Educational Technology Research and Development manuscript No.
(will be inserted by the editor)

Managing Items and Knowledge Components: Domain Modeling in Practice

Radek Pelánek

Received: date / Accepted: date

Abstract Adaptive learning systems need large pools of examples for practice—thousands of items that need to be organized into hundreds of knowledge components within a domain model. Domain modeling and closely related student modeling are intensively studied in research studies. However, there is a gap between research studies and practical issues faced by developers of scalable educational technologies. The aim of this paper is to bridge this gap by connecting techniques and notions used in research papers to practical problems in development. We put specific emphasis on scalability—on techniques that enable relatively cheap and fast development of adaptive learning systems. We summarize conceptual questions in domain modeling, provide an overview of approaches in the research literature, and discuss insights based on the development and analysis of a widely used system. We conclude with recommendations for both developers and researchers in the area of adaptive learning systems.

1 Introduction

Instructional design theory predicts trade-offs in the design, development, and application of educational technology (Reigeluth & Carr-Chellman, 2009b; Honebein & Honebein, 2015). The overall theme of this work is the balancing of these trade-offs in the practical development of online learning systems. Specifically, we focus on the issue of domain modeling—designing an appropriate organization of individual learning objects to higher-level units and specification of relations among these units. Many aspects of our discussion are quite general, but to attain clear focus, we consider adaptive practice systems, particularly systems employing as their key instructional methods the basic drill and practice enhanced with personalized features (e.g., mastery learning,

Faculty of Informatics, Masaryk University Brno; pelanek@fi.muni.cz, Phone +420-549 49 6991, Fax +420-549 491 820

adaptive sequencing, spaced repetition). Such systems are today widely applied and used by millions of students; specific well-known examples are Khan Academy, IXL, or Duolingo.

As a conceptual basis for our discussion, we use the Knowledge-Learning-Instruction framework (Koedinger et al., 2012)—a recently proposed framework that builds upon previous instructional design approaches (Gagné, 1985; Sweller, 1994), connects them with cognitive theories of learning, and provides specific impulses for the development of educational technology in the form of mapping knowledge types to suitable instructional methods. A key aspect of the Knowledge-Learning-Instruction framework is the decomposition of knowledge into knowledge components. The concept of knowledge components also is featured in many other theories of learning and instructional design, just under different names (e.g., skills, concepts, schemas).

From the perspective of the development of practical educational technology, we need to organize available practice items into these knowledge components. Simple examples of knowledge components and items are the addition of fraction ($\frac{1}{2} + \frac{2}{3}$), European states ("Where is Poland?"), and capitalization rules in English ("monday or Monday?"). We are concerned with questions concerning the management of knowledge components and items:

- *How are knowledge components defined?* Some knowledge components, like "European states", are quite natural. However, in cases like English vocabulary for second language learners, the suitable choice of knowledge components is much less clear.
- *What is a suitable granularity of knowledge components?* Should we use "addition of fraction" or rather more fine-grained components like "addition of fraction with the same denominator" and "addition of fraction with different denominators"?
- *How should we deal with items of different difficulty?* In the practice of African states, Egypt and Guinea-Bissau have widely different difficulty (at least for students from other continents). Should they be part of the same knowledge component? How should differences in item difficulty be incorporated in the domain model?
- *How do we model relations among knowledge components?* Do we use a taxonomy for specifying a hierarchy of knowledge components? Do we use a tree, a directed acyclic graph, or even a general ontology for the representation? Do we model prerequisite relations among knowledge components? Can one item belong to several knowledge components?

Answers to these questions are crucial in the practical development of learning systems since they have an important impact on many aspects of these systems. Knowledge components are used within the user interface of a system to present the content to students; they can also be used for adaptive navigation (Brusilovsky, 1998), e.g., personalized recommendations of topics to practice. Domain modeling is closely connected to student modeling, i.e., the estimation of skills of students (Pelánek, 2017; Abyaa et al., 2019). Student modeling serves as a basis for adaptive practice techniques like mastery learning, spaced repetition, and adaptive choice of items (Roediger & Pyc, 2012; Pelánek & Řihák, 2018), and the visualization of progress to students in the form of progress bars or open learner modeling (Bull & Kay, 2007). Moreover, the domain model is used by system developers "behind the scenes" for content management and data analysis (Baker, 2016).

Domain modeling is intensively studied by researchers, e.g., Käser et al. (2013, 2014); Huang et al. (2016); González-Brenes et al. (2014); Pelánek (2017). Researchers focus mainly on the accuracy of models and their conceptual clarity but often make implicit assumptions that ignore practical issues. For example, the common implicit assumption in research papers is that knowledge components are well-defined and homogeneous (i.e., all items within a knowledge component are similar). As we discuss in this paper, this assumption is quite hard to satisfy with realistic educational content. Moreover, research studies often consider relatively small domain models. The aim of this paper is to bridge research and practice.

We consider the issue of domain modeling from the perspective of the practical development of educational technology, with an emphasis on scalability.

As described by Reigeluth & Carr-Chellman (2009b), practical issues associated with instructional conditions (learner, content, learning environment, and constraints) are important, particularly when we are developing adaptive systems for unique conditions, such as a relatively small target audience, a small team, a modest budget, and specialized content areas. A large portion of domain modeling research considers learning in English-speaking countries and on content that is studied by most students (e.g., elementary math)—in these cases, it is feasible and realistic to build very detailed domain models. However, when we consider learning in smaller countries or more specialized subjects, it may not be feasible to develop "idealistic" domain models with extensive item pools[1]. In these contexts, we need to develop "usable" domain models that can be obtained within constraints of a particular project.

In the development of such domain models, it is necessary to make tradeoffs between the conceptual correctness and clarity of domain models and costs associated with their development. A realistic learning system contains hundreds of knowledge components and tens of thousands of items. For a small target audience, the domain model needs to be created and managed by a small team and with a limited budget. The development thus needs to be pragmatic. Nevertheless, it can still be informed by research. The aim of this paper is to bridge the gap between theoretically oriented research and pragmatically oriented development by providing an overview of techniques that are based on research and yet directly applicable to the development. Specifically, we focus on the above-presented questions and seek answers applicable to realistic, constrained instructional situations.

[1] Moreover, the technological landscape is shifting quickly—consider for example the rise and decline of Adobe Flash technology for educational applications. Even in context like practice of elementary math for English-speaking audience, it is advantageous to be able to react quickly and develop models usable for a particular technological application.

2 Background and Terminology

Before discussing the domain modeling itself, we clarify the context of our discussion and the used terminology.

2.1 Context and Methodology

The aim of this paper is to bridge the gap between research papers and practical problems in the development of adaptive systems. The used methodology corresponds to this aim—it was based on an iterative process involving analysis of research literature and practical system development, specifically on these steps:

- Design (revision) of a domain model and preparation of content (practice items).
- Collection of data in the adaptive system.
- Analysis of data, identification of problems that need to be addressed.
- Overview of research literature with the goal to find suitable solutions to identified problems.

The work is based on practical experience with the development of an adaptive practice system Umíme (`umimeto.org`). The system is aimed at Czech elementary and high-school students. It offers practice in a variety of domains, including elementary mathematics, programming, Czech grammar and orthography, English grammar and vocabulary, and factual knowledge in geography and biology. The system has hundreds of knowledge components, thousands of items, and is used by thousands of students each day. The focus of the system is on the basic learning objectives like remembering facts, recognizing categories, and understanding and applying simple rules. Many aspects of the system are similar to well-known systems that target an English speaking audience primarily (e.g., Khan Academy, IXL, Duolingo). The main difference is that the Umíme system is developed by a small team and under a severely limited budget.

