Information and Software Technology 110 (2019) 92–107

Contents lists available at ScienceDirect

Information and Software Technology

journal homepage: www.elsevier.com/locate/infsof

CaMeLOT: An educational framework for conceptual data modelling

Daria Bogdanova*, Monique Snoeck

Research Center for Management Informatics, KU Leuven, Naamsestraat 69, 3000 Leuven, Belgium

ARTICLE INFO

Keywords:
Bloom's taxonomy
Conceptual data modelling
Educational framework
Software engineering education

ABSTRACT

Context: Teaching conceptual data modelling (CDM) remains a challenging task for educators. Despite the fact that CDM is an integral part of software engineering curricula, there is no generally accepted educational framework for the subject. Moreover, the existing educational literature shows significant gaps when it comes to pursued learning outcomes and their assessment.
Objective: In this paper, we propose an educational framework for conceptual data modelling, based on the revised Bloom's taxonomy of educational objectives, and provide necessary examples of systemized learning outcomes.
Method: We utilized the revised Bloom's taxonomy to develop an adapted framework specifically for learning outcomes related to CDM. We validated the framework by mapping learning outcomes distilled from the existing course material to the framework, by presenting the framework for feedback to the experts in the field and further elaborating and refining it based on the feedback and experiences from these validation activities.
Results: CaMeLOT is an adaptation of the Bloom's taxonomy specifically for learning outcomes related to CDM. We identified different content areas and indicated the necessary scaffolding. Based on the framework, we worked out 17 example tables of learning outcomes related to content areas at different levels of scaffolding, exemplifying the different knowledge and cognitive levels. We clarify the differences in learning outcomes related to different knowledge and cognitive levels and thereby provide a domain specific clarification of the classification guidelines.
Conclusion: CaMeLOT gives educators an opportunity to enhance the CDM part of software engineering curricula with a systemized set of learning outcomes to be pursued, and open the path for creating more complete, useful and effective assessment packages. The adoption of our educational framework may reduce the time spent on designing educational material and, at the same time, improve its quality.

1. Introduction

Conceptual data modelling can be considered a crucial part of software engineering curricula, being "the phase of the information systems development process that involves the abstraction and representation of the real world data pertinent to an organization" [1]. Such development process involves solving ill-structured problems and implies activities at high levels of abstraction, which poses a substantial challenge for educators. Moreover, due to the ill-structured nature of modelling problems, which are in most cases context-dependent, software engineering students have to grasp not only the modelling techniques themselves, but also certain specifics of the domain and context in which the task is situated [2].

Though data modelling is part of the majority of software engineering curricula, no generally accepted educational framework for data modelling exists to this moment. Subsequently, educators become the ones responsible for setting the learning outcomes to be pursued in the data-modelling course and for designing the entire course, based mostly on their own experience and professional judgement. However, not every educator has necessary experience and time resources to come up with a full set of learning outcomes, even though clearly set learning outcomes are essential for the learning process [3]. The path between the two states of "novice modeler" and "expert modeler" remains relatively unclear despite the fact that the differences between them have been already explored in a large number of scientific studies (e.g. [4–6]). Such differences include the stages of modelling implemented by the experts, the differences in modelling patterns, and strategies.

While many educators rely on existing educational literature on data modelling as a basis for their course material and design, the initial study of current educational resources on conceptual data modelling has shown a considerable number of gaps in the learning outcomes presented in those sources [7]. As the research showed, one of the most important types of tasks – evaluation of models, – was significantly underrepresented in the practical educational materials. Moreover, while the tasks related to the ultimate goal – creation of the model, were present in all types of educational resources, the scaffolding steps and the lower levels leading to the mastery of creative skills in modelling remain unequal and lack consistency.

* Corresponding author.
E-mail addresses: daria.bogdanova@kuleuven.be (D. Bogdanova), monique.snoeck@kuleuven.be (M. Snoeck).

https://doi.org/10.1016/j.infsof.2019.02.006
Received 21 December 2017; Received in revised form 28 January 2019; Accepted 22 February 2019
Available online 23 February 2019

Student experiences from conceptual modelling courses have also not been flawless. Despite the fact that the history of conceptual modelling education spans more than several decades, the most common difficulties and mistakes in novice modellers' solutions persist: the difficulties they faced more than 20 years ago, such as understanding the requirements, or choosing the right multiplicity [8], are very similar to those they faced 10 and 20 years later [9,10]. These persistent difficulties and errors correspond to the initial skills that the students have to acquire before being able to create fully functional models.

These concerns suggest that the scaffolding process in conceptual modelling education requires a thorough review and improvement. This study aims at filling the identified gaps by presenting a systematic educational framework for data modelling education, based on the revised Bloom's taxonomy of educational objectives [11]. The work includes the identification of scaffolded content areas, a domain-specific definition of the Bloom's taxonomy's levels, the development of recommendations for learning outcomes classification as well as examples of learning outcomes to be pursued in the identified particular content areas of the course.

2. Background and related work

In the last 2 decades, numerous studies have been made on software engineering pedagogy, exploring various approaches to curriculum development and enhancement (e.g. [12–14]). However, very few of them explore the pedagogy related to the data-modelling phase of software engineering in depth. In 2005, Cowling [15] argued that modelling in general had not been given sufficient attention in the software engineering curriculum, while the role of modelling is crucial in the software development process. Though the state of affairs for modelling pedagogy has slightly changed since then, the currently used standards such as MSIS 2006 [16] or IEEE SE2014 curriculum guidelines for software engineering education [17], do not address conceptual data modelling in detail, giving only the general overview of learning objectives related to the entire field of software engineering. Such widespread professional frameworks as SFIA [18] also provide only basic description of competences and learning goals related to modelling: for example, SFIA guide describes five competence levels related to data modelling and design, however, each level is described only by a short list of professional activities that a software engineer or a similar specialist should perform.

The challenges software engineering educators face when teaching conceptual data modelling are not always intrinsic to this particular field – educators in many other fields face a wide variety of unsystematic resources and approaches, as well as the lack of a generally accepted set of learning outcomes to pursue. To resolve these issues, the educational community applies various classifications, the most prominent of which are Bloom's taxonomy [19] and the revised Bloom's taxonomy [11] of educational objectives.

Developed in late 1960s, Bloom's taxonomy of educational objectives has been used as one of the primary means of classifying and generating systematic educational frameworks and assessment tools. The implications of the taxonomy and its revised version produced tens of frameworks in a wide variety of fields. The University of Washington applied the Blooming Biology Tool as a unified evaluation kit for developing a framework of learning outcomes in the domain of biology [20]. The tool was later applied to the standardized biology exams to assess whether the balance between the tasks requiring lower-order thinking and those requiring higher-order thinking is kept [21]. Later on, the "Blooming" tools expanded to more narrow fields: consequently, the Blooming Anatomy Tool [22] and the Bloom's Taxonomy Histology Tool [23] were successfully applied in the higher education context.

Bloom's taxonomy has been successfully applied in other fields, and, which is of particular interest for this study, in software engineering education. Starr, Manaris and Stalvey [24] proposed general guidelines to create (rather than assess) the learning outcomes in computer science courses based on the taxonomy. Though the guidelines provided a good outline for applying Bloom's taxonomy, they did not go into the specifics of classification process. Moreover, the authors reported on several problems they faced in the implementation of the taxonomy: they observed a phenomenon of "concept shifting", when a concept for which the new learning outcomes are developed is switched to a related one: for example, an abstract concept of "iterative process" or "iteration" may shift to a much more concrete concept of "loop" or "for loop" in a programming course. In addition, according to the same authors, the application of Bloom's taxonomy requires considerable amount of practice and memory efforts. Nevertheless, despite certain shortcomings, the use of taxonomy for designing new learning material reportedly improved course delivery and assessment [24]. In addition, Bloom's taxonomy was applied for assessing the software engineering curriculum and the IEEE software engineering body of knowledge [25].

Regarding the application of Bloom's taxonomy in teaching modelling, the Technical University of Dresden successfully applied Bloom's taxonomy along with constructive alignment [26] to an undergraduate object-oriented modelling course, improving the overall exam results throughout the 3 years of implementation.

3. Design methodology

CaMeLOT's design was performed using the design science approach adapted from the Information Systems Research Framework developed by Hevner et al. [27]. The choice of the approach was determined by the nature of the aim of this study: development of a taxonomy of learning outcomes for future practical use by the conceptual modelling educators, taking into account the current state of the art and needs of the modelling community.

As described by Hevner et al., design science research (DSR) is conducted according to the following three cycles of activities:

- the Relevance Cycle, which embodies the initialization of research: the requirements, such as the actual problem that has to be addressed, and the possible ways of evaluation of the solution against the problems identified in the environment.
- the Rigor Cycle, which represents the selection and application of the existing methods, theories, frameworks or other artefacts in the construction and evaluation of the new solution.
- the Design Cycle – the core of design science research that involves creation of various alternative solutions and their evaluation against the requirements.

Following the three cycle principle and the corresponding DSR checklist [28], CaMeLOT (a **C**onceptu**a**l **M**od**e**lling **L**earning **O**utcomes **T**axonomy) was developed, as presented in Fig. 1.

