Oppl et al. International Journal of Educational Technology in Higher Education (2017) 14:2
DOI 10.1186/s41239-017-0039-0

International Journal of Educational Technology in Higher Education

RESEARCH ARTICLE Open Access

A flexible online platform for computerized adaptive testing

Stefan Oppl[1,3*], Florian Reisinger[1], Alexander Eckmaier[1] and Christoph Helm[2]

* Correspondence: stefan.oppl@jku.at
[1]Department of Business Information Systems–Communications Engineering, Johannes Kepler University of Linz, Altenberger Straße 69, Linz 4040, Austria
[3]Department of Software Science, Radboud University, Mailbox 47, PO Box 9010, Nijmegen 6500, GL, The Netherlands
Full list of author information is available at the end of the article

Abstract

Computerized Adaptive Testing (CAT) is a field of research originating in psychometrics that has been adopted in the last years for formative and summative assessment activities in educational settings in general and in online learning in particular. While a variety of platforms is available for designing and deploying CAT the challenge of providing the flexibility in test and item design required for domain-specific assessment formats in education has hardly been addressed so far. The present article introduces a software architecture to fill this gap and enable the development of fully customizable CAT tools with respect to domain-specific item design and visualization as well as deployed CAT algorithms. A prototypical implementation of the architecture and a set of domain-specific item types are presented to demonstrate the feasibility of the proposed approach and outline future directions of development and research.

Keywords: Computerized Adaptive Testing, Learning platform, Domain-specific Testing

Introduction

Computerized Adaptive Testing (CAT) is a concept dating back to the 1970s (Reckase, 1974), and has been applied in educational psychology ever since then. Due to progress in psychometric research and rising capabilities of the technical support platform, the conceptual design and technical development of appropriate testing environments remains a topic of engineering research until today (Kröhne & Frey, 2011). With the advent of the world-wide-web, several platforms have been developed that could administer adaptive tests over the web. Commercial and non-commercial products are available,[1] with the platform Concerto (Scalise & Allen, 2015) probably being the most widely recognized effort in this field.

In recent years, interest has risen to use CAT in the context of online learning processes in order to adapt the difficulty level of proposed learning materials (Salcedo, Pinninghoff, & Contreras, 2005) or to perform summative evaluation of learning outcomes (Guzmán & Conejo, 2005). It is also seen as a potentially important component in Massive Open Online Courses (MOOCs) (Meyer & Zhu, 2013).

Currently available CAT systems focus on items (i.e., questions presented to the examinees) that can be answered dichotomously or on a multi-part scale (e.g., similar to Likert-scales). While this is appropriate for latent-trait-testing, the primary use-case of CAT in psychometrics (Bortolotti, Tezza, Andrade, Bornia, & Sousa Júnior, 2013), the

evaluation of learning outcomes might require more complex and open answering options (Guzmán & Conejo, 2005). Some systems, such as Concerto (Scalise & Allen, 2015) or SIETTE (Conejo et al., 2004), enable such item types by providing means to specify simple HTML forms. However, items requiring the presentation and evaluation of answers comprising of multiple components (e.g. several input fields, locating multiple errors in conceptual drawings such as electronic circuits or process models) are not supported. Furthermore, items containing dynamic elements that rely on user interaction (such as assessing the behavior of a physical system) can only be administered when using third-party technological solutions, e.g., based on Adobe Flash (Triantafillou, Georgiadou, & Economides, 2008) or Java Applets (Conejo et al., 2004), which do not integrate with the testing system.

The present work addresses this limitation and aims at providing a flexible architecture for enabling CAT with arbitrarily complex item types, putting a particular focus on integration in online learning settings. The architecture is designed in a way that does not only allow to alter the types of items, but also the testing strategies, the algorithms for item selection, and the user interface. Using the proposed architecture enables technology-proficient users to integrate CAT in their online learning platform and provides a light-weight, XML-based, item specification format to domain experts responsible for maintaining the item pool.

The remainder of this paper is structured as follows: in the next section, we briefly summarize the properties of CAT to establish the design frame that guides the identification of requirements on the framework to be developed. Related work section gives an account on related work and reviews it with respect to these requirements. In Platform development section, we describe the architecture of our platform and give explain the extension points that are provided for customization. Design of domain-specific items section reports on prototypical instances we have implemented for the field of testing accounting skills in vocational business schools and business process management skills in a bachelor's program in information systems. We close with a discussion of the current status of the framework, its limitations and directions of future research.

Computerized Adaptive Testing for online learning

In order to establish a research framework to examine the design space for a flexible platform for online CAT, we need to draw on the underlying kernel theories. We thus briefly discuss item response theory in the context of psychometric testing to establish the context of our research, before we give an overview about the development of CAT.

These descriptions will provide the foundation deriving the requirements on the platform to be developed at the end of this section.

Item response theory

Item response theory (IRT) is the most commonly used form of psychometric theories today (Chen & Wang, 2010). Its origins date back to Rasch and Lord in the 1950s (ibid.). IRT is a "family of mathematical models that describe how people interact with test items" (Čisar, Radosav, Markoski, Pinter, & Čisar, 2010). It can be used with a

variety of item selection algorithms and scoring procedures. They try to estimate an examinee's skill level and therefore find a connection between an examinee's answers to particular items and their skill level (Chen & Wang, 2010).

These estimation approaches mainly differ in the number of parameters the estimation is based on. The most wide-spread approach, the 1-parameter-logistics-model (or Rasch-model), only requires to determine the *difficulty* level of each item (Reckase, 2010). The remaining parameters–*discrimination* (the amount of information the item provides for skill estimation) and *guessing* (the probability of guessing the right answer)–remain fixed here. 2- and 3-parameter-logistics are considered to give a more exact and/or faster estimation of an examinee's skill level, but require to determine the additional parameters for each item (Reckase, 2010).

CAT process

CAT aims at implementing a testing process, which adjusts to an examinee's skill level via dynamically selecting appropriate testing items (Moosbrugger & Kelava, 2012). The difficulty of the next selected item depends on all previously answered items. The next item should be selected in a way which provides the most information regarding the currently estimated skill level (Linacre, 2000).

Items are drawn from an item pool that contains all items that can be used for testing. The items for a specific test are selected from the item pool based on the examinee's estimated skill level (Veldkamp & Matteucci, 2013). In order to enable this selection, the items need to be calibrated. Calibration refers to the process of determining the parameters for each item that are necessary for skill estimation during testing (Krass & Williams, 2003).

CAT is primarily used in combination with IRT models (Wainer & Mislevy, 2000). Items prepared for use in IRT can take different forms. If an item can be evaluated unambiguously to be either true or false, it is called "dichotomous" (Weiss, 2004). If there are more than two response options, the items are referred to as "polytomous" (Bortolotti et al., 2013). Moreover, IRT models exist for unidimensional CAT (UCAT) as well as for multidimensional CAT (MCAT). "Unidimensionality" refers to the test estimating exactly one skill of the tested person. "Multidimensionality" means that the test estimates multiple person parameters representing more than one skill (H. Lin, 2012). Literature further distinguishes selected-response formats from constructed-response formats to differentiate between items with prepared answers that need to be selected by examinees, from items that provide open answering options. The latter require examinees to specify their answers themselves (Bennet & Ward, 1993).