The system collects data on students' actions within the system. The primary type of data that was used for the presented discussion is the data on students' answers to multiple-choice questions and basic constructed response items. The data are collected within an exercise that presents students a set of related items in a randomized order and uses a mastery criterion to terminate the practice (Pelánek & Řihák, 2018). The used analysis of data involves mainly analysis of the difficulty of items (e.g., success rate, median response time) and similarity of items (Pelánek, 2019). The used system also provides recommendations and collects data on student navigation within this system. These data also informed the iterative processes outlined above.

2.2 Items

Learning systems can contain a variety of learning objects. Churchill (2007) presents a classification of learning objects into six types (presentation, practice, simulation, conceptual models, information, and contextual representation). In this work, we consider only practice objects, which are fundamental for adaptivity—by interacting with practice objects, students provide information that can be used for adaptivity. Practice objects can take many forms (e.g., multiple-choice questions, short text answers, interactive multi-step problems). In this work, we denote practice objects by the generic and commonly used term *items*.

Most of our discussion is general and relevant to a variety of item formats and domains; through the paper, we use illustrative examples from mathematics, language learning, and geography. We consider items relevant mainly in elementary practice, i.e., bottom levels in the Bloom's taxonomy of educational objectives (Bloom et al., 1956; Anderson et al., 2000), particularly recall, recognition, classification, and execution concerning factual, conceptual, and procedural knowledge. With items of this type, students can typically answer items in 2-to-60 seconds, and the answer can be automatically evaluated by a learning system.

Many of the discussed issues are relevant also to more complex items, e.g., multi-step, interactive problems targeting higher level skills. However, our focus is on practical scalability with limited resources. Even for elementary items, the development of a scalable, adaptive learning system is quite a challenge.

2.3 Knowledge Components and Their Types

Instructional theories and student modeling approaches use many different notions for the organization units in learning systems, e.g., skills, concepts, abilities, content constructs, or content of instruction (Reigeluth et al., 1980; Reigeluth & Carr-Chellman, 2009b; Porter, 2002; De Ayala, 2013; Pelánek, 2017).

In this work, we build upon the Knowledge-Learning-Instruction (KLI) framework (Koedinger et al., 2012) and we use the notion "knowledge component" (KC), which the KLI framework defines as "an acquired unit of cognitive function or structure that can be inferred from performance on a set of related tasks" (Koedinger et al., 2012, p. 764). The term knowledge component is closely related to many other terms used in both research and practice. Terms like skills, abilities, or concepts are mostly used as synonymous. A related, but slightly different notion is a learning objective. Learning objectives differ from knowledge components in their focus, purpose, and typical time scale. Whereas learning objectives focus on outcomes (what the student should be able to do), knowledge components (particularly in the KLI framework definition) highlights the cognitive aspect of performance. The mental processes described by knowledge components are at the time scale of "unit tasks" that

Table 1 Basic types of knowledge components, according to the KLI framework (Koedinger et al., 2012), simplified

Common labels	Application condition	Response	Examples
facts, associations	constant	constant	European states, fruits vocabulary
categories, concepts	variable	constant	capitalization, determiners, geometric objects names
rules, principles, schemas	variable	variable	fraction addition, linear equations, word order in sentences

take students time at the order of 10 seconds (Koedinger et al., 2012). Learning outcomes are typically formulated at a coarser level of granularity and concern processes at the order of minutes or hours, spanning multiple knowledge components. As a specific example, consider a learning objective "Students will be able to add and subtract fractions" and a knowledge component "addition of fraction with like denominators".

The KLI framework discusses knowledge components from the point of view of cognitive science and pedagogy. In this work, we consider knowledge components from a more pragmatic perspective of the development of educational technologies and treat knowledge components mostly as "organizational units that group together related items". Note that from the conceptual point of view, "a unit of cognitive function" and "a unit of organization within software" are significantly different notions. However, for purposes of our discussion, this distinction is mostly not consequential, and we gloss over it.

The KLI framework presents a key idea for the development of educational technologies: When making a decision about which instructional methods to use, it is better to base the decision on the type of knowledge components rather than the domain (mathematics, language learning, geography). The KLI framework discusses several types of knowledge components; Table 1 presents a simplified view of the categorization. The table describes only three basic types of knowledge components, which are widely used in practical applications.

The type of knowledge component has consequences for practice within a learning system. Some instructional methods are relevant only for some types of knowledge components (Koedinger et al., 2012; Reigeluth & Carr-Chellman, 2009a):

- For facts, it is important to take into account forgetting. Practice within the system should support spaced repetition (Roediger & Pyc, 2012).
- For rules, interleaved practice (a mixed practice of several KCs together) is essential, so that students can practice the recognition of "application conditions" (Roediger & Pyc, 2012).
- For rules, the order of the presentation of knowledge components to students should respect prerequisite relations between KCs. This aspect is not fundamental in the practice of facts.

The type of KC also has practical consequences for the preparation of items. Facts often have a limited number of potential items within a KC; in many cases, we want the list of items to be exhaustive (e.g., "days and months" in second language vocabulary). For rules, there is typically an unlimited number of potential items, so we need to decide which items to choose and how to specify them.

2.4 Domain Modeling

The design of adaptive learning systems is often decomposed into several interacting models (Nkambou et al., 2010; Sottilare et al., 2016):

– a domain model, which models the content of the concerned domain,
– a student model, which models the state of a student, particularly students' knowledge with respect to the domain model; it can also model other aspects of student state (e.g., affect, meta-cognition),
– a tutor (pedagogical) model (instructional method), which specifies the choice of pedagogical actions based on the domain model and the student model.

Domain modeling and student modeling are closely intertwined (Pelánek, 2017). In this work, we focus on domain modeling, making brief remarks about the consequences of different choices on student modeling. In instructional design literature, domain modeling is often denoted by different terms, e.g., Reigeluth et al. (1978) provide an early example of discussion closely relevant to the current work; they use the term "structure of subject matter content".

Some authors consider the domain model in a general way: "The domain model contains the set of skills, knowledge, and strategies/tactics of the topic being tutored. It normally contains the ideal expert knowledge and also the bugs, mal-rules, and misconceptions that students periodically exhibit." (Sottilare et al., 2016, p. 3). We focus on the core of domain modeling that is necessary for the practical development of a learning system: mapping between items and KCs and relations among KCs. We consider the additional aspects of domain modeling mentioned in the above-given definition (strategies, bugs, mal-rules, misconceptions) to be "nice-to-have features", which are not completely necessary for the development, particularly under a limited budget.

2.5 Uses of Domain Models

The domain model can be used in many ways, for example:

– personalization of practice for students, e.g., mastery learning (Pelánek & Řihák, 2018), adaptive choice of items (Pelánek et al., 2017), spaced repetition (Roediger & Pyc, 2012),

Table 2 Illustration of knowledge components of different granularity

granularity	geography facts	grammar categories	mathematics rules
coarse	names and locations of countries	capitalization rules	fractions
medium	names and locations of African countries	capitalization of nouns	addition of fractions
fine	location of Egypt	capitalization of days of the week	addition of fractions with the same denominator

- organization of the content, presentation to students, adaptive navigation (Brusilovsky, 1998),
- feedback to students on the progress of their learning, e.g., using open learner modeling (Bull & Kay, 2007) or gamification features like badges (Dicheva et al., 2015),
- support for teachers, e.g., dashboards for teachers (Molenaar & Knoop-van Campen, 2017).

