Data & Knowledge Engineering 153 (2024) 102342

Contents lists available at ScienceDirect

Data & Knowledge Engineering

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

Evaluating quality of ontology-driven conceptual models abstractions

Elena Romanenko [a,*], Diego Calvanese [a,b], Giancarlo Guizzardi [c]

[a] KRDB Research Centre on Knowledge and Data, Free University of Bozen-Bolzano, Bolzano, Italy
[b] Department of Computing Science, Umeå University, Umeå, Sweden
[c] Semantics, Cybersecurity & Services (SCS), University of Twente, Enschede, The Netherlands

ARTICLE INFO

Dataset link: https://w3id.org/ExpO/github

Keywords:
Conceptual model abstraction
Ontology-driven conceptual models
Quality evaluation of abstractions
Unified foundational ontology (UFO)
FAIR model catalog
User studies in conceptual modeling

ABSTRACT

The complexity of an (ontology-driven) conceptual model highly correlates with the complexity of the domain and software for which it is designed. With that in mind, an algorithm for producing ontology-driven conceptual model abstractions was previously proposed. In this paper, we empirically evaluate the quality of the abstractions produced by it. First, we have implemented and tested the last version of the algorithm over a FAIR catalog of models represented in the ontology-driven conceptual modeling language OntoUML. Second, we performed three user studies to evaluate the usefulness of the resulting abstractions as perceived by modelers. This paper reports on the findings of these experiments and reflects on how they can be exploited to improve the existing algorithm.

1. Introduction

Conceptual models (CMs) are concrete artifacts representing conceptualizations of particular domains. Ontology-driven conceptual modeling is a paradigm lying at the intersection of conceptual modeling and ontology engineering [1]. *Ontology-driven conceptual models* (ODCMs) are usually considered a special class of conceptual models, namely, those that benefit from reusing foundational ontologies to guide their development.

Both ODCMs and traditional CMs alike play a fundamental role in organizing communication between people with different backgrounds, such as programmers, ontology engineers, and domain experts. However, the complexity of a model highly correlates with the complexity of the domain and software for which it is designed. Thus, although (OD)CMs are developed for communication, are human-centered, and are aimed at human comprehension [2], one of the most challenging problems is "to understand, comprehend, and work with very large conceptual schemas" [3].

It is widely accepted that one way to reduce complexity is through processes of abstraction [4]. The intuition about CM abstractions is to provide the user with a bird's-eye view of the model by filtering out some details. Some of the existing algorithms for abstraction are based on classic modeling notations (UML, ER) and use topological properties of the graphs (see [5,6]), while others leverage the ontological semantics offered by ontology-driven conceptual modeling languages (see [7,8]).

In [8], we have proposed an algorithm for building ODCM abstractions for the language OntoUML. In this paper, we focus on assessing the quality of the ODCM abstractions produced by this algorithm from different perspectives. The paper is an extension of [9] and elaborates in much more detail the quality assessments put forth there. First, we elaborate on the notion of conceptual model quality and on the nature of the abstraction process from a cognitive point of view. This allows us to more precisely characterize and discuss our empirical study w.r.t. the quality dimensions of abstractions that it addresses.

* Corresponding author.
E-mail address: eromanenko@unibz.it (E. Romanenko).

https://doi.org/10.1016/j.datak.2024.102342
Received 15 December 2023; Received in revised form 4 April 2024; Accepted 6 July 2024
Available online 14 July 2024
0169-023X/

Second, we elaborate on the implementation and testing of this algorithm over a recently created FAIR catalog of OntoUML conceptual models [10,11]. This testing over the FAIR catalog provides evidence for the *correctness* of the algorithm's implementation, i.e., that it correctly implements the model transformation rules prescribed by the algorithm, and for its *effectiveness*, i.e., that it is able to achieve high compression (summarization) rates over these models.

However, in addition to these properties, it is fundamental to understand the *validity* (appropriateness, usefulness) of this algorithm, i.e., that it achieves what it is intended to do, namely, provide summarizing abstractions over the input models whilst preserving the gist of the conceptualization being represented. We thus elaborate on three experiments conducted with modelers/model users to evaluate the validity of the resulting abstractions and the process that leads to their creation. The analysis of the results produced by these studies (much more elaborated here) as well as of the *process of abstracting* (not discussed at all in the original paper) provide important lessons learned to be systematically employed in the future for improving our algorithm.