Here we provide only a brief summary of the first iteration, as it was described in full in [7]. In the first iteration, we developed an initial set of criteria for classification based on the revised Bloom's taxonomy [11] – a means of learning outcome classification that involves two dimensions: cognitive dimension including six cognitive processes: Remember, Understand, Apply, Analyze, Evaluate and Create, and knowledge dimension including four types of knowledge: Factual, Conceptual, Procedural and Metacognitive knowledge (see Appendix A for the summary of the revised Bloom's taxonomy). Using the initial set of classification guidelines, we examined educational resources (course books [29–32], MOOCs (Massive Open Online Courses) [33–35] and university exams [36–40]) on conceptual data modelling to classify and assess the learning outcomes pursued in the modern educational practice as well as to validate the current version of the taxonomy. At the initial stage, the revised Bloom's taxonomy was implemented without changes to classify the learning outcomes. However, this first attempt to apply the taxonomy provided results that would be non-informative for educators, as the criteria of the taxonomy are too general to help teachers improve the courses with necessary level of precision. Thus, an additional dimension – content area – was introduced, so a more specific identification of gaps would be possible (this dimension is described

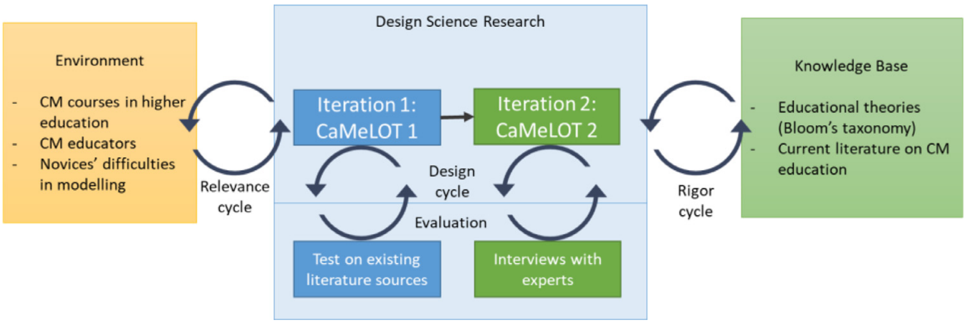


Fig. 1. Design science research methodology.

in Section 4.1). The study identified significant gaps in certain levels of the revised Bloom's taxonomy: metacognitive knowledge level was not represented, in neither of the educational resources; tasks related to procedural knowledge were heavily underrepresented. In the cognitive dimension, there was a significant lack of tasks related to evaluation – a very important high-level cognitive process. This research resulted in an initial version of CaMeLOT, consisting of a domain-specific reinterpretation of the Bloom's taxonomy and the addition of the content area dimension.

Given the gaps identified in Iteration 1, the framework was further refined by the creation of assessment items, which also led to the formulation of more crisp classification criteria. The outcome of the current iteration is an instantiation of the educational framework for conceptual data modelling education, which was performed by working out complete examples for a total of six content areas covering two subdomains of CDM. Afterward, this improved version of CaMeLOT was evaluated by three independent experts with more than 10 years of experience in conceptual modelling education (see Section 6).

4. CaMeLOT: a revised Bloom's taxonomy for conceptual data modelling

Next to an adapted definition of the knowledge and cognitive levels, CaMeLOT also proposes the identification of a set of distinct content areas and their scaffolding.

4.1. Scaffolding the learning process

In the first development cycle, we attempted to classify the assessment tasks directly into the Bloom's taxonomy. However, soon we faced the problem that certain tasks require prerequisite knowledge. This scaffolding of knowledge is not adequately captured by either the cognitive process levels or the knowledge levels. Therefore, based on the existing learning paths for domain modelling education, we decided to identify different content areas and created the following scaffolding tree (Fig. 2), according to which the modelling levels addressed by learning outcomes were determined.

The arrows should be read "A is a prerequisite for B", if the arrow starts in A and points at B. Double-headed arrows represent a co-requisite, while the dashed arrows represent a possible co-requisite. The bold lines and arrows represent the pre-requisites based not on one concept, but the whole content level (e.g. understanding every notion at the Class level is necessary to understand a Simple Model). Thus, Fig. 2 can be read as follows: understanding the concept of Object is necessary to understand the concept of Class; learning the concept of an Attribute is impossible without knowing the concept of Class; Complex Models require knowledge from all the levels above, etc.

To modify the revised Bloom's taxonomy for use in the field of conceptual data modelling and to ease the orientation in the learning material, we have added an additional dimension to the classification framework: the content areas. Each learning outcome is associated with one or several content areas (e.g. the topics or parts of the course). Each content area, in turn, has a list of terms and notions associated with it. Such terms and notions in application to learning outcomes can be viewed as keywords characterizing this learning outcome. Thus, each learning outcome can be categorized by the following parameters:

- Content area (which is defined by means of Related keywords)
- Knowledge level
- Cognitive level

Based on the analysis of samples of existing course material, and the way it is organized [7], we identified a set of general content areas in domain modelling (see Fig. 3 for the structure). It is important to note that this set of content areas is not exhaustive, and can be augmented by adding more content areas and corresponding keywords if necessary.

4.1.1. Model creation

The area of model creation represents the conceptual aspect of modelling and includes only notions and elements that are strictly related to developing a new model as such, not relating to the requirements collection process (this aspect is out of the scope of this framework and might be included in the future extension work) or to general knowledge about modelling (see further). It is further subdivided in the areas "Classes", "Relationships" and "Models".

4.1.1.1. Classes. The area of knowledge can be subdivided into subcategories of "pure" class-related concepts and inheritance-related concepts.

Related keywords (in alphabetical order): abstract class, attribute, class, class eliciting techniques, complex attribute, concept, constraint, disjoint set, domain, entity, generalization, generalization set, hierarchy, inheritance, instance, multiple inheritance, object, overlapping, overriding, single inheritance, specialization, subclass, substitutability, subtype, superclass, supertype.

4.1.1.2. Relationships. The Relationships area includes all kinds of associations between classes, including association classes.

Related keywords: aggregation, association, association class, binary association, multiplicity, consistency, life cycle, multiplicity, n-ary association, recursive association.

In addition, it is possible to expand the framework to a specific modelling method (e.g. MERODE [41], or Entity-Relationship Modelling [42]), thus, a collection of other related keywords will be added to the default ones (e.g. existence dependency, propagation, weak entity, weak relationship).

4.1.1.3. Models. The Models area involves model building, from simple to complex models, and the use of common analysis and design patterns.

Related keywords: analysis patterns, complex model, conceptual data model, design patterns, domain model, generic domain model, reference data model, simple model.

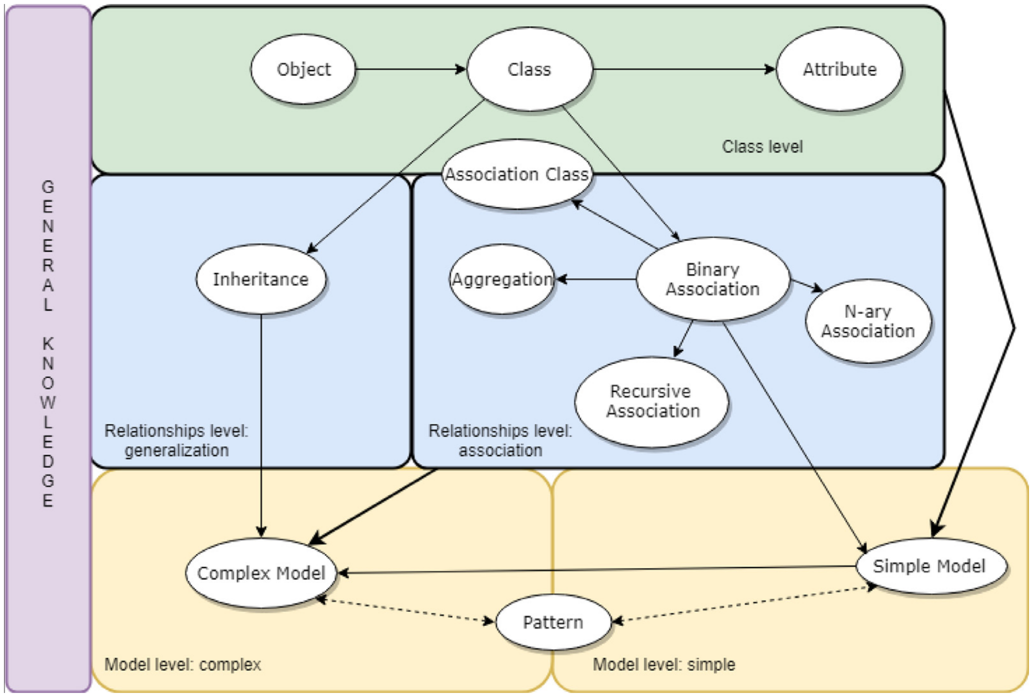


Fig. 2. Scaffolding tree for domain modelling education.

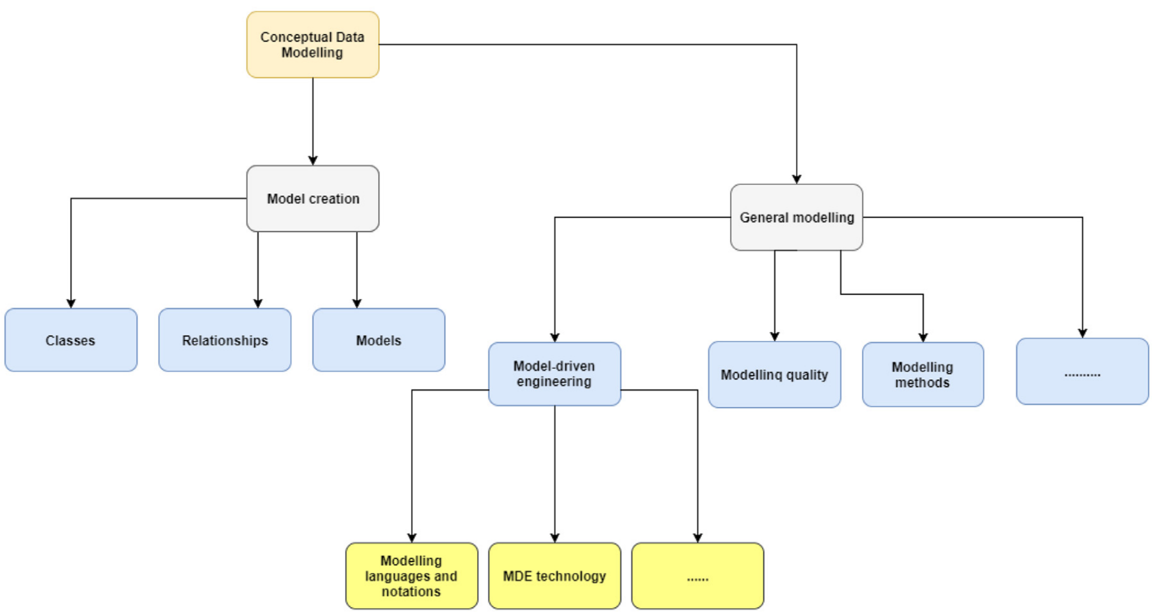


Fig. 3. Content areas tree.