The appropriate choice of domain model depends on the specific use of the domain model within an application. However, most of our discussion is relevant to all of these uses.

3 Specification of Knowledge Components

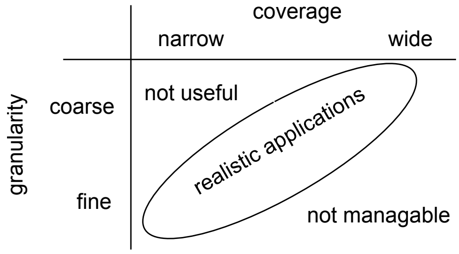
Research studies in student modeling often (implicitly) assume that items in the same knowledge component are homogeneous, i.e., that they are equivalent from the point of estimation of students skills. However, this assumption is never completely satisfied in practice. In a practical system, we need to make trade-offs between the homogeneity of KCs and the feasibility of their creation, management, and use.

3.1 Granularity of Knowledge Components

Table 2 provides examples of different granularity for several domains and types of KC. Which level of granularity should we use? The appropriate choice of KC granularity depends on the application, e.g., what is the specific aim of the system, how wide is the coverage of the system, how large pool of items is available (or can be prepared under given circumstances). The target audience significantly influences decisions about KC granularity. Consider, for example, the practice of anatomy. For high school students, it may be sufficient to have "muscles" as a single KC. For medical students a more fine-grained division is necessary (e.g., muscles on the head, back, neck, abdomen).

Table 3 depicts (in a simplified manner) the interaction between the coverage of the system and the granularity of KCs. A coarse granularity with a narrow coverage would mean just a few KCs—this setting is not very useful, at least for adaptation. A fine granularity with a broad coverage leads to a very large number of KCs—such a setting is challenging to manage. The realistic area is diagonal in between, which includes, for example, an optimized fraction tutor with a narrow scope, but with fine-grained KC modeling, or our Umíme system with a broad coverage (including the practice of English, mathematics, and programming), but coarse-grained KCs.

Table 3 Granularity and coverage tradeoff



The choice of KC granularity is an important decision, which influences not only the presentation of the content to students but also the choice of modeling techniques within the system. Specifically, the granularity of KCs interacts with student modeling. When we use very coarse-grained KCs, we can assume that student knowledge is (nearly) constant, i.e., we do not need to model short-term learning and models from item response theory are applicable (De Ayala, 2013). When we use very fine-grained KCs, modeling short-term learning is essential. In this situation, it may be applicable to use a Bayesian knowledge tracing model (Corbett & Anderson, 1994), which assumes a direct transition from the unknown to the known state. For knowledge components of intermediate granularity, learning happens, but more slowly. In these cases, it may be appropriate to use one of the logistic models (Pelánek, 2017), e.g., one of the variations on additive factors models (Cen et al., 2006).

Several research studies have compared student models of different knowledge component granularity (Koedinger et al., 2016; Feng et al., 2006; Pardos et al., 2010). From the research perspective, there is typically a preference for more fine-grained KCs, as these offer a better fit to data and have better cognitive interpretation. However, from a practical perspective, there are issues with KCs that are too fine-grained. When items are too similar, repeated practice within a KC does not make much sense, particularly for multiple-choice questions, where all items can have the same type of answer. With very fine-grained KCs, it is typically necessary to use some variant of interleaved practice (mixing practice of several KCs). This is a meaningful and potentially useful approach. However, it complicates the implementation of the learning system. The implementation is much more straightforward when the practice of KCs can be standalone.

3.2 Difficulty of Items

Items within a knowledge component differ in their difficulty. For illustration, consider the following examples:

– one-digit multiplication: 2×3 versus 8×7,
– English determiners: "[a/an] car" versus "[a/an] hour",
– African countries: Egypt versus Guinea-Bissau.

In all these cases, there are substantial differences in difficulty—the first item from each pair is significantly more difficult than the second one (as measured by success rates and response times of students). We can treat these differences in several ways.

The principled solution is to *divide the KC* into several more fine-grained knowledge components. For example, in the case of English determiners, the difficult examples are related to a clear rule ("The sound, not the spelling is important."). In other cases, the suitable division of a KC may be less clear but can be found with the use of data mining over student data, as done in the "closing the loop" studies (Koedinger et al., 2013; Liu et al., 2014; Cen et al., 2007; Koedinger & McLaughlin, 2016). Such division is a principled solution, but it can lead to practical problems with too fine-grained KCs mentioned above: insufficient number of items within KCs, necessary interleaving of practice, and more difficult management of the domain model. This approach is suitable when we have a lot of time and a large budget, but in many cases, we need a more pragmatic approach.

At the other extreme, a very pragmatic solution is simply to *ignore* differences in difficulty. This approach, however, can have a nontrivial negative impact on the application. Ignoring difficulty makes student models less precise, and this loss of precision impacts all uses of student models. It can negatively impact the user experience. For example, when a system uses mastery criterion based on streaks (series of consecutive correct answers) and a student's streak is ruined by an excessively difficult example, the student can easily get frustrated.

One compromise possibility is to keep items with different difficulty in the same KC and *take the difficulty of individual items into account.* Student modeling techniques can incorporate the difficulty of items. Some student modeling approaches include item difficulty in their basic form, e.g., item response theory models (De Ayala, 2013) or models based on the Elo rating system (Pelánek, 2016). Bayesian knowledge tracing, an often used student modeling approach, does not consider item difficulty in its basic form (Corbett & Anderson, 1994), but it can be extended to include difficulty (Pardos & Heffernan, 2011; González-Brenes et al., 2014). The difficulty of items can be incorporated into an algorithm for the choice of items, i.e., we can use an algorithm that chooses items of suitable difficulty for a particular student (Pelánek et al., 2017).

Based on the experience with the development of our system, we propose another compromise option: to *divide items into several levels of similar difficulty* and then ignore differences in difficulty within individual levels. This solution is quite advantageous. The division can be done by an offline analysis; the implementation within the production system is simple, as there is no need to track the difficulties of items or use complex models. Items within levels can be chosen randomly, which limits methodological problems in data analysis due to feedback loops in data collection (Pelánek, 2018). This solution also leads to limited loss of precision and degradation of user experience since the division into levels captures the most critical differences in difficulty. In our experience, three levels of difficulty are sufficient.

The division with respect to difficulty should not be done mechanically as a simple algorithmic division could cause artifacts with unintended consequences. As a specific example, consider the comparison of fractions. Our data show that the difficulty of items is closely related to a simple (and wrong) heuristic "the result of the comparison is the same as the comparison of numerators", i.e., items like $\frac{3}{5} < \frac{6}{7}$ have much higher success rate than $\frac{3}{6} < \frac{2}{3}$. If we split items simply by their difficulty, we would create a group of items that would reinforce the wrong heuristic.

So far, we assumed that the difficulty of items corresponds to the performance of students (success rate, median response time). This approach, however, compounds the intrinsic difficulty of items with difficulty caused by extraneous cognitive load (Sweller, 1994), e.g., poor wording of a question, the format of presentation. Different forms of difficulty require different actions: intrinsic difficulty should be taken into account in KC modeling, difficulty caused by extraneous cognitive load should lead to the improvement of items. Differentiating these two kinds of difficulty in an algorithmic way is, however, nontrivial. One possible approach is to utilize the item discrimination factor as used in item response theory (De Ayala, 2013).