The CAT-process (cf. Fig. 1) requires that all items of an item pools are calibrated by administering them to examinees with the intent to estimate the associated item parameters in a pre-testing phase. Item pools are usually segmented into intervals (also "bags" or "bins"), to which items are assigned according to their determined difficulty parameter. During testing, the items are selected from these "bags" based on to the currently estimated skill level (Reckase, 2010).

The CAT-process starts with the selection of the first item from the item pool. The selection of the first item cannot rely on information gained from previous answers and

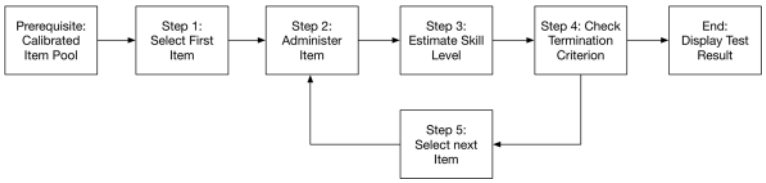

Fig. 1 CAT Process

thus usually either is selected randomly or taken from an item bag of medium difficulty (Veldkamp & Matteucci, 2013).

Secondly, the selected item is administered, i.e., presented, to the examinee. Based on the answer, the current skill level is (re-)estimated using the selected IRT model. Literature proposes choosing mechanisms that attempt to select the item that maximizes the discriminatory value for the given skill level of the examinee (H. Lin, 2012). In general, if an examinee answers correctly, the next question will be slightly more challenging than the previous one, and vice versa.

The final step is to check whether the stopping criterion has been met. If this is not the case, another item is drawn from the pool, administered, and the skill level is computed again. The procedure is repeated until the termination criterion is finally met and the test concludes (Veldkamp & Matteucci, 2013). In "fixed-length testing", a fixed number of items is administered to the examinee. In "variable-length testing", a certain level of measurement accuracy or precision is used as a termination criterion (Segall, 2004). Other criteria for ending a test comprise a maximum test time that has been reached, or if all available items have been consumed. Arbitrary combinations of these criteria can be made (Segall, 2004) following the purpose of the test (ibid.).

Requirements on testing platform support

Considering all of the design dimensions of CAT described above, the premier requirement on a platform enabling to deliver CAT to learners in general is to provide a *flexible* and *configurable* architecture that can be instantiated in a structured way for different use cases. The following generic requirements can be derived from the plethora of approaches proposed to be used in CAT:

- R1: *Flexibility in testing strategy and item pool design*–the platform has to be able to operate with arbitrary testing strategies and different item pool designs.
- R2: *Flexibility in item selection algorithm*–the platform has to be able to use different item selection strategies for the identification of first item and for succeeding items.
- R3: *Flexibility in specifying the termination criterion*–the platform has to allow for flexible specification under which constraints the test is executed und when it is considered to be finished.

For usage in online learning settings, further requirements on the platform can be identified, which are not directly related to CAT:

- R4: *Possibility of technical integration with learning platforms on different layers*–the platform should allow to be integrated in arbitrary learning platform on user interface, functional and/or data layer, enabling external platforms to provide data handing, presentation of tests to users or functional integration with other features of the external platform to support examinees during testing (Pellegrino, 2010).
- R5: *Ability to display and evaluate items stemming from arbitrary domains*–the platform has to support the presentation and evaluation of items that require domain-specific display and data representation. It furthermore has to enable domain-specific ways of interaction for examinees to provide answers (Achtenhagen, 2012).

In the following section, we review related work with respect to these requirements, and identify the gap in the state-of-the-art to be addressed with the present work.

Related work

The challenge of supporting CAT in online educational settings has been addressed in several approaches over the last years. Related work has been identified via two sources. First, the products listed on the website of the International Association for Computerized Adaptive Testing[2] have been considered for inclusion. Those products which are openly available for adaptation for specific educational use cases have been included. Furthermore, we have conducted a literature study in the field of learning technologies that use CAT for formative or summative assessment. Studies have been included that explicitly mention the development of CAT software and describe its features. The review is structured along the aims of the platforms and separates the body of available work in two categories–platforms that focus on web-based delivery of CAT (mainly in the context of psychometrics), and platforms that explicitly use CAT in an educational context. The platforms are qualitatively compared within their categories in the following two sections and subsequently are assessed with respect to the identified requirements.

Platforms for web-based delivery of CAT

The platforms reviewed in this section use the web as a delivery channel for CAT-based psychometric tests. They mainly differ in the flexibility they offer for testing strategies and item selection. Concerto[3] (Scalise & Allen, 2015) is the most flexible platform to that respect. It is designed for stand-alone deployment and uses the catR-library (Magis & Raîche, 2012), which offers different IRT testing strategies, several methods for next item selection, and three stopping rules. Tests and items can be designed by specifying HTML-templates using a drag-and-drop editor. While this allows for easy editing, test answering options are limited to standard HTML input forms. This is a limitation for domain-specific tests, where more complex answering options might be required.

The IRT-CAT[4] project develops a platform, which–in contrast to Concerto–only requires a PHP web-stack. This makes deployment easier, but also limits the flexibility of the platform. Items can only be stored as text-based single-choice questions. Three different IRT-models can be selected when setting up a test.

A single-technology approach has also been chosen in CAT-MD (Triantafillou et al., 2008). It focusses on mobile delivery of tests and has been implemented using the then-state-of-the-art Adobe Flash technology, which today limits its potential for real-world deployment. Its testing framework is of limited flexibility and supports multiple-choice-items and true-false-items that are evaluated using a non-configurable CAT-algorithm based on the Rasch-model.

Learning platforms with CAT features

This section discusses platforms that have been developed from an educational perspective and include CAT features for formative or summative testing.

The SIETTE platform (Conejo et al., 2004; Guzmán & Conejo, 2005) is part of a system that comprehensively supports web-based teaching and learning systems. The testing system–similarly to Concerto–allows to specify items and configure tests via a web-interface. Items can contain dynamically determined parameters (e.g., to change numeric values in a calculation task) and interactive elements (realized by then-state-of-the-art Java Applets) for item presentation and evaluation of provided answers. The system allows to select one of three predefined item selection strategies based in IRT. A similar, yet less flexible, system is proposed by (Huang, Lin, & Cheng, 2009), which provides CAT functionality for eLearning scenarios. It has a modular architecture based on ASP.NET and thus can be integrated with other platforms building on the same technology stack.

The MISTRAL eLearning platform (Salcedo et al., 2005) includes CAT components for formative testing and should allow to determine the optimal learning content for a particular student. It uses an IRT-approach for item selection and differs from other systems in that it considers tasks that need to be assessed by a human due to their qualitative nature. The evaluation results of such items are then manually entered into the skill estimation algorithm.

The issue of integration with external e-learning platforms is addressed in the software framework proposed by Duda and Walter (2012). While they constrain their system to only work with multiple-choice questions, they explicitly consider different types of items and discuss requirements on items that should be evaluated automatically by a technical system. They thus do not focus on a single assessment mechanism, but rather discuss a service-oriented middleware that can be used to flexibly configure the system and integrate it with other platforms.

Discussion

The description of related work shows that all requirements identified in the last section have already been addressed in related work. While this indicates the relevance of the requirements, there is no single approach that meets all of them. Table 1 gives an overview about properties of the approaches discussed above. Empty cells indicate that the requirement is not explicitly addressed.

In the next section, we will address the identified gap by developing an architecture and prototypical implementation of a CAT platform designed for usage in online educational settings.

