Enterprise Modelling and Information Systems Architectures
Vol. 18, No. 2 (2023). DOI:10.18417/emisa.18.2

Special Issue on Teaching and Learning Conceptual Modeling

Automated Assessment of Conceptual Models in Education

A Systematic Literature Review

Meike Ullrich[*,a], Constantin Houy[b], Tobias Stottrop[c], Michael Striewe[c], Brian Willems[b], Peter Fettke[b], Peter Loos[b], Andreas Oberweis[a]

[a] Karlsruhe Institute of Technology (KIT), Institute of Applied Informatics and Formal Description Methods (AIFB), Kaiserstr. 89, 76133 Karlsruhe
[b] German Research Center for Artificial Intelligence (DFKI) and Saarland University, Institute for Information Systems (IWi), Campus D3 2, 66123 Saarbrücken
[c] University of Duisburg-Essen, paluno - The Ruhr Institute for Software Technology, Gerlingstr. 16, 45127 Essen

Abstract. *In Computer Science, Software Engineering, Business Informatics or Information Systems, conceptual modeling is an important tool and as such also contained in the respective curricular recommendations. Especially in large university courses, an automated assessment of models can improve the quality of teaching and learning. While there are many different approaches to automatically assess conceptual models, these approaches, however, often only tackle a single aspect or a single type of conceptual model. In this paper, we aim to take a comprehensive perspective on the topic and shed light on the current state of the art and technique. Furthermore, as assessment approaches have to be developed in accordance with appropriate teaching or learning activities and desired learning outcomes, we inquire in which settings automated assessment approaches are included and to which extent didactic aspects are taken into account. To this end, we have conducted a systematic literature review in which we identified 110 relevant publications on the topic which we have analyzed in a structured way. The results provide answers to five relevant research questions and pinpoint open issues which should be inquired in further research.*

Keywords. Systematic Literature Review • Technology-enhanced Learning • Education • Didactics • Automated Assessment • Conceptual Models • Modeling Languages

Communicated by Kristina Rosenthal, Estefanía Serral, Monique Snoeck & Stefan Strecker. Received 2021-05-31. Accepted after 2 revisions on 2023-01-23.

1 Introduction

Conceptual modeling is an important tool for developing systems in research and practice (Fettke 2009), such as organizational systems or information systems. Against this background many related fields of research, e. g. Computer Science, Software Engineering, Business Informatics or Information Systems, put major emphasis on the significance and value of a proper conceptual modeling education in their curricula (Association for Computing Machinery (ACM) 2018). Furthermore, learning and teaching conceptual modeling approaches at universities and other institutions of higher education have gained more and more relevance (Recker and Rosemann 2009). In this context, graphical and graph-based conceptual modeling approaches, such as Unified Modeling Language (UML), Entity-Relationship diagrams (ERD), Petri nets, Event-driven process chains (EPC) or Business Process Model and Notation

* Corresponding author.
E-mail. meike.ullrich@kit.edu
This research has been funded by the German Federal Ministry of Education and Research (BMBF) within the project KEA-Mod (Kompetenzorientiertes E-Assessment für die grafische Modellierung) under FKZ: 16DHB3022-16DHB3026, Website: https://keamod.gi.de/

(BPMN) are essential core components of the mentioned curricula.

However, conceptual modeling can be considered a complex task as several different aspects need to be addressed in order to develop models of good quality (e. g. syntactic and semantic correctness and also pragmatic aspects) (Nelson et al. 2012), which makes learning as well as teaching conceptual modeling a demanding task. Moreover, there can be more than only one correct solution for a particular modeling problem. Against this backdrop, providing detailed individual feedback and coaching for every student can overstrain lecturers and teachers, especially in large university course settings, e. g., in Germany with sometimes several hundreds of participants each semester. Automated assessment systems can significantly mitigate this situation and effectively support individualized learning and teaching of conceptual modeling providing customized feedback on students' modeling efforts. Furthermore, they can significantly relieve teachers in their manual assessment tasks which are typically cumbersome and tedious. Furthermore, automated assessment is inherently more objective and consistent in comparison to the more subjectively affected manual grading of conceptual modeling tasks.

In the following, we present the results of a systematic literature review focusing on approaches and systems for automated assessment of conceptual models in education. In this context we have systematically identified 110 relevant contributions on the topic which we have analyzed in a structured way.

There are two issues which we would like to particularly address in this contribution. (1.) There are many different approaches to automatically assess conceptual models, but they often only tackle a single aspect or a single type of conceptual model. In this paper, we take a more global perspective on existing approaches and available assessment systems, hence, on the current state of the art and technique. (2.) Furthermore, as assessment approaches have to be developed in accordance with appropriate teaching or learning activities and desired learning outcomes, e. g. competences to be acquired, we inquire in which teaching and learning settings assessment approaches are included and to what extent they take didactic aspects into account.

In this paper, we aim at identifying the potential of combining existing approaches and analyze the current application scenarios. Initially, this literature review was motivated by the up-coming technical development and realization of assessment algorithms and methods for conceptual models in form of graph-based diagrams (see Sec. 2) in our ongoing research project KEA-Mod (see footnote on page 1). In this particular development, we focus on ERD, different diagram types of the UML as well as process models in form of EPCs, Petri nets and BPMN as relevant diagram types in the context of Software Engineering, Databases and Business Process Modeling.

There is some current related work presenting the results of comprehensive literature reviews, such as the review by Deeva et al. (2021) which systematically investigates automated feedback systems for digital teaching and learning scenarios in general but not particularly in the domain of conceptual modeling. Moreover, Keuning et al. (2018) present a systematic literature review treating automated feedback generation for programming exercises which is more closely related but does not focus on conceptual modeling either. While we agree that it can generally be a good idea to seek inspiration from approaches presented in other application domains, our literature review concretely focuses on automated assessment of conceptual models. We see significant benefits and a further contribution to the current state of research in addressing the following research questions in this article:

RQ 1: *Which aspects of conceptual models are addressed through automated assessments?*

RQ 2: *Which technical methods are used to realize the automated assessment of conceptual models?*

RQ 3: *Are didactic aspects like competences or learning objectives for conceptual modeling* *considered in the context of the automated assessment?*

RQ 4: *Which form(s) of feedback for learners (if any) does the automated assessment of conceptual models produce?*

RQ 5: *At what stages of the learning process (formative, summative) and in which context (teaching method, course settings) are automated assessments of conceptual models involved?*

By addressing these research questions, we would like to provide a comprehensive overview of the state of the art but also new insights for colleagues involved in the education of conceptual modeling in higher education and even in organizational practice, especially those interested in applying or designing systems for the automated assessment of conceptual models in education.

The remainder of this article is structured as follows. In Sec. 2, the conceptual foundations of the treated topics will be introduced, before Sec. 3 introduces the applied literature review method especially focusing on the relevance criteria and the literature search process. Sec. 4 presents the review results regarding each of the developed research questions which are discussed in Sec. 5 before Sec. 6 summarizes the findings and gives an outlook on future work.

2 Conceptual Foundations

This section presents an introduction of the conceptual foundations of the treated topics.

2.1 Graphical and graph-based (conceptual) models

45 years after Chen (1976) proposed his Entity-Relationship-Model which is often regarded as one of the first cornerstones for the domain of conceptual modeling, there is no generally accepted definition for the term *conceptual model*. Researchers are still struggling for consensus by examining existing definition proposals or exploring the nature and anatomy of conceptual models (e. g., Guarino et al. 2020; Thalheim 2018). Thus, in the

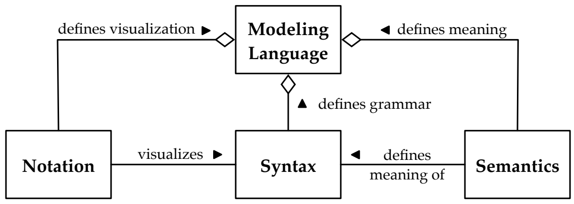

Figure 1: Modeling language components. Excerpt from original figure in Karagiannis and Kühn (2002, p. 3).

following we present a simple description of our understanding of models. This notion is derived especially from the viewpoint of the considered disciplines Software Engineering, Databases and Business Process Modeling.

Models are the central artifact of (conceptual) modeling. A model is a representation of relevant aspects of an object described in a modeling language. It is intended for a specific usage scenario and serves a specific purpose (Stachowiak 1973). The modeled object does not necessarily have to exist physically or virtually (i. e. model serving as a copy of the modeled object), it can also be something that has yet to be constructed (i. e. model serving as a design plan for the modeled object). Examples for typical objects which are modeled in the considered disciplines are information systems, data and organization structures or business processes. According to Karagiannis and Kühn (2002), a modeling language defines modeling elements together with composition rules (syntax) and describes the meaning or rather corresponding element types from the context of the objects to be modeled (formal semantics) as well as the form of representation (notation) (see Fig. 1).

Typically the notation of modeling languages from the considered disciplines provides graphical symbols yielding in two-dimensional visual representations (Moody 2009). Such representations are also referred to as diagrams (Larkin and Simon 1987), which is often reflected by the name of specific model types, for example Entity-Relationship diagram or UML class diagram. When graphical symbols are used in a model, they are usually

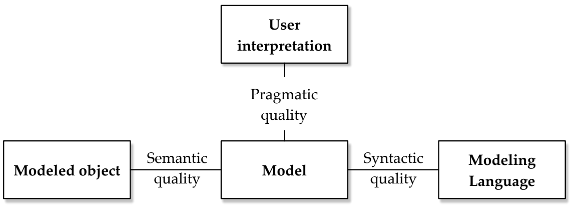

Figure 2: Core quality aspects in models, following Krogstie et al. (1995).

annotated with textual labels to define their correspondence in the context of the modeled object. The modeling languages in the initial focus of the literature search (ERD, UML, EPC, BPMN and Petri nets[1]) are examples for such graphical (or diagrammatic) modeling languages. Moreover, they exhibit another common characteristic: they are graph-based. Accordingly, the corresponding models exhibit a graph-structure where modeling elements are specific nodes, which are connected via specific edges (e. g., to define relations or control-flow). While there exist both non-graphical (e. g., WS Business Process Execution Language, BPEL) and non-graph-based (e. g., F-Logic) conceptual modeling languages, the focus of our literature review lies specifically on models which are graphical and graph-based likewise, in short: graph-based diagrams.

In order to reason about the quality of models, established frameworks (such as Krogstie et al. (1995, 2006) and Overhage et al. (2012)) distinguish three essential quality aspects: syntactic, semantic and pragmatic quality (see Fig. 2). Those have been derived from the semiotic theory of Morris (1988)[2] .

Syntactical aspects address the conformance of a given model and the used modeling language (Krogstie et al. 1995). Hence, to assess the syntactic quality of a model, it has to be checked whether the model uses only modeling elements which are part of the modeling language (e. g., transitions, places and directed edges for basic Petri nets, e. g., as described in Peterson (1981)) and also whether the model adheres to the composition rules defined by the modeling language (e. g., using directed edges to connect only transitions with places or vice versa. This also applies to the aforementioned basic Petri nets).

Semantic aspects cover the correspondence of a given model with the modeled object. As for the semantic quality of a model, Krogstie et al. (1995) distinguish between validity and completeness. A model is valid, if the expressed information is correct and relevant with regard to the modeled object. It is furthermore complete, if the model expresses all information that is correct and relevant with regard to the modeled object. As the term "semantics" already appears in the description of a modeling language, it is necessary to distinguish between two concepts here: 1) *formal semantics* of a modeling language e. g., (e. g., Karagiannis and Kühn 2002) and 2) *semantic quality* of a model (e. g., Krogstie et al. 1995; Overhage et al. 2012). The further used terms "semantic aspects", "semantic errors" and "semantic correctness" refer to the latter concept (semantic quality).

Pragmatic aspects arise as soon as a given model is intended to be interpreted by humans which is the case for most typical conceptual modeling usage scenarios where models serve as communication instrument[3] . Then, the pragmatic quality describes the understandability of the given model by a specific user (Houy et al. 2014). Traditionally, this is achieved by empirically measuring both objective and subjective dimensions of understandability effectiveness and efficiency during experiments (Houy et al. 2012). However, a general measure which abstracts from the experience

[1] Petri nets are to be considered graphical models when they are described by a visual graph representation. While this is the general case, it is not mandatory at all. In principle, specifying the element sets according to the formal mathematical definition is sufficient.

[2] German translation of the original source: Charles W. Morris, Foundations of the Theory of Signs, International Encyclopedia of Unified Science. Vol. 1, Nr. 2, Chicago, 1938

[3] Mentionably, pragmatic aspects can be neglected when a process model is to be interpreted by means of an interchange format in automated fashion only, e. g., by specifying the process flow in a workflow management system.

of a specific user has to consider established criteria which are associated with a positive effect on understandability, as they can be found for example in modeling recommendations like the Guidelines of Modeling (Schütte and Rotthowe 1998) and the Seven Process Modeling Guidelines (Mendling et al. 2010) or drawn from graph aesthetics (Purchase 2002). Exemplary criteria are model clarity, label and flow direction consistency or minimal number of edge crossings.

One of the main obstacles to the assessment of model quality in practice is the less tangible nature of many aspects (beyond syntactic quality) a model has to be compared against. For example, information about the object to be modeled is hard to grasp and yet a prerequisite for the assessment of semantic quality. In the case of automated assessment, that information additionally has to be available and interpretable in digital form. However, the assessment of a student submission to a simple model creation task where a given textual description has to be modeled using a specified modeling language is different from the assessment of models created in a real-world modeling context. With respect to semantic quality, information about the object to be modeled is primarily derived from the task description. It is thus feasible for a task designer to (digitally) capture the required contents of a model (e. g., in form of rules or sample solutions) and perform corresponding checks. Since a task description may (intentionally) include ambiguities, contextual information (e. g. from the lecture) may serve as a secondary source of information when creating rules, sample solutions, or alike. The challenge to compare model contents with the (probably undocumented) (real-)world object (or even worse a future world object, e. g. a new information system) that has been modeled, for example via investigations, interviews or user questionnaires does not apply here.

Moreover, it should be emphasized that model quality is not limited to the core quality aspects syntactic, semantic and pragmatic quality discussed here. For example, depending on the intended application context and model purpose, a model can be evaluated against the modeling goal (Krogstie et al. 2006) refer to the so-called "organizational quality"). A multi-perspective framework addressing the quality of reference models providing a large set of aspects (also including aspects relevant for conceptual models in general) is presented by Frank (2006). Such aspects can only be assessed in student's models if the corresponding task is designed accordingly which is usually not the case for the aforementioned simple model creation tasks.[4] .