3.3 Type of Knowledge Component

As mentioned before, the KLI framework highlights the point that instructional methods should reflect the types of knowledge component, not domains (Koedinger et al., 2012; Reigeluth & Carr-Chellman, 2009a). Within one domain we can have several different types of knowledge components, for example, in English as a second language we have vocabulary KCs (facts), but also grammar KCs about the usage of tenses (rules). In mathematics, typical KCs are rules (e.g., fraction comparison, solving equations), but some aspects have to be remembered in the long-term memory to achieve fluency, and these can be treated as facts (e.g., one-digit multiplication). The type of KC has consequences for the suitable choice of instructional methods (as discussed in Section 2.3), and also for the choice of a student modeling approach. For example, the Bayesian knowledge tracing model can be used for modeling learning of rules, but it is not directly usable for modeling learning of facts since it uses a strong binary assumption about the state of knowledge, which is not suitable for factual knowledge (Pelánek, 2017).

It is thus advantageous to have KCs homogeneous with respect to the type of content and to have the type of KC explicitly specified within the domain model. For example, a KC that contains facts to be remembered should not contain rules to be understood, and the other way around. However, this is not always easy to achieve. Even within a single KC, we can have different types of items. This occurs particularly in grammar, where we often encounter a mix of rules and idioms (facts to be memorized), e.g., past tense in English, which combines rules for regular verbs and facts about irregular verbs, or English prepositions. From the conceptual point of view, this situation can be solved by splitting the KC into more fine-grained and "cleaner" components, but such splitting has the disadvantages discussed before. Moreover, students often have expectations that correspond to these mixed components.

3.4 Making KC Decisions

The choice of knowledge components is a difficult step that requires balancing pedagogical and pragmatic considerations. In a practical application, it is an iterative process, which can significantly benefit from insights from data. An appropriate balance of different considerations depends on the setting of a particular project. For a project with a large budget and long-term outlook, it may be feasible to base KCs on a detailed cognitive analysis of a domain by experts. In order to achieve scalability in more modest settings, pragmatics and data insights have to play the central role.

In all cases, the basic outline of KCs has to be based on pedagogical considerations: pedagogical intuition, textbook conventions within a domain, at least basic homogeneity of types of KCs. A deeper, but also much more time-intensive (and thus expensive) approach is to use cognitive task analysis to analyze the domain (Clark et al., 2007).

A pragmatic view of KC definitions is concerned mainly with the size of knowledge components: How many items should belong to a KC? Within one practice session, we do not want students to see the same item multiple times[2]. Assuming that a practice session takes at least a few minutes, we can get some coarse lower bound estimates on the size of KCs. For simple items that take less than 10 seconds (e.g., multiple-choice questions), we need at least 40 items for a meaningful KC. For more complex items (e.g., word problems), we need at least 15 items.

The choice of KCs is a difficult process and can benefits greatly from iteration based on insight from data collected by the application. The primary

[2] In the case of facts it is meaningful to practice the same fact multiple times in a short sequence. But even in this case the question could be presented in slightly different form and thus correspond to different items.

use of data analysis is to focus on principles—we can find ways to improve definitions of KC by analyzing the difficulty of items, comparing results of different student models, or studying the similarity of students' performance on items (Řihák & Pelánek, 2017). Specific examples of such analysis are given by "closing the loop" studies (Koedinger et al., 2013; Liu et al., 2014; Cen et al., 2007; Koedinger & McLaughlin, 2016). More pragmatic use of data is to focus on popularity. If some KC receives significant attention from students, it deserves the attention of the system authors as well, e.g., in the form of labor-intensive manual improvement of items within the KC.

4 Relations among Knowledge Components

Once we choose knowledge components, we need to specify relations among them.

4.1 Modeling KC Relations

Researchers have described many approaches that can be used for modeling KC relations. One general approach for describing domain models is to use an ontology (Al-Yahya et al., 2015). Another systematic approach is the use of Bayesian networks (Millán et al., 2010; Conati et al., 2002; Käser et al., 2014; Carmona et al., 2005; Käser et al., 2013), which are used together with student models for skill estimation. Another approach based on formal foundations is the knowledge space theory (Doignon & Falmagne, 2012); this approach is useful particularly for modeling prerequisite relations.

These research works provide principled ways of dealing with KC relations. From the practical perspective, however, they are quite complex and difficult to use in a practical system, although it should be noted that for example a commercial system ALEKS is based on the knowledge space theory (Stillson & Alsup, 2003). A realistic system typically contains hundreds of KCs—it is difficult to build and maintain a Bayesian model or full-fledged ontology of this size.

The basis of KC relations is typically the modeling of a *hierarchy* (taxonomy) of KCs. Typically, KCs form a natural hierarchy. This hierarchy is necessary at least for the organization of KCs in the user interface and navigation within the system. The relevant practical question is whether it is sufficient to represent relations using a tree data structure or whether a more complex approach is necessary, e.g., a directed acyclic graph, a weighted graph, a general ontology. A specific example, where a tree is insufficient, is the KC "equations with fractions". Should this KC be placed in the "equations" subtree or the "fractions" subtree? Thus, from a principled point of view, the tree structure is not entirely sufficient. However, thanks to its simplicity it has many advantages in practical development. So unless there are numerous exceptions which do not fit into the tree structure, it may be beneficial to use a simple tree and ignore the exceptions or solve such cases in an ad-hoc fashion.

A hierarchy captures subsumptive relations. Another kind of relations are *prerequisite relations*, i.e., one KC requires (should be preceded by) another KC. We may also specify *follow-up relations*, i.e., one KC should follow after another; follow-up relations are often inverse to prerequisite relations, but it may be beneficial to specify them separately. These relations are important mainly for rules. For facts, prerequisites are typically not necessary, but even for these KCs, it may be useful to have "soft" versions of prerequisite and follow-up relations that can be used for navigation and recommendations of content, e.g., after learning fruit vocabulary, it is natural to recommend vegetable vocabulary.

Another practical approach that leads to indirect relations among KCs is tagging of KCs, specifically with information about the relevant audience (e.g., school grade, the age of students) or relations to external standards (e.g., national curricula, Common European Framework of Reference for Languages). It may also be possible to use collaborative tagging by users to create a folksonomy (Bieliková et al., 2014).

4.2 Relations Through Shared Items

In addition to explicit relations, knowledge components can be related implicitly through shared items. For example, consider an item "$87 - 6 \times 7$". This item can be naturally mapped to "one-digit multiplication" and "subtraction under 100". Although such mapping is natural, it brings nontrivial practical problems. An important decision in domain modeling is thus whether we allow items to belong to multiple knowledge components ($N : M$ mapping between items and KCs), or whether we use a simpler approach where each item belongs to a single knowledge component ($1 : N$ mapping).

The assignment of items to multiple knowledge components is well-studied in the research literature. A standard approach to specifying the mapping of items to multiple knowledge components is a Q-matrix (Tatsuoka, 1983). There exists extensive research focused on specifying and refining these matrices (Barnes, 2005; Desmarais, 2011). However, many student modeling approaches (e.g., most variants of Bayesian knowledge tracing) consider only a single knowledge component per item (Pelánek, 2017). The major problem with multiple KCs per item is the "credit/blame assignment problem". When a student answers an item correctly (incorrectly), which knowledge components should be assigned credit (blame) for the answer? The answer to this question is difficult and depends on the type of concerned items and knowledge components. Researchers have proposed many approaches to student modeling with Q-matrices, including compensatory (additive) model (Ayers & Junker, 2006), conjunctive (product) model (Cen et al., 2008; Koedinger et al., 2011; Beck et al., 2008), logistic regression (Xu & Mostow, 2012), or taking the weakest skill (Gong et al., 2010). Moreover, the research literature focuses mainly on student modeling but neglects practical issues of Q-matrix application. How should the Q-matrix be used for guiding the adaptive behavior of a learning system, e.g., in the context of mastery learning, spaced repetition, or interleaving of items?