Table 1 Comparison of related work

	Platform	R1–Flexible testing strategy	R2–Configurable item selection algorithm	R3–Flexible termination criterion	R4–Integration with other platforms	R5–Arbitrary item types and domains
Platforms for web-based delivery of CAT	Concerto	Yes	Yes	Yes	No–standalone platform	Limited–answers necessary via HTML forms
	IRT-CAT		Limited–three pre-specified algorithms		No–standalone platform	
	CAT-MID	No–fixed	No–fixed	No–fixed	Possible via Adobe Flash	Limited–two different item types
Learning Platforms with CAT features	SIETTE	Limited–two different testing modes	Limited–three pre-specified algorithms	Limited–max. number of items can be spec.	No–standalone platform	Yes–via self-assessing Java applets
	(Huang et al., 2009)	No–fixed	No–fixed	No–fixed	Possible because of modular architecture	
	MISTRAL	No–fixed	No–fixed	No–fixed		Discussed, but impl. not elaborated
	(Duda & Walter, 2012)				Possible via service-oriented architecture	Limited–different types of multiple choice

Platform development

Based on the requirements identified in Computerized Adaptive Testing for online learning section, we propose an architecture for a computerized adaptive testing system that can be used in the course of online learning activities. In the following subsection, we describe the architecture and identify, where domain- or application-specific extensions and adaptations can be performed. Afterwards, we revisit the current implementation of the architecture, which relies on catR for item selection. This section closes with a comparison of the platform's features with the requirements identified above.

Architecture

The architecture focuses on providing flexibility in terms of testing domain, testing strategy, data storage, and user interface display. Figure 2 gives an overview about the fundamental components. Components with a dashed outline can be altered according to the intended testing scenario. Components with a grey background need to be adapted to the domain the test is carried out in.

The central component of the architecture is the *Item Management Core*. It provides the central services for coordinating the components necessary to load, start, and carry out a test. It interlinks the other components using interfaces and abstract classes, making them exchangeable without needing to change the implementation of the core.

The *Test Engine* implements the control flow for a test. This comprises the management of the item pool and storing all information necessary to carry out a test (e.g., storing the history of administered items and their answers).

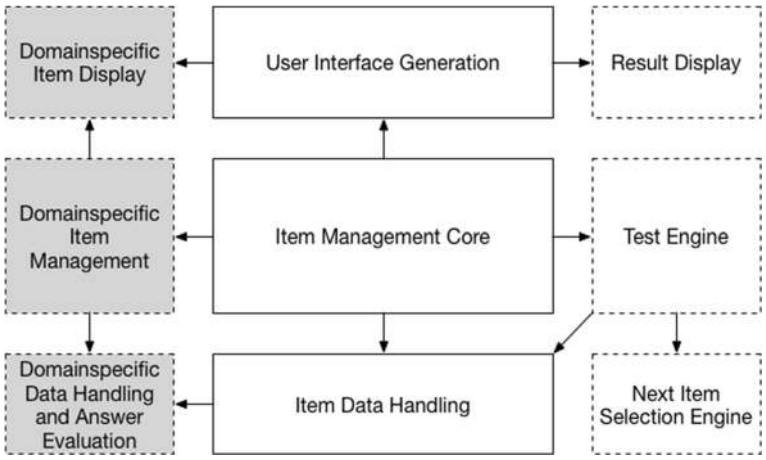

Fig. 2 Architecture of the CAT platform

The *Next Item Selection Engine* implements the adaptiveness of the tests via selecting an item from the item pool. As this selection usually relies on advanced statistical algorithms, the platform here provides an interface to the R software package (R Development Core Team, 2009) that can be used, e.g., in combination with the catR-library (Magis & Raîche, 2012). Using the interface to external software is not required. Adaptive or non-adaptive selection algorithms, such as a fixed branch strategy (Moosbrugger & Kelava, 2012), could also be implemented directly. The platform thus is not restricted to CAT-applications but can be used for testing in general.

The *Item Data Handling* component is used to load and manage the items that are to be used in a test. The test engine accesses this component when retrieving data necessary for the selection of a next item. As a test not necessarily only comprises items of a single type, the ways to address items and evaluate answers given by examinees need to be generic. This component thus consists of a set of interfaces that need to be implemented for domain-specific item types.

User Interface Generation provides the fundamental functionality to render HTML output for the test. It is controlled by the core component and relies on further domain- and application-specific components to provide the test interface to examinees via a web browser. It feeds user inputs back to the core component, which forwards it to the test engine for evaluation.

Result Display is used by the user interface generation component to render the result and feedback page for a completed test. It has been established as a separate component, as the feedback for a particular test is application-specific and requires to include different kinds and amount of information.

The remaining components are domain-specific, i.e., have to be adapted to the domain of the test. *Domain-specific item management* inherits its functionality from the core component and extends it in terms of domain-specific user support measures, i.e., the provision of supporting information that should be made available to examinees when answering the administered items. Furthermore, this component determines the data format of domain-specific items. In this way, items can be retrieved flexibly from different storage formats, such as XML, or from a relational database, but could also be loaded via JSON from remote locations, e.g., via a web-service.

Domain-specific data handling and answer evaluation comprises classes that represent different domain-specific item types. This includes the data necessary to display the question, information about expected answer formats, and procedures to evaluate the correctness of an answer. These classes can also contain polytomous or multi-stage evaluation procedures. The domain-specific item management component loads the available items and instantiates respective item objects. The item objects are then added to the item pool of the test engine via the core component.

Domain-specific item display comprises classes that render an item's question to HTML and provide means of user input to collect the examinees' answers. The management component instantiates the respective objects, when the user interface generation component requests them for display.

The components described above are the main conceptual building blocks of the platform architecture. Its interplay with external components is shown in Fig. 3. Platform implementations can be integrated with external platforms on a data level (for item retrieval), on a functional level (for using externally provided test support resources) and on an UI level (for integrating tests in an external platform's genuine UI).

In the following subsection we demonstrate the use of the architecture in a concrete platform implementation. It is domain-agnostic, i.e., can be adapted for tests with arbitrary item types. Examples for concrete item types are described in Design of domain-specific items section.

Current implementation

The current implementation allows to administer IRT-based CAT usable with dichotomous or polytomous items. It contains implementations of testing routines, data storage and a UI, and thus is designed for stand-alone deployment. The architecture is visualized in the UML class diagram in Fig. 4. In the following, we outline the