4.1.2. General modelling

General knowledge involves a wide variety of topics related to modelling and models. As in the previous section, this content area can be divided into several subcategories, and further subcategorized if necessary. Here we provide just several illustrative examples, which can be further expanded according to the needs of educators.

4.1.2.1. Modelling quality. This important aspect of modelling includes all the aspects related to model quality, validity, quality assessment, consistency, etc. [43, 44].

Related keywords: conceptual modelling quality framework, pragmatic quality, semantic completeness, semantic quality, semantic validity, semiotics, SEQUAL, syntactic quality, etc.

4.1.2.2. Modelling methods. This subcategory refers to the process of modelling, and more in particular the development of modelling methods as "how to guidance" for modellers.

Related keywords: MERODE, method engineering, modelling guideline, modelling strategy, OO-Method, Open Method, Rational Unified Process, etc.

4.1.2.3. Model-driven engineering. This subcategory involves many different aspects relating other content areas such as modelling languages and notations, and modelling tools, but also containing its own specific set of concepts such as transformation languages, transformation technologies, and comprising both theory and "technical" materials, which are necessary for building professional and practical knowledge.

4.1.3. Modelling languages and notations

This subcategory refers to the many aspects relating to the use of notational languages to write down models. It will refer to specific languages as well as to the theories behind modelling languages, e.g. [45].

Related keywords: Crow's foot notation, Entity Relationship notation, MERODE notation, physics of notation, UML…

4.1.4. Model-driven engineering technology

This category refers to the specific concepts to put model-driven engineering at work.

Related keywords: meta-model, meta-object facility, Model-Driven Architecture, platform-independent model, platform-specific model, repository, transformation (M2C), transformation (M2M), etc.

These lists can be enlarged by the necessary notations, modelling methods and modelling tools used by educators.

4.1.4.1. Other. A separate set of keywords can be introduced for metacognitive learning outcomes to ease the future search and categorization:

Related keywords: portfolio, self-regulation, planning, learning strategy, motivation, knowledge implementation, experience, self-reflection, learning progress.

4.2. Revised Bloom's taxonomy: conceptual data modelling interpretation

4.2.1. Knowledge levels

4.2.1.1. Factual. On the factual level, the student deals with the basic elements of domain modelling, such as:

- Terms and their definitions according to the textbooks/lecture materials
- Modelling notation(s); knowledge of how a notation element corresponds to a modelling term
- General modelling conventions – how to name classes, attributes, associations, etc.
- Sources of information on domain modelling (such as textbooks, standards and thematic websites).

Due to the nature of the subject, purely factual knowledge-related learning outcomes are not as widely represented in the field of domain modelling and conceptual modelling in general, as in some other subjects, such as biology [20], medicine [23], etc. Despite the traditional allocation of terms' knowledge to the "factual" level, in domain modelling everything that goes beyond the basic elements listed above, can be considered conceptual or procedural knowledge. As an example, the definition of the concept of class according to a particular textbook belongs to factual knowledge, but the concept of class itself belongs to the conceptual level.

Depending on the content area, the basic elements will differ: they will be the terms, notation elements and patterns that are relevant for the given content area. For example, for Class level, the terms will not include any forms of associations, and the Associations level will not introduce any terms that were already introduced at Class level, as knowledge of Class level terminology and concepts is a pre-requisite for acquisition of Associations level.

4.2.1.2. Conceptual. Conceptual knowledge implies knowledge of the relationships among the notions learned on factual level, including classifications, categories, etc. Conceptual knowledge of domain modelling includes:

- Classification of information – e.g. identification of various types of information in the requirements document.
- Conceptual differences between various terms and types of information.
- Knowledge of fundamental principles of modelling
- Generalization of models/patterns.
- Commonly used modelling patterns and sample models – mostly part of the conceptual level. When patterns are introduced at a high level of abstraction, they may be considered factual knowledge.
- Relationships between different basic elements (e.g. class overlapping).

4.2.1.3. Procedural. Procedural knowledge in domain modelling is directly related to the core question: "How to build a model?" We refer here to the detailed level of creating a model, even very small. Modelling methods, which typically provide more coarse-grained advice on phases in a (software) development process, belong to separate content area.

Procedural knowledge may come in various shapes and forms. A student gets acquainted with the following (not exclusive) kinds of information:

- Step-by-step guidelines or algorithms for performing a modelling task at any stage (from eliciting the classes to building the UML model).
- Criteria for use of a modelling concept when solving a particular kind of modelling task – which concept to use in which case (e.g. ternary association or additional class ?) and why.
- Modelling quality evaluation procedures.

4.2.1.4. Metacognitive. Metacognitive knowledge implies the responsibility of a student for his/her own knowledge when learning a subject. A student may grasp the following kinds of knowledge on the metacognitive level:

- Strategic knowledge for learning the subject – organizational (planning), mnemonic (how to remember terms/notations/patterns), heuristics for ill-defined problem solving.
- Knowledge of cognition as such – e.g. which kinds of modelling exercises/tasks may improve the level of comprehension of the subject.
- Self-knowledge – knowledge of what learning strategies a student prefers, what content areas/types of tasks she/he is more proficient in, what motivation a student has to do a task or follow the subject.

4.2.2. Cognitive levels

4.2.2.1. Remember. This section is associated with memorizing and retrieving relevant knowledge about the basic elements a student must know to be introduced to the subject or be able to understand further notions that are more complex.

Typical kinds of tasks:

- Matching the terms with definitions/recognizing the terms.
- Giving definitions of terms.
- Drawing a graphical notation element corresponding to the given term.
- Duplicating a model, pattern, or notation element.
- Listing types of certain concepts/terms (e.g., all types of associations).

4.2.2.2. Understand. This section is related to understanding the basic concepts and terms of conceptual data modelling.

Typical kinds of tasks:

- Exemplification – giving examples representing learned modelling terms.
- Linking concepts with corresponding modelling notation elements.
- Explaining a given modelling concept with own words.
- Translation tasks – from notation to text.
- Summarizing tasks – e.g. summarizing the requirements document/a case in own words.
- Discussion tasks.

4.2.2.3. Apply. The “Apply” level is closely related to procedural knowledge, with rare cases of factual knowledge application, thus, the “Apply” level will not always be introduced in the “Factual” section.

“Apply” level both in Conceptual and Procedural knowledge implies usage of procedures or generally accepted methods to solve problems. In case of domain modelling, such methods may include class elicitation techniques, naming conventions, generalization techniques and usage of patterns. Given that “Apply” in the conceptual level is almost indistinguishable from the procedural level, the “Apply” level will not always be introduced in the “Conceptual” section either.

Typical kinds of tasks:

- Use of a given pattern to solve a modelling task.
- Application of previously learnt heuristics or guidelines in practice.
- Translation of a model from one notation to another according to the guidelines.
- Demonstration of use of a particular method.

4.2.2.4. Analyze. This section represents outcomes related to differentiation, organizing and attributing processes applied to the basic elements of domain modelling. This level can be considered an extension of a corresponding “Understand” level.

Typical kinds of tasks:

- Ordering and comparison tasks.
- Determining completeness or incompleteness of a statement/element/attribute.
- Distinguishing relevant information from irrelevant (e.g. in the requirements document).
- Explaining the modelling choice (e.g. modelling something as a class or as an attribute).
- Determining which modelling pattern is used in a given model.
- Mapping a given analysis pattern to a given problem.

4.2.2.5. Evaluate. In this section, a student should make judgements regarding the usage of basic terms/modelling elements.

Typical kinds of tasks:

- Checking the correctness/finding mistakes in used notation.
- Identifying conventions/rules violations.
- Critiquing a solution/the choice of example.
- Identifying the most suited analysis & design pattern for a modelling problem.

4.2.2.6. Create. This section is associated with creative tasks involving the use of basic terminology and concepts.

Typical kinds of tasks:

- Developing a solution for a given simple modelling task.
- Generating a class/association/group of classes for given requirements.
- Designing a model element according to the given specifications.
- Defining a modelling pattern based on a series of problems and solutions.

5. Example sets of learning outcomes

To validate the classification framework as a practical tool to be used by educators, we have developed example sets of learning outcomes for six content areas covering two subdomains of conceptual data modelling. For each content area, one or two tables with a line of example learning outcomes per knowledge level is given. Not every content area would have all the knowledge levels presented and certain intersections between content areas are expected in many cases. The exception is Metacognitive knowledge: this type of knowledge is universal for the entire course/field and “wraps around” all the content areas; it thus does not have to be subcategorized.

5.1. Class

5.1.1. Factual knowledge

Factual knowledge about classes is related to the definitions given by the course author or the author of the textbook. In the particular case given in Table 1, not every cognitive dimension is presented, due to the conceptual nature of knowledge related to classes. This set of learning outcomes is designed around the definitions from particular textbooks, and the knowledge is built gradually from “Remember” level up to the “Analyse” level.

It is important to note that a learning outcome formulated as “give a definition of “class” in your own words” will fall into the Conceptual knowledge level rather than factual, as the very concept of “class” has to be understood by the student prior to formulating the definition. For the same reason, in this specific set, learning outcomes from higher cognitive dimensions such as “Evaluate” and “Create” fall into Conceptual knowledge level rather than Factual.

5.1.2. Conceptual knowledge

The following tables present learning outcomes related to the differences between classes, objects and attributes (Table 2), and generalization (Table 3).