From the practical perspective, it is much simpler to use the basic approach "each item belongs to a single KC". For many items, this approach is entirely sufficient. Items that naturally fit into multiple KCs can be treated by using integrative KCs (discussed below), or by duplicating them into individual KCs. The duplication approach even has some advantages—difficulty of items can sometimes depend on the context in which they appear; this effect is easier to detect and take into account when we use of duplicated items.

4.3 Combining Knowledge Components

Elementary KCs can be combined into more complex KCs in two basic ways:

– an integrative knowledge component, which can be viewed as a "sequential composition of KCs": solving an item requires the application of all constituent KCs,
– a union of knowledge components, which can be viewed as a "parallel composition of KCs": solving an item requires the application of only one constituent KC.

Integrative knowledge components (Koedinger et al., 2012) correspond to a combination of two (or more) KCs that is more difficult than just a sum of the parts. A specific example is the composition effect in mathematics story problems (Heffernan & Koedinger, 1997; Koedinger & McLaughlin, 2010). The integrative effect is also crucial in learning programming, where students often manage the basic control structures (conditions, loops), but struggle to combine these structures; Huang et al. (2016) discuss a modeling approach for this case.

Some integrative effect is present nearly always, but only sometimes it is strong enough to warrant special attention. The effect of integrative KCs could be captured by using advanced student modeling techniques. From a practical perspective, however, it may be better to use a more pragmatic approach: adding an ad-hoc integrative KCs when the integration effect is strong and ignoring the effect otherwise.

The second type of combination is the union of KCs, which serves mainly for the purpose of *interleaved practice*—combining practice from different KCs supports learning of "knowing when to apply what", which is useful particularly for rules. Practical examples of such combined KCs are English tenses (combining practice for past, present perfect, and present tense) or computing area in geometry (using different formulas for different objects). The pedagogical usefulness of interleaved practice is well-supported by research comparing interleaved and blocked practice (Taylor & Rohrer, 2010; Rau et al., 2010). This kind of KC combination can be practically realized simply by specifying the combined KC as a union of elementary KCs. The usage of such a KC can

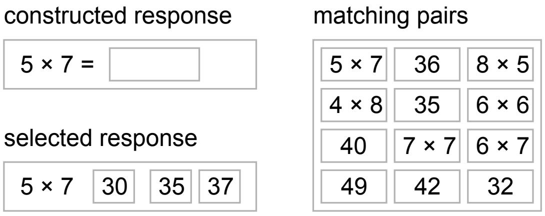


Fig. 1 Example of different activities. Each of these activities can be used without timing, with a strict time limit, or with "reward" based on the speed of response.

be treated by an algorithm for suitably interleaved selection of items from individual KCs, but even random sampling often achieves sufficient interleaving effect.

4.4 Different Practice Activities

The same topic can often be practiced in several ways. Consider, for example, one-digit multiplication or vocabulary about fruits. These topics can be practiced using constructed-response items, selected response items, the pairing of cards, or some game-like activity with timing limits (see Fig. 1 for illustration). Each of these types of practice has its advantages and disadvantages. For example, answering multiple choice questions often requires only recognition of answers (instead of recall). However, answering multiple-choice questions is faster; in our data students are 2 to 3 times faster compared to the same items with a constructed response. This difference can be more pronounced on mobile devices. Moreover, research shows that when multiple-choice questions are used with plausible distractors, they can also exercise retrieval processes (Little et al., 2012). Students also have different preferences for practice activities, and many prefer to alternate different types of activities.

For these reasons, it is beneficial to have several types of activities in a learning system. This leads to practical problems in domain modeling that have not gotten much attention in research so far. How should we treat different practice activities in domain modeling? Should the different activities for one-digit multiplication illustrated in Fig. 1 be considered as a single KC or as different KCs? On the one hand, the performance in different activities for the same topic is clearly closely related. On the other hand, student performance on different activities may nontrivially differ—game-like activities requiring fast reactions require quick recognition of correct answers, whereas constructed-response exercise requires recall without speed.

The relation between different activities of the same topic definitely needs to be captured in the domain models so that the learning system can provide students with meaningful navigation and recommendations. It is not clear when and how these relations should be used for the estimation of students' skills. We consider this to be an interesting open question in student modeling.

5 Creating and Managing Items

Having discussed knowledge components and their relations, we now look at individual items that make up these knowledge components. We consider issues concerning the creation and management of items, again with a focus on scalability.

5.1 Templates versus Separate Items

There are two basic approaches to specifying items within a KC: we can either specify them individually or generate them dynamically using a template. Templates are most directly applicable in mathematics—instead of specific numbers, problems contain variables, which are dynamically instantiated by numbers from a specified interval.

The use of templates is also called automatic item generation or item models (Attali, 2018). An advantage of templates is that we obtain a large number of items with easy management as any changes are easy to incorporate. Templates, however, do not provide a universal approach, since they may be difficult to specify. Templates are natural in mathematics, whereas in other domains their usage is often impossible or impractical—consider for example grammar exercises. Another disadvantage is that the properties of items generated from a single template can significantly differ. For example, in comparison of fractions, "$\frac{1}{2} > \frac{2}{10}$" is much easier than "$\frac{4}{6} > \frac{5}{8}$". When templates are used, such differences are typically ignored, which has a negative impact on skill estimation and user experience.

Individually specified items are universally applicable, and we can analyze the properties of specific items. The obvious disadvantage is that this approach is very labor-intensive.

A practical, hybrid approach is to use a template to generate a set of items and then fix these items for use within the application. This approach is more easily and widely applicable than full-fledged templates—in this case, templates can be heuristic, since special cases can be covered or corrected manually. Since items are fixed, we can now analyze their properties and take them into account for refinement of KCs. A disadvantage, compared to fully dynamic templates, is more difficult management, e.g., making small presentation changes to items is more cumbersome.

5.2 Item Analysis

Analysis of items and students' performance can provide us with useful insights. The primary analysis is the difficulty of items, which can be measured in different ways, depending on the specific type of items, e.g., success rate, median response time, the average number of attempts, hint usage rate.

A practical approach to domain modeling is the use of measures of item similarity. Researchers have proposed measures of item similarity based both

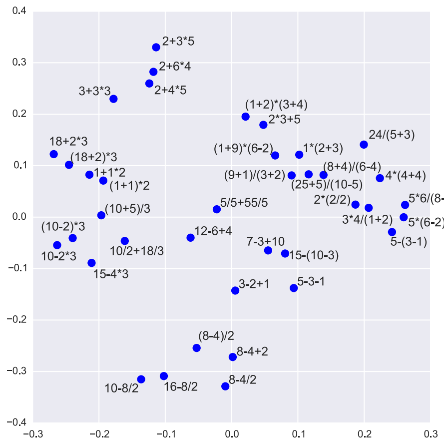


Fig. 2 Example of item projection (PCA projection based on content similarities computed by a Levensthein distance).

on the content of items (Hosseini & Brusilovsky, 2017) and students' performance data (Řihák & Pelánek, 2017). Using item similarity, we can create visualizations of the items pool (projections into the 2D plane), compute item clusters, or identify outlying items. These results are useful for the refinement of knowledge components, deleting duplicated or outlying items, and identifying suitable candidates for new items. Fig. 2 shows an example of such a projection.