2.2 Didactic aspects

In recent decades, educational theory has formed the idea of output-oriented learning and competence-based assessments. Following these ideas, each learning activity should be associated with the expected learning outcome that describe what learners know or are able to do on completion of the learning activity. These goals are usually described in terms of competences and thus not on the level of knowledge facts, but on the level of skills that are necessary to solve practically relevant, complex tasks and problems within a specific domain (Bachmann 2018; Weinert 2001). As a consequence, there also needs to be an alignment between the learning activities on the one hand and the assessment activities on the other hand (Biggs and Tang 2011), since competences must be assessed using domain-specific tasks (Koeppen et al. 2008). Hence, (automated) assessment tools should not be developed independently of the actual contents of learning activities. Furthermore, assessment tools should not be restricted to universal types of assessment items that can be used in any domain of study, but must also provide domain-specific item types.

Bloom (1956) has proposed a taxonomy of learning objects that has later been revised by

[4] Yet, this raises the legitimate question whether simple modeling creation tasks are sufficient in the context of teaching and learning conceptual modeling or whether more elaborate task types corresponding to competencies required in a real-world modeling context are required. However, this question is beyond the scope of this literature review.

Anderson et al. (2000). These taxonomies help to classify learning objectives according to their required knowledge (e. g. factual knowledge, conceptual knowledge, or procedural knowledge) and cognitive processes (e. g. remember, understand, apply, analyze, evaluate, or create). Due to the intended alignment between learning objectives and assessments, these taxonomies can also be used to classify assessment items and thus check whether an assessment actually assesses the competences as it is intended to do. Notably, it may be necessary to use more than one single assessment item to assess whether a learner has a specific competency and it may also be possible that a single assessment item requires several aspects that are classified in different places in the taxonomy. For example, some tasks may require learners to remember syntactical rules for some modeling language and at the same time require them to analyze a given process.

Assessment can happen in many different stages of the learning process (see Fig. 3). According to Mctighe and O'Connor (2005), one can in particular distinguish between formative and summative assessments. Formative assessments take place during the learning process and are primarily intended to inform the learners about their current progress within the learning process. Typical examples are lab exercises and coursework. A formative assessment can be conducted by a teacher or an assessment system, but also by another learner (peer assessment) or the learners themselves (self-assessment). Summative assessments take place at the end of the learning process and are primarily intended to make a final evaluation, i. e. to assign grades. Typical examples are thus oral or written exams. They are typically conducted by teachers, possibly with the help of an automated assessment system.

2.3 Feedback

According to Hattie and Timperley (2007), feedback in the context of education is basically any kind of information that is provided by some agent and that informs learners about their performance or level of understanding. The information can be of very different kind, such as corrective information, alternative strategies, additional facts, or encouragement. However, feedback has to be related to the previous performance in some way in any case. More precisely, feedback is able to improve learning, if its main purpose is to reduce the discrepancies between the current performance and the desired performance.

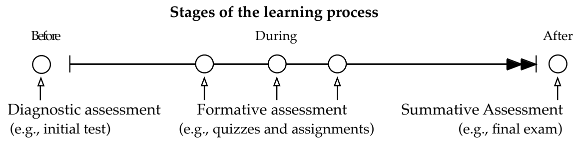

Figure 3: Assessment in the context of the learning process.

Giving effective feedback is not a trivial task. The feedback model by Hattie and Timperley (2007) defines three questions, effective feedback should answer: "Where am I going?", "How am I going?" and "Where to next?". Each of these questions clarifies a specific aspect of the discrepancies between the current performance and the desired performance. The first question relates to the goals and thus the desired performance. The second question relates to the current performance. The third question provides some helpful information to reduce the gap. In the context of conceptual modelling, effective feedback could for example point out that a model is incomplete because of some missing relation between two elements and suggest to look up a specific syntactic element. In that case, it would name the goal (creation of a model that expresses some relationship), the current performance (the model is not yet complete with respect to the goal) and provide helpful information (suggest an element that might help to express the missing relation).

Since feedback can convey very different kind of information, feedback can also appear in different forms. A very common form of feedback in educational contexts is numerical feedback in terms of grades, marks or credit points. However, such feedback is inherently incomplete with respect to the above mentioned feedback model, because it only expresses the current performance in relation to the best expected performance, but provides no information on how to reduce the gap. That part can be taken by textual feedback, e. g. by listing errors or omissions and thus naming concrete aspects of the current performance that need improvement. Especially in the context of graph-based diagrams (but also in other domains of study), feedback can also be graphical. For example, missing elements can be added directly to a diagram or wrong elements can be highlighted. That kind of feedback is more concrete than textual feedback, since it not only tells learners what to improve, but also demonstrates how to do it. In that sense, revealing a sample solution is also a valid kind of feedback. However, even if graphical feedback does not reveal a solution directly, it is proven to be more effective in specific situations (Rieber et al. 1996), but the effect may also depend on individual learning styles (Parvez and Blank 2008). According to Kleij et al. (2015), elaborate feedback that provides explanations appears to be more effective than providing a correct solution, which is in turn more effective than feedback regarding the correctness of the answer.

2.4 Automated Assessment

The process of grading submissions and giving feedback can be supported by technology-enhanced assessment systems. They are today used in many contexts in higher education, while the oldest known automated assessment systems in computer science date back to the 1960s (Hollingsworth 1960) and also automated assessment of diagrams is known for about 20 years (Tsintsifas 2002). Both examples belong to the group of automated assessment systems that are able to handle open-ended tasks in which students submit some artifact they have created while working on their assignment. Different to that, there are also many automated assessment systems that ask closed questions such as multiple-choice questions.

If the primary focus of automated assessment is on giving grades, it can be considered a classification task. In case of closed questions, it is easily solved by defining a grade for each possible answer. In open-ended tasks it can be solved by determining the position of a particular submission within the solution space. Depending on the kind of task, there is one point or some region (often even multiple points/regions) within the space of all possible submissions that represents solutions receiving full credit. Other regions may receive partial credit and some no credit at all. That view on automated assessment is particularly common in conjunction with automated essay grading (Valenti et al. 2003), but can also be used in other domains like computer programming (Gross et al. 2012). Usually, representative sample solutions for the different grades are used in these approaches and statistical methods are employed to determine the closest sample solution for each newly arriving submission. In the particular domain of modeling, the computation of similarity or distance between models can make use of graph-based properties of the submission artifacts (Sousa and Leal 2015).

If automated assessment should also include detailed feedback on specific errors or if a classification approach is not feasible for some reason (e. g. because computing plain similarities between submission artifacts is not meaningful), rule-based approaches are often used. Again, closed tasks can be handled quite easily by associating specific feedback with every possible answer. For open-ended tasks, most approaches automatically apply a list of criteria to each submission and provide specific feedback for each criterion that is not met. Common examples of criteria are test cases in the domain of computer programming (Keuning et al. 2018), evaluation of formulas in mathematics and natural sciences (Muñoz de la Peña et al. 2013; Sangwin 2013), or matching of specific terms in short answers (Leacock and Chodorow 2003). In the particular domain of modeling, graph pattern matching can be used to check the fulfilment of individual criteria (Striewe 2014; Thomas et al. 2008b). Notably, models typically include labels in natural language, and thus mechanisms can be included that can handle spelling errors or synonyms (Jayal and Shepperd 2009; Smith et al. 2013).

3 Method

The systematic literature review presented in this article primarily draws on the guidelines of Brocke et al. (2009, 2015) and considers further methodological recommendations where applicable, e. g., a review protocol as suggested by Kitchenham and Charters (2007). According to the framework for literature reviews proposed in Brocke et al. (2009), the review scope has to be defined first. This has been done in Sec. 1, where also the research questions RQ1 - RQ5 were defined precisely. Second, the topic of research has to be conceptualised before in order to reason about it and draw conclusions from analyzing relevant publications in a systematic way, which we addressed with Sec. 2. Third, the literature search process has to be performed and documented (this section). Fourth, the literature analysis and synthesis is carried out and the results are presented in Sec. 4. Finally, a research agenda should be drawn from the conclusions, which is included in the discussion in Sec. 5 as well as in the outlook in Sec. 6.

3.1 Relevance criteria

To identify publications providing evidence with regard to the intended scope of the review and the chosen research questions (Kitchenham and Charters 2007), four relevance criteria were decided in the planning phase of the review. For a publication to be included in the literature review, all of the following four relevance criteria C1-C4 have to be fulfilled.

C1: *The publication addresses graph-based diagrams.*

C2: *The publication describes an (semi) automated assessment of such models.*

C3: *The described approach is intended for application in an educational context.*

C4: *The addressed model type is taught in the disciplines computer science, business informatics or information systems.*

As intended, the application of those criteria during the screening process led to the exclusion of approaches for model types outside the scope of this literature review, e. g., the publication of Wijesinghe et al. (2017), which have developed an automated assessment of Venn and Euler diagrams. These are graphical but not graph-based per se and thus violating criterion C1. With criterion C2, we excluded (i) publications not assessing models (e. g., the approach of Fan (2007), where the modeling process itself is assessed) and (ii) publications where the assessment process is not directly supported through automation. This means, we included publications describing approaches assisting the assessment process by e. g., clustering similar student solutions (Batmaz and C. Hinde 2006; Al-Hoqani 2018) but ruled out approaches with the mere application of a digital modeling tool or a web-based learning environment while the assessment of models is completely performed manually (e. g., Krusche et al. 2020). While there are undoubtedly also powerful assessment approaches for conceptual models in form of graph-based diagrams outside the context of (higher) education, our research questions are mainly concerned with the application of automated assessment in an educational context. Thus, we excluded for example the publication of Javed and Lin (2020), which describe an approach for the automated assessment of ER diagrams using domain knowledge. Due to criterion C4, we also excluded publications concerning diagrams not related to the disciplines in the focus of this literature review, e. g., truss and free body diagrams used in the fields of Statics and Physics (Runyon et al. 2020).

3.2 Search process

The initial search was conducted in 02/2021 using the Scopus[5] search engine, which covers relevant publishers of the targeted disciplines Computer

[5] http://scopus.com

Science, Business Informatics and Information Systems, i. e. Elsevier, Springer, IEEE and many more. Inspired by the systematic development of search strings for automated search proposed by Zhang et al. (2011), we first compiled a so-called quasi-gold standard with 26 already known relevant papers fulfilling all relevance criteria which were obtained through previous manual searches (see Table 5). In an iterative process we tried to create a search string that yields in a search result containing as much papers from our quasi-gold standard as possible while the number of papers in the result set was still manageable for the subsequent manual analysis. The final search string matched 24 of 26 papers of our quasi-gold standard and included the following simplified search terms which are made up of a combination of relevant keywords.

- `automatic marking OR automatic grading OR automatic assessment OR assessment system OR tutoring system OR e-learning system OR e-learning tool`
- `(automatic OR tool OR system) AND (teaching OR education) AND (checking OR testing)`
- `diagram* OR UML OR petri net OR process modeling OR process models OR database modeling OR dfa OR bpmn OR business processes OR epc`

The first two search terms were combined by an OR-operator (meaning only one of them had to be resolved to true) while the last search term was added through an AND-operator (meaning those keywords were required). Some keywords were also varied by including both singular and plural or regional spelling differences e. g., "modeling" vs. "modelling": The search string was then applied in the Scopus search and enriched with additional database-specific filters (e. g., English language, relevant subject areas) to search in title, abstract and keywords. The complete search string including the filters is presented in Listing 1 in the appendix.

The result set consisted of 691 publications of which 9 duplicates were excluded in a first step (see Fig. 4 for an overview of the complete search process). In the next step we screened each publication with regard to the relevance criteria presented in Sec. 3.1. To achieve this, we looked at the title and abstract and if necessary at the full text. This led to a much smaller result set consisting of 68 publications fulfilling all four relevance criteria. On this set we performed an iterative backward search with three iterations where 32, 3 and 1 new relevant publications could be retrieved. Subsequently, we performed a forward search on the resulting set of relevant publications during which 17 new relevant publications were found. On these 17 publications we performed another backward search leading to 2 further relevant publications. While the next backward search did not bring up new relevant publications, we were able to obtain one further relevant publication with a last forward search. From here on, no new relevant publications could be retrieved with further iterations. The resulting set of 124 papers included all 26 publications of our quasi-gold standard and was analyzed in detail.

For the literature analysis, we defined a data extraction form in the context of the review protocol as proposed by (Kitchenham and Charters 2007) that allowed us to capture 25 aspects of each publication in a systematic way. Some of the aspects are phrased as questions that can be answered either with "yes" or "no" for each paper (e. g. "Does the paper report on checking syntactic aspects?"), while others allow for lists (e. g. "Which scientific disciplines does the paper refer to?") or free-form notes (e. g. "What approach is used for feedback generation?"). Of those 25 aspects, only 11 aspects are directly related to the five research questions addressed in this article. Those aspects are presented in Table 1 alongside with further fields regarding metadata and general aspects that were presented as general findings in Section 4.1.