In Table 2, the student is supposed to start with recalling the differences between classes and attributes, then explain both notions with a concrete example. Further, she applies her knowledge deciding whether to model a specific item as a class or as an attribute. In the higher level cognitive processes, all the previous knowledge is involved: the student has to analyse the requirements document, evaluate a given list of classes and to create a list of classes and attributes based on the requirements document.

In Table 3, a similar process of grasping the relationships between super- and subclasses in the context of generalization is presented.

5.1.3. Procedural knowledge

Table 4 gives the learning outcomes related to the procedures of eliciting classes. It is important to note that in the “apply” level the emphasis is on applying guidelines. Eliciting classes from a given requirements document was given as an example of “Conceptual-Create” for the content area “Class”.

5.2. Relationships

5.2.1. Factual knowledge

In this case, factual knowledge table from the Class area can be adapted – featuring the definitions of various relationship types from literature.

5.2.2. Conceptual knowledge

Table 5 presents learning outcomes related to the notion of association multiplicity. The first cognitive level may be skipped here, as here it will refer to factual knowledge about multiplicities, such as the related notation.

5.2.3. Procedural knowledge

Table 6 presents the learning outcomes related to the procedures and rules related to the usage of association classes, including the transformation rules and the differences between usage of associations and association classes in certain cases.

5.3. Models

5.3.1. Factual knowledge

Table 7 presents the learning outcomes related to modelling patterns viewed as the basic elements of conceptual models (as opposed to the “class” content area where classes were viewed as the basic elements). On the “apply” cognitive level, there is no identified outcome as the

Table 1
Example learning outcomes related to the theoretical background (Class – Factual).

Keywords: Class, definition, object Cognitive dimension:	The student should be able to:
Remember	Recall a definition of a class according to the provided textbook/author
Understand	Summarize the general characteristics of a class, according to the definition provided in a textbook.
Apply	– (conceptual level)
Analyse	Compare the definitions of a class given by Author A and Author B
Evaluate	– (conceptual level)
Create	– (conceptual level)

Table 2
Example learning outcomes related to classes, objects and attributes (Class – Conceptual).

Keywords: Attribute, class, object, requirements document Cognitive dimension:	The student should be able to:
Remember	State the conditions for when to model something as an attribute or a class.
Understand	Give an example of something that should be modelled as a class instead of an attribute.
Apply	Based on a set of precise requirements, decide whether something should be added as an attribute or a class.
Analyse	Highlight the sections of the requirements document containing information about classes, attributes and objects.
Evaluate	Decide which classes of the given list are excessive (according to the requirements document).
Create	Based on a given requirements document, create a draft list of classes and attributes.

Table 3
Example learning outcomes related to inheritance (Class – Conceptual).

Keywords: Class, subclass, superclass, overlapping, inheritance, generalization set, disjoint set Cognitive dimension:	The student should be able to:
Remember	State the rules of what a subclass inherits from a superclass, and what modifications are allowed by the subclass.
Understand	Give an example of subclass overlapping.
Apply	Based on a set of precise requirements, decide whether something should be added as a subclass or a separate class.
Analyse	In a given inheritance diagram, identify whether a generalisation set is overlapping or disjoint.
Evaluate	Given requirements and an (incorrect) inheritance tree, find a mistake in the interpretation of requirements.
Create	Given requirements, create a diagram representing the inheritance of classes according to the given requirements.

Table 4
Example learning outcomes related to classes eliciting techniques (Class – Procedural).

Keywords: Requirements document, modelling guidelines, class eliciting techniques, modelling strategy Cognitive dimension:	The student should be able to:
Remember	List the stages of eliciting classes from a requirements document.
Understand	Discuss a class eliciting technique.
Apply	Given a requirements document, elicit classes according to the given guidelines.
Analyse	Check the stages that were used by his/her peer in eliciting classes and determine which class eliciting strategy was applied.
Evaluate	Find a mistake in a class eliciting strategy and justify his/her answer.
Create	Write down his/her own class elicitation technique.

Table 5
Example learning outcomes related to multiplicity (Relationships – Conceptual).

Keywords: Association, multiplicity, requirements document Cognitive dimension:	The student should be able to:
Remember	— (factual knowledge)
Understand	Given a diagram with an association, explain the multiplicity of the association ends by drawing a representative instance-level diagram.
Apply	Based on a representative instance-level diagram, decide upon the correct multiplicity of an association end.
Analyse	Given a textual description of an association, analyse what multiplicities are implied by the description.
Evaluate	Given a requirements document and a partially incorrect class diagram, find a mistake in the given multiplicities.
Create	Given a requirements document and a list of classes, come up with the list of associations and their multiplicities.

Table 6
Example learning outcomes related to the association classes (Relationships – Procedural).

Keywords: Association, association class, multiplicity Cognitive dimension:	The student should be able to:
Remember	Recall the rules of transforming an association class to a class and associations.
Understand	Explain the logic behind the transformation rules.
Apply	Translate an association class into a class and associations with corresponding multiplicities using the transformation rules.
Analyse	Compare two solutions using transformation rules and derive the stages of transformation.
Evaluate	Check whether his/her peer applied the transformation rules correctly for a given model.
Create	Create a decision tree that helps a modeller to decide when to use an association class versus an association.

Table 7
Example learning outcomes related to modelling patterns (Models – Factual).

Keywords: Modelling patterns, simple models	
Cognitive dimension:	The student should be able to:
Remember	List the different forms for describing patterns.
Understand	Find an example of usage of a modelling pattern from a given list of models.
Apply	— (Conceptual/Procedural level)
Analyse	Compare the descriptions of two similar patterns to identify similarities and differences.
Evaluate	Evaluate the completeness of a pattern description, compared to the Coplien Form [46].
Create	Complete the missing sections in the description of a modelling pattern.

Table 8
Example learning outcomes related to modelling patterns (Models – Conceptual).

Keywords: Modelling patterns, simple models, complex models	
Cognitive dimension:	The student should be able to:
Remember	Given the solution part of a particular pattern, recall the typical problem that is addressed by this pattern.
Understand	Given a modelling pattern, create your own example of instantiation of the pattern.
Apply	Given a modelling pattern and a requirements document, apply the pattern to create a model.
Analyse	Compare two modelling patterns for a given problem and find the main conceptual differences in how they address the given problem.
Evaluate	Decide, which modelling pattern is better to be implemented in modelling, according to the given requirements.
Create	Given three similar modelling problems and their solution, create a modelling pattern that represents the generic solution for the re-occurring problem.

Table 9
Example learning outcomes related to building a model (Models – Procedural).

Keywords: Modelling guidelines, simple model, complex model, requirements	
Cognitive dimension:	The student should be able to:
Remember	Recall the stages of building a model
Understand	Discuss the stages of model building and prove the necessity of each stage
Apply	Perform the first three (n) stages of building a model, provided detailed requirements
Analyse	Examine a solved full case and elicit the stages of modelling implemented during the solution
Evaluate	Check, whether his/her peer followed the modelling guidelines when modelling to evaluate the clarity of the guidelines
Create	Improve the modelling guidelines based on your experience

Table 10
Learning outcomes related to knowledge of notation (General modelling – Factual).

Keywords: Notation, requirements, requirements document, UML	
Cognitive dimension:	The student should be able to:
Remember	Match the given terms with their modelling notation symbols.
Understand	Explain the meaning of a given symbol.
Apply	Transform a model from one notation to another (requires knowledge about two notations).
Analyse	Compare two modelling notations ((E)ER and UML) by indicating the symbols that have an identical meaning or not.
Evaluate	Decide which notation has the strongest expressive power.
Create	Develop a domain specific notation for a specific type of modelling problems.

application of a pattern requires deep understanding of relationships between different elements in a given context, and subsequently falls under conceptual or procedural knowledge levels.

5.3.2. Conceptual knowledge

Building “on top” of the factual knowledge about modelling patterns (Table 7), Table 8 represents learning outcomes related to the conceptual knowledge about modelling patterns.

5.3.3. Procedural knowledge

Table 9 presents learning outcomes related to the procedures of building a model applying provided modelling guidelines. This table can be viewed as the most typical example of procedural learning outcomes on each cognitive level.

5.4. General modelling: model-driven engineering

5.4.1. Factual knowledge

Table 10 presents learning outcomes related to the use of modelling languages and notations.

5.4.2. Conceptual knowledge

Table 11 provides examples relating to the conceptual understanding of meta-modelling.

5.4.3. Procedural knowledge

The table below gives only specific examples about model-to-code transformation procedures. Obviously, there are many factual and conceptual learning outcomes related to model transformation (Table 12).

5.5. General modelling: modelling quality

5.5.1. Factual knowledge

Table 13 presents learning outcomes related to modelling quality. Given the fact that syntactic errors do not require model understanding (they are comparable to spelling errors), learning goals that target syntactic quality are positioned in the factual knowledge layer. In contrast, other types of errors and quality (e.g. semantic quality) are of a conceptual nature and are therefore positioned in the conceptual knowledge layer.

Table 11
Example learning outcomes related to meta-modelling (MDE – Conceptual).

Keywords: Meta-model, OMG, meta-model hierarchy	
Cognitive dimension:	The student should be able to:
Remember	List the model layers of an OMG meta-model 4-layer hierarchy.
Understand	Illustrate the layers of OMG meta-model hierarchy.
Apply	Given a list of concepts about models and meta-models, place them in the right layer of the 4-layer hierarchy.
Analyse	Compare two proposed meta-models for the same modelling language.
Evaluate	Given a modelling language and a meta-model, evaluate whether the meta-model correctly and completely defines the modelling language.
Create	Given a modelling language, create a meta-model for that language.

Table 12
Example learning outcomes related to model transformation (MDE – Procedural).

Keywords: Model transformation, MDE, prototype	
Cognitive dimension:	The student should be able to:
Remember	Recall the steps required for transforming a model to a prototype (code).
Understand	Describe the stages of model transformation.
Apply	Given a transformation tool, transform a given conceptual model into code.
Analyse	Compare two model-to-code transformation procedures.
Evaluate	Evaluate the ease of use of a model-to-code transformation procedure.
Create	Create a step by step plan to transform a model into code.