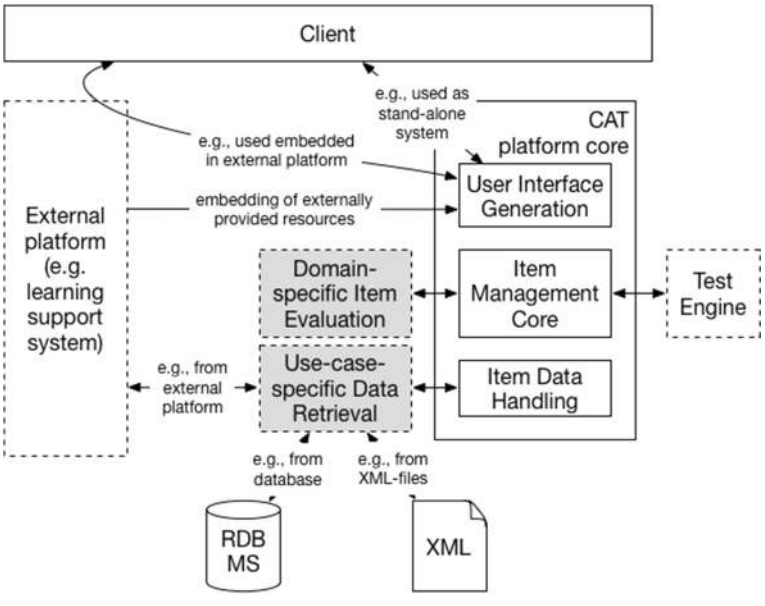

Fig. 3 Interplay of platform architecture with external components

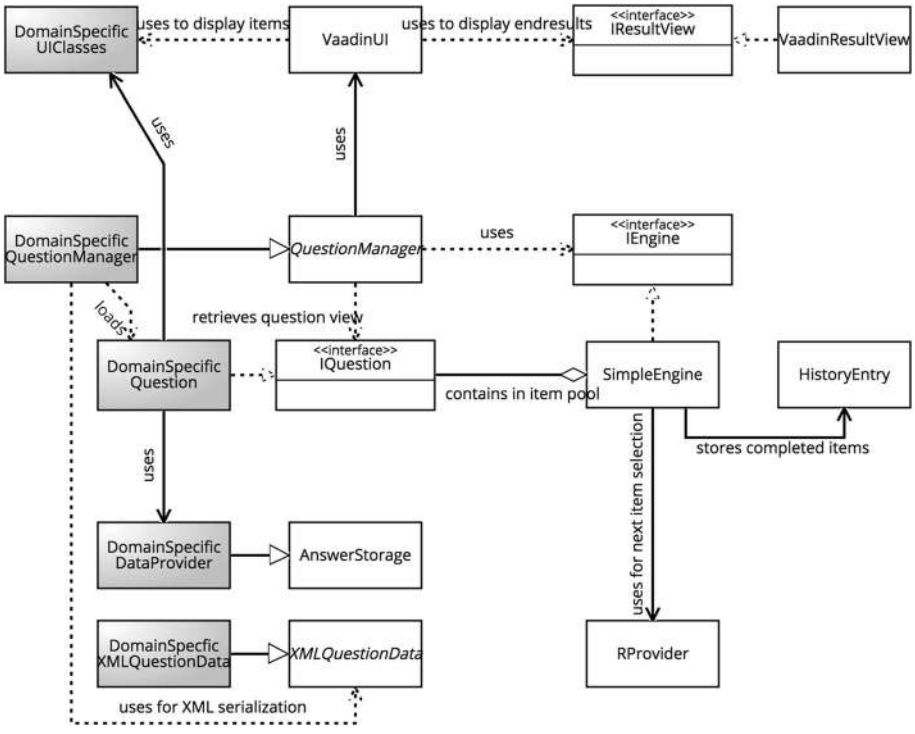

Fig. 4 UML Class diagram of current implementation

properties of this implementation and describe, how it can be adapted to items from a particular domain.

Figure 4 follows the standard notation for UML class diagrams (Rumbaugh, Jacobson, & Booch, 2004). At the same time, its layout roughly resembles the component structure shown in Fig. 2 and described above.

User interface generation

The implementation of the user interface relies on the Vaadin[5] framework. Vaadin focusses on single-page web-apps, i.e., does not reload the page after an item has been answered. This inherently avoids that examinees can use the "back"-button of the browser to revisit a former item. The test state is maintained completely on the server. The single-page approach also allows to show overlays without losing the current state of the test. This can be used to dynamically provide supporting information for an item, e.g., reference materials that might be required to answer an item, or even integrate information or functionality provided by an external learning platform.

Item data handling

Item storage has been implemented using the JAXB-mechanism.[6] JAXB enables a transparent binding between java objects and XML structures. The XML-schema is generated from the (annotated) domain-specific item classes. These schemas can be used for offline preparation of items directly in XML via schema-aware editors.

Test engine & item selection

A test engine (SimpleEngine) has been implemented using an item pool with a flexible number of bags. The number of bags and their difficulty bounds are specified when the engine is instantiated.

During a test, the engine uses the selection component for the next item. In the current implementation, this is realized by interfacing the catR-library (Magis & Raîche, 2012), that is also used in the Concerto platform. catR is used to estimate the required difficulty of the next item to be administered based on the history of answers. The engine is currently configured for use with dichotomous items but can be altered to use all features of catR. The result calculated by catR is used to determine the bag from which the next item should be drawn. The actual item is then drawn randomly from the respective bag.

Discussion of requirements R1-R4

In this section, we have described the architecture of our proposed CAT platform and have introduced a prototypical implementation to be used as a stand-alone implementation for IRT-based testing of dichotomous items. The current version of the source code can be obtained via Github.[7]

The requirements identified in Computerized Adaptive Testing for online learning section have been met by the platform in its current implementation as follows: *R1* and *R3* are met via the exchangeable test engine that can be implemented to handle arbitrary item pools, conduct different testing strategies and check diverse stopping criteria. *R2* has been implemented in the architecture by providing a dedicated interface that item selection modules can implement. In the current implementation, this flexibility is maintained by integrating the catR library that offers different skill level estimation algorithms. *R4* has been met by conceptually separating test management from user interface generation and data storage. In the current implementation, the Vaadin library enables to embed the user interface in any learning platform that is based on Java servlet technology. *R5* is enabled by providing a generic interface for question handling on all levels of architecture from data handling over evaluation logic to user interface generation. In the current implementation, the use of XML for data representation enables domain-specific item types with different data formats for each item.

Focus has been put on maintaining flexibility in terms of applied testing strategies and item types. The ability to administer arbitrary item types in particular is a feature that sets apart our approach from available related work. In the following section, we thus focus on demonstrating the flexibility of the platform in terms of administering item types that differ in content presentation as well as form and amount of user interaction required when providing answers.

Design of domain-specific items

In the former section, we have introduced our architecture and a prototypical implementation with a test engine using catR. We were able to show that the CAT-oriented requirements R1–R3 could be met by our architecture. The technical feasibility of integrating the platform with other systems on presentation or data layer (R4) has been shown in principle by adopting the Vaadin framework for displaying information and XML as one potential form of item data representation.

R4, however, also has a functional integration dimension. The platform should enable to integrate externally provided content or functionality in the test. This can be used to provide domain-specific support measures (e.g., searchable language references for

programming tests, etc.) during tests. Furthermore, the ability to display and evaluate items stemming from arbitrary domains (R5) has yet to be shown. Both requirements relate to the domain-specific part of the architecture. In the following, we thus show how R4 and R5 are addressed in different item types.

Item-type 1: business administration–impact on profit

The first example for an item-type is taken from an item pool designed for testing accounting skills of students in vocational business schools in Austria. Details about the results of these tests are described in (Helm, 2016). The item-type is prototypical for dichotomous items with simple display and answer option requirements, which are also supported in existing platforms.

Aim of item-type

Items of this type confront examinees with a brief textual description of a business transaction and ask them to decide, whether this transaction impacts the profit of the company. There are three potential answering options, of which exactly one is correct for any given business transaction. The items thus are to be considered dichotomous.

User interface

The user interface for this item type (as shown in Fig. 5) comprises the textual description of the business case. The answering options are presented in a drop-down list.

Data representation and assessment

The domain-specific data representation in XML for the item type is shown in Fig. 4 exemplified for the same item as used in Fig. 6.