In a first pass, each paper was read by one member of the research team and the respective entries according to the data extraction form were made

Short name	Description/Question	Coding/Values
Metadata & General aspects		
Author(s), Title and Year	Retrieved through Scopus or extracted manually in case of other source during backward/forward search	self-explanatory
Citekey	Key to bibtex database entry containing full citation	self-explanatory
Source title	What is the name of the journal or conference/workshop series where the publication originates?	name (strip year, issue, volume, iteration etc.)
Document type	Does the publication originate from a journal (Article) or conference/workshop proceedings (Conference Paper)	One of: Article, Conference Paper
Quasi Gold Standard (QGS)	Is the publication included in the QGS compiled manually prior to the literature search?	One of: yes, no
Source	In which step of the literature search was the publication retrieved first?	#id + description of step
Model type(s)	Which types of models are covered by the approach described in the publication? (Types were consolidated bottom-up)	List of: UML class diagram, ER diagram, UML sequence diagram, UML use case diagram, DFA, digital circuit, UML activity diagram, UML state machine, flowchart, coloured Petri net or individual
Universal	Is the approach described as universal?	One of: yes, no
RQ1: Addressed aspects		
Syntactic aspects	Does the automated assessment described in the publication address syntactic aspects?	One of: not specified, no, yes, indirectly
Semantic aspects	Does the automated assessment described in the publication address semantic aspects?	One of: not specified, no, yes, indirectly
Pragmatic aspects	Does the automated assessment described in the publication address pragmatic aspects?	One of: not specified, no, yes, indirectly
RQ2: Technical realization		
Technique or method	Which techniques or methods are employed by the approach for automated assessment described in the publication? (Classes were derived bottom-up)	List of: model-based, rule-checking, constraint-based, machine learning, simulation, testing, clustering or individual
Label analysis	Does the publication report on employing a specific technique on model element labels?	One of: yes, no (short description if yes)
Degree of automation	What is the degree of automation of the approach described in the publication?	One of: fully automated, semi automated (short description if semi automated)
RQ3: Competences and learning objectives		
Didactic aspects	Are didactic aspects like competences or learning objectives explicitly mentioned in the publication?	One of: yes, no, indirectly (short description if yes or indirectly)
RQ4: Feedback		
Textual feedback	Does the automated assessment described in the publication generate textual feedback?	One of: not specified, no, yes
Graphical feedback	Does the automated assessment described in the publication generate graphical feedback?	One of: not specified, no, yes
Score	Does the automated assessment described in the publication provide a score?	One of: not specified, no, yes, indirectly
RQ5: Educational context		
Stage and Context	Which stages of the learning process are reported as suitable for the automated assessment described in the publication? What is the application context?	List of: formative, summative, (short description of application context) or not specified

Table 1: Data extraction form as part of the review protocol

in a spreadsheet with a column for each of the considered aspects. In a second pass, each column in that table was re-checked and consolidated by a member of the research team. With this procedure we were able to detect inconsistencies between the way different members of the research team worked and could resolve these via joint discussions and subsequent adjustments of the coding of the entries in the spreadsheet. As most questions were phrased as closed questions, the coding procedure was top down and rather straightforward, except for the fields regarding addressed aspects (Table 1, RQ1). Here, we decided to add "indirectly" as possible value next to "yes" and "no" (and "not specified") to account for the cases where for example syntax rules are not checked explicitly but the comparison with a sample solution implicitly addresses syntactic correctness (see Sec. 4.2). For the openly coded fields (Table 1, Metadata: Model type(s), RQ2: Technique or method) the coding values were derived in a bottom-up fashion through identification of consistent designators respectively suitable classes. Moreover, all fields presented in the data extraction form (Table 1) are explained further during the presentation of the results (see Sec. 4), for example our understanding of "semi automated" and "fully automated" (Table 1, RQ2: Degree of automation) is revealed there.

Post analysis, we applied a filter on the original set of 124 papers in order to exclude publications describing only conceptual work which has not been implemented yet with regard to the chosen research questions of this paper aiming at examining the current state of the art. This led to the exclusion of 14 publications and thus a final result set of 110 selected publications. All publications are referenced in Table 5 in the appendix. Because of the large result set, we have opted to publish the spreadsheet containing our analysis online so that it is publicly accessible (Ullrich et al. 2022). This both saves space in the article itself and allows the interested reader to conveniently browse (and reuse) the presented data.

3.3 Limitations

During the search process, we opted to exclude publications other than journal articles and conference papers. The contributions of a few PhD theses we encountered were also published in other form (and thus represented in the result set) and some Master theses and technical reports could not be retrieved.

Also, the forward search was limited to publications accessed via Scopus, which allowed for an immediate display of a list of articles citing the selected publication (“Cited by”).

Moreover, due to the original focus of our literature review on specific diagram types UML, Petri nets, BPMN and EPC which is also partly reflected in the search string, there is a risk of bias, i. e. possibly leading to an overrepresentation of these diagram types in the result set. However, the final results with regard to the covered diagram types (see Sec. 4.1) do not support this assumption.

4 Results

This section is structured into subsections dedicated to the research questions presented in Sec. 1. It should be pointed out here, that the literature analysis was performed on the basis of the selected publications without aggregating approaches belonging to the same assessment system. Consequently, some approaches might be over-represented in the set of selected publications. This should be kept in mind when looking at the numbers given in this section.

4.1 General findings

The set of 110 selected publications consists of 81 conference papers (ca. 74 %) and 29 articles (ca. 26 %) (see. Fig. 5). Table 2 shows outlets where two or more papers ($N >= 2$) of the result set have been published. Notably, the selected publications are scattered across 77 distinct outlets. These results indicate that the automated assessment of models is a niche topic, mostly placed in journals or at conferences with an educational focus. However it is also observed that

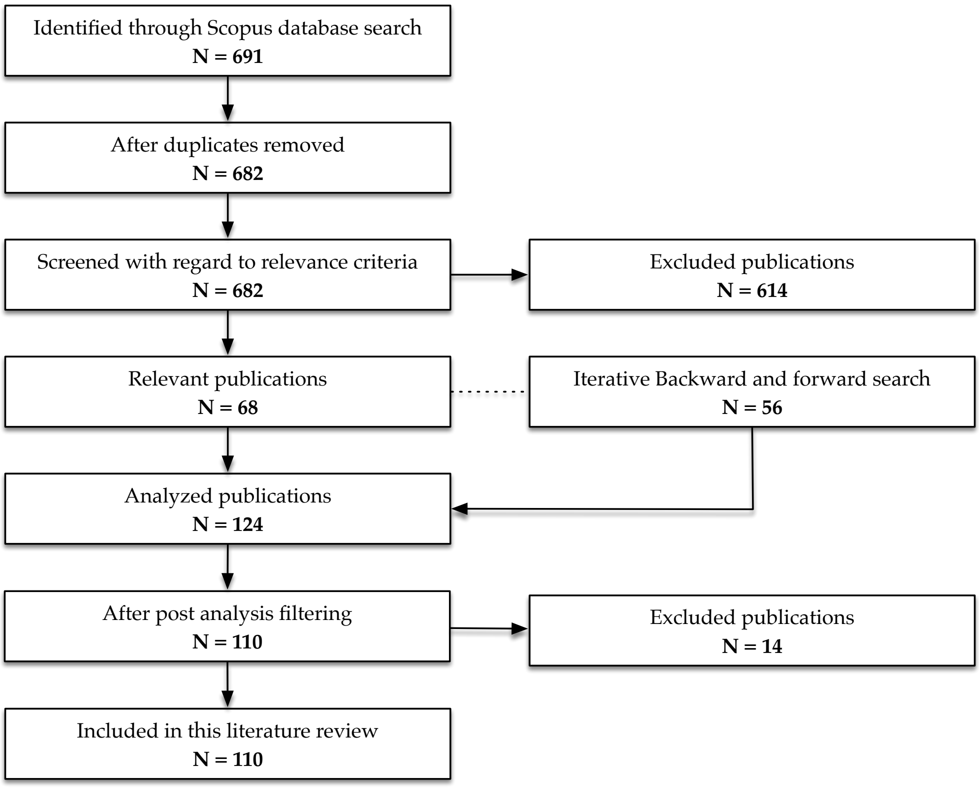

Figure 4: Overview of the literature search process resulting in the set of papers used as basis for the review.

respective contributions are published at subject-specific conferences in the context of modeling, e. g., MODELS, Diagrams or SLATE.

The distribution of the selected publications over the years is presented in Fig. 6. While there is a peak of the number of publications observed in the range of the years 2007 and 2011 and a decrease afterwards, in 2018 the number of publications is rising again. Possibly the curve is following the general interest in the topic e-learning as suggested e. g., by the Web of Science[6] publication data for the keyword "e-learning". The year 2021 should not be taken into account here because the literature search was performed in February of 2021.

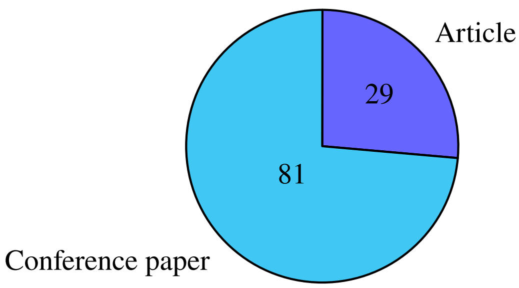

Figure 5: Share of type of the selected publications

Table 3 presents an overview about the model types covered by the publications in the set of selected papers. Again, those numbers have to be interpreted with care, because (i) some approaches are over-represented in this set by being included with multiple publications and (ii) some publications describe approaches for more than one model type. In addition, model types reported only once are: (generic) automata diagram, CASE diagram, BPMN, decision tree diagram, non deterministic finite automaton (NFA), push down automaton (PDA), block diagram, EPC and Metra Potential net plan. Moreover, 27 publications strongly emphasize that the suggested approach is

[6] https://www.webofknowledge.com

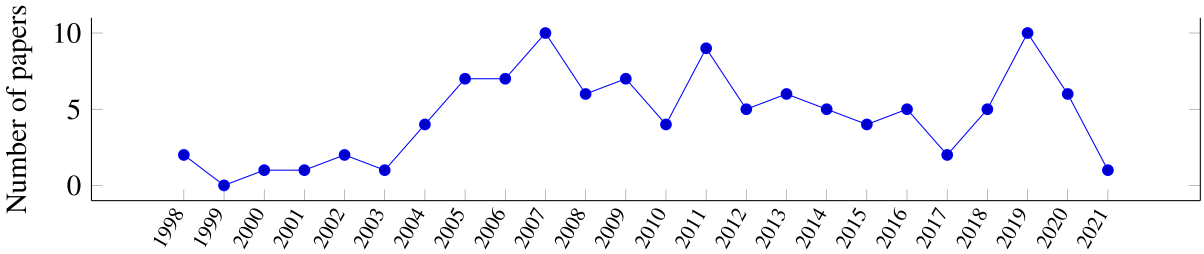

Figure 6: Publication per year in the set of selected publications

of universal nature for graph-based diagrams (e. g., by using a generic term in the title like "graph-based diagram" or just "diagram" or by explicitly stating this in the text). Of those 27 publications, 22 describe an implementation and evaluation of their approach for a specific model type (and thus were counted respectively in Table 3) while 5 completely abstract from a concrete model type. The individual analysis for each publication is presented in Table 5 in the appendix (see column “Model type(s)”).

4.2 RQ 1: Addressed aspects

In general, the 110 analyzed publications present research approaches on the automated assessment of conceptual models, especially on available system implementations supporting automated model evaluation and assessment which address and focus on a variety of different aspects. There are, e. g., approaches focusing on particular design patterns (Reischmann and Kuchen 2016, 2018, 2019; Van Doorn et al. 2019), structural aspects of evaluated models (Batmaz and C. J. Hinde 2007; Smith et al. 2010; Soler et al. 2010b; Sousa and Leal 2015; Thomas et al. 2009a) or on the identification of defects (Hasker and Rowe 2011) in student models. In the following, we analyze the automated assessment of the model quality aspects syntax, semantics and pragmatics.

The automated assessment of the model syntax is supported in 71 of the 110 analyzed contributions. While 36 of these 71 contributions describe dedicated and language-specific approaches for syntax checking concerning the addressed conceptual model type, the other 35 contributions report on matching student models with available sample solutions regarding the modeling problem to be tackled in a modeling task and thus implicitly also check syntactic correctness. 35 of the 110 contributions do not provide automated model syntax checks and 4 of the 110 papers do not allow for a secured statement about their system’s provision of syntax checks.

The automated assessment of the model semantics is supported by the approaches presented in 101 of the 110 analyzed papers. 56 of these 101 papers present dedicated approaches to check the semantic correctness of student models in relation to the modeling problem to be addressed in a modeling task. 45 of the 101 papers implicitly check for semantic correctness through matching student models with sample solutions concerning the modeling task. 7 of the 110 contributions do not provide automated checks for semantic correctness and for 2 of the 110 papers no secured conclusion about their system’s provision of syntax checks can be made.

In regard to the pragmatic aspect of conceptual models such as model understandability, only 6 of the 110 analyzed papers present an implemented approach for automated assessment. Two of the six papers assess pragmatic aspects of the model quality, e. g., using established understandability metrics (Thaler et al. 2016) or design patterns (Coelho and Murphy 2007). The other four of the six papers describe indirect solutions for understandability assessment based on manual assessment or matching model solutions with given models which are considered to be easy to understand, e. g., in the contributions by Boubekeur et al.

Journal	N
International Journal of Artificial Intelligence in Education	5
ACM Journal on Educational Resources in Computing	2
IEEE Transactions on Learning Technologies	2
International Journal of Distance Education Technologies	2
Conference	N
ACM Innovation and Technology in Computer Science Education Conference (ITiCSE)	10
International Conference on Intelligent Tutoring Systems (ITS)	5
International Computer Assisted Assessment Conference (CAA)	4
International Conference on Model Driven Engineering Languages and Systems (MODELS)	4
ASEE Annual Conference & Exposition	2
World Conference on Educational Multimedia, Hypermedia and Telecommunications	2
International Conference on Artificial Intelligence in Education (AIED)	2
International Conference on Computer Supported Education (CSEDU)	2
International Conference on Computers in Education (ICCE)	2
International Conference on the Theory and Application of Diagrams (Diagrams)	2
Koli Calling Conference on Computing Education Research	2
Symposium on Languages, Applications and Technologies (SLATE)	2

Table 2: Outlets with two or more publications from the result set.

Model type	N
UML class diagram	40
ER diagram	38
UML sequence diagram	7
UML use case diagram	6
deterministic finite automaton	5
digital circuit	4
UML activity diagram	4
UML state machine	3
flowchart	3
coloured Petri net	2
EER diagram	2

Table 3: Model types covered by the approaches described in the set of selected publications with $N >= 2$.

(2020), Sanchez-Ferreres et al. (2020), and Westergaard et al. (2012, 2013). 99 of the 110 papers do not provide automated checks for pragmatic model aspects and 5 of the 110 papers do not make any statements and do not allow for a secured statement about their provision of automated checks for pragmatic model aspects.

Hence, regarding the research question concerning the addressed aspects of conceptual models in automated assessment systems it can be stated that current available approaches mostly address semantic and syntactic correctness - in both cases either as reported explicitly - or implicitly by matching student solutions with sample solutions. Pragmatic aspects currently play a subordinate role in the automated model assessment.

4.3 RQ 2: Technical realization

In this section we analyze aspects regarding the technical realization of the approaches presented in the 110 reviewed papers from their presentation of the implementation details. Three papers did not reveal enough information about the implementation to be included in this analysis. In the remaining 107 papers, we (i) identified several technical methods used in the respective approaches and analyzed (ii) the use of specific techniques addressing the problem of label matching in graph-based diagrams and (iii) the degree of automation in the automated assessment approach. The results are presented in the following paragraphs.