Similarities of items can also be included directly into the domain model, possibly in a discretized fashion by keeping data only about closely similar items. These data can be used for item sequencing. There are many possible instructional strategies for content sequencing (Reigeluth & Keller, 2009), e.g., concrete-abstract sequencing, easy-difficult sequencing, or spaced repetition. As a basic step, we want to avoid consecutive items that are too similar. For example, in a practice of geography facts, an item "Where is Poland?" is not suitable directly after an item "What is the name of this country?" with a correct answer Poland. Another possible use of similarity is for recommendations, e.g., similarity relations can be used to recommend worked-out examples, which can serve as hints (Hosseini & Brusilovsky, 2017).

5.3 Automatic Creation and Reuse of Items

Manual creation of items is labor-intensive and error-prone. Automation of the item creation process is welcomed since it improves the efficiency and effectiveness of the instruction. The suitability of a particular type of exercise for automatic item creation can thus be seen as one of the conditions that drive the choice of instructional methods (Reigeluth & Carr-Chellman, 2009b).

Creation of items can be partially automated with the use of above-mentioned item templates. Researchers have also proposed techniques for completely automatic item generation, e.g., generating multiple-choice questions from texts (Wang et al., 2018; Agarwal & Mannem, 2011). However, the practical realization of these techniques is difficult, and even state-of-the-art techniques struggle to achieve items of comparable quality as those produced by humans.

A pragmatic approach is to generate new items from manually created seed items, specifically to transfer items into new contexts (different practice activities). We provide an example of a particular workflow:

1. Create an initial set of items with a constructed response.
2. Collect student answers to these items and find the most common wrong answers.
3. Create multiple-choice questions where distractors are the common wrong answers.
4. Reuse multiple-choice questions for additional practice activities, e.g., game-like exercises where students need to select an answer quickly.

Only the first step requires manual effort; all other steps can be fully automated. The third step is based on the observation that wrong answers have a very skewed distribution (Wang et al., 2015; Pelánek & Řihák, 2016), i.e., many students produce the same, common wrong answer. Such a wrong answer is typically a good distractor for a multiple-choice question (see Table 4 for examples).

Table 4 Examples of most common wrong answers from several mathematics knowledge components.

item	correct answer	most common wrong answer
$12-6+4$	10	2
$8-(2-5)$	11	5
24×3	72	8
$2+6\times 4$	26	32
$2.05+1.1$	3.15	3.06
$9.8+0.7$	10.5	9.15

6 Discussion and Conclusions

Finally, we conclude with a high-level discussion of the main topics of the paper and their consequences for both research and practice.

6.1 Don't Be Stupid

Our idealistic aim in the development of adaptive learning systems is to make them "intelligent". In reality, however, it is quite nontrivial even to avoid looking "stupid". The pursuit of principled, clean solutions is often at the expense of more essential improvements at other parts of the system, particularly in cases of systems with a modest budget. As predicted by instructional design theory (Reigeluth & Carr-Chellman, 2009b; Honebein & Honebein, 2015), constraints such as small target audience, a small development team with a modest budget, and specialized content areas, force designers to make trade-offs and sacrifices. Thus, the first step in not being stupid is to clearly understand the design situation so that pragmatic solutions, like those that we have suggested in this paper, are embraced and valued by stakeholders.

The development of an intelligent learning system is difficult. For an adaptive learning system to be successful, it has to incorporate many aspects including high-quality items, domain model, student model, instructional methods, and user interface. The overall impression of the system is to a large degree influenced by the weakest link. It is not very useful to have a state-of-the-art Bayesian student model, while the support for spaced repetition is entirely missing.

Related arguments have been made in previous works. McNee et al. (2006) proposed the "don't look stupid" principle in the context of recommender systems; in personalized systems, the first step is to avoid actions that look wrong to the user. Baker (2016) calls for "stupid tutoring system, intelligent humans"—the use of relatively simple methods, which are iteratively refined using data and experience. Aleven et al. (2016) call this approach "design-loop adaptivity" and argue that it is an important kind of adaptivity.

6.2 Recommendations for System Development

Based on the presented review of research and our practical experience with the development of a practically used adaptive learning system, we provide several recommendations for the development and management of a domain model:

- Base the choice of knowledge components (and their granularity) on a mix of pedagogical and domain knowledge, practical considerations (particularly the number of items available), and data analysis.
- Map each item into a single knowledge component, i.e., avoid sharing items among several knowledge components (using Q-matrix and similar methods).
- Consider the difficulty of items since it often varies widely even among closely related items. There are several approaches to addressing item difficulty. A simple solution is to split knowledge components by difficulty and to ignore difficulty within these split knowledge components.
- In capturing relations among knowledge components, focus on the basic taxonomy of knowledge components (the subsumption relation) and the prerequisite relation. Avoid complex Bayesian models and full-fledged ontologies, which are too expensive to develop and maintain.

- Automate the creation of items when feasible. Use item templates as a heuristic for item creation, but not as a tool for dynamically serving items to students.
- Consider different practice activities for the same topic.
- Revise the domain model periodically based on data about students' performance. Focus primarily on simple types of analysis that can be used widely across all types of knowledge components.

These recommendations try to capture a suitable trade-off between complex issues involved in the development of learning systems. The usefulness and applicability of these recommendations depend on a specific application. Our review and experience are relevant mainly for systems focusing on the practice of recall, recognition, classification, and execution concerning factual, conceptual, and procedural knowledge, particularly when developed under a limited budget. It is definitely useful to challenge these recommendations in future research, e.g., by developing techniques based on Q-matrix or hierarchical Bayesian models that are easily scalable and applicable in practical settings.

6.3 Research-Practice Gap

Researchers often choose a single aspect of adaptive educational technology and study this aspect in great depth. Practitioners need to cover a wide breadth of issues and must often opt for shallow solutions. The aim of this paper is to help bridge these two perspectives since both sides can benefit from each other.

For practitioners, it is useful to be aware of related research. Whenever it is realistic to implement research-based solutions, these should be preferred as they lead to both better behavior and better scalability than ad-hoc solutions. Some approaches described in research papers, like the use of the Elo rating system in educational technology (Pelánek, 2016), are simple to implement and achieve good accuracy. Closing the loop studies (Koedinger et al., 2013; Liu et al., 2014; Cen et al., 2007; Koedinger & McLaughlin, 2016) provide a specific example of research with direct practical application. In other cases, the research literature provides a useful warning. For example, the basic idea of mapping items to multiple knowledge components seems natural and quite simple; however, the literature on the "credit/blame assignment problem" presents a warning—it may be quite challenging to use the mapping in the system.

For researchers, it is useful to be aware of issues faced by developers of practical systems and properties of real data. The (hidden) assumptions used in research papers may not be satisfied in practical systems, often in a significant way. Practical issues often present interesting research problems, e.g., the question of modeling student skills across different practice activities is not sufficiently addressed by current student modeling techniques. It would also be useful to explore in more detail applicability of techniques proposed in existing work, with respect to both instructional considerations (e.g., along the lines of the Knowledge-Learning-Instruction framework (Koedinger et al., 2012)) and practical considerations (e.g., to consider not only the question whether a proposed technique can be efficient for learning, but also when it is cost-effective to apply). Finally, considerations of scalability and applicability of techniques under limited budgets can provide interesting impulses for the development of novel techniques.