The XML-question format comprises three second-level tags that remain stable for all item types. The tag dataStorage comprises the data necessary to evaluate the answer of the examinee. The tag question holds the content to be displayed on the user interface. Here, this content is represented by plain text. The tag difficulty holds the parameters necessary for sorting the item into the correct bag in the item pool and for subsequent evaluation of the estimated skill level of the examinee. Items for this test are described using the Rasch-model. If more complex testing models are used, the respective item-data-classes need to be extended with the necessary information types.

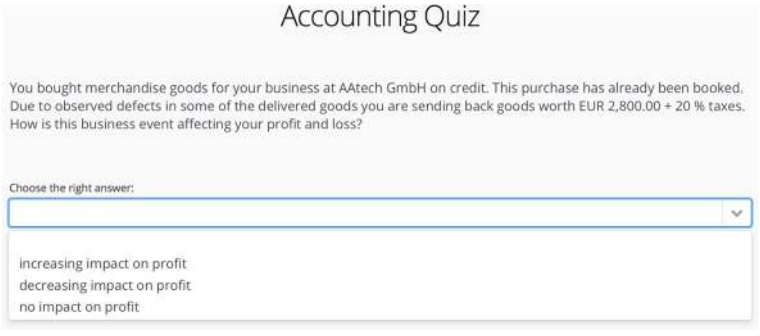

Fig. 5 User interface of item type for assessing a business transaction's impact on profit

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<profitQuestionDataStorage>
    <dataStorage xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="profitDataStorage">
        <answer>Increase</answer>
    </dataStorage>
    <question>
        You bought merchandise goods for your business at AAtech GmbH on credit. This purchase has
already been booked. Due to observed defects in some of the delivered goods you are sending back goods
worth EUR 2,800.00 + 20 % taxes. How is this business event affecting your profit and loss?
    </question>
    <difficulty>-1.28</difficulty>
</profitQuestionDataStorage>
```

Fig. 6 XML for storing data of items for assessing a business transaction's impact on profit

Item-type 2: business administration–accounting

The second item-type is taken from the same test as described above. Both types have been used in combination to assess accounting skill of students in vocational business education schools in Austria (Helm, 2016). The expected answering format of this item-type is more complex. Items follow a constructed-response format, requiring examinees to provide answers in a pre-specified structure.

Aim of item-type

Items of this type again confront examinees with a brief textual description of a business transaction, identical to the items-type described above.

Examinees are required to correctly describe the accounting record in double-entry bookkeeping for the business transaction. They have to identify the affected accounts, assign them to either debit or credit and calculate the respective amount of money to be booked on each account. An answer is only correct, if all components are correctly described (i.e., all affected accounts assigned correctly to either debit or credit listed with the correct amounts of money).

Students are allowed to use a searchable chart of accounts when solving the task. The searchable chart of accounts is stored as an external resource separated from the testing platform and is functionally embedded in the user interface only for items of the present type (following R4).

User interface

The top of the user interface remains identical to item-type 1. The lower part designated for providing the constructed answer resembles so-called T-accounts, which are a common notation for accounting records in double-entry bookkeeping (cf. Fig. 7).

The chart of accounts is provided by two means. First it is encoded as filtering drop-down list for each field taking an account name, retrieving only data from the external component (cf. Fig. 8).

The second support measure is the actual chart of accounts (cf. Fig. 9). It provides information on the numerical account IDs, which need to be entered in the T-account. Students are allowed to consult the chart at any time. Here, the external component also provides the presentation, which is only embedded in the test display.

Data representation and assessment

The fundamental structure of the XML representation of the items remains identical to the format presented above. A fundamental difference, however, can be found in the contents of the dataStorage-Tag. It holds a domain-specific structure containing the correct answer, i.e. the filled T-account serialized to an XML-representation (cf. Fig. 10).

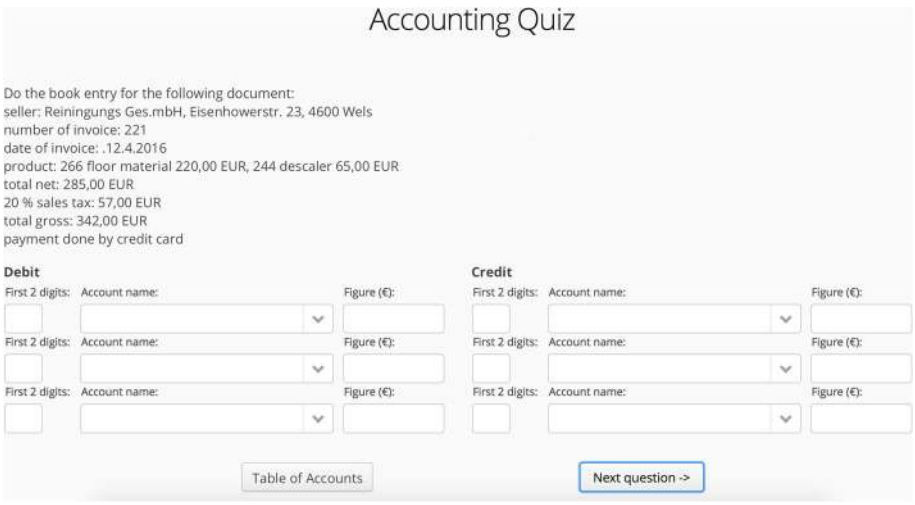

Fig. 7 User interface of item type for filling a T-account for a business transaction

The information contained in the tags debit and credit is assessed procedurally in the evaluation method implemented in the domain-specific item-class. This enables to assess answers according to domain-specific rules. In the present case, e.g., answers need to be rated correct independently of the sequence the accounts are listed in the T-account. Such domain-specific evaluation behavior cannot be easily implemented in existing platforms.

Item-type 3: business process modeling–syntax of conceptual models

The third item-type is part of an ongoing effort to create a test for assessing skills in business process modeling. In this case, not only the answer format requires multi-step interaction with the examinee, but also question presentation requires more complex visualizations. It thus is used as an example here to demonstrate the flexibility of the item-specification approach followed in the present work.

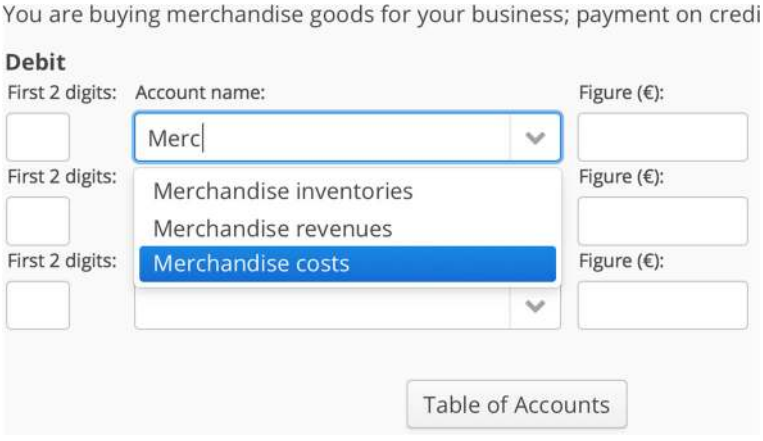

Fig. 8 Dynamically filtered account list as a support measure for correctly filling the T-account

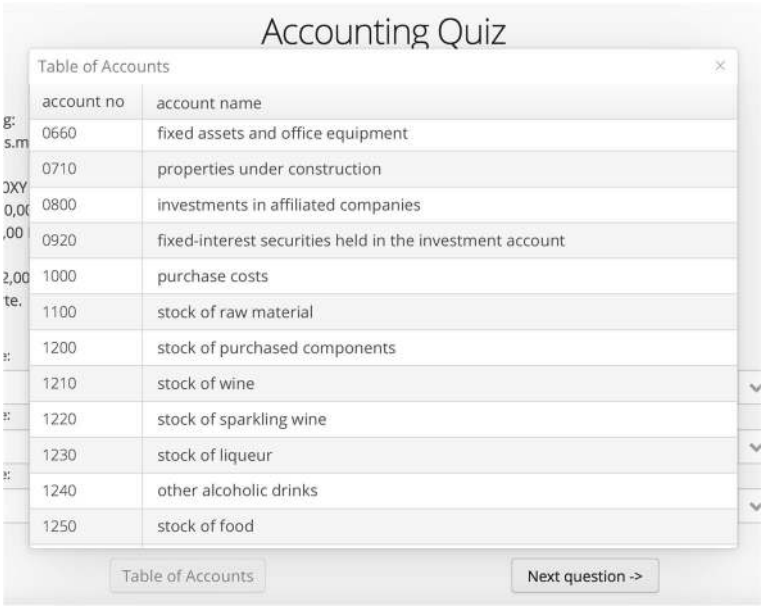

Fig. 9 Chart of accounts as a support measure for correctly filling the T-account

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<accountingtQuestionDataStorage>
    <dataStorage xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="accountingDataStorage">
        <debit>
            <accountName>Cleaning Materials</accountName>
            <amount>285.00</amount>
            <accountNumber>54</accountNumber>
        </debit>
        <debit>
            <accountName>Taxes</accountName>
            <amount>57.00</amount>
            <accountNumber>25</accountNumber>
        </debit>
        <debit>
            <amount>0.0</amount>
            <accountNumber>0</accountNumber>
        </debit>
        <credit>
            <accountName>Credit Card</accountName>
            <amount>342.00</amount>
            <accountNumber>31</accountNumber>
        </credit>
        <credit>
            <amount>0.0</amount>
            <accountNumber>0</accountNumber>
        </credit>
        <credit>
            <amount>0.0</amount>
            <accountNumber>0</accountNumber>
        </credit>
    </dataStorage>
    <question>
    Do the book entry for the following document:\n
    seller: Reiningungs Ges.mbH, Eisenhowerstr. 23, 4600 Wels\n
    number of invoice: 221\n
    date of invoice: .12.4.2016\n
    product: 266 floor material 220,00 EUR, 244 descaler 65,00 EUR\n
    total net: 285,00 EUR\n
    20 % sales tax: 57,00 EUR\n
    total gross: 342,00 EUR\n
    payment done by credit card
</question>
    <difficulty>-0.49</difficulty>
</accountingtQuestionDataStorage>
```