78 papers report on employing *model comparison* methods. Here, the assessed model is compared against one (or more) solution model(s). This has been achieved through e. g., similarity measures (Fauzan et al. 2020; Shukur and Mohamed 2008; Sousa and Leal 2015; Vachharajani and Pareek 2019) or graph matching (Outair et al. 2021; Reischmann and Menezes 2019). *Rule-based approaches* employing the checking of criteria have been reported in 41 papers. Examples are approaches based on rules in form of graph queries (Striewe and Goedicke 2011), properties (Westergaard et al. 2012), metrics (Lino et al. 2019), defects (Hasker and Rowe 2011) or searching patterns (Van Doorn et al. 2019) in the assessed model. Also the so-called constraint-based approaches (e. g., Baghaei and Mitrovic 2007a; Mitrovic and Suraweera 2016) from the intelligent tutoring systems research area (15 papers) were attributed towards rule-based approaches. Four recent papers present approaches exploiting the potential resting within *machine learning* methods (Boubekeur et al. 2020; Lino et al. 2019; Reischmann and Menezes 2019; Stikkolorum et al. 2019). Further examples for specific techniques reported in rare cases are e. g., simulation of the assessed model (Jayawardena et al. 2018; Ogata and Kayama 2019; Ogata et al. 2019), testing strategies (Beck et al. 2015; Shekhar et al. 2014), clustering of models (Batmaz and C. Hinde 2006; Batmaz et al. 2010; Al-Hoqani 2018) and the alignment of the assessed model with an annotated textual description (Sanchez-Ferreres et al. 2020). Notably, the aforementioned categories are not exclusive. While the majority of papers (70 of 107) has been mapped to a single category of methods or technique, there are 37 papers combining several of those. Often a model comparison approach is enhanced with some kind of rule-checking, e. g., to assess syntactic correctness (Correia et al. 2018; Thomas et al. 2008c).

We also analyzed the use of specific tools and techniques to address the problem of label matching in graph-based diagrams. 38 papers explicitly report on employing techniques to handle spelling errors or synonyms within the labels of model elements. This is useful e. g., for approaches using model comparison methods when matching model elements of a student solution with a model element of a sample solution or for rule-based approaches using criteria checking to be able to identify modeling elements where checks should be performed upon. Specifically, examples for employed tools and techniques are calculating the edit-distance (Thomas et al. 2009a,c), regular expressions (Demuth and Weigel 2009; Reischmann and Kuchen 2019), spell checking dictionaries, stemming algorithms, Soundex database (Vachharajani et al. 2012), generic (WordNet) (Bian et al. 2019) or domain-specific synonym databases (Thomas et al. 2009b). Three publications in our result set are focused on the problem of label matching in graph-based diagrams and present elaborate approaches employing most of the above mentioned tools and techniques (Jayal and Shepperd 2009; Thomas et al. 2009b; Vachharajani et al. 2012).

Regarding the degree of automation, 88 of the 110 analyzed papers report on fully automated assessment approaches. The remaining 22 papers could be classified as semi automated because they require manual activity during the assessment process. For example, unknown or ambigous model elements are manually assessed or mapped by tutors (Batmaz and C. J. Hinde 2007; Al-Hoqani 2018; Ichinohe et al. 2019; Siepermann 2016) or manual review and enhancement of automated assessment is required explicitly (Coelho and Murphy 2007; Ogata et al. 2019). The four identified machine learning approaches were also counted towards this category, because training data by previous manual assessments is required here.

In summary, the reported findings with regard to RQ 2 reveal that most implementations of automated assessment approaches involve the comparison of a student model with one or more modeling solutions. Also rule-based approaches are used commonly. Moreover, a variety of other techniques, also specifically addressing the problem of label matching in graph-based diagrams has been listed in the above results.

4.4 RQ 3: Competences and learning objectives

Most papers found during the review seem to tackle the subject from a technical perspective. In 99 of the 110 papers there are no or only very vague and abstract references to competences, learning objectives or any other kind of didactic foundations. The need for automated assessment is in many cases motivated by a large amount of students and the burden of grading coursework or exams manually, without naming specific competences to be assessed or skills to be trained. Some papers are slightly more specific, e. g. Bañeres et al. (2014) refer to a concrete curriculum and the related learning objectives on a rather abstract level. Others (like e. g. Siepermann et al. (2013)) at least refer to the possibility to rank tasks according to their difficulty and required knowledge.

Four papers are more specific and name actual skills. Reischmann and Kuchen (2016) motivate their work with the need for assessing "higher-level software engineering skills" and specifically work on assessing the use of software design patterns, but give neither references nor detailed definitions on how software engineering skills are ranked and what defines a higher level. Ichinohe et al. (2019) refer to Sendall and Kozaczynski (2003) when they state that modeling requires "the ability to create a model", "the ability to understand requirements of software" and "the ability to preserve the status when there are changes in the model". They argue that novices specifically miss the ability to understand requirements of software and thus cannot review their own work, which motivates the creation of an automated feedback tool. Two papers from the same group of authors (Baghaei and Mitrovic 2007b; Baghaei et al. 2007) refer to collaboration on modeling and problem solving and include a larger set of references to previous work on computer-supported collaborative learning.

Another set of four papers is even more specific with respect to educational theories on competences or learning objectives by referring explicitly to the taxonomy of educational objectives by Bloom (1956) or its revised version by Anderson et al. (2000). All four papers motivate their work by the aim to reach either generally "higher levels" within these taxonomies or more specifically the "application" level. These references stress the fact that conceptual modeling not only requires theoretical knowledge, but also practical skills that can only by obtained and strengthened through application. They can also be used to distinguish between general purpose automated assessment systems that ask and grade closed questions on the level of knowledge and understanding on the one hand, and more sophisticated automated assessment systems that grade models created by students on the other hand. However, most papers found during the review actually do the latter, but without giving references to the application level of the taxonomies.

In three papers, major references to educational aspects other than competences or learning objectives could be found. Zakharov et al. (2005) name "learning from performance errors" as learning theory, Ifland et al. (2012) mention the Cognitive Apprenticeship Model, and Py et al. (2013) refer to empirical research on difficulties in object-oriented modeling. Similar to the rare references to the taxonomy of educational objectives, there are much more papers that actually make use of some learning theory or teaching model, but do not make these references explicit.

Hence, with respect to RQ 3 it can be stated that didactic aspects like competences or learning objectives for conceptual modeling are only rarely named explicitly in the context of the automated assessment. Notably, that does not mean that they are not considered at all or under-researched in general. It is possible that didactic aspects are published in other places without digging into the details of system descriptions (or before an actual system has been implemented) and have thus been excluded by the relevance criteria. Moreover, many papers implicitly include didactic aspects, but without naming theories explicitly or giving references to foundational work.

4.5 RQ 4: Feedback

Feedback in any form is crucial for learners to improve their understanding of the modeling domain. Some of the papers do not describe the type of feedback explicitly or only in a short paragraph. Nevertheless, in many cases, it was possible to infer the type of feedback used by looking at the illustrations. However, in 7 of the 110 found papers, the provided feedback is not described explicitly and could not be derived from illustrations of the assessment system. There are many ways, in which form feedback for a learner can be presented. Many of the reviewed papers use either a) textual feedback, b) graphical feedback or c) numerical feedback.

Textual feedback are (often) pre-defined text blocks, which are given to the student as feedback based on a sample solution or rule/constraint-based comparison. It can be divided into two different categories: Providing general information about the whole diagram and feedback to a specified model element. Bian et al. (2019) use textual feedback to provide the learner with the found model element matches compared to a sample solution. Another approach is shown by Reischmann and Kuchen (2019). They try to identify the used design patterns in a model. Based on the recognized design pattern, they deliver textual feedback two-staged. Correia et al. (2018) use a multi-stage approach to give more detailed textual feedback every time a wrong solution is uploaded by a learner. Smith et al. (2013) utilize the textual feedback to deliver general information like the number of identified model elements and if these match the number of model elements in the sample solution.

The term graphical feedback describes annotations in the learners solution. These could be simple feedback such as texts at the corresponding model element, highlighting of model elements, or overlaying the learners solution with the sample solution. Thomas et al. (2007b) show an approach in which they overlay the learners solution with the model solution. The learner can gain deeper insight by selecting different parts of the diagram which will highlight the corresponding part of the task text. A set of four papers from the same authors describe a semi-automatic system, in which an evaluator marks the modeled elements with a three-color-encoding (Batmaz and C. Hinde 2006; Batmaz and C. J. Hinde 2007; Batmaz et al. 2010; Stone et al. 2009). Werf and Steehouwer (2019) also uses color-encoding and detailed feedback, when a learner selects a specific model element. In contrast to the previously described the feedback is not delivered after the submission of the task but live.

Most of the papers found generate either a score in percent (for example Echeverria et al. (2017) and Siepermann et al. (2013) or total score (such as Bian et al. 2019). Also, often these scores are shown to the learner as a mark (like Shukur and Mohamed 2008; Thomas et al. 2007a). Several papers generate a grading based on the degree of similarity to a sample solution (Echeverria et al. 2017; Ichinohe et al. 2019; Simanjuntak 2015). It seems that most papers focus on generating a single score, even if the grading is based on different categories.

44 publications were found which concentrated on one of the above-mentioned feedback types in their assessments. Especially in the context of formative assessment, it can be advantageous to give the student feedback in as many ways as possible. From the 110 papers found, only 11 (ca. 12,1%) mentioned textual, numerical, and graphical feedback connected. One application, which uses all feedback types mentioned above, is shown by Bian et al. (2019). The feedback is generated by comparing against a sample solution. The textual feedback includes all matches found while the similarity quantification is shown highlighted in the learners solution. Thomas et al. (2008c) implemented various tools for automated marking of diagrams with different types of feedback. One tool, which uses all feedback types is called Sequence Diagram Drawing Exerciser. Incorrect model elements are drawn with

dashed lines. The learner can get additional textual feedback by clicking a button. A mark is calculated based on so-called minimal meaningful units. Another paper found using all feedback types is shown by Tselonis and Sargeant (2007). All feedback is shown in the modeling area of the tool via a popup.

There seems to be a lack of systems, which take learning objectives or competencies into account for the feedback generation. Higgins et al. (2002) concluded that not only the type of feedback but also the amount of feedback in particular can be crucial for the learning process. Sanchez-Ferreres et al. (2020) explicitly addressed syntax, semantics, and pragmatics as error types . In five more recent papers, a multi-level feedback was described (Correia et al. 2017, 2018; Reischmann and Kuchen 2018, 2019; Werf and Steehouwer 2019). Not only feedback in the form of error naming supports the learner in his learning process. Three papers name hints (D'Antoni et al. 2015; Demuth and Weigel 2009; Hasker and Rowe 2011), two paper mentioned questions (Elmadani et al. 2015; Parvez and Blank 2008) as feedback. Another aspect of feedback is shown by D'Antoni et al. (2015). They use counterexamples to enable the learner to improve his or her learning additionally. Referring to RQ3, Parvez and Blank (2008) do not use learning competencies but try to identify different learning types (e. g., Active, Reflective) and use the appropriate textual feedback like examples or questions.

In summary, it can be said regarding RQ 4 that various types of feedback were reported. However, the feedback is often not explicitly described, but named in a short paragraph or only visible through figures while the paper focuses other aspects of e-assessment. Therefore it is possible, that the feedback types named are not complete. Also, the feedback type itself does not take the feedback quality into account. Since in many publications the feedback was not directly described it is difficult to distinguish whether given feedback only contains a list with found errors or the learner is guided to a correct solution by hints and explanations.

4.6 RQ 5: Educational context

Usage scenarios for the automated assessment of conceptual models reported in the literature are clearly dominated by formative settings (57 papers), while only seven papers report solely about summative use. Twelve papers report on both kinds of settings and 34 papers make no clear statement about the usage scenarios at all.

There seems to be some alignment between the reported use and the feedback modes as discussed in the previous section. All papers that describe summative scenarios report on feedback in terms of numerical grades or marks (except two papers that make no statement on that aspect), while only 21 out of the 57 papers describing exclusively formative use report on that kind of feedback. In turn, eleven out of the 19 papers that describe summative scenarios report on textual or graphical feedback, while 60 out of 68 papers on formative use report on that kind of feedback.

While the summative use is clearly focused on grading examination submissions, the use in formative settings is much more diverse. 20 papers report on scenarios in conjunction with coursework and thus focus on feedback that is provided after students have finished working on their submission. However, re-submissions of revised work is usually possible in these scenarios, which distinguishes them from summative grading. Papers that report on both summative and formative use also mostly take coursework as an example for the latter. Another set of 20 papers reports on scenarios related to automated tutoring, although the systems used in that context are not necessary naming themselves "(intelligent) tutoring systems". The focus in these scenarios is on a close interrelation between the creation of a model and the provision of feedback. Hence, feedback is not related to an explicit submission, but can also be provided while working on a diagram. Seven papers report on lab-classes as usage scenarios for their systems, which offers to option for additional human tutoring in addition to the automated feedback. One of those (Coelho and Murphy 2007) focuses on collaborative learning in class and there are others (Constantino-Gonzalez et al. 2003; Constantino-González and Suthers 2000; Echeverria et al. 2017) that also focus on collaborative learning in general.

Seven papers describe more general learning tools which can be used in a variety of contexts. For example, Hasker and Rowe (2011) describe a tool that allows students to check any UML model for defects, and Shekhar et al. (2014) provide feedback not only to automatically generated practice problems from their tool, but also to problems taken from other sources such as textbooks. Notably, that latter case of general learning tools is the only case in which an automated assessment tool can be seen solely as an additional learning aid. In contrast to that, the tools in all other scenarios so far are intended to be at least a partial replacement for human tutors and teachers, e. g. to save them from the burden of tedious grading tasks and thus help them to focus on individual feedback, or to provide feedback in cases in which not enough human resources are available anyway.

Notably, the share of papers that report on in-class usage of automated assessment systems in our review is substantially smaller than it is reported by Deeva et al. (2021) in their domain-independent review of automated feedback systems. In that review, 54% of the papers have reported in-class usage and 12% have reported blended usage. However, that review explicitly focused on papers that use the word "feedback" in the tile and hence excluded all papers that use other terms like "assessment" or "grading". In our survey, we also include grading for exams and coursework, that usually does not happen in class. For the remaining, formative scenarios, 12% report explicitly on in-class usage and 53% can potentially be used in blended scenarios, so the total share for in-class and blended usage is almost the same in our results, but the emphasis is different.

As a summary for RQ 5 it can be stated that automated assessment systems for conceptual modeling appear in any stage of the learning process and that they are used in a variety of different settings. There is an emphasis on tools for formative feedback and these tools are often not limited to a particular scenario, but can be applied in different contexts.