Table 13
Example learning outcomes related to modelling errors (Modelling quality – Factual).

Keywords: Semantic quality, syntactic quality, pragmatic quality, modelling errors	
Cognitive dimension:	The student should be able to:
Remember	Recall the types of possible syntactic errors
Understand	Give an example of a syntactic error in a UML model
Apply	–
Analyse	Determine whether a described error is syntactic or not.
Evaluate	Determine whether a given UML model contains a syntactic error
Create	–

Table 14
Example learning outcomes related to quality types (Modelling quality – Conceptual).

Keywords: Semantic quality, syntactic quality, pragmatic quality, modelling errors, completeness, validity, UML	
Cognitive dimension:	The student should be able to:
Remember	Give a definition of syntactic, semantic and pragmatic quality
Understand	Explain the interrelations between syntactic, semantic and pragmatic quality
Apply	Given an error in a model, determine the type of error.
Analyse	Compare two given models with regards to their pragmatic quality
Evaluate	Evaluate a model in terms of its completeness and validity, according to the given requirements
Create	Modify a given UML class diagram to correct syntactic and semantic errors it contains

Table 15
Example learning outcomes related to quality assessment procedures (Modelling quality – Procedural).

Keywords: Quality assessment	
Cognitive dimension:	The student should be able to:
Remember	Recall all the points of the quality assessment checklist
Understand	Discuss the strategies for modelling quality assessment
Apply	Assess the quality of a model using the quality assessment checklist
Analyse	Compare two quality assessment strategies
Evaluate	Determine which quality assessment strategy would serve better in a given case
Create	Improve a given quality assessment checklist with your own bullet points

5.5.2. Conceptual knowledge

In contrast to Factual knowledge presented in Table 13, which concerned only syntactic quality of the model and possible syntactic errors, Table 14 presents the learning outcomes related to the next level of knowledge – knowledge about various types of quality and their interrelations.

5.5.3. Procedural knowledge

Table 15 presents learning outcomes related to quality assessment strategies and guidelines. It is important to note that the learning outcome under "Apply" could be also positioned as "Conceptual-Evaluate", as it comprises both conceptual and procedural knowledge.

5.6. Metacognitive knowledge

Metacognitive knowledge is a separate category that "wraps around" all the content areas, and can be adopted for each of them in the course of teaching. Table 16 presents learning outcomes related to motivation, while the learning outcomes of Table 17 are related to the learning strategies a student can utilize to succeed in the course.

Table 16
Example learning outcomes related to motivation (Metacognitive).

Keywords: Motivation, knowledge implementation, experience, self-reflection	
Cognitive dimension:	The student should be able to:
Remember	Recall his/her favourite tasks from the course so far
Understand	Discuss how domain modelling may be implemented in his/her future professional life
Apply	—
Analyse	Rate the parts of the course according to how interesting they are for the student
Evaluate	Evaluate his/her overall experience from today's class
Create	Describe some possible improvements of the course.

Table 17
Example learning outcomes related to learning strategies (Metacognitive).

Keywords: Learning strategy, learning progress, self-reflection	
Cognitive dimension:	The student should be able to:
Remember	State your teacher's advice for learning conceptual modelling
Understand	Discuss the learning strategy for domain modelling.
Apply	Implement the learning strategy for domain modelling education for a week of studies.
Analyse	Compare your learning progress with your peer's and list the experienced difficulties.
Evaluate	Reflect on your progress by going through your learning achievements.
Create	Describe your conceptual modelling skills in a self-reflective report.

Table 18
Expert interviews attempts summary.

Interviewee	Status	Length	Recording
Expert 1 (Belgium)	Conducted in person	50 min	Audio recording + concurrent notes
Expert 2 (USA)	Conducted by phone	45 min	Audio recording + concurrent notes
Expert 3 (Germany)	Conducted by phone	45 min	Audio recording + concurrent notes
Expert 4 (Belgium)	Excluded as reported that changed the pedagogical focus	–	–
Expert 5 (Russia)	No final response	–	–
Expert 6 (Spain)	No response	–	–
Expert 7 (Belgium)	No response	–	–

6. Evaluation

To perform the preliminary evaluation of the current version of the framework, seven experts in the field of conceptual modelling education were contacted and asked to participate in the framework evaluation in a form of an interview that followed the presentation of the framework and the experts' familiarization with the guidelines. Three experts were found willing to participate in an in-depth interview. The summary of the interview gathering process is presented in Table 18.

As a result of the requests and negotiations, three semi-structured individual interviews were conducted.

6.1. Participants

The criteria for inviting the experts to participate in the interviews were the following:

- More than 10 years of experience in teaching conceptual data modelling and/or other conceptual modelling disciplines at an institution of higher education, for an undergraduate and graduate engineering education program;
- Being fully employed as a professor at an accredited university;
- Being involved in research activities in the field of conceptual modelling;
- Having articles published in peer-reviewed academic journals in the fields of computer science and/or enterprise modelling and/or business engineering.

The three experts who answered the invitation to participate in the interviews, are employed as full professors at the universities from three different countries: Belgium, USA, and Germany. The summary of their teaching experience and research interests is presented in Table 19.

6.2. Interview methodology and structure

The interview questions were designed based on the Technology Acceptance Model as designed by Davis et al. [47]. The model implies that faced with a new technology or method, the potential users will be influenced by their perceptions of the ease-of-use of the technology, as well of its usefulness. These perceptions, in their turn, result in the attitude toward using and, finally, the intention to use the technology.

In case of this particular framework, though it cannot be considered a technology, it can be viewed as an instrument or a method. According to [48], "theoretical models used to explain and predict user acceptance of information technology may be adapted to explain and predict the adoption of methods". Thus, the interview questions were designed to get insights on the three key aspects of TAM: perceived ease of use, perceived usefulness and intention of use.

The interviews were structured as follows:

1) The interviewees were provided with the textual description of the framework before the interview.
2) Before the start of the interview, the framework was presented to the experts, including the explanation of the dimensions and examples of the learning outcomes generated using the framework.
3) The first part of the interview (~15 min) included questions related to the educational experience and the research interests of the experts.
4) The rest of the interview was dedicated to the closed and open questions related to the framework, and were grouped according to their topic:

Perceived Ease of Use:

- *How simple/difficult do you find the procedure for applying the framework?*
- *What is your opinion on how easy/difficult it is to learn the guidelines?*

Table 19
Experts' background.

Experts	Pedagogical experience	Research topics	Taught courses	Students' specialization
Expert 1 (Belgium)	16 years	CM quality, CM patterns, Enterprise Architecture Models	ER modelling (+ UML as notation), Business Process Modelling, Goal Modelling, Value modelling, Business Motivation Modelling	Applied Economics Business Engineering
Expert 2 (USA)	28 years	Database support, query languages and optimization, data warehouses, object-oriented data modelling, computer science education	Database systems	Software Engineering, Computer Science
Expert 3 (Germany)	13 years	Enterprise modelling, domain-specific modelling languages, business process modelling, business informatics	Information systems modelling, development of business information systems	Business Information Systems

- *Overall, how understandable is the framework for you as a teacher?*

Perceived Usefulness

- *Do you think the use of the framework may increase the variety (or improve the gradual scaffolding) of exercises you would develop for conceptual modelling courses?*
- *Do you think the use of the framework may reduce the effort necessary to develop exercises for conceptual modelling courses?*
- *What do you think about the speed of curriculum and individual task design using such framework?*
- *How useful do you find this framework overall?*

Intention of Use

- *Would you use the framework if you needed to design a conceptual modelling course from scratch?*
- *Would you adapt or customize the framework or its elements for your own pedagogical activities?*

5) After the interview, the experts were provided with a summary of their interviews and given the opportunity to review and finalize their answers.

6.3. Evaluation results

6.3.1. Expert 1

6.3.1.1. Perceived ease-of-use of CaMeLOT. The expert expressed the opinion that the framework is well-structured and may be helpful in designing a new course on conceptual modelling. It may make it easier to create new types of questions. Though it will not make it easier to create new questions as such, it will make the process more structured and systematic.

The framework is quite understandable; however, the interviewee was already familiar with Bloom's taxonomy.

It may be time consuming for the teachers to familiarize themselves with the framework for the first time; there is a danger that the teachers will look immediately to the tables without building up a good understanding of the framework, thus they might misinterpret certain guidelines. Overall, the ease-of-use can be considered moderate.

6.3.1.2. Perceived usefulness of CaMeLOT. The expert expressed the opinion that CaMeLOT can be useful to set an ambition level for the conceptual modelling course: as it is not necessary that every course should be aiming at the highest cognitive levels (such as creation), the framework could help setting the right ambition, so that the students will not be overloaded with unnecessary information. Thus, it can help distinguish between a basic version and an advanced version of a course. The examples provided were also perceived as useful and making the framework more "tangible". The framework could also be useful for evaluating the existing course in case of poor results or poor student evaluation in the end of the year. Once the initial investment is made, the expert states that the framework can be applied to many different courses. Overall, the framework is found useful by the interviewee.

6.3.1.3. Intention of use. The interviewee expressed the intention of using the framework to create new exercises for the currently taught courses.

6.3.1.4. Suggestions. The expert suggested development of additional prescriptive guidelines for different pedagogical purposes, such as evaluating the existing course, creating a new course, etc., as a possible direction of future improvement of the framework.

6.3.2. Expert 2

6.3.2.1. Perceived ease-of-use of CaMeLOT. Though it may be slightly time-consuming to learn the guidelines, for an educator the procedure for applying the framework is expected to be simple once the guidelines are learnt. The framework is understandable for teachers and, except for the initial familiarization with the Bloom's taxonomy principles and the framework guidelines, should be easy to use in the day-to-day activities, both for creating new material and evaluating the possible gaps in the current curriculum. Overall, the ease-of-use can be considered moderate.