Fig. 10 XML for storing data of items for filling a T-account for a business transaction

Aim of item-type

Business process modeling is an activity performed in the field of business process management (Weske, 2010). It is a conceptual modeling task that aims at making organizational processes representable and supportable by information systems (Recker, Rosemann, Indulska, & Green, 2009) and more transparent to members of the organization (van der Linden, Proper, & Hoppenbrouwers, 2014). Assessment of conceptual modeling skills is still in its infancy and has only been a subject of research in recent years, with a focus on formative research exploring potential dimensions for assessment and their operationalization during testing (Bider & Perjons, 2015; Frederiks & van der Weide, 2006).

One skill considered necessary in this field is the ability to transform natural language statements to abstract conceptual structures that adhere to particular syntactic structures specified in a modeling language (Frederiks & van der Weide, 2006). One prerequisite for this ability is to be able to determine whether a given model is syntactically correct (Overhage, Birkmeier, & Schlauderer, 2012).

For testing this ability, the items contain a graphical representation of a business process model, using the modeling language BPMN (White & Miers, 2008). Examinees assess these items in a two-step approach. First, they locate a potential error in the model by clicking on the erroneous model construct. In the currently chosen approach, each model either is correct or only contains one error at maximum. If an error has been identified, examinees need to select the assumed type of error from a set of pre-specified options. The two-stage approach here thus combines a constructed-response format (the click on an arbitrary position in the model to locate the error) with a selected-response format (the selection of the error type from a provided list). Correctness is currently evaluated dichotomously (i.e., both, the located error and the reason need to be correct). An extension toward polytomous answers (e.g., by including several errors, or by considering partially correct answers) is currently being assessed from a test-theoretical point-of-view and is currently not part of the implementation.

User interface

Figure 11 shows the user interface in the state after a complete answer has been provided and before the next question button is clicked.

As described above, the item type requires a two-stage answer. First, an assumed error in the model must be selected by clicking on it. The selected element is marked using a red outline. The elements available for selection can be specified in the item definition–the difficulty of the item thus can be tuned.

The answering options are dynamically loaded in the drop-down-list depending on the selected element. For this purpose, each selectable item is typed with the class of elements it belongs to. If, as shown in the example, a “closing gateway” is selected, a set of error types is loaded, which in principle could occur at such an element. Examinees are then required to select the correct option.

The item type also contains a support measure that is similar to the chart of accounts presented for item-type 2. Here a graphical summary of the BPMN notation can be loaded as an external resource and displayed as an overlay.

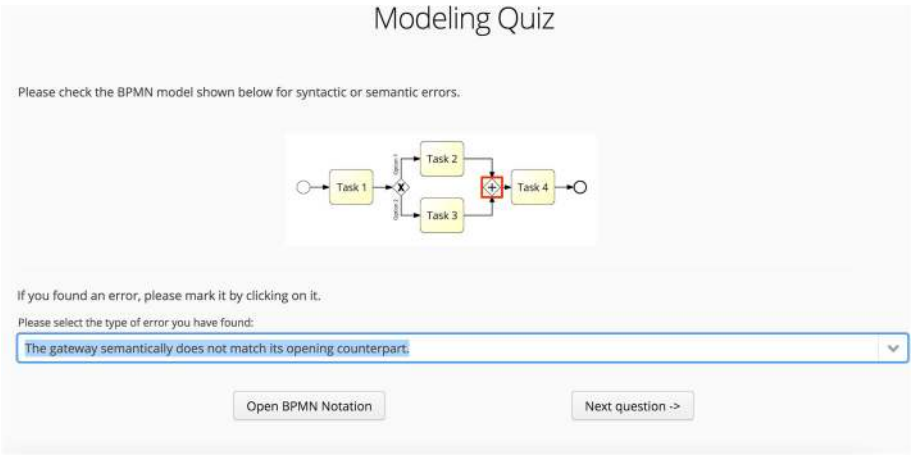

Fig. 11 User interface of item type for assessing BPMN modeling skills

Data representation and assessment

In order to realize the interaction possibilities on the user interface described above, the data stored for representing the question is more complex than in the former items. Figure 12 shows this representation.

The semantics of the task description is specified on a more fine-grain level, which allows to construct the user interface more flexibly. The tag model links to the image of the graphical model to be displayed and specifies an arbitrary number of selectable areas of the model with their respective type of element. The element selection in Fig. 10 is represented by the element-tag with id 6 and is stored with type = "gw-cl", which represents a "closing gateway".