5 Discussion

In this section, the previously presented results are discussed. We also intend to reflect on limitations and shortcomings in the context of our analysis and possible strategies to overcome those. Obviously, a common limitation to this study is that the analyzed set of publications is likely to miss a few relevant sources due to the restrictions of the search. However, there is clearly a trade-off between search repeatability using search-strings and the completeness of the set of relevant articles. As this article should provide an overview about the addressed aspects rather than an in-depth analysis of a complete set of relevant articles, we opted for search repeatability so that it is feasible to repeat the search (possibly adapted) in a follow-up study.

5.1 RQ 1: Addressed aspects

The investigation of RQ 1 has revealed that currently available assessment systems mostly address semantic and syntactic aspects. Notably, there is an obvious interdependency between syntactic and semantic aspects which is also mentioned by Moody et al. (2002). A student model exhibiting syntactic errors consequently has ambiguos semantics, thus posing a challenge for automated assessment approaches that needs to be addressed and investigated further.

While pragmatic aspects, such as automated model understandability assessment currently play a subordinate role, it could be argued that particularly taking pragmatic aspects into account can significantly improve the impact of modeling education for practical modeling projects and conceptual modeling endeavors in companies, or digital government projects. Conceptual models can only fulfill their function and purpose properly if they possess an appropriate pragmatic quality e. g., understandability (Lindland et al. 1994). Furthermore, model understandability is particularly important when conceptual models are used in order to support the communication about and a collective understanding of the functionality and structure of information systems or organizational structures to be developed in a project (Krogstie 2007). If users or stakeholders of a development project in practice or research do not correctly comprehend a conceptual model, the communication about the model content or the verification of whether central requirements of the project are effectively met is not possible (Houy et al. 2012). In order to avoid such situations, it is of utmost importance to develop pragmatically usable conceptual models which are easy to understand. This is why pragmatic model aspects should play a more significant role in learning and teaching conceptual modeling. Hence, this aspect could and should be addressed more strongly in future implementations of automated assessment systems for conceptual modeling. To this end, a closer look at the identified approaches considering pragmatic aspects that were described in Sec. 4.2 is advisable. Moreover, both results from the research area of model understandability (Mendling et al. 2007; Vanderfeesten et al. 2008) and modeling guidelines might serve as a good starting point. While two such guidelines (Mendling et al. 2010; Schütte and Rotthowe 1998) have already been cited in Sec. 2.1, there is further relevant literature on this topic, e. g. pragmatic guidelines for business process modeling (Moreno-Montes de Oca and Snoeck 2015).

5.2 RQ 2: Technical realization

Tentatively, the results of RQ 2 are intended to give a broad overview about the technical methods employed in the context of the automated assessment of models. However, due to the often very rudimentary and simplified description of the technical implementation in scientific publications it was not feasible to derive a precise classification system and achieve a reliable mapping of the analyzed papers to specific categories. This would require further investigation (e. g., by interviewing developers or examining source code where possible).

Also, however promising an automated assessment of models may be, the technical realization of such still presents a challenge due to the inherent degree of freedom in classical open-ended modeling tasks (e. g., individual abstraction, labeling and layout choices have to be made) and thus the lack of an unique correct solution. This is reflected by comparative evaluations of automated assessment and human graders as reported in the analyzed papers. Those clearly show that while some results are quite impressive, overall there is still space for improvement with regard to the accuracy of automated assessment approaches (e. g., Ogata et al. 2019; Stikkolorum et al. 2019; Thomas 2013). To this end, a comparison of the identified approaches with regard to the accuracy of the automated assessment is desirable. Ideally, this comparison reveals, which categories of methods and techniques perform best (and for which model types) and which strengths and weaknesses exist. This would require at minimum a deeper analysis of the evaluations described in the publications in the form of a meta study or preferably, the conduction of a large comparative evaluation where several (publicly available/accessible) approaches are tested against each other on a common data basis of student models.

With this knowledge gathered, we see an especially big opportunity in combining and refining already existing approaches for the automated assessment of models. For example, some of the elaborate approaches which were proposed for the problem of label matching could be incorporated in either existing or upcoming assessment approaches to overcome common obstacles instead of reinventing the wheel each time. This also applies for the transfer of existing approaches to other model types or the even the development of cross-language and universal assessment solutions (for examples for the latter, see Table 5 in the appendix).

5.3 RQ 3: Competences and learning objectives

The most striking result with respect to competences and learning objectives was that only a few papers make explicit references to educational theories. It might be true that this result is somewhat biased due to the design of our inclusion criteria and that in fact more tools have been designed with educational theories in mind. Many papers indeed create the impression that there are links to theory that are not made explicit. Thus, a detailed content analysis that maps key terms and concepts from the papers to educational theories seems to be advisable to come to more solid conclusions.

Nevertheless, the missing explicit links also point towards another direction: Many papers motivate system development with the burden of tedious and error-prone manual grading and thus strive to automate what educators currently do. Omitting explicit references to educational theories thus means to miss the chance to reflect on what is currently done in education. However, that reflection seems to be advisable if automated assessment should not only be designed to replace manual work but to actually enhance the learning experience. Since assessing competences and aligning assessments with learning objectives are complex tasks, automated systems create the opportunity to use specialized systems that partially solve these tasks and thus substantially improve teaching, learning and assessment. Indeed we have found some examples for systems that are specifically tailored for some learning objective (e. g. collaborative modeling) and that thus successfully demonstrate that this is a meaningful way to go. The further development of automated assessment systems for conceptual modeling is thus a chance to actually support more aspects of modeling in education than it is currently done.

Because competences and learning objectives do not only play a role in automated assessment but are of utmost importance for teaching, learning and assessment in general, further related research activity regarding learning objectives and competence frameworks for conceptual modeling (e. g., Bogdanova and Snoeck 2019; Bork 2019; Soyka et al. 2022) is also worthwhile pursuing.

5.4 RQ 4: Feedback

As stated in Sec. 4.5, the type of feedback was not always clearly described and had to be inferred by illustrations. However, we could identify various types of feedback used in automated assessment systems grouped into textual, graphical and numerical feedback.

The investigation showed that feedback is often delivered only through one of the feedback types. However, the type, quality and amount of feedback used are rarely discussed in the reviewed papers. When the feedback is not only used for grading, these aspects should adapt to the learners needs. It could be possible, that these aspects were taken into account while designing the feedback of the automated assessment system, but are not described in the publication. Therefore, it is unfortunately not possible to conclude whether these aspects have been taken into account.

There is a lack of papers that discuss, how feedback should be presented to the learner. It could be beneficial to present the feedback graphically because the learner can easily identify which feedback belongs to which model element. However not every generated feedback can be traced to a specific model element, for example semantic errors like missing model elements, which requires the use of other feedback representations. Therefore, the combined use of different feedback types should be favored.

Another aspect which is only discussed rarely is the time, at which the learner receives feedback. In most automatic assessment systems the learner has to submit his solution and gets feedback afterward, even if the modeling tool is integrated into the system. For formative scenarios, live feedback by the modeling tool could further enhance the learning process. Especially syntax errors could be detected by the modeling tool for specific modeling languages.

5.5 RQ 5: Educational context

The consequences of the observations on educational contexts are two-fold and closely linked to what we have already discussed for RQ 3 (Sec. 5.3): On the one hand, the emphasis on formative tools that are not limited to a specific context means that there are universal tools available that are ready to be used by teachers independent of the actual teaching scenario. That is a promising result for everyone who wants to use such systems for teaching. On the other hand, it means that more research may be necessary to develop systems that support specific scenarios, like the existing ones we have seen specifically for collaborative modeling. Hence again, there seems to be a missing, deeper link between the design of automated assessment systems and the theory behind a particular way of teaching. While available systems are able to provide support for formative and summative scenarios in general, only a few of them provide more support than the plain provision of feedback. We have already discussed above that there are many options for detailed feedback and tailoring those options to specific scenarios is an additional effort that seems to be valuable.

Notably, that does not mean that we advocate in favor of the creation of a lot of specialized systems. Instead, we see the need for more carefully designed configuration options that allow teachers and students to adapt a general assessment system to the particular needs of their scenario. Such options could relate to the amount and type of feedback, but also to e. g. time limits for submissions, modeling aids in an integrated modeling editor, or options for collaboration or peer-review.

6 Summary and Outlook

Besides the reported results, the performed literature review has also produced valuable questions for future research as desired by Brocke et al. (2009). First of all, an aggregation of the reviewed publications would allow to answer more precise questions about automated assessment systems in terms of quantification (e. g., "How many systems cover an automated assessment of UML class diagrams?"). Also, the results of the considered research questions indicate further directions of research. An overview contrasting the current state with the open issues for the considered research questions RQ 1-5 is given in Table 4 and has been summarized below.

In regard to RQ 1, most of the reviewed papers reported on automated assessment systems addressing semantic and syntactic aspects. Furthermore, it was found that pragmatic aspects, such as automated model understandability assessment, play a rather subordinate role. We have discussed in more detail why a stronger focus on pragmatic aspects in learning and teaching conceptual modeling could significantly improve the practical impact of conceptual modeling, especially when conceptual models are used for human communication purposes in organizations or in development projects. Looking at current system implementations for automated assessment of conceptual models it can, nevertheless, be stated that there is a whole lot more potential to involve pragmatic model aspects and, thus, to improve teaching conceptual modeling by supporting learning and teaching with a focus on pragmatic model quality aspects.

The analysis of the selected papers in regard to RQ 2 led to the identification of numerous methods and techniques used for the technical realization of an automated assessment of conceptual models. Yet, as discussed, the inquiry has also shown that a precise category system of existing methods and a meta study analyzing the accuracy of specific methods is desirable. Such a knowledge base would allow the identification of suitable and promising approaches for application purposes in teaching and learning scenarios as well as for refinement with the goal to create more powerful tools for the automated assessment of conceptual models.

The examination regarding RQ 4 showed that a variety of feedback types are used in automated assessment systems. A large set of papers focus on grading the learner solutions and provide feedback based on the found errors. There seems to be a lack of automated assessment systems that provide feedback taking the state of learning into account. For the most part, no distinction is made between formative and summative scenarios in the type of assessment. There has also been limited research

Current state	Open issues
RQ 1: *Which aspects of conceptual models are addressed through automated assessments?*	
• Most publications present approaches addressing semantic and syntactic aspects • Pragmatic aspects such as understandability are addressed only in rare cases	• Stronger emphasis on pragmatic aspects in automated assessment (also consequently represented in learning content and objectives) • Further investigation of the interdependency between syntactic and semantic aspects in the context of automated assessment
RQ 2: *Which technical methods are used to realize the automated assessment of conceptual models?*	
• Predominantly used methods are model comparison and rule-based approaches • Recent emergence of machine learning approaches • For label processing, elaborated approaches have been proposed	• Need for a meta-study comparing the accuracy of specific methods • Stronger focus on developing cross-language, i. e. universal assessment approaches for graph-based diagrams • Take advantage of the opportunity to incorporate and improve already existing solutions
RQ 3: *Are didactic aspects like competences or learning objectives for conceptual modeling considered in the context of the automated assessment?*	
• Need for automated assessment primarily motivated by burden of manual grading in courses with a large amount of participants • Didactic aspects only rarely named explicitly (and if, only vague references)	• References to educational theory should be made explicit, e. g., by creating links between assessment items and desired competences or learning objectives
RQ 4: *Which form(s) of feedback for learners (if any) does the automated assessment of conceptual models produce?*	
• Feedback in textual, graphical and numerical form has been reported (often inferred through figures)	• Stronger consideration and inclusion of the type and level of detail of feedback depending on the application context
RQ 5: *At what stages of the learning process (formative, summative) and in which context (teaching method, course settings) are automated assessments of conceptual models involved?*	
• Automated assessment systems appear in any stage of the learning process • Emphasis on tools for formative feedback • Tools often not limited to a particular scenario	• Need for more carefully designed configuration options to adapt a general system to particular needs or a specific scenario (formative, summative)

Table 4: Overview contrasting current state and open issues regarding the considered research questions

into what type and level of detail of feedback is best suited to an automated assessment approach. In particular, formative assessment should focus on the learning progress of the individual. Accordingly, feedback given in this scenario should be more detailed than in a summative scenario and allow the learner to deepen their knowledge.

Based on the results for RQ 3 and 5 it can be concluded that more explicit connections between automated assessment systems on the one hand and educational settings and theories on the other hand can be created. A large set of papers made no explicit and deep references to educational theories and did not design their systems for a particular scenario. Thus, more detailed work is necessary to find out which actual competences in the area of conceptual modeling can already be assessed with automated systems and how that aligns with the scenarios that are actually used to teach these competences. Any gap that may be found during that analysis is then to be closed with further research on automated assessment systems that are specifically tailored to these competences and scenarios.