Acknowledgement

The author thanks Petr Jarušek, the chief developer of Umíme systems, for long-term developement and thorough discussions, on which this paper is based. The author also thanks members of the Adaptive Learning group at Masaryk University for their insights and feedback.

References

Abyaa, A., Idrissi, M. K., & Bennani, S. (2019). Learner modelling: systematic review of the literature from the last 5 years. *Educational Technology Research and Development*, pages 1–39.

Agarwal, M. & Mannem, P. (2011). Automatic gap-fill question generation from text books. In *Proceedings of innovative use of NLP for building educational applications*, pages 56–64. Association for Computational Linguistics.

Al-Yahya, M., George, R., & Alfaries, A. (2015). Ontologies in e-learning: review of the literature. *International Journal of Software Engineering and Its Applications*, 9(2):67–84.

Aleven, V., McLaughlin, E. A., Glenn, R. A., & Koedinger, K. R. (2016). *Handbook of research on learning and instruction*, chapter Instruction based on adaptive learning technologies. Routledge.

Anderson, L. W., Krathwohl, D. R., Airasian, P. W., Cruikshank, K. A., Mayer, R. E., Pintrich, P. R., Raths, J., & Wittrock, M. C. (2000). *A taxonomy for learning, teaching, and assessing: A revision of Blooms taxonomy of educational objectives, abridged edition*. Pearson.

Attali, Y. (2018). Automatic item generation unleashed: An evaluation of a large-scale deployment of item models. In *Proceedings of artificial intelligence in education*, pages 17–29. Springer.

Ayers, E. & Junker, B. (2006). Do skills combine additively to predict task difficulty in eighth grade mathematics. In *Proceedings of educational data mining*.

Baker, R. S. (2016). Stupid tutoring systems, intelligent humans. *International Journal of Artificial Intelligence in Education*, 26(2):600–614.

Barnes, T. (2005). The q-matrix method: Mining student response data for knowledge. In *Proceedings of american association for artificial intelligence 2005 educational data mining workshop*, pages 1–8.

Beck, J. E., Pardos, Z. A., Heffernan, N. T., & Ruiz, C. (2008). The composition effect: Conjunctive or compensatory? an analysis of multi-skill math questions in its. In *Proceedings of educational data mining*, pages 147–156.

Bieliková, M., Šimko, M., Barla, M., Tvarožek, J., Labaj, M., Móro, R., Srba, I., & Ševcech, J. (2014). ALEF: From application to platform for adaptive collaborative learning. In Manouselis, N., Drachsler, H., Verbert, K., & Santos, O., editors, *Recommender systems for technology enhanced learning*, pages 195–225. Springer.

Bloom, B. S., Engelhart, M. B., Furst, E. J., Hill, W. H., & Krathwohl, D. R. (1956). *Taxonomy of educational objectives. The classification of educational goals. Handbook 1: Cognitive domain.* Longmans Green, New York.

Brusilovsky, P. (1998). Methods and techniques of adaptive hypermedia. In *Proceedings of adaptive hypertext and hypermedia*, pages 1–43. Springer.

Bull, S. & Kay, J. (2007). Student models that invite the learner in: The smili: Open learner modelling framework. *International Journal of Artificial Intelligence in Education*, 17(2):89–120.

Carmona, C., Millán, E., Pérez-de-la Cruz, J.-L., Trella, M., & Conejo, R. (2005). Introducing prerequisite relations in a multi-layered bayesian student model. In *Proceedings of user modeling*, pages 347–356. Springer.

Cen, H., Koedinger, K., & Junker, B. (2006). Learning factors analysis–a general method for cognitive model evaluation and improvement. In *Proceedings of intelligent tutoring systems*, pages 164–175. Springer.

Cen, H., Koedinger, K., & Junker, B. (2008). Comparing two irt models for conjunctive skills. In *Proceedings of intelligent tutoring systems*, pages 796–798. Springer.

Cen, H., Koedinger, K. R., & Junker, B. (2007). Is over practice necessary?-improving learning efficiency with the cognitive tutor through educational data mining. In *Proceedings of artificial intelligence in education*, pages 511–518.

Churchill, D. (2007). Towards a useful classification of learning objects. *Educational Technology Research and Development*, 55(5):479–497.

Clark, R. E., Feldon, D., vanMerrienboer, J., Yates, K., & Early, S. (2007). *Handbook of research on educational communications and technology*, chapter Cognitive Task Analysis. Lawrence Erlbaum Associates.

Conati, C., Gertner, A., & Vanlehn, K. (2002). Using bayesian networks to manage uncertainty in student modeling. *User Modeling and User-Adapted Interaction*, 12(4):371–417.

Corbett, A. T. & Anderson, J. R. (1994). Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Modeling and User-Adapted Interaction*, 4(4):253–278.

De Ayala, R. J. (2013). *The theory and practice of item response theory.* Guilford Publications.

Desmarais, M. (2011). Conditions for effectively deriving a q-matrix from data with non-negative matrix factorization. In *Proceedings of educational data mining*, pages 41–50.

Dicheva, D., Dichev, C., Agre, G., & Angelova, G. (2015). Gamification in education: A systematic mapping study. *Journal of Educational Technology & Society*, 18(3).

Doignon, J.-P. & Falmagne, J.-C. (2012). *Knowledge spaces.* Springer Science & Business Media.

Feng, M., Heffernan, N., Mani, M., & Heffernan, C. (2006). Using mixed-effects modeling to compare different grain-sized skill models. In *Proceedings of educational data mining.*

Gagné, R. (1985). *The conditions of learning and theory of instruction.* Holt, Rinehart and Winston.

Gong, Y., Beck, J. E., & Heffernan, N. T. (2010). Comparing knowledge tracing and performance factor analysis by using multiple model fitting procedures. In *Proceedings of intelligent tutoring systems*, pages 35–44. Springer.

González-Brenes, J., Huang, Y., & Brusilovsky, P. (2014). General features in knowledge tracing: Applications to multiple subskills, temporal item response theory, and expert knowledge. In *Proceedings of educational data mining*, pages 84–91.

Heffernan, N. T. & Koedinger, K. R. (1997). The composition effect in symbolizing: The role of symbol production vs. text comprehension. In *Proceedings of the nineteenth annual conference of the cognitive science society*, pages 307–312.

Honebein, P. C. & Honebein, C. H. (2015). Effectiveness, efficiency, and appeal: Pick any two? the influence of learning domains and learning outcomes on designer judgments of useful instructional methods. *Educational Technology Research and Development*, 63(6):937–955.

Hosseini, R. & Brusilovsky, P. (2017). A study of concept-based similarity approaches for recommending program examples. *New Review of Hypermedia and Multimedia*, 23(3):161–188.

Huang, Y., Guerra-Hollstein, J. D., & Brusilovsky, P. (2016). Modeling skill combination patterns for deeper knowledge tracing. In *Proceedings of user modeling adaptation and personalization (extended proceedings).*

Käser, T., Busetto, A. G., Solenthaler, B., Baschera, G.-M., Kohn, J., Kucian, K., von Aster, M., & Gross, M. (2013). Modelling and optimizing mathematics learning in children. *International Journal of Artificial Intelligence in Education*, 23(1-4):115–135.

Käser, T., Klingler, S., Schwing, A. G., & Gross, M. (2014). Beyond Knowledge Tracing: Modeling Skill Topologies with Bayesian Networks. In *Proceedings of intelligent tutoring systems*, pages 188–198.