6.3.2.2. Perceived usefulness of CaMeLOT. The use of the framework may increase the variety of exercises and activities in the currently taught conceptual modelling courses. The framework may reduce the effort necessary to develop exercises for new educators – though for experienced educators it will not change the time spent on curriculum and individual task design, however, it might increase the quality of those and improve the pedagogical process. Overall, the interviewee anticipates that the framework will be very useful for course design activities.

6.3.2.3. Intention of use. The expert believes that the framework could definitely be used to design a conceptual modelling course from scratch, if necessary. The interviewee expressed a strong intent to implement the framework to her own pedagogical activities starting from this academic year, as well as to evaluate the effectiveness of such implementation.

6.3.2.4. Suggestions. The expert suggested providing additional examples of learning outcomes, and possible expansion of the framework to other areas of conceptual modelling.

6.3.3. Expert 3

6.3.3.1. Perceived ease-of-use of the framework. The expert believes that the framework would be very hard to understand and use by educators, as they would have to spend substantial amount of time to understand the framework; in addition, due to the possible overlaps in categories,

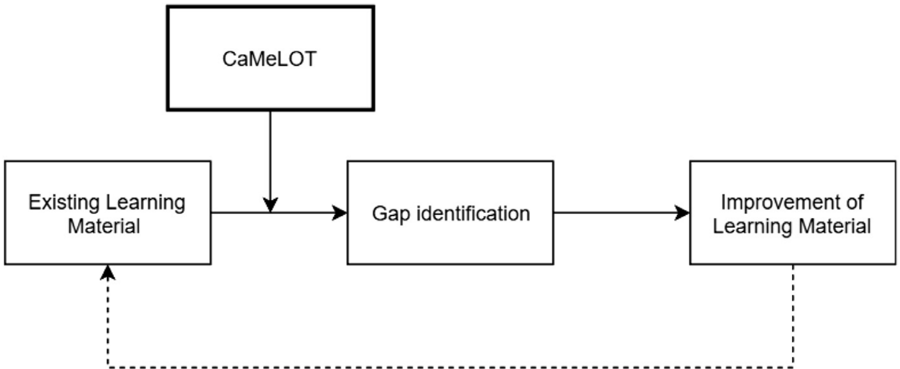


Fig. 4. Course of action for assessment of an existing course using CaMeLOT.

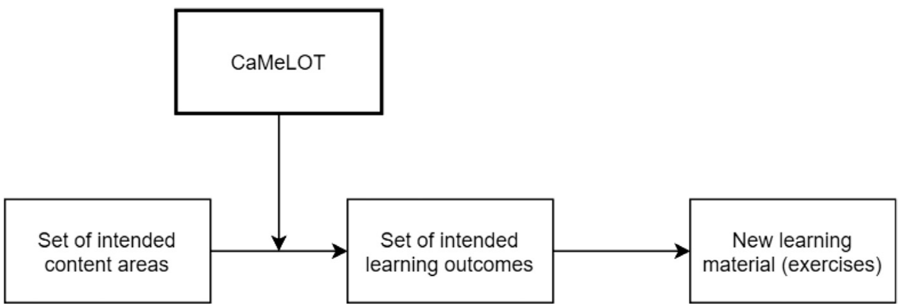


Fig. 5. Course of action for creating a new course using CaMeLOT.

the framework may confuse the teachers trying to classify the existing materials, or creating the new ones – thus the ease of use can be considered low.

6.3.3.2. Perceived usefulness of the framework. The framework can be viewed only as a starting point for creation of a more pragmatic framework. At the current stage the usefulness is perceived as low.

6.3.3.3. Intention of use. The intention of use is low, due to the concerns mentioned above.

6.3.3.4. Suggestions. The expert suggests to further specify the framework to avoid the overlaps, possibly by applying a different set of dimensions.

6.4. Evaluation conclusion

Two of the interviewed experts perceived the framework as a useful potential means to improve their courses and ease the pedagogical activity. At the same time, they rate the ease of use as moderate given that it could be difficult and time-consuming to learn the guidelines. This is confirmed by the third expert, who found the framework hard to use and understand by educators. In line with the TAM, the ease of use affects the perceived usefulness and intention to use. The two experts who were at ease with the framework, evaluated it as useful and had a high intention to use. However, for the third expert, the difficulty in use negatively affected the perceived utility, resulting in a lack of intention to use.

7. Discussion

The set of tables presented in Section 5 provides examples and guidelines to conceptual data modelling educators wishing to enrich their educational practice with new types of exercises and assessment items or to identify the gaps in the existing curriculum.

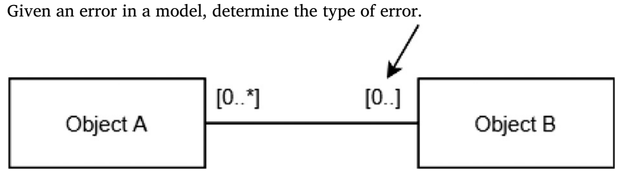
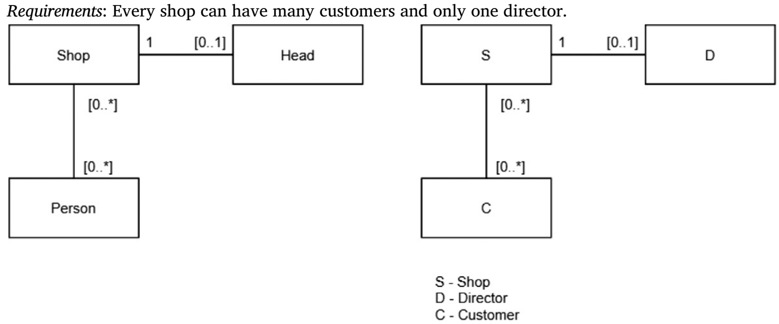
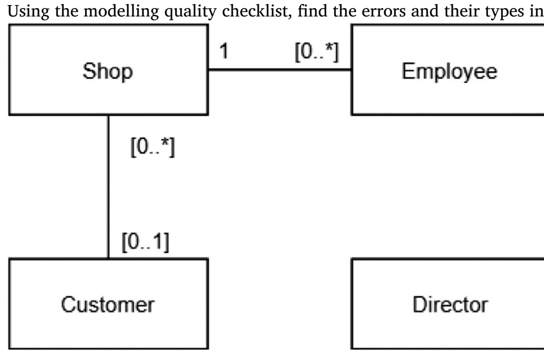
7.1. Limitations

While the CaMeLOT provides an opportunity for arranging the learning outcomes in a structured and coherent way, several limitations should be considered by those who wish to implement it. The first concern is certain ambiguity of classification in some cases – one rater may perceive a learning outcome as conceptual, while the other may argue it is factual based on a slight difference in the formulation. The same concern was expressed by Expert 3 at the evaluation stage. Such discussions, however, stimulate a careful analysis of the envisaged learning outcome and a more precise formulation of corresponding assessment questions. The second concern, which was already noticed by the authors of [24], is the concept shifting that may occur when generating new learning outcomes: the content areas are rather broad, and certain concepts or terms may be switched to similar ones. For example, the factual knowledge of classes and associations, requires knowledge of the notation to write down models. At the same time this kind of knowledge also belongs to the general domain of “Modelling languages and notations”. These two problems could be solved in the future by introducing stricter criteria of classification. On the other hand, in some cases the ambiguity of the framework may be viewed as its flexibility. From this point of view, further restriction of the classification criteria might not be necessary, and the ambiguities may be resolved by introduction of a second (or third) opinion.

7.2. Suggestions for use

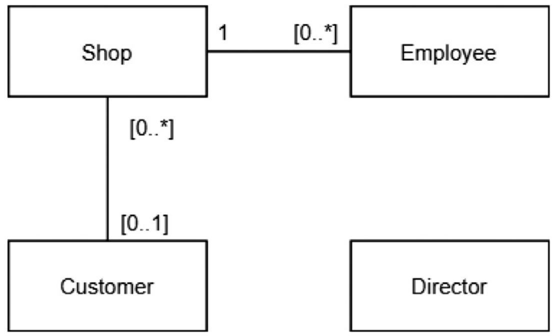
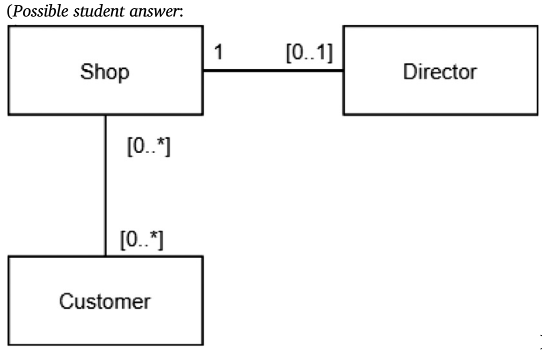
As current educational literature on conceptual data modelling lacks tasks related to procedural knowledge [7], the framework may help educators designing more tasks at this knowledge level and further increase the efficiency of their courses, providing the novice modellers with the means to adapt their modelling strategies and get closer to the state of “modelling experts”. According to the opinions of two of the experts, the framework may make the process of curriculum development more structured and provide valuable means both for assessment of an existing course’s learning objectives and the creation of new learning mate-

Table 20
Modelling quality exercises examples.