References

Alur R., D'Antoni L., Gulwani S., Kini D., Viswanathan M. (2013) Automated grading of DFA constructions. In: Proc. of Int. Joint Conf. on Artificial Intelligence (IJCAI), pp. 1976–1982

Anderson L. W., Krathwohl D. R., Airasian P. W., Cruikshank K. A., Mayer R. E. (Dec. 19, 2000) A Taxonomy for Learning, Teaching, and Assessing. Pearson Education (US) pp. 384

Association for Computing Machinery (ACM) (2018) Curricula Recommendations. https://www.acm.org/education/curricula-recommendations

Auxepaules L., Py D., Lemeunier T. (2008) A diagnosis method that matches class diagrams in a learning environment for object-oriented modeling. In: Proc. of 8th IEEE Int. Conf. on Advanced Learning Technologies (ICALT). IEEE, pp. 26–30

Bachmann H. (2018) Formulating learning outcomes. In: Bachmann H. (ed.) Competence-Oriented Teaching and Learning in Higher Education. hep, Bern, Switzerland, pp. 41–56

Baghaei N., Mitrovic A. (2005) COLLECT-UML: Supporting individual and collaborative learning of UML class diagrams in a constraint-based intelligent tutoring system. In: Proc. of 9th Int. Conf. on Knowledge-Based Intelligent Information and Engineering Systems (KES). Springer, pp. 458–464

Baghaei N., Mitrovic A. (2006) A Constraint-Based Collaborative Environment for Learning UML Class Diagrams. In: Proc. of 8th Int. Conf. on Intelligent Tutoring Systems (ITS). Springer, pp. 176–186

Baghaei N., Mitrovic A. (2007a) Evaluating a Collaborative Constraint-based Tutor for UML Class Diagrams.. In: Proc. of 13th Int. Conf. on Artificial Intelligence in Education (AIED). ACM, pp. 533–535

Baghaei N., Mitrovic A. (2007b) From modelling domain knowledge to metacognitive skills: Extending a constraint-based tutoring system to support collaboration. In: Proc. of 11th Int. User Modeling Conf. (UM). Springer, pp. 217–227

Baghaei N., Mitrovic A., Irwin W. (2005) A constraint-based tutor for learning Object-Oriented analysis and design using UML. In: Proc. of Int. Conf. on Computers in Education (ICCE). IEEE, pp. 9–16

Baghaei N., Mitrovic A., Irwin W. (2006) Problem-Solving Support in a Constraint-based Tutor for UML Class Diagrams. In: Technology, Instruction, Cognition and Learning Journal 4(2)

Baghaei N., Mitrovic A., Irwin W. (2007) Supporting collaborative learning and problem-solving in a constraint-based CSCL environment for UML class diagrams. In: Int. Journal of Computer-Supported Collaborative Learning 2(2-3), pp. 159–190

Bañeres D., Clarisó R., Jorba J., Serra M. (2014) Experiences in Digital Circuit Design Courses: A Self-Study Platform for Learning Support. In: IEEE Transactions on Learning Technologies 7(4), pp. 360–374

Batmaz F., Hinde C. (2006) A diagram drawing tool for semi-automatic assessment of conceptual database diagrams. In: Proc. of 10th Int. Computer Assisted Assessment Conf. (CAA). Loughborough University, pp. 71–84

Batmaz F., Hinde C. J. (2007) A web-based semi-automatic assessment tool for conceptual database diagram. In: Proc. of 6th Int. Conf. on Web-Based Education (WBE), pp. 427–432

Batmaz F., Stone R., Hinde C. (2010) Personalised Feedback With Semi-Automatic Assessment Tool For Conceptual Database Model. In: Innovation in Teaching and Learning in Information and Computer Sciences 9(1), pp. 105–109

Beck P.-D., Mahlmeister T., Ifland M., Puppe F. (2015) COCLAC - Feedback generation for combined UML class and activity diagram modeling tasks. In: Proc. of 2nd Workshop "Automatische Bewertung von Programmieraufgaben" (ABP). CEUR-WS Vol. 1496

Bian W., Alam O., Kienzle J. (2019) Automated grading of class diagrams. In: Proc. of 22nd ACM/IEEE Int. Conf. on Model Driven Engineering Languages and Systems Companion (MODELS). IEEE, pp. 700–709

Bian W., Alam O., Kienzle J. (2020) Is automated grading of models effective?: Assessing automated grading of class diagrams. In: Proc. of 23rd ACM/IEEE Int. Conf. on Model Driven Engineering Languages and Systems (MODELS). ACM, pp. 365–376

Biggs J. B., Tang C. (2011) Teaching for quality learning at university: What the student does. 4. ed. Open University Press, Maidenhead

Blank G., Parvez S., Wei F., Moritz S. (2005) A Web-based ITS for OO Design. In: Proc. of Workshop on Adaptive Systems for Web-based Education at 12th Int. Conf. on Artificial Intelligence in Education (AIED). IOS Press, pp. 59–64

Bloom B. S. (1956) Taxonomy of educational objectives. Handbook 1: Cognitive domain. Longmans, Green and Company, New York

Bogdanova D., Snoeck M. (2019) CaMeLOT: An educational framework for conceptual data modeling. In: Information and software technology 110, pp. 92–107

Bolloju N., Lee K. W. K., Banerjee P. K. (2011) Formative and summative assessment of class diagrams: Development and evaluation of a prototype. In: Proc. of 3rd Int. Conf. on Computer Supported Education (CSEDU). SciTePress, pp. 402–410

Bork D. (2019) A framework for teaching conceptual modeling and metamodeling based on Bloom's revised taxonomy of educational objectives. In: Proc. of 52nd Hawaii Int. Conf. on System Sciences (HICSS), pp. 7701–7710

Boubekeur Y., Mussbacher G., McIntosh S. (2020) Automatic assessment of students' software models using a simple heuristic and machine learning. In: Proc. of 23rd ACM/IEEE Int. Conf. on Model Driven Engineering Languages and Systems (MODELS). ACM, pp. 84–93

vom Brocke J., Simons A., Niehaves B., Riemer K., Plattfaut R., Cleven A. (2009) Reconstructing the giant: On the importance of rigour in documenting the literature search process. In: Proc. of 17th European Conf. on Information Systems (ECIS). AISeL, pp. 2206–2217

vom Brocke J., Simons A., Riemer K., Niehaves B., Plattfaut R., Cleven A. (2015) Standing on the Shoulders of Giants: Challenges and Recommendations of Literature Search in Information Systems Research. In: Communications of the Association for Information Systems 37, pp. 205–224

Chen P. P. (1976) The Entity-Relationship Model – Toward a Unified View of Data. In: ACM Transactions on Database Systems 1(1), pp. 9–36

Coelho W., Murphy G. (2007) ClassCompass: A software design mentoring system. In: Journal on Educational Resources in Computing 7(1)

Constantino-Gonzalez M., Suthers D. D., De los Santos J. G. E. (2003) Coaching Web-based Collaborative Learning based on Problem Solution Differences and Participation. In: Int. Journal of Artificial Intelligence in Education 13(2-4), pp. 263–299

Constantino-González M., Suthers D. D. (2000) A Coached Collaborative Learning Environment for Entity-Relationship Modeling. In: Proc. of 5th Int. Conf. on Intelligent Tutoring Systems (ITS). Springer, pp. 324–333

Correia H., Leal J. P., Paiva J. C. (2017) Enhancing feedback to students in automated diagram assessment. In: Proc. of 6th Symposium on Languages, Applications and Technologies (SLATE) Vol. 56. Dagstuhl Publishing

Correia H., Leal J. P., Paiva J. C. (2018) Improving diagram assessment in Mooshak. In: Proc. of 20th Int. Conf. on Technology Enhanced Assessment (TEA) Vol. 829. Springer, pp. 69–82

D'Antoni L., Kini D., Alur R., Gulwani S., Viswanathan M., Hartmann B. (2015) How Can Automatic Feedback Help Students Construct Automata? In: ACM Transactions on Computer-Human Interaction 22(2)

D'Antoni L., Helfrich M., Kretinsky J., Ramneantu E., Weininger M. (2020) Automata Tutor v3. In: Proc. of 32nd Int. Conf. on Computer Aided Verification (CAV). Springer, pp. 3–14

Daehli O., Kristoffersen B., Lauvås P. J., Myrbakken H. (2018) A supportive web-based tool for learning basic data modeling skills. In: Proc. of 17th European Conf. on e-Learning (ECEL). Academic Conf.s Limited, pp. 116–124

Darshan K. M., Kumar V. (2016) Hybrid ITS for DFA construction problems. In: Proc. of 24th Int. Conf. on Computers in Education (ICCE). Asia-Pacific Society for Computers in Education, pp. 96–98

Deeva G., Bogdanova D., Serral E., Snoeck M., Weerdt J. D. (2021) A review of automated feedback systems for learners: Classification framework, challenges and opportunities. In: Computers & Education 162, p. 104094

Demuth B., Weigel D. (2009) Web based software modeling exercises in large-scale software engineering courses. In: Proc. of 22nd Conf. on Software Engineering Education and Training (CSEET). IEEE, pp. 138–141

Echeverria V., Falcones G., Castells J., Martinez-Maldonado R., Chiluiza K. (2017) Exploring on-time automated assessment in a co-located collaborative system. In: Proc. of 4th Int. Conf. on eDemocracy and eGovernment (ICEDEG). IEEE, pp. 273–276

Elmadani M., Mitrovic A., Weerasinghe A., Neshatian K. (2015) Investigating student interactions with tutorial dialogues in EER-Tutor. In: Research and Practice in Technology Enhanced Learning 10(1)

Fan S. B. (2007) Helping teachers automate student sketch assessment. In: Proc. of IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC). IEEE, pp. 256–257

Fauzan R., Siahaan D., Rochimah S., Triandini E. (2020) A Different Approach on Automated Use Case Diagram Semantic Assessment. In: Int. Journal of Intelligent Engineering and Systems 14(1), pp. 496–505

Fettke P. (2009) How Conceptual Modeling Is Used. In: Communications of the Association for Information Systems 25(43), pp. 571–592

Frank U. (2006) Evaluation of Reference Models. In: Fettke P., Loos P. (eds.) Reference Modeling for Business Systems Analysis. Idea Group, pp. 118–140

Gross S., Mokbel B., Hammer B., Pinkwart N. (2012) Feedback Provision Strategies in Intelligent Tutoring Systems Based on Clustered Solution Spaces. In: Proc. of 10th e-Learning Fachtagung Informatik (DeLFI 2012). GI e.V., pp. 27–38

Guarino N., Guizzardi G., Mylopoulos J. (2020) On the Philosophical Foundations of Conceptual Models. In: Information Modelling and Knowledge Bases XXXI Frontiers in Artificial Intelligence and Applications 321, pp. 1–15

Guo X., Wu C., Chen Y., Xie M. (2011) Interactive items-based online examination system. In: Proc. of Int. Conf. on Internet Technology and Applications (iTAP). IEEE, pp. 1–4

Hall L., Gordon A. (1998) A Virtual Learning Environment for Entity Relationship Modelling. In: Proc. of 29th Technical Symposium on Computer Science Education (SIGCSE). ACM, pp. 345–349

Hasker R. W. (2011) UMLGrader: An Automated Class Diagram Grader. In: Journal of Computing Sciences in Colleges 27(1), pp. 47–54

Hasker R. W., Rowe M. (2011) UMLint: Identifying defects in UML diagrams. In: Proc. of ASEE Annual Conf. & Exposition. ASEE PEER, pp. 22.1558.1–22.1558.14

Hasker R. W., Shi Y. (2014) Teaching basic class diagram notation with UMLGrader. In: Proc. of ASEE Annual Conf. & Exposition. ASEE PEER, pp. 24.1157.1–24.1157.11

Hattie J., Timperley H. (2007) The Power of Feedback. In: Review of educational research 77(1), pp. 81–112

Higgins C., Symeonidis P., Tsintsifas A. (2002) The Marking System for CourseMaster. In: Proc. of 7th Annual Conf. on Innovation and Technology in Computer Science Education (ITiCSE). 3 Vol. 34. ACM, pp. 46–50

Higgins C. A., Bligh B. (2006) Formative computer based assessment in diagram based domains. In: Working Group Reports on Innovation and Technology in Computer Science Education (ITiCSE 2006), pp. 98–102

Hoggarth G., Lockyer M. (1998) An automated student diagram assessment system. In: Proc. of 6th Annual Conf. on the Teaching of Computing and 3rd Annual Conf. on Integrating Technology into Computer Science Education (ITiCSE). ACM, pp. 122–124

Holland J., Baghaei N., Mathews M., Mitrovic A. (2011) The effects of domain and collaboration feedback on learning in a collaborative intelligent tutoring system. In: Proc. of 15th Int. Conf. on Artificial Intelligence in Education (AIED). Springer, pp. 469–471

Hollingsworth J. (1960) Automatic graders for programming classes. In: Communications of the ACM 3(10), pp. 528–529

Al-Hoqani W. M. (2018) Recommendation of new semi-automated approach of decision tree diagram. In: Proc. of Computing Conf.. Institute of Electrical and Electronics Engineers Inc., pp. 1123–1130

Houy C., Fettke P., Loos P. (2012) Understanding Understandability of Conceptual Models – What Are We Actually Talking about? In: Proc. of 31st Int. Conf. on Conceptual Modeling (ER). Springer, pp. 64–77

Houy C., Fettke P., Loos P. (2014) On the Theoretical Foundations of Research into the Understandability of Business Process Models. In: Proc. of 22nd European Conf. on Information Systems (ECIS). AISeL

Ichinohe Y., Hashiura H., Tanaka T., Hazeyama A., Takase H. (2019) Effectiveness of automated grading tool utilizing similarity for conceptual modeling. In: Proc. of 12th Joint Conf. on Knowledge-Based Software Engineering (JCKBSE) Vol. 108. Springer, pp. 117–126

Ifland M., Hörnlein A., Ott J., Puppe F. (2012) Guided tutoring system with increasing support: Autonomy for modeling UML class diagrams. In: Proc. of 10. e-Learning Fachtagung Informatik (DeLFI). GI e.V., pp. 75–86

Javed M., Lin Y. (2020) Automated Assessment of ER Model Using the Domain Knowledge In: Computer and Information Science Springer, pp. 143–162

Jayal A., Shepperd M. (2009) The problem of labels in e-assessment of diagrams. In: Journal of Educational Resources in Computing 8(4)

Jayawardena R. R. A. M. P., Thiwanthi G. A. D., Suriyaarachchi P. S., Withana K. I., Jayawardena C. (2018) Automated Exam Paper Marking System for Structured Questions and Block Diagrams. In: Proc. of 9th IEEE Int. Conf. on Information and Automation for Sustainability (ICIAfS). IEEE, pp. 1–5

Karagiannis D., Kühn H. (2002) Metamodelling platforms. In: Proc. of 3rd Int. Conf. EC-Web. Springer, pp. 182–196

Keuning H., Jeuring J., Heeren B. (Sept. 2018) A Systematic Literature Review of Automated Feedback Generation for Programming Exercises. In: ACM Transactions on Computing Education 19(1)

Kitchenham B., Charters S. (2007) Guidelines for performing Systematic Literature Reviews in Software Engineering. Technical Report No. EBSE-2007-01, Keele University

van der Kleij F. M., Feskens R. C. W., Eggen T. J. H. M. (2015) Effects of Feedback in a Computer-Based Learning Environment on Students' Learning Outcomes: A Meta-Analysis. In: Review of Educational Research 85(4), pp. 475–511

Koeppen K., Hartig J., Klieme E., Leutner D. (2008) Current Issues in Competence Modeling and Assessment. In: Zeitschrift für Psychologie / Journal of Psychology 216(2), pp. 61–73

Krogstie J. (2007) Modelling of the People, by the People, for the People In: Conceptual Modelling in Information Systems Engineering Springer, pp. 305–318

Krogstie J., Lindland O. I., Sindre G. (1995) Defining quality aspects for conceptual models. In: Proc. of IFIP Int. Working Conf. on Information Systems Concepts (ISCO). Springer, pp. 216–231

Krogstie J., Sindre G., Jørgensen H. (2006) Process models representing knowledge for action: A revised quality framework. In: European Journal of Information Systems 15(1), pp. 91–102

Krusche S., von Frankenberg N., Reimer L. M., Bruegge B. (2020) An Interactive Learning Method to Engage Students in Modeling. In: Proc. of 42nd ACM/IEEE Int. Conf. on Software Engineering: Software Engineering Education and Training (ICSE-SEET), pp. 12–22