The correct answer represented in the dataStorage-tag consequently also is more complex, as the item solution comprises two parts. The data format in principle allows to store models with several errors. This option, however, would require polytomous item handling and is currently not used.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<bpmnQuestionDataStorage>
    <dataStorage xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:type="bpmnDataStorage">
        <errorelement id=1>
            <element id=6 />
            <descr>The gateway semantically does not match its opening counterpart.</descr>
        </errorelement>
    </dataStorage>
    <question>
        <task>
            Please check the BPMN model shown below for syntactic or semantic errors.
        </task>
        <model path="./bpmn_test_1.png">
            <element id=1 type="ev-st" x=21 y=114 w=48 h=48 />
            <element id=2 type="task" x=106 y=90 w=125 h=100 />
            <element id=3 type="gw-op" x=272 y=114 w=48 h=48 />
            <element id=4 type="task" x=340 y=17 w=125 h=100 />
            <element id=5 type="task" x=340 y=161 w=125 h=100 />
            <element id=6 type="gw-cl" x=504 y=114 w=48 h=48 />
            <element id=7 type="task" x=572 y=90 w=125 h=100 />
            <element id=8 type="ev-en" x=732 y=114 w=48 h=48 />
        </model>
        <supporttext>
            If you found an error, please mark it by clicking on it.
        </supporttext>
    </question>
    <difficulty>-0.74</difficulty>
</bpmnQuestionDataStorage>
```

Fig. 12 XML for storing data of items for assessing BPMN modeling skills

Discussion of requirements R4 and R5

The three examples described above have demonstrated that the platform architecture enables flexible provision of domain-specific item in CAT with respect to data representation, answer evaluation and item presentation. It also allows to functionally include interactive external resources in a test.

Item data storage is flexible not only with respect to storing domain-specific details required for presentation and answer evaluation, but also with respect to storing item parameters required for different IRT-algorithms. The same is true for parameters required to correctly evaluate polytomous items.

Answer evaluation is performed procedurally specific for each item type. Procedural evaluation enables to easily formulate flexible answer constraints (such as the variable order of accounts in example 2) or enable multi-step answers (as used for example 3), which are currently not available in existing platforms and contribute towards fulfilling R5.

Item presentation and including external support measures is realized by embedding item-type-specific rendering of items as well as answer structures (such as the prepared empty T-account for item-type example 2) in the platform. The platform also enables to provide interactive item visualizations (such as the list of potential errors in a model in item-type example 3, which changes dynamically depending on the selected model element). The same mechanisms are used to integrate externally provided support measures for examinees during the test in a controlled manner (e.g., only providing them for specific items). Examples of such support measures include the interactive chart of accounts in example 2 or the BPMN notation in example 3. Support options integrated more deeply in an item type, such as the dynamically filtered list of accounts to support correct input of account names used in example 2, can be realized by retrieving data from external sources when creating item-specific display routines.

In all, we could show how the architecture of the platform in general and the current implementation in particular enable to meet the design requirements R4 and R5. The presented platform thus provides an amount of flexibility for designing and embedding CAT in online learning approaches that has not yet been available in any other platform discussed in related work. It aims at contributing to spread CAT as a practically feasible approach for educational assessment activities. Deploying CAT still causes high upfront effort for setting up a valid, calibrated item pool. The effort for the technical implementation of a testing tool tailored for the specific scenario, however, is significantly reduced with the platform presented in this article.

Conclusions

We have introduced a platform for flexibly implementing CAT tools for arbitrary testing scenarios and application domains. We have derived a set of requirements to be met by such a platform from existing concepts in the field of CAT and IRT. In a study of related work, we could show that—although all requirements have been previously addressed in earlier work—there is no single platform that meets all of the identified requirements. In an effort to fill this gap, we have proposed an architecture for such a platform and have described its prototypical implementation. By showing examples of different domain-specific item types, we could demonstrate the flexibility of our approach in terms of data representation, answer evaluation, item presentation and

functional integration of resources provided by external platforms. In this way, we could show that our approach meets the identified requirements.

The present work has several limitations. First, the platform currently enables to administer tests and facilitates the CAT-process, but does not yet enable easy test- or item-administration. This limitation will be addressed in future iterations of our platform. Second, the platform has yet to be integrated in a comprehensive online learning system to further examine the feasibility of the embedding mechanisms that are part of the architectural design on data- and UI-level. Third, the current implementation of the test-engine is only prepared to work with dichotomous items that are evaluated with the 1-parameter-logistics model. The version deployed as open-source via the GitHub-repository, however, will be equipped with a more comprehensive test engine that can be configured to work with different IRT-approaches.

In our future work, we aim at addressing the current limitations described above. The platform furthermore is currently being prepared for deployment in large-scale evaluations in real-world case studies in different application domains. Our research in this area will focus on examining the potential effects of the CAT-platform on the assessment process in presence-based and online learning settings.

Endnotes

[1]http://iacat.org/content/cat-software

[2]http://iacat.org/content/cat-software

[3]http://www.psychometrics.cam.ac.uk/newconcerto

[4]https://sourceforge.net/projects/irt-cat/

[5]https://vaadin.com/home

[6]Java Architecture for XML Binding, https://jcp.org/en/jsr/detail?id=222

[7]https://github.com/win-ce/AdaptiveTesting2

Abbreviations

CAT: Computerized Adaptive Testing; HTML: Hypertext Markup Language; IRT: Item response theory; JAXB: Java XML Binding; MOOC: Massive Open Online Courses; UI: User interface; XML: Extensible Markup Language

Availability of data and materials

The software implementation of the presented platform can be obtained as open source software under the GNU Lesser Public License at https://github.com/win-ce/AdaptiveTesting2.

Authors' contributions

SO has developed the conceptual framework presented in this paper, has carried out the study of related work and has developed usage scenario 3. He has furthermore acted as the main author of the article. FR has developed the platform architecture and the prototypical implementation and has contributed to its description. AE has contributed to the study and description of the kernel theories the requirements are based upon. CH has provided the usage scenarios used for demonstration of the platform features. He has furthermore contributed to the study of related work. All authors read and approved the final manuscript.

Competing interests

The authors declare that they have no competing interests.

Author details