Keywords: Semantic quality, syntactic quality, pragmatic quality, modelling errors, completeness, validity Cognitive dimension:	Exercise:
Remember	List the types of modelling quality according to the Lindland, Sindre and Solbverg's framework. (*Answer*: Syntactic, Semantic, Pragmatic).
Understand	Explain the interrelations between syntactic, semantic and pragmatic quality of a model using the notions of "language", "domain" and "actor". (*Possible student answer*: Syntactic quality of the model ensures that it corresponds to the conventions of the given modelling language. A model can be syntactically correct, while being incorrect semantically. Semantic quality refers to the extent to which a model reflects the real world/domain situation. However, a model can be semantically correct, while being incorrect pragmatically, in case if the actors that read the model cannot understand it.)
Apply	Given an error in a model, determine the type of error.  *(Answer*: Syntactic error)
Analyse	Compare two given models with regards to their pragmatic quality *Requirements*: Every shop can have many customers and only one director.  (*Possible student answer*: Both of these models could be improved in terms of pragmatic quality. The left model is easily readable, however, the names of the classes do not allow to easily link them to the requirements. On the other hand, the model to the right has appropriately named classes, however, those names are not easily readable.)
Evaluate	*Requirements*: Every shop can have many customers and only one director. Using the modelling quality checklist, find the errors and their types in the given model.  (*Possible student answer:* There is one syntactic error in the model: class Director is not connected to any other classes. There are also several semantic errors: according to the requirements, every shop can have many customers, however, the multiplicity in the given model allows for maximum one customer per shop. Class "Employee" is excessive as it was not explicitly mentioned in the requirements).

(*continued on next page*)

Table 20 (*continued*)

Keywords: Semantic quality, syntactic quality, pragmatic quality, modelling errors, completeness, validity	
Cognitive dimension:	Exercise:
Create	Modify the given UML class diagram to correct syntactic and semantic errors it contains. *Requirements*: Every shop can have many customers and only one director.  (*Possible student answer*: )

rials and question types to ensure the proper scaffolding in the learning process.

Another aspect of utility of the framework is a possibility for step-by-step assessment of knowledge of novice modellers: the gradual assessment of knowledge within and across the scaffolded content areas will allow teachers to discover the points where their students experience most difficulties with much more precision. For example, when a student experiences problems for creating simple models, the scaffolding of content areas suggests to first verify whether the students masters the create level of the "Class" and "Binary Associations" content areas. Finally, the framework may stimulate creativity and ease the development of examples and exercises for any content area.

To facilitate the adoption of the framework by conceptual modelling educators, we propose the three possible uses of CaMeLOT:

1) *Assessment and improvement of an existing course*. The course of action for this use is presented in Fig. 4. Educators who wish to assess their conceptual modelling course can apply CaMeLOT to existing learning material to identify which cognitive and knowledge levels are not yet addressed, as well as to check the content areas that lack attention, and, consequently, can improve the learning material by filling the gaps. Such assessment may allow to reflect on the learning goals set for the course and to find whether the intended learning outcomes are constructively aligned with the learning material. It also allows for continuous assessment of learning material.
2) *Creation of new exercises and exams for an existing course in a systematic way*. If new exercises are required for an existing course, or the teacher has to design exam tasks, CaMeLOT can be used as a basis for creation of such exercises. The examples in the tables may serve as a source of inspiration, however, educators are free to invent various types of exercises tailored to the needs of their students.

Example:

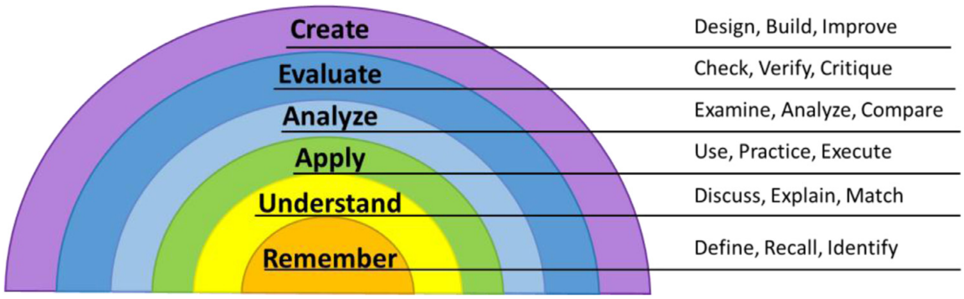
A set of additional practice exercises is needed for a module of a conceptual data modelling online course. In particular, at the end of the module students should be familiar with the grounds of modelling quality and the main quality types – in order to continue to more advanced subsequent modules of the course. Based on the given goal and the prerequisites to the module (the knowledge that students already possess), a set of learning sub-outcomes can be developed using CaMeLOT framework. Table 14 provides an example of such sub-outcomes related to the class eliciting techniques, and a set of exercises corresponding to those sub-outcomes is given below in Table 20.

1) *Creation of a modelling course from scratch.* This option might be of particular help for novice educators in the field of conceptual data modelling. The possible course of action is presented in Fig. 5. After determining a set of content areas that should be addressed in the given course, the educator can use CaMeLOT for determining the learning outcomes of the course, and, using the set of examples in the tables for inspiration, create new learning material based on the intended learning outcomes.

It is important to note that not every course will require all cognitive or knowledge levels, thus it is the teacher's duty to decide which learning outcomes are the most relevant for the students of a particular course.

8. Conclusion

In this study, we have developed CaMeLOT as a revised Bloom's taxonomy adapted for conceptual data modelling. Besides a reinterpretation of the knowledge and cognitive levels, the framework also identifies and scaffolds content areas in the domain of conceptual modelling. While the content area of Model Creation was elaborated specifically



for data modelling, it can easily be extended for other types of modelling (e.g. business process modelling). The content area of General Modelling, as its name suggests, is already encompassing different types of conceptual modelling. Additionally, a set of examples of learning outcomes related to conceptual data modelling is proposed to explain the use of CaMeLOT. The classification guidelines provided in the description of CaMeLOT's knowledge and cognitive levels (Section 4) and further exemplified in Section 5 are applicable not only for evaluation of the current curriculum, but also for generating new learning items related to conceptual data modelling. The obtained results give educators an opportunity to enhance the conceptual (data) modelling part of software engineering curricula with a systemized set of learning outcomes to be pursued, and open the path for creating useful and effective assessment packages. In addition, the framework can be used for continuous evaluation of the learning materials, including textbooks and supplementary online courses, which are more and more often introduced in the context of higher education. According to the initial evaluation (Section 6), the adoption of our educational framework may reduce the time spent on designing educational material and, at the same time, improve its quality.

Future work will include the further refinement and clarification of CaMeLOT to increase its ease of use and address the shortcomings of the revised Bloom's taxonomy by applying other educational frameworks, e.g. 4C/ID model [49]. Furthermore, the effectiveness of the framework for teaching could be formally demonstrated in the future in the higher education context and its effectiveness for improvement of students results should be assessed. The scaffolding process, as well as the relationships between different learning outcomes as prerequisites to one another can be further investigated. Finally, the possible expansion of CaMeLOT to the areas of modelling other than data modelling, such as business process modelling should be considered.

Acknowledgment

This research is funded by grant C24/16/002 of the KU Leuven Research Council.

Appendix A

The revised Bloom's taxonomy[1] structure can be viewed as a 6 × 4 matrix representing Cognitive and Knowledge domains of learning.

The Cognitive levels of the taxonomy are defined by the action verbs that characterize the learning tasks associated with them. The following picture presents some of the action verbs and how the levels are positioned related to each other.

The Knowledge levels of the taxonomy are characterized by the types of knowledge the learning outcome or task aims at.

Factual knowledge is the basics of the studied disciplines, such as basic terminology and/or notation.

Conceptual knowledge implies understanding of the connections and interrelationships between the basic elements learned on factual level.

Procedural knowledge refers to the subject-specific methods, procedures and rules.

Metacognitive knowledge is strategic knowledge related to the learning process and the student's awareness of his own knowledge.

The initial version of taxonomy[2] did not include the intersection between the Knowledge and Cognitive dimensions, but viewed knowledge as one of the general levels in the Cognitive dimension.

[1] Adapted from Krathwohl, David R. A revision of Bloom's taxonomy: An overview. Theory into practice 41.4 (2002), pp. 212-218.

[2] Bloom, B. S., Engelhart, M. D., Furst, E. J., Hill, W. H., & Krathwohl, D. R. (1956). Taxonomy of educational objetives: the classification of educational goals: handbook I: cognitive domain (No. 373.19 C734t). New York, US: D. Mckay.