Larkin J. H., Simon H. A. (1987) Why a Diagram is (Sometimes) Worth Ten Thousand Words. In: Cognitive Science 11(1), pp. 65–100

Le N.-T. (2006) A Constraint-based Assessment Approach for Free- Form Design of Class Diagrams using UML. In: Proc. of Workshop on Intelligent Tutoring Systems for Ill-Defined Domains, pp. 11–19

Leacock C., Chodorow M. (2003) C-rater: Automated Scoring of Short-Answer Questions. In: Computers and the Humanities 37(4), pp. 389–405

Lindland O., Sindre G., Sølvberg A. (1994) Understanding Quality in Conceptual Modeling. In: IEEE Software 11(1), pp. 42–49

Lino A., Rocha Á., Macedo L., Sizo A. (2019) LabDER - Relational database virtual learning environment. In: Proc. of Joint ER Forum and Poster and Demos Session. CEUR-WS Vol. 2469, pp. 42–55

Mctighe J., O'Connor K. (2005) Seven practices for effective learning. In: Educational Leadership 3(63), pp. 10–17

Mendling J., Reijers H. A., van der Aalst W. M. P. (2010) Seven process modeling guidelines (7PMG). In: Information and Software Technology 52(2), pp. 127–136

Mendling J., Reijers H. A., Cardoso J. (2007) What makes process models understandable? In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 4714 LNCS, pp. 48–63

Mitrovic A., Mayo M., Suraweera P., Martin B. (2001) Constraint-Based Tutors: A Success Story. In: Engineering of Intelligent Systems. Springer, pp. 931–940

Mitrovic A., Suraweera P. (2016) Teaching database design with constraint-based tutors. In: Int. Journal of Artificial Intelligence in Education (AIED) 26(1), pp. 448–456

Moody D. L. (2009) The "Physics" of Notations: Toward a Scientific Basis for Constructing Visual Notations in Software Engineering. In: IEEE Transactions on Software Engineering 35(6), pp. 756–779

Moody D. L., Sindre G., Brasethvik T., Sølvberg A. (2002) Evaluating the quality of process models: Empirical testing of a quality framework. In: Proc. of 21st Int. Conf. on Conceptual Modeling (ER 2002) Vol. 2503. Springer, pp. 380–396

Moreno-Montes de Oca I., Snoeck M. (2015) Pragmatic Guidelines for Business Process Modeling. KU Leuven, Faculty of Economics and Business

Moritz S. H., Wei F., Parvez S. M., Blank G. D. (2005) From objects-first to design-first with multimedia and intelligent tutoring. In: Proc. of 10th Annual Conf. on Innovation and Technology in Computer Science Education (ITiCSE). ACM, pp. 99–103

Morris C. W. (1988) Grundlagen der Zeichentheorie. Fischer Verlag

Muñoz de la Peña A., González-Gómez D., Muñoz de la Peña D., Gómez-Estern F., Sánchez Sequedo M. (2013) Automatic Web-Based Grading System: Application in an Advanced Instrumental Analysis Chemistry Laboratory. In: Journal of Chemical Education 90(3), pp. 308–314

Nelson H. J., Poels G., Genero M., Piattini M. (2012) A conceptual modeling quality framework. In: Software Quality Journal 20(1), pp. 201–228

Němec M., Fasuga R. (2011) The automatic evaluation testing: Tasks with graphics character. In: Proc. of Int. Conf. on Mobile, Hybrid, and On-line Learning (eLmL), pp. 19–24

Ogata S., Kayama M. (2019) SML4C: Fully automatic classification of state machine models for model inspection in education. In: Proc. of 22nd ACM/IEEE Int. Conf. on Model Driven Engineering Languages and Systems Companion (MODELS). IEEE, pp. 720–729

Ogata S., Kayama M., Okano K. (2019) Approach to testing many state machine models in education. In: Proc. of 11th Int. Conf. on Computer Supported Education (CSEDU). SciTePress, pp. 481–488

Outair A., Lyhyaoui A., Tanana M. (2014) Towards a semi automatic assessment of UML diagrams by graph transformation. In: Proc. of Int. Conf. on Multimedia Computing and Systems. IEEE, pp. 668–673

Outair A., Mariam T., Abdelouahid L. (2021) New method for summative evaluation of UML class diagrams based on graph similarities. In: Int. Journal of Electrical and Computer Engineering 11(2), pp. 1578–1590

Overhage S., Birkmeier D., Schlauderer S. (2012) Quality Marks, Metrics, and Measurement Procedures for Business Process Models. In: Business Information System Engineering (BISE) 4, pp. 229–246

Parvez S. M., Blank G. D. (2008) Individualizing tutoring with learning style based feedback. In: Proc. of 9th Int. Conf. on Intelligent Tutoring Systems (ITS), pp. 291–301

Peterson J. L. (1981) Petri Net Theory and the Modelling of Systems. Prentice Hall

Prados F., Boada I., Soler J., Poch J. (2006) A Web-Based Tool for Entity-Relationship Modeling. In: Proc. of Computational Science and Its Applications (ICCSA). Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 364–372

Purchase H. C. (2002) Metrics for Graph Drawing Aesthetics. In: Journal of Visual Languages & Computing 13(5), pp. 501–516

Py D., Auxepaules L., Alonso M. (2013) Diagram, a learning environment for initiation to object-oriented modeling with UML class diagrams. In: Journal of Interactive Learning Research 24(4), pp. 425–446

Recker J., Rosemann M. (2009) Teaching business process modelling: experiences and recommendations. In: Communications of the Association for Information Systems 25(1), p. 32

Reischmann T., Kuchen H. (2016) Towards an e-assessment tool for advanced software engineering skills. In: Proc. of 16th Koli Calling Int. Conf. on Computing Education Research (Koli Calling). ACM, pp. 81–90

Reischmann T., Kuchen H. (2018) An interactive learning environment for software engineering design patterns. In: Proc. of 18th Koli Calling Conf. on Computing Education Research (Koli Calling). ACM, pp. 1–2

Reischmann T., Kuchen H. (2019) A web-based e-assessment tool for design patterns in UML class diagrams. In: Proc. of 34th Annual ACM Symposium on Applied Computing (SAC). ACM, pp. 2435–2444

Reischmann T., Menezes B. (2019) Application of Swarm-intelligent Methods to Optimize Error-tolerant Graph Matching for Automatic E-Assessment. In: Proc. of 6th IEEE Latin American Conf. on Computational Intelligence (LA-CCI). IEEE, pp. 1–6

Rieber L. P., Smith M., Al-Ghafry S., Strickland B., Chu G., Spahi F. (1996) The role of meaning in interpreting graphical and textual feedback during a computer-based simulation. In: Computers & Education 27(1), pp. 45–58

Runyon M., Williford B., Linsey J., Hammond T. (2020) An Intelligent Interface for Automatic Grading of Sketched Free Body Diagrams. In: Proc. of 25th Int. Conf. on Intelligent User Interfaces Companion. ACM, pp. 136–137

Sanchez-Ferreres J., Delicado L., Andaloussi A. A., Burattin A., Calderon-Ruiz G., Weber B., Carmona J., Padro L. (2020) Supporting the Process of Learning and Teaching Process Models. In: IEEE Transactions on Learning Technologies 13(3), pp. 552–566

Sangwin C. (2013) Computer Aided Assessment of Mathematics. Oxford University Press

Schramm J., Strickroth S., Le N.-T., Pinkwart N. (2012) Teaching UML skills to novice programmers using a sample solution based intelligent tutoring system. In: Proc. of 25th Int. Florida Artificial Intelligence Research Society Conf. (FLAIRS), pp. 472–477

Schütte R., Rotthowe T. (1998) The Guidelines of Modeling – An Approach to Enhance the Quality in Information Models. In: Proc. of 17th Int. Conf. on Conceptual Modeling (ER). Springer, pp. 240–254

Sendall S., Kozaczynski W. (2003) Model Transformation: The Heart and Soul of Model-Driven Software Development. In: IEEE Software 20(5), pp. 42–45

Shekhar V. S., Agarwalla A., Agarwal A., Nitish B., Kumar V. (2014) Enhancing JFLAP with automata construction problems and automated feedback. In: Proc. of 7th Int. Conf. on Contemporary Computing (IC3). IEEE, pp. 19–23

Shukur Z., Mohamed N. F. (2008) The design of ADAT: A tool for assessing automata-based assignments. In: Journal of Computer Science 4(5), pp. 415–420

Siepermann M. (2016) Database engineering game. In: Proc. of 5th Int. Conf. on Games and Learning Alliance (GALA) Vol. 10056. Springer, pp. 70–79

Siepermann M., Siepermann C., Lackes R. (2013) Electronic exercises for the metra potential method. In: Proc. of 15th Int. Conf. on Enterprise Information Systems (ICEIS) Vol. 2. SciTePress, pp. 435–442

Simanjuntak H. (2015) Proposed framework for automatic grading system of ER diagram. In: Proc. of 7th Int. Conf. on Information Technology and Electrical Engineering (ICITEE). IEEE, pp. 141–146

Smith N., Thomas P., Waugh K. (2004) Interpreting imprecise diagrams. In: Proc. of Int. Conf. on Theory and Application of Diagrams (Diagrams). Springer, pp. 239–241

Smith N., Thomas P., Waugh K. (2010) Diagram interpretation and e-learning systems. In: Proc. of 6th Int. Conf. on the Theory and Application of Diagrams (Diagrams). Springer, pp. 331–333

Smith N., Thomas P., Waugh K. (2013) Automatic grading of free-form diagrams with label hypernymy. In: Proc. of 1st Int. Conf. on Learning and Teaching in Computing and Engineering (LaTiCE), pp. 136–142

Soler J., Boada I., Prados F., Poch J., Fabregat R. (2010a) A Formative Assessment Tool for Conceptual Database Design Using UML Class Diagram. In: 5, pp. 27–33

Soler J., Boada I., Prados F., Poch J., Fabregat R. (2010b) A web-based e-learning tool for UML class diagrams. In: Proc. of IEEE Education Engineering Conf. (EDUCON), pp. 973–979

Soler J., Boada I., Prados F., Poch J., Fabregat R. (2011) A web-based e-learning tool for database design courses. In: Int. Journal of Engineering Education 27(1 PART 1), pp. 61–69

Sousa R., Leal J. P. (2015) A structural approach to assess graph-based exercises. In: Proc. of 4th Int. Symposium on Languages, Applications and Technologies (SLATE) Vol. 563. Springer, pp. 182–193

Soyka C., Schaper N., Bender E., Striewe M., Ullrich M. (2022) Toward a Competence Model for Graphical Modeling. In: ACM Transactions on Computing Education 23(1)

Stachowiak H. (1973) Allgemeine Modelltheorie. Springer, Wien

Stikkolorum D. R., Van Der Putten P., Sperandio C., Chaudron M. R. V. (2019) Towards automated grading of UML class diagrams with machine learning. In: Proc. of 31st Benelux Conf. on Artificial Intelligence and 28th Belgian Dutch Conf. on Machine Learning (BNAIC/BENELEARN). CEUR-WS Vol. 2491

Stone R., Batmaz F., Hinde C. (2009) Drawing and marking graph diagrams. In: Innovation in Teaching and Learning in Information and Computer Sciences 8(2), pp. 45–52

Striewe M. (2014) Automated Assessment of Software Artefacts - A Use Case in E-Assessment. PhD thesis, University of Duisburg-Essen

Striewe M., Goedicke M. (2011) Automated checks on UML diagrams. In: Proc. of 16th Annual Conf. on Innovation and Technology in Computer Science (ITiCSE). ACM, pp. 38–42

Striewe M., Goedicke M. (2014) Automated assessment of UML activity diagrams. In: Proc. of 19th Annual Innovation and Technology in Computer Science Education Conf. (ITiCSE). ACM, p. 336

Suraweera P., Mitrovic A. (2002) Kermit: A constraint-based tutor for database modeling. In: Proc. of 6th Int. Conf. on Intelligent Tutoring Systems (ITS). Springer, pp. 377–387

Suraweera P., Mitrovic A. (2004) An intelligent tutoring system for entity relationship modelling. In: Int. Journal of Artificial Intelligence in Education 14(3-4), pp. 375–417

Suraweera P., Mitrovic A., Martin B. (2004) The use of ontologies in ITS domain knowledge authoring. In: Proc. of 2nd Int. Workshop on Applications of Semantic Web for E-learning (SWEL). University of Canterbury, pp. 41–49

Suraweera P., Mitrovic A., Martin B. (2010) Widening the knowledge acquisition bottleneck for constraint-based tutors. In: Int. Journal of Artificial Intelligence in Education (AIED) 20(2), pp. 137–173

Thaler T., Houy C., Fettke P., Loos P. (2016) Automated assessment of process modeling exams: Basic ideas and prototypical implementation. In: Proc. of 1st Workshop on Modeling in Higher Education (MoHoL). GI e.V., pp. 63–70

Thalheim B. (2018) Conceptual Model Notions – A Matter of Controversy: Conceptual Modelling and its Lacunas. In: Enterprise Modelling and Information Systems Architectures (EMISA) 13(02), pp. 9–27

Thomas P. (2013) Online Automatic Marking of Diagrams. In: 26(4), pp. 349–359

Thomas P., Smith N., Waugh K. (2007a) Computer assisted assessment of diagrams. In: Proc. of 12th Annual Conf. on Innovation and Technology in Computer Science Education (ITiCSE). ACM, pp. 68–72

Thomas P., Smith N., Waugh K. (2007b) Learning and automatically assessing graph-based diagrams. In: Proc. of 14th Int. Conf. of the Association for Learning Technology (ALT-C). Taylor & Francis, pp. 249–267

Thomas P., Smith N., Waugh K. (2008a) Automatic assessment of sequence diagrams. In: Proc. of 12th Int. Computer Assisted Assessment Conf. (CAA). Loughborough University

Thomas P., Smith N., Waugh K. (2008b) Automatically assessing graph-based diagrams. In: Learning, Media and Technology 33(3), pp. 249–267

Thomas P., Smith N., Waugh K. (2009a) Automatically assessing diagrams. In: Proc. of IADIS Int. Conf. e-Learning, pp. 263–270

Thomas P., Smith N., Waugh K. (2009b) The role of labels in the automatic assessment of graph-based diagrams. In: Proc. of 23rd World Conf. on Open Learning and Distance Education

