilakis, H., and Magoulas, G. (2001). INSPIRE: An INtelligent System for Personalized Instruction in a Remote Environment. In: *Proceedings of Third workshop on Adaptive Hypertext and Hypermedia*, Sonthofen, Germany, July 14, 2001, Technical University Eindhoven, pp. 13-24.

Hammond, N. (1989). Hypermedia and learning: Who guides whom? In: H. Maurer (ed.) *Proceedings of 2-nd International Conference on Computer Assisted Learning*, ICCAL'89, Berlin, May 9-11, 1989, Springer-Verlag, pp. 167-181.

Henze, N. and Nejdl, W. (2001). Adaptation in open corpus hypermedia. *International Journal of Artificial Intelligence in Education, 12* (4), 325-350.

Hoog, R. d., Wielinga, B., Kabel, S., Anjewierden, A., Verster, F., Barnard, Y., PaoloDeLuca, Desmoulins, C., and Riemersma, J. (2002). Re-using technical manuals for instruction: document analysis in the IMAT project. In: Y. Barnard (ed.) *Proceedings of Workshop on integrating technical and training documentation held in conjuction with ITS'02 conference,* San Sebastian, Spain, June 3, 2002, pp. 15-25.

Koedinger, K. R., Corbett, A. T., and Perfetti, C. (2012). The knowledge-learning-instruction framework: Bridging the science-practice chasm to enhance robust student learning. *Cognitive Science, 36* (5), 757–798.

Kumar, A. (2006). The Effect of Using Problem-Solving Tutors on the Self-Confidence of Students. In: *Proceedings of 18th Annual Psychology of Programming Workshop (PPIG 06)*, Brighton, U.K., September 7-8, 2006, pp. 275-283.

Loboda, T., Guerra, J., Hosseini, R., and Brusilovsky, P. (2014). Mastery Grids: An Open Source Social Educational Progress Visualization. In: *Proceedings of 9th European Conference on Technology Enhanced Learning (EC-TEL 2014)*, Graz, Austria, September 16-19, 2014, pp. 235-248.

Manouselis, N., Drachsler, H., Verbert, K., and Duval, E. (eds.) (2013). *Recommender Systems for Learning*. Berlin: Springer.

McArthur, D., Stasz, C., Hotta, J., Peter, O., and Burdorf, C. (1988). Skill-oriented task sequencing in an intelligent tutor for basic algebra. *Instructional Science, 17* (4), 281-307.

Millán, E., Loboda, T., and Pérez de la Cruz, J. L. (2010) Bayesian networks for student model engineering. *Computers & Education ,55*, 1663-1683.

Mitrovic, A. (2003) An Intelligent SQL Tutor on the Web. *International Journal of Artificial Intelligence in Education, 13* (2-4), 173-197.

Mitrovic, A. and Devedzic, V. (2004) A Model of Multitutor Ontology-based Learning Environments. *Continuing Engineering Education and Life-Long Learning, 14* (3), 229-245.

Murray, T. (2003) MetaLinks: Authoring and affordances for conceptual and narrative flow in adaptive hyperbooks. I*nternational Journal of Artificial Intelligence in Education, 13* (2-4), 199-233.

Ohlsson, S. (1992) Constraint-based student modeling. *Journal of Artificial Intelligence in Education, 3* (4), 429-447.

Papanikolaou, K. A., Grigoriadou, M., Kornilakis, H., and Magoulas, G. D. (2003) Personalising the interaction in a Web-based Educational Hypermedia System: the case of INSPIRE. *User Modeling and User Adapted Interaction, 13* (3), 213-267.

Rowe, J. P. and Lester, J. C. (2015). Improving Student Problem Solving in Narrative-Centered Learning

Environments: A Modular Reinforcement Learning Framework. In: C. Conati, N. Heffernan, A. Mitrovic and M. F. Verdejo (eds.): 17th International Conference on Artificial Intelligence in Education, AIED 2015. *Lecture Notes in Computer Science*, Madrid, Spain, pp. 419-428.

Rus, V., Baggett, W., Gire, E., Franceschetti, D., Conley, M., and Graesser, A. (2013). Towards Learner Models based on Learning Progressions in DeepTutor. In: R. Sottilare (ed.) *Learner Models*. Army Research Lab.