References

[1] D. Batra, G.M. Marakas, Conceptual data modelling in theory and practice, Eur. J. Inf. Syst. 4 (1995) 185–193, doi:10.1057/ejis.1995.21.
[2] D.H. Jonassen, Instructional design models for well-structured and III-structured problem-solving learning outcomes, Educ. Technol. Res. Dev. 45 (1) (1997) 65–94, doi:10.1007/BF02299613.
[3] R.M. Harden, Learning outcomes and instructional objectives: is there a difference? Med. Teach. 24 (2002) 151–155, doi:10.1080/0142159022020687.
[4] G. Sedrakyan, M. Snoeck, J. De Weerdt, Process mining analysis of conceptual modeling behavior of novices–empirical study using JMermaid modeling and experimental logging environment, Comput. Hum. Behav. 41 (2014) 486–503.
[5] S. Powell, T. Willemain, How novices formulate models. Part I: qualitative insights and implications for teaching, J. Oper. Res. Soc. 58 (2007) 983–995, doi:10.1057/palgrave.jors.2602275.
[6] J.R. Venable, Teaching novice conceptual data modellers to become experts, in: Proc. 1996 Int. Conf. Softw. Eng. Educ. Pract., IEEE Comput. Soc. Press, 1996, pp. 50–56, doi:10.1109/SEEP.1996.533980.
[7] D. Bogdanova, M. Snoeck, in: Domain Modelling in Bloom: Deciphering How We Teach It, Springer, Cham, 2017, pp. 3–17, doi:10.1007/978-3-319-70241-4_1.
[8] D. Batra, S.R. Antony, Novice errors in conceptual database design, Eur. J. Inf. Syst. 3 (1) (1994) 57–69, doi:10.1057/ejis.1994.7.
[9] F. Leung, N. Bolloju, Analyzing the quality of domain models developed by novice systems analysts, in: Proc. Thirty-Eighth Annu. Hawaii Int. Conf. Syst. Sci., IEEE, n.d.: p. 188b–188b. doi:10.1109/HICSS.2005.98.
[10] M. Kayama, S. Ogata, K. Masymoto, M. Hashimoto, M. Otani, A practical conceptual modeling teaching method based on quantitative error analyses for novices learning to create error-free simple class diagrams, in: 2014 IIAI Third Int. Conf. Adv. Appl. Inf., IEEE, 2014, pp. 616–622, doi:10.1109/IIAI-AAI.2014.131.
[11] D.R. Krathwohl, A Revision of Bloom's Taxonomy, Theory Into Practice, 2002, doi:10.1207/s15430421tip4104.
[12] A.J. Cowling, A multi-dimensional model of the software engineering curriculum, in: Proc. Eleventh Conf. Softw. Eng. Educ., IEEE Comput. Soc, n.d.: pp. 44–55. doi:10.1109/CSEE.1998.658297.
[13] R. Adcock, E. Alef, B. Amato, M. Ardis, L. Bernstein, B. Boehm, P. Bourque, J. Brackett, M. Cantor, L. Cassel, R. Edson, R. Fairley, D. Frailey, G. Hafen, T. Hilburn, G. Hislop, D. Klappholz, P. Kruchten, P. Laplante, Q. (Liz) Li, S. Lucero, J. McDermid, J. McDonald, E. McDuffie, B. Michael, W. Milam, K. Nidiffer, A. Pyster, P. Robitaille, M. Shaw, S. Sheard, R. Suritis, M. Towhidnejad, R. Thayer, J.B. Thompson, G. Travassos, R. Turner, J. Urban, R. Valerdi, O. Vikman, D. Weiss, M.J. Willshire, Curriculum guidelines for graduate degree programs in software engineering, 2009 accessed 13.12.2017. https://dl.acm.org/citation.cfm?id=2593248.
[14] J. Lee, Y.C. Cheng, Change the face of software engineering education: a field report from Taiwan, Inf. Softw. Technol. 53 (1) (2011) 51–57, doi:10.1016/J.INFSOF.2010.08.005.
[15] A.J. Cowling, The role of modelling in the software engineering curriculum, J. Syst. Softw. 75 (1–2) (2005) 41–53, doi:10.1016/J.JSS.2004.02.021.
[16] J.T. Gorgone, P. Gray, E.A. Stohr, J.S. Valacich, R.T. Wigand, MSIS 2006, ACM SIGCSE Bull 38 (2006) 121, doi:10.1145/1138403.1138448.

[17] M. Ardis, D. Budgen, G. Hislop, J. Offutt, M. Sebern, W. Visser, Curriculum guidelines for undergraduate degree programs in software engineering, Comput. Curric. Ser. (2014), doi:10.1093/bib/bbp050.
[18] Data modelling and design DTAN — SFIA, (n.d.). https://www.sfia-online.org/en/framework/sfia-7/en/framework/sfia-7/skills/solution-development-and-implementation/systems-development/data-analysis (accessed 14.09.2018).
[19] B.S. |An, O. Bloom, Handbook on Formative and Summative Evaluation of Student Learning, 1971 https://eric.ed.gov/?id=ED049304 (accessed 18.07.2017).
[20] A. Crowe, C. Dirks, M.P. Wenderoth, Biology in bloom: implementing Bloom's taxonomy to enhance student learning in biology., CBE Life Sci. Educ. 7 (2008) 368–381, doi:10.1187/cbe.08-05-0024.
[21] A.Y. Zheng, J.K. Lawhorn, T. Lumley, S. Freeman, Application of Bloom's taxonomy debunks the "MCAT, Science 319 (2008) 414–415. http://science.sciencemag.org/content/319/5862/414. accessed 17.07.2017.
[22] A.R. Thompson, V.D. O'Loughlin, The Blooming anatomy tool (BAT): a discipline-specific rubric for utilizing Bloom's taxonomy in the design and evaluation of assessments in the anatomical sciences, Anat. Sci. Educ. 8 (2015) 493–501, doi:10.1002/ase.1507.
[23] N.B. Zaidi, C. Hwang, S. Scott, S. Stallard, J. Purkiss, M. Hortsch, Climbing Bloom's taxonomy pyramid: lessons from a graduate histology course, Anat. Sci. Educ. 00 (2017), doi:10.1002/ase.1685.
[24] C.W. Starr, B. Manaris, R.H. Stalvey, C.W. Starr, B. Manaris, R.H. Stalvey, Bloom's taxonomy revisited, in: Proc. Thirty-Nineth SIGCSE Tech. Symp. Comput. Sci. Educ. - SIGCSE '08, ACM Press, New York, USA, 2008, p. 261, doi:10.1145/1352135.1352227.
[25] P. Dolog, L.L. Thomsen, B. Thomsen, Assessing problem-based learning in a software engineering curriculum using Bloom's taxonomy and the IEEE software engineering body of knowledge, ACM Trans. Comput. Educ. 16 (2016) 1–41, doi:10.1145/2845091.
[26] J. Biggs, Enhancing teaching through constructive alignment, High Educ. 32 (1996) 347–364, doi:10.1007/BF00138871.
[27] A.R. Hevner, S.T. March, J. Park, S. Ram, Design science in information systems research, MIS Q. 28 (2004) 75–105.
[28] A. Hevner, S. Chatterjee, in: Design Science Research in Information Systems, Springer, Boston, MA, 2010, pp. 9–22, doi:10.1007/978-1-4419-5653-8_2.
[29] A. van Lamsweerde, Requirements Engineering: from System Goals to UML Models to Software Specifications, John Wiley, 2009 https://www.researchgate.net/publication/220693667_Requirements_Engineering_From_System_Goals_to_UML_Models_to_Software_Specifications (accessed 10.07.2017).
[30] M. Blaha, W. Premerlani, Object-Oriented Modeling and Design For Database Applications, Prentice Hall, 1998 http://dl.acm.org/citation.cfm?id=264777 (accessed 02.05.2017).
[31] A. Olivé, Conceptual modeling of information systems, 2007 (accessed 19.05.2017). http://download.springer.com/static/pdf/352/bok%253A978-3-540-39390-0.pdf?originUrl=http%3A%2F%2Flink.springer.com%2Fbook%2F10.1007%2F978-3-540-39390-0&token2=exp=1495185531~acl=%2Fstatic%2Fpdf%2F352%2Fbok%25253A978-3-540-39390-0.pdf%3ForiginUrl%3Dhttp%25.
[32] R. Elmasri, S. Navathe, Fundamentals of Database Systems (2011). http://www.mypearsonstore.com/bookstore/fundamentals-of-database-systems-0136086209. (accessed 10.07.2017).
[33] M. Snoeck, UML class diagrams for software engineering | edX, 2017 (accessed 10.07.2017). https://www.edx.org/course/uml-class-diagrams-software-engineering-kuleuvenx-umlx.
[34] The Open University, Modelling object-oriented software – an introduction, (n.d.). http://www.open.edu/openlearn/science-maths-technology/computing-and-ict/modelling-object-oriented-software-introduction/content-section-0 (accessed 10.07.2017).
[35] J. Widom, DB9 Unified Modelling Language|Stanford Lagunita, 2014 (accessed 12.07.2017). https://lagunita.stanford.edu/courses/DB/UML/SelfPaced/info.
[36] M. Snoeck, Architecture and Modelling of Management Information Systems, KU Leuven, (n.d.). https://onderwijsaanbod.kuleuven.be/syllabi/e/D0I71AE.htm#activetab=doelstellingen_idp548768 (accessed 18.07.2017).
[37] S. Poelmans, Design of a Business Information System, KU Leuven, (n.d.). https://onderwijsaanbod.kuleuven.be/syllabi/v/e/HMH28EE.htm#activetab=doelstellingen_idp1415488 (accessed 18.07.2017).
[38] M. Kolp, A. Pirotte, UCL/IAG/ISYS - Unité de Systèmes d'Information (ISYS) – UCL, (n.d.). http://www.isys.ucl.ac.be/etudes/cours/geti2101/ (accessed 18.07.2017).
[39] P. Heymans, 2016 (accessed 25.07. 2017). https://directory.unamur.be/teaching/courses/IHDCB335/2015.
[40] S. Faulkner, Bases de données - Université de Namur, 2016 (accessed 25.07.2017). https://directory.unamur.be/teaching/courses/EIMIB212/2016.
[41] M. Snoeck, Enterprise Information Systems Engineering: The MERODE Approach, Springer Publishing Company, Incorporated, 2014.
[42] P.P.-S. Chen, P. Pin-Shan, The entity-relationship model—toward a unified view of data, ACM Trans. Database Syst. 1 (1976) 9–36, doi:10.1145/320434.320440.
[43] O.I. Lindland, G. Sindre, A. Solvberg, Understanding quality in conceptual modeling, IEEE Softw. 11 (1994) 42–49, doi:10.1109/52.268955.
[44] H.J. Nelson, G. Poels, M. Genero, M. Piattini, A conceptual modeling quality framework, Softw. Qual. J. 20 (2012) 201–228, doi:10.1007/s11219-011-9136-9.
[45] D. Moody, The "Physics" of notations: toward a scientific basis for constructing visual notations in software engineering, IEEE Trans. Softw. Eng. 35 (2009) 756–779, doi:10.1109/TSE.2009.67.
[46] J. Coplien, Advanced C++ programming styles and idioms, in: Proceedings. Technol. Object-Oriented Lang. Syst. TOOLS 25 (Cat. No.97TB100239), IEEE Comput. Soc., n.d.: pp. 352–352. doi:10.1109/TOOLS.1997.681881.
[47] F.D. Davis, Perceived usefulness, perceived ease of use, and user acceptance of information technology, MIS Q. (1989), doi:10.2307/249008.
[48] D.L. Moody, The method evaluation model : a theoretical model for validating information systems design methods, in: ECIS 2003 proceedings, 2003, p. 79.
[49] J.J.G. van Merriënboer, L. Kester, F. Paas, Teaching complex rather than simple tasks: balancing intrinsic and germane load to enhance transfer of learning, Appl. Cognit. Psychol. 20 (2006) 343–352, doi:10.1002/acp.1250.