Thomas P., Waugh K., Smith N. (2005) Experiments in the automatic marking of ER-diagrams. In: Proc. of 10th Annual Conf. on Innovation and Technology in Computer Science Education (ITiCSE). ACM, pp. 158–162

Thomas P., Waugh K., Smith N. (2006) Using patterns in the automatic marking of ER-diagrams. In: Proc. of 11th Annual Joint Conf. Integrating Technology into Computer Science Education. ACM, pp. 83–87

Thomas P., Waugh K., Smith N. (2007c) Tools for supporting the teaching and learning of data modelling. In: Proc. of World Conf. on Educational Multimedia, Hypermedia and Telecommunications. AACE, pp. 4014–4023

Thomas P., Waugh K., Smith N. (2008c) A revision tool for teaching and learning sequence diagrams. In: Proc. of World Conf. on Educational Multimedia, Hypermedia and Telecommunications

Thomas P., Waugh K., Smith N. (2009c) Generalised diagram revision tools with automatic marking. In: Proc. of 14th Annual Conf. on Innovation and Technology in Computer Science Education (ITiCSE). ACM, pp. 318–322

Thomas P., Waugh K., Smith N. (2011) Generalised diagramming tools with automatic marking. In: Innovation in Teaching and Learning in Information and Computer Sciences 10(1), pp. 22–34

Thomas P., Waugh K., Smith N. (2012) Automatically assessing free-form diagrams in e-assessment systems. In: Proc. STEM Annual Conf. Imperial College London

Tselonis C., Sargeant J. (2007) Domain-specific formative feedback through domain-independent diagram matching. In: Proc. of 11th Int. Computer Assisted Assessment Conf. (CAA). Loughborough University

Tselonis C., Sargeant J., McGee Wood M. (2005) Diagram matching for human-computer collaborative assessment. In: Proc. of 9th Int. Computer Assisted Assessment Conf. (CAA). Loughborough University

Tsintsifas A. (2002) A Framework for the Computer Based Assessment of Diagram Based Coursework. PhD thesis, University of Nottingham

Ullrich M., Houy C., Stottrop T., Striewe M., Willems B., Fettke P., Loos P., Oberweis A. (2022) Spreadsheet containing SLR analysis. https://docs.google.com/spreadsheets/d/1HWcWvtuC46C-aauCxHTOdzCOW4RuTFNq/edit?usp=sharing&ouid=104591697375237020838&rtpof=true&sd=true

Vachharajani V., Pareek J. (2019) Framework to approximate label matching for automatic assessment of use-case diagram. In: Int. Journal of Distance Education Technologies 17(3), pp. 75–95

Vachharajani V., Pareek J. (2020) Effective structure matching algorithm for automatic assessment of use-case diagram. In: Int. Journal of Distance Education Technologies 18(4), pp. 31–50

Vachharajani V., Pareek J., Gulabani S. (2012) Effective Label Matching for Automatic Evaluation of Use – Case Diagrams. In: Proc. of 4th IEEE Int. Conf. on Technology for Education. IEEE, pp. 172–175

Valenti S., Neri F., Cucchiarelli A. (2003) An Overview of Current Research on Automated Essay Grading. In: Journal of Information Technology Education: Research 2(1), pp. 319–330

Van Doorn E., Stuurman S., Van Eekelen M. (2019) Static detection of design patterns in class diagrams. In: Proc. of 8th Computer Science Education Research Conf. (CSERC). ACM, pp. 79–88

Vanderfeesten I., Reijers H. A., Mendling J., Van Der Aalst W. M., Cardoso J. (2008) On a quest for good process models: The cross-connectivity metric. In: Proc. of 20th Int. Conf. on Advanced Information Systems Engineering (CAiSE), pp. 480–494

Waugh K., Thomas P., Smith N. (2004) Toward the automated assessment of entity-relationship diagrams. In: Proc. of 2nd Workshop of the Learning and Teaching Support Network - Information and Computer Science

Waugh K., Thomas P., Smith N. (2007) Teaching and learning applications related to the automated interpretation of ERDs. In: Proc. of 24th British National Conf. on Databases (BNCOD), pp. 39–47

Weinert F. E. (2001) Concept of competence: A conceptual clarification. In: Rychen D. S., Hersh Salganik L. (eds.) Defining and selecting key competences. Hogrefe & Huber, Seattle, pp. 45–65

van der Werf J. M. E. M., Steehouwer L. (2019) CoRA: An Online Intelligent Tutoring System to Practice Coverability Graph Construction. In: Proc. of 40th Int. Conf. on Application and Theory of Petri Nets and Concurrency (Petri Nets). Springer, pp. 91–100

Westergaard M., Fahland D., Stahl C. (2012) Grade/CPN: Semi-automatic support for teaching petri nets by checking many petri nets against one specification. In: Proc. of Int. Workshop on Petri Nets and Software Engineering 2012 (PNSE). CEUR-WS Vol. 851, pp. 32–46

Westergaard M., Fahland D., Stahl C. (2013) Grade/CPN: A tool and temporal logic for testing colored Petri net models in teaching. In: Proc. of 33rd Int. Conf. on Application and Theory of Petri Nets and Other Models of Concurrency (Petri Nets) Vol. 8100, pp. 180–202

Wijesinghe D. B., Kadupitiya J., Ranathunga S., Dias G. (2017) Automatic Assessment of Student Answers Consisting of Venn and Euler Diagrams. In: Proc. of 17th IEEE Int. Conf. on Advanced Learning Technologies (ICALT). IEEE, pp. 243–247

Zakharov K., Mitrovic A., Ohlsson S. (2005) Feedback Micro-Engineering in EER-Tutor. In: Proc. of Conf. on Artificial Intelligence in Education. IOS Press, pp. 718–725

Zhang H., Babar M. A., Tell P. (2011) Identifying relevant studies in software engineering. In: Information and Software Technology 53(6) Special Section: Best papers from the APSEC, pp. 625–637

Appendix

Table 5: Resulting set of papers regarded in the systematic literature review. The column QGS indicates whether a paper was included in the quasi-gold standard used for the development of the search string.

Publication	QGS	Year	Model type(s)
Hall and Gordon (1998)		1998	ER diagram
Hoggarth and Lockyer (1998)	x	1998	CASE diagram
Constantino-González and Suthers (2000)		2000	ER diagram
Mitrovic et al. (2001)		2001	ER diagram
Higgins et al. (2002)		2002	flowchart, digital circuit
Suraweera and Mitrovic (2002)	x	2002	ER diagram
Constantino-Gonzalez et al. (2003)		2003	ER diagram
Smith et al. (2004)		2004	universal
Suraweera et al. (2004)		2004	ER diagram
Suraweera and Mitrovic (2004)		2004	ER diagram
Waugh et al. (2004)		2004	ER diagram
Baghaei and Mitrovic (2005)		2005	UML class diagram
Baghaei et al. (2005)		2005	UML class diagram
Blank et al. (2005)		2005	UML class diagram
Moritz et al. (2005)		2005	UML class diagram (further UML diagram types possible)
Thomas et al. (2005)	x	2005	universal (ER diagram)
Tselonis et al. (2005)		2005	universal (ER diagram, UML class diagram)
Zakharov et al. (2005)		2005	EER diagram
Baghaei et al. (2006)		2006	UML class diagram
Baghaei and Mitrovic (2006)		2006	UML class diagram
Batmaz and C. Hinde (2006)	x	2006	ER diagram
Higgins and Bligh (2006)	x	2006	universal (ER diagram)
Le (2006)		2006	UML class diagram
Prados et al. (2006)		2006	ER diagram
Thomas et al. (2006)	x	2006	ER diagram
Baghaei et al. (2007)		2007	UML class diagram
Baghaei and Mitrovic (2007b)		2007	UML class diagram
Baghaei and Mitrovic (2007a)		2007	UML class diagram
Batmaz and C. J. Hinde (2007)		2007	ER diagram
Coelho and Murphy (2007)		2007	UML class diagram
Thomas et al. (2007b)		2007	universal (ER diagram)
Thomas et al. (2007c)		2007	universal (ER diagram)
Thomas et al. (2007a)	x	2007	universal (ER diagram)
Tselonis and Sargeant (2007)		2007	universal (ER diagram)
Waugh et al. (2007)	x	2007	ER diagram
Auxepaules et al. (2008)		2008	UML class diagram
Parvez and Blank (2008)		2008	UML class diagram
Shukur and Mohamed (2008)	x	2008	automata diagram
Thomas et al. (2008b)		2008	universal (ER diagram)
Thomas et al. (2008c)		2008	universal (UML sequence diagram)
Thomas et al. (2008a)	x	2008	universal (UML sequence diagram)
Batmaz et al. (2010)		2009	ER diagram
Demuth and Weigel (2009)		2009	UML class diagram
Jayal and Shepperd (2009)	x	2009	universal (UML activity diagram)
Stone et al. (2009)		2009	universal (ER diagram)

continued on next page

Table 5 – continued from previous page

Publication	QGS	Year	Model type(s)
Thomas et al. (2009b)		2009	universal (ER diagram)
Thomas et al. (2009a)	x	2009	universal (ER diagram)
Thomas et al. (2009c)		2009	universal (ER diagram, UML sequence diagram)
Smith et al. (2010)	x	2010	universal (UML class diagram, UML sequence diagram)
Soler et al. (2010a)		2010	UML class diagram
Soler et al. (2010b)	x	2010	UML class diagram
Suraweera et al. (2010)		2010	universal (ER diagram)
Bolloju et al. (2011)		2011	UML class diagram
Guo et al. (2011)		2011	universal (ER diagram, flowchart)
Hasker and Rowe (2011)		2011	UML class diagram, UML use case diagram
Hasker (2011)		2011	UML class diagram
Holland et al. (2011)		2011	UML (diagram type not specified)
Němec and Fasuga (2011)		2011	universal (ER diagram, UML class diagram, digital circuit)
Soler et al. (2011)		2011	ER diagram
Striewe and Goedicke (2011)	x	2011	UML class diagram (further UML diagram types possible)
Thomas et al. (2011)	x	2011	universal (ER diagram, UML sequence diagram)
Ifland et al. (2012)		2012	UML class diagram
Schramm et al. (2012)		2012	UML class diagram, UML activity diagram
Thomas et al. (2012)		2012	universal
Vachharajani et al. (2012)		2012	UML use case diagram
Westergaard et al. (2012)	x	2012	coloured Petri net
Alur et al. (2013)	x	2013	deterministic finite automaton (DFA)
Py et al. (2013)		2013	UML class diagram
Siepermann et al. (2013)		2013	Metra Potential Method net plan
Smith et al. (2013)	x	2013	universal (ER diagram, UML sequence diagram)
Thomas (2013)		2013	universal (ER diagram)
Westergaard et al. (2013)	x	2013	coloured Petri net
Bañeres et al. (2014)		2014	digital circuit
Hasker and Shi (2014)		2014	UML class diagram
Outair et al. (2014)		2014	UML class diagram, UML state machine, UML sequence diagram, UML use case diagram
Shekhar et al. (2014)		2014	deterministic finite automaton (DFA)
Striewe and Goedicke (2014)	x	2014	UML activity diagram
Beck et al. (2015)		2015	UML class diagram, UML activity diagram
D'Antoni et al. (2015)		2015	deterministic finite automaton (DFA)
Elmadani et al. (2015)		2015	EER diagram
Sousa and Leal (2015)	x	2015	universal
Darshan and Kumar (2016)		2016	deterministic finite automaton (DFA)
Mitrovic and Suraweera (2016)		2016	ER diagram
Reischmann and Kuchen (2016)	x	2016	UML class diagram
Siepermann (2016)		2016	ER diagram
Thaler et al. (2016)	x	2016	EPC
Correia et al. (2017)	x	2017	universal
Echeverria et al. (2017)		2017	ER diagram
Al-Hoqani (2018)		2018	decision tree diagram
Correia et al. (2018)	x	2018	universal
Daehli et al. (2018)		2018	ER diagram
Jayawardena et al. (2018)		2018	flowchart, digital circuit, block diagram
Reischmann and Kuchen (2018)		2018	UML class diagram
Bian et al. (2019)		2019	UML class diagram
Ichinohe et al. (2019)		2019	UML class diagram
Lino et al. (2019)		2019	ER diagram

continued on next page

Table 5 – continued from previous page

Publication	QGS	Year	Model type(s)
Ogata et al. (2019)		2019	UML state machine diagram
Ogata and Kayama (2019)		2019	UML state machine diagram
Reischmann and Menezes (2019)		2019	UML class diagram
Reischmann and Kuchen (2019)		2019	UML class diagram
Stikkolorum et al. (2019)		2019	UML class diagram
Vachharajani and Pareek (2019)		2019	UML use case diagram
Van Doorn et al. (2019)		2019	UML class diagram
Bian et al. (2020)		2020	UML class diagram
Boubekeur et al. (2020)		2020	UML class diagram
D'Antoni et al. (2020)		2020	deterministic and non deterministic finite automaton (DFA and NFA), push-down automaton (PDA)
Fauzan et al. (2020)		2020	UML use case diagram
Sanchez-Ferreres et al. (2020)		2020	BPMN (EPC and Petri net extension possible)
Vachharajani and Pareek (2020)		2020	UML use case diagram
Outair et al. (2021)		2021	UML class diagram

Listing 1: Complete search string applied to the Scopus database

```
TITLE-ABS-KEY (( "automatic marking" OR "automated marking" OR "automatic grading"
OR "automated grading" OR "automatic assessment" OR "automated assessment" OR
"assessment system" OR "tutoring system" OR "e-learning system" OR "e-learning tool"
OR ( ( automatic OR automated OR tool OR system ) AND ( teaching OR education )
AND ( checking OR testing ) ) ) AND ( diagram* OR uml OR "petri nets" OR
"petri net" OR "process modeling" OR "process modelling" OR "process models" OR
"database modeling" OR "database modelling" OR dfa OR bpmn OR "business processes"
OR epc ) AND ( EXCLUDE ( DOCTYPE , "cr" ) ) AND ( LIMIT-TO ( LANGUAGE ,
"English" ) ) AND ( LIMIT-TO ( SUBJAREA , "COMP" ) OR LIMIT-TO ( SUBJAREA ,
"ENGI" ) OR LIMIT-TO ( SUBJAREA , "MATH" ) OR LIMIT-TO ( SUBJAREA , "DECI" )
OR LIMIT-TO ( SUBJAREA , "BUSI" ) OR LIMIT-TO ( SUBJAREA , "MULT" ) OR
LIMIT-TO ( SUBJAREA , "ECON" ) OR LIMIT-TO ( SUBJAREA , "Undefined" ) )
```