[1]Department of Business Information Systems–Communications Engineering, Johannes Kepler University of Linz, Altenberger Straße 69, Linz 4040, Austria. [2]Department of Educational Psychology, Johannes Kepler University of Linz, Altenberger Straße 69, Linz 4040, Austria. [3]Department of Software Science, Radboud University, Mailbox 47, PO Box 9010, Nijmegen 6500, GL, The Netherlands.

Received: 9 June 2016 Accepted: 5 January 2017
Published online: 20 January 2017

References

Achtenhagen, F. (2012). The curriculum-instruction-assessment triad. *Empirical Research in Vocational Education Training, 4*(1), 5–25.

Bennet, S., & Ward, W. (1993). *Construction versus choice in cognitive measurement: issues in constructed response, performance testing, and portfolio assessment*. Lawrence Erlbaum Associates, Inc.

Bider, I., & Perjons, E. (2015). Design science in action: developing a modeling technique for eliciting requirements on business process management (BPM) tools. *Software & Systems Modeling, 14*(3), 1159–1188.

Bortolotti, S. L. V., Tezza, R., Andrade, D. F., Bornia, A. C., & Sousa Júnior, A. F. (2013). Relevance and advantages of using the item response theory. *Quality & Quantity, 47*(4), 2341–2360. http://doi.org/10.1007/s11135-012-9684-5.

Chen, J., & Wang, L. (2010). Computerized Adaptive Testing: A New Trend in Language Testing. In *International Conference on Artificial Intelligence and Education (ICAIE)* (pp. 725–728).

Čisar, S. M., Radosav, D., Markoski, B., Pinter, R., & Čisar, P. (2010). *Computer Adaptive Testing for Student's Knowledge in C ++ Exam*. Presented at the 11th IEEE International Symposium on Computational Intelligence and Informatics, Budapest, Hungary.

Conejo, R., Guzmán, E., Millán, E., Trella, M., Pérez-Del-La-Cruz, J. L., & Rios, A. (2004). SIETTE: A Web-Based Tool for Adaptive Testing. *International Journal of Artificial Intelligence in Education, 14*(1), 1–33.

Duda, I., & Walter, T. (2012). *A software framework for e-testing*. Presented at the IADIS International Conference e-Learning 2012, Lisbon, Spain.

Frederiks, P. J. M., & van der Weide, T. P. (2006). Information modeling: The process and the required competencies of its participants. *Data & Knowledge Engineering, 58*(1), 4–20. http://doi.org/10.1016/j.datak.2005.05.007.

Guzmán, E., & Conejo, R. (2005). Self-assessment in a feasible, adaptive web-based testing system. *IEEE Transactions on Education, 48*(4), 688–695.

Helm, C. (2016). Berufsbildungsstandards und Kompetenzmodellierung im Fach Rechnungswesen. In *Bildungsstandards und Kompetenzorientierung. Herausforderungen und Perspektiven der Bildungs- und Berufsbildungsforschung* (pp. 149–168). Bonn: W. Bertelsmann Verlag.

Huang, Y.-M., Lin, Y.-T., & Cheng, S.-C. (2009). An adaptive testing system for supporting versatile educational assessment. *Computers & Education, 52*(1), 53–67.

Krass, I. A., & Williams, B. (2003). *Calibrating CAT Pools and Online Pretest Items Using Nonparametric and Adjusted Marginal Maximum Likelihood Methods*. Presented at the Annual Meeting of the National Council on Measurement in Education, Chicago, IL, USA.

Kröhne, U., & Frey, A. (2011). *Multidimensional Adaptive Testing Environment (MATE)–Software for the Implementation of Computerized Adaptive Tests*. Presented at the International Conference on Computerized Adaptive Testing, Pacific Grove, CA, USA.

Lin, H. (2012). *Item selection methods in multidimensional computerized adaptive testing adopting polytomously-scored items under multidimensional generalized partial credit model*. Doctoral dissertation. University of Illinois at Urbana-Champaign.

Linacre, J. M. (2000). Computer-Adaptive Testing : A Methodology Whose Time Has Come. In S. Chae, U. Kang, E. Jeon, & J. M. Linacre (Eds.), *Development of Computerized Middle School Achievement Test* (pp. 1–58). Seoul: Komesa Press.

Magis, D., & Raîche, G. (2012). Random generation of response patterns under computerized adaptive testing with the R package catR. *Journal of Statistical Software, 48*(8), 1–31.

Meyer, J. P., & Zhu, S. (2013). Fair and equitable measurement of student learning in MOOCs: An introduction to item response theory, scale linking, and score equating. *Research & Practice in Assessment, 8*, 26–39.

Moosbrugger, H., & Kelava, A. (2012). Testtheorie und Fragebogenkonstruktion. In *Statistik für Human- und Sozialwissenschaftler* (pp. 7–26). Berlin, Heidelberg: Springer. http://doi.org/10.1007/978-3-642-20072-4_2.

Overhage, S., Birkmeier, D., & Schlauderer, S. (2012). Quality marks, metrics, and measurement procedures for business process models. *Business & Information Systems Engineering, 4*(5), 229–246.

Pellegrino, J. W. (2010). *The design of an assessment system for the race to the top: A learning sciences perspective on issues of growth and measurement*. Proceedings of the Exploratory Seminar: Measurement Challenges Within the Race to the Top Agenda. Educational Testing Service, Princeton.

R Development Core Team (2011). *R: A Language and Environment for Statistical Computing*. Vienna, Austria: the R Foundation for Statistical Computing. ISBN: 3-900051-07-0. Available online at http://www.R-project.org/.

Reckase, M. D. (1974). An interactive computer program for tailored testing based on the one–parameter logistic model. *Behavior Research Methods and Instrumentation, 6*(2), 208–212.

Reckase, M. D. (2010). Designing Item Pools to Optimize the Functioning of a Computerized Adaptive Test. *Psychological Test and Assessment Modeling, 52*(2), 127–141.

Recker, J. C., Rosemann, M., Indulska, M., & Green, P. (2009). Business process modeling-a comparative analysis. *Journal of the Association for Information Systems, 10*(4), 333–363.

Rumbaugh, J., Jacobson, I., & Booch, G. (2004). *The Unified Modeling Language Reference Manual*. London: Pearson Higher Education.

Salcedo, P., Pinninghoff, M. A., & Contreras, R. (2005). Computerized adaptive tests and item response theory on a distance education platform. In J. Mira & J. R. Álvarez (Eds.), *Artificial Intelligence and Knowledge Engineering Applications: A Bioinspired Approach* (pp. 613–621). Heidelberg: Springer.

Scalise, K., & Allen, D. D. (2015). Use of open-source software for adaptive measurement: Concerto as an R-based computer adaptive development and delivery platform. *British Journal of Mathematical and Statistical Psychology, 68*(3), 478–496.

Segall, D. O. (2004). Computerized Adaptive Testing. In K. Kempf-Leonard (Ed.), *Encyclopedia of Social Measurement* (pp. 429–438). San Diego, CA: Academic.

Triantafillou, E., Georgiadou, E., & Economides, A. A. (2008). CAT-MD: Computerized adaptive testing on mobile devices. *International Journal of Web-Based Learning and Teaching Technologies (IJWLTT), 3*(1), 13–20.

van der Linden, D., Proper, H. A., & Hoppenbrouwers, S. J. B. A. (2014). Conceptual Understanding of Conceptual Modeling Concepts: A Longitudinal Study among Students Learning to Model. In L. Iliadis, M. Papazoglou, & K. Pohl (Eds.), *Advanced Information Systems Engineering Workshops* (pp. 213–218). Berlin: Springer International Publishing.

Veldkamp, B. P., & Matteucci, M. (2013). Bayesian computerized adaptive testing. *Ensaio: Avaliação e Políticas Públicas em Educação, 21*(78), 57–82. http://doi.org/10.1590/S0104-40362013005000001.
Wainer, H., & Mislevy, R. J. (2000). Item Response Theory, Item Calibration, and Proficiency Estimation. In H. Wainer (Ed.), *Computerized Adaptive Testing A Primer* (pp. 61–100). New York: Lawrence Erlbaum Associates.
Weiss, D. J. (2004). Computerized adaptive testing for effective and efficient measurement in counseling and education. *Measurement and Evaluation in Counseling and Development, 37*(2), 70–84.
Weske, M. (2010). *Business process management: concepts, languages, architectures*. Heidelberg: Springer.
White, S. A., & Miers, D. (2008). BPMN Modeling and Reference Guide: Understanding and Using BPMN. *The Journal of Strategic Information Systems, 3*, 23–40. Florida, USA: Future Strategies Inc. http://doi.org/10.1016/0963-8687(94)90004-3.