Koedinger, K. & McLaughlin, E. (2010). Seeing language learning inside the math: Cognitive analysis yields transfer. In *Proceedings of the annual meeting of the cognitive science society.* Cognitive Science Society.

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

Koedinger, K. R. & McLaughlin, E. A. (2016). Closing the loop with quantitative cognitive task analysis. In *Proceedings of educational data mining*,

pages 412–417.
Koedinger, K. R., Pavlik Jr, P. I., Stamper, J. C., Nixon, T., & Ritter, S. (2011). Avoiding problem selection thrashing with conjunctive knowledge tracing. In *Proceedings of educational data mining*, pages 91–100.
Koedinger, K. R., Stamper, J. C., McLaughlin, E. A., & Nixon, T. (2013). Using data-driven discovery of better student models to improve student learning. In *Proceedings of artificial intelligence in education*, pages 421–430. Springer.
Koedinger, K. R., Yudelson, M. V., & Pavlik, P. I. (2016). Testing theories of transfer using error rate learning curves. *Topics in Cognitive Science*, 8(3):589–609.
Little, J. L., Bjork, E. L., Bjork, R. A., & Angello, G. (2012). Multiple-choice tests exonerated, at least of some charges: Fostering test-induced learning and avoiding test-induced forgetting. *Psychological Science*, 23(11):1337–1344.
Liu, R., Koedinger, K. R., & McLaughlin, E. A. (2014). Interpreting model discovery and testing generalization to a new dataset. In *Proceedings of educational data mining*, pages 107–113.
McNee, S. M., Kapoor, N., & Konstan, J. A. (2006). Don't look stupid: avoiding pitfalls when recommending research papers. In *Proceedings of computer supported cooperative work*, pages 171–180. ACM.
Millán, E., Loboda, T., & Pérez-de-la Cruz, J. L. (2010). Bayesian networks for student model engineering. *Computers & Education*, 55(4):1663–1683.
Molenaar, I. & Knoop-van Campen, C. (2017). Teacher dashboards in practice: Usage and impact. In *Proceedings of european conference on technology enhanced learning*, pages 125–138. Springer.
Nkambou, R., Mizoguchi, R., & Bourdeau, J. (2010). *Advances in intelligent tutoring systems*, volume 308. Springer Science & Business Media.
Pardos, Z. A. & Heffernan, N. T. (2011). KT-IDEM: Introducing item difficulty to the knowledge tracing model. In *Proceedings of User Modeling, Adaption and Personalization*, pages 243–254. Springer.
Pardos, Z. A., Heffernan, N. T., Anderson, B., Heffernan, C. L., & Schools, W. P. (2010). *Handbook of educational data mining*, chapter Using fine-grained skill models to fit student performance with Bayesian networks, pages 417–426. Chapman & Hall/CRC Press.
Pelánek, R. (2016). Applications of the Elo rating system in adaptive educational systems. *Computers & Education*, 98:169–179.
Pelánek, R. (2017). Bayesian knowledge tracing, logistic models, and beyond: an overview of learner modeling techniques. *User Modeling and User-Adapted Interaction*, 27(3):313–350.
Pelánek, R. (2018). The details matter: methodological nuances in the evaluation of student models. *User Modeling and User-Adapted Interaction.*
Pelánek, R. (2019). Measuring similarity of educational items: An overview. *IEEE Transactions on Learning Technologies.*
Pelánek, R., Papoušek, J., Řihák, J., Stanislav, V., & Nižnan, J. (2017). Elo-based learner modeling for the adaptive practice of facts. *User Modeling*

and User-Adapted Interaction, 27(1):89–118.

Pelánek, R. & Řihák, J. (2018). Analysis and design of mastery learning criteria. *New Review of Hypermedia and Multimedia*, 24:133–159.

Pelánek, R. & Řihák, J. (2016). Properties and applications of wrong answers in online educational systems. In *Proceedings of educational data mining*.

Porter, A. C. (2002). Measuring the content of instruction: Uses in research and practice. *Educational Researcher*, 31(7):3–14.

Rau, M. A., Aleven, V., & Rummel, N. (2010). Blocked versus interleaved practice with multiple representations in an intelligent tutoring system for fractions. In *Proceedings of intelligent tutoring systems*, pages 413–422. Springer.

Reigeluth, C. M. & Carr-Chellman, A. A. (2009a). Situational principles of instruction. In Reigeluth, C. M. & Carr-Chellman, A. A., editors, *Instructional-Design Theories and Models, Volume III: Building a Common Knowledge Base*, pages 57–68. Routledge.

Reigeluth, C. M. & Carr-Chellman, A. A. (2009b). Understanding instructional theory. In Reigeluth, C. M. & Carr-Chellman, A. A., editors, *Instructional-Design Theories and Models, Volume III: Building a Common Knowledge Base*, pages 3–26. Routledge.

Reigeluth, C. M. & Keller, J. B. (2009). Understanding instruction. In Reigeluth, C. M. & Carr-Chellman, A. A., editors, *Instructional-Design Theories and Models, Volume III: Building a Common Knowledge Base*, pages 27–39. Routledge.

Reigeluth, C. M., Merrill, M. D., & Bunderson, C. V. (1978). The structure of subject matter content and its instructional design implications. *Instructional Science*, 7(2):107–126.

Reigeluth, C. M., Merrill, M. D., Wilson, B. G., & Spiller, R. T. (1980). The elaboration theory of instruction: A model for sequencing and synthesizing instruction. *Instructional Science*, 9(3):195–219.

Řihák, J. & Pelánek, R. (2017). Measuring similarity of educational items using data on learners' performance. In *Proceedings of educational data mining*, pages 16–23.

Roediger, H. L. & Pyc, M. A. (2012). Inexpensive techniques to improve education: Applying cognitive psychology to enhance educational practice. *Journal of Applied Research in Memory and Cognition*, 1(4):242–248.

Sottilare, R. A., Graesser, A. C., Hu, X., Olney, A., Nye, B., & Sinatra, A. M. (2016). *Design Recommendations for Intelligent Tutoring Systems: Volume 4-Domain Modeling*, volume 4. US Army Research Laboratory.

Stillson, H. & Alsup, J. (2003). Smart ALEKS ... or not? teaching basic algebra using an online interactive learning system. *Mathematics and Computer Education*, 37(3):329–340.

Sweller, J. (1994). Cognitive load theory, learning difficulty, and instructional design. *Learning and Instruction*, 4(4):295–312.

Tatsuoka, K. K. (1983). Rule space: An approach for dealing with misconceptions based on item response theory. *Journal of Educational Measurement*, 20(4):345–354.

Taylor, K. & Rohrer, D. (2010). The effects of interleaved practice. *Applied Cognitive Psychology*, 24(6):837–848.

Wang, Y., Heffernan, N. T., & Heffernan, C. (2015). Towards better affect detectors: effect of missing skills, class features and common wrong answers. In *Proceedings of learning analytics and knowledge*, pages 31–35. ACM.

Wang, Z., Lan, A. S., Nie, W., Waters, A. E., Grimaldi, P. J., & Baraniuk, R. G. (2018). QG-net: a data-driven question generation model for educational content. In *Proceedings of learning at scale*, pages 7:1–7:10. ACM.

Xu, Y. & Mostow, J. (2012). Comparison of methods to trace multiple subskills: Is LR-DBN best? In *Proceedings of educational data mining*, pages 41–48.