Sangineto, E., Capuano, N., Gaeta, M., and Micarelli, A. (2008). Adaptive course generation through learning styles representation. *Universal Access in the Information Society, 7* (1-2), 1-23.

Sanrach, C. and Grandbastien, M. (2000). ECSAIWeb: A Web-based authoring system to create adaptive learning systems. In: P. Brusilovsky, O. Stock and C. Strapparava (eds.) *Proceedings of Adaptive Hypermedia and Adaptive Web-based Systems*, AH2000, Trento, Italy, August 28-30, 2000, Springer-Verlag, pp. 214-226.

Sao Pedro, M. A., Baker, R. S., and Gobert, J. D. (2013). Incorporating Scaffolding and Tutor Context into Bayesian Knowledge Tracing to Predict Inquiry Skill Acquisition. In: *Proceedings of The 6th International Conference on Educational Data Mining (EDM 2013)*, Memphis, Tennessee, July 6 - 9, 2013, pp. 185-192.

Sleeman, D. H. (1985). UMFE: a user modeling front end system. International Journal on the Man-*Machine Studies, 23*, 71-88.

Sosnovsky, S. and Brusilovsky, P. (2015). Evaluation of Topic-based Adaptation and Student Modeling in QuizGuide. *User Modeling and User-Adapted Interaction, 25* (4), 371-424.

Sosnovsky, S. and Dicheva, D. (2010). Ontological technologies for user modelling. I*nternational Journal of Metadata, Semantics and Ontologies, 5* (5), 32-71.

Specht, M. and Oppermann, R. (1998). ACE - Adaptive Courseware Environment. *The New Review of Hypermedia and Multimedia, 4*, 141-161.

Steinacker, A., Faatz, A., Seeberg, C., Rimac, I., Hörmann, S., Saddik, A. E., and Steinmetz, R. (2001). MediBook: Combining semantic networks with metadata for learning resources to build a Web based learning system. In: *Proceedings of ED-MEDIA'2001 - World Conference on Educational Multimedia, Hypermedia and Telecommunications*, Tampere, Finland, June 25-30, 2001, AACE, pp. 1790-1795.

Tang, S., Gogel, H., McBride, E., and Pardos, Z. (2015). Item Ordering Effects using Online Tutoring Data. In: O. Santos, et al. (eds.) *Proceedings of the 8th International Conference on Educational Data Mining (EDM 2015)*, Madrid, Spain, June 26-29, 2015.

Trausan-Matu, S., Maraschi, D., and Cerri, S. A. (2002). Ontology-centered personalized presentation for knowledge extracted from the Web. In: S. A. Cerri, G. Gouardères and F. Paraguaçu (eds.) *Proceedings of 6th International Conference on Intelligent Tutoring Systems (ITS'2002)*, Berlin, June 2-7, 2002, Springer-Verlag, pp. 259-269.

Trella, M., Conejo, R., and Bueno, D. (2002). An autonomous component architecture to develop WWW-ITS. In: P. Brusilovsky, N. Henze and E. Millán (eds.) P*roceedings of Workshop on Adaptive Systems for Web-Based Education at the 2nd International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (AH'2002)*, Málaga, Spain, May 28, 2002, pp. 69-80.

VanLehn, K. (2006). The behavior of tutoring systems. *International Journal of Artificial Intelligence in Education, 16* (3), 227-265.

Vassileva, J. (1998). DCG + GTE: Dynamic Courseware Generation with Teaching Expertise. *Instructional Science, 26* (3/4), 317-332.

Vassileva, J. and Deters, R. (1998). Dynamic courseware generation on the WWW. *British Journal of Educational Technology, 29* (1), 5-14.

Weber, G. and Brusilovsky, P. (2001). ELM-ART: An adaptive versatile system for Web-based instruction. *International Journal of Artificial Intelligence in Education, 12* (4), 351-384.

Weber, G., Kuhl, H.-C., and Weibelzahl, S. (2002). Developing adaptive internet based courses with the authoring system NetCoach. In: S. Reich, M. M. Tzagarakis and P. M. E. De Bra (eds.): *Hypermedia: Openness, Structural Awareness, and aptivity*. Berlin: Springer-Verlag, pp. 226-238.

Wescourt, K. T., Beard, M., and Gould, L. (1977). Knowledge-based adaptive curriculum sequencing for CAI: application of a network representation. In: *Proceedings of 1977 annual ACM conference*, Seattle, October 1977, pp. 234-240.