The remainder of the paper is organized as follows: Section 2 presents our baseline and background; Section 3 describes in detail the conducted experiments, including user studies; Section 4 assesses the quality of the resulting models; Section 5 elaborates on final considerations and future work.

2. Background

2.1. Ontology-driven conceptual models

By a conceptual model, one could refer to a UML Class Diagram, an i* Goal Model, or a Business Process Model. This is because conceptual models are high-level abstractions used to capture information about the domain and all these languages (among many others) are employed for that. *Ontology-driven conceptual models* are usually considered a special class of conceptual models that utilize foundational ontologies to ground modeling elements, modeling languages, and tools [12].

In the literature, there are a number of approaches that connect ontologies and conceptual models in many different ways. In fact, as early as 2008, there were special issues exploring the multiple relations between these topics [13]. These range from the use of the so-called ontology specification languages (e.g., OWL or, more generally, Description Logics) to reason with conceptual models [14–16], to the use of standard conceptual modeling languages for representing ontologies [17], to the use of ontological theories (in the philosophical sense) to analyze conceptual modeling constructs [18,19]. Here, by *ontology-driven conceptual modeling languages* (ODCML), we mean something stronger than these three dimensions albeit addressing aspects of all of them. By an ODCML, we mean a language that commits to a foundational ontology (i.e., a domain-independent system of axiomatic ontological theories) in a strong sense, namely: *(i)* the language has a grammar (e.g., a meta-model) comprising modeling primitives that reflect the ontological distinctions put forth by this underlying ontology; *(ii)* the grammar incorporates semantically-motivated syntactical constructs that reflect the axiomatization of that underlying ontology [20]. As discussed in depth by Guizzardi [20,21], a conceptual modeling language can have its abstract syntax (e.g., a meta-model enriched by grammatical constraints) and semantics systematically designed according to an underlying reference ontology.

In principle, ODCMs are not bound to any specific foundational ontology, so one can choose the most appropriate one to the task at hand. A recent special issue of the Applied Ontology journal [22] describes seven of them — BFO [23], DOLCE [24], GFO [25], GUM [26], TUpper [27], UFO [28], and YAMATO [29]. We chose the *Unified Foundational Ontology* (UFO) [28] for developing our abstraction algorithm because it is the only one of these mainstream foundational ontologies that has an associated ODCML (termed OntoUML).

OntoUML is an ODCML in the strong sense as described above *(i-ii)*. It is technically implemented as a UML profile, i.e., a lightweight extension of the UML meta-model endowed with formal OCL constraints [30,31]. In the OMG MOF jargon,[1] the OntoUML meta-model is an M2-level model and, hence, the language is meant to represent M1-level models. Finally, the language is complemented by an ecosystem of tools for model engineering, validation, verification, code generation, complexity management, etc., which leverage this explicit language meta-model and the associated semantics of the language [32]. For an in-depth discussion, philosophical justification, and formal characterization of UFO and OntoUML, we refer to [28,30–32], while here we briefly review only some of its notions that are germane for the purposes of this article.

The first distinction that UFO makes is highlighting the existence of both *endurants* and *perdurants*. Endurants are object-like individuals that exist in time and are able to qualitatively change while maintaining their identities [31]. Examples include ordinary objects, i.e., 'Person' and 'Organization', as well as existentially dependent entities, e.g., 'Symptom' or 'Enrollment'. In contrast, perdurants are entities that unfold in time, accumulating temporal parts, including, but not limited to *events*.

Endurants in UFO can instantiate different types, which are distinguished by the formal meta-properties of *rigidity* and *sortality*. Rigidity is a property that describes the dynamics of how the type may be instantiated. From this perspective, types may be classified as *rigid*, *anti-rigid*, and *semi-rigid*. Rigid types classify their instances necessarily (i.e., in all possible situations these instances exist), e.g., stating that 'Person' is rigid means that no person may cease to be a person and still exist. Anti-rigid types, including *roles* and *phases*, classify their instances contingently, for example, 'Registered Person'. Lastly, semi-rigid types classify some of their instances necessarily and some of their instances contingently.

To define sortality, we first need to introduce the notion of a *principle of identity*. A *principle of identity* is principle that establishes what makes two individuals the same and, by the same token, what kind of changes an individual can undergo and still be the same

[1] https://www.omg.org/ocup-2/documents/Meta-ModelingAndtheMOF.pdf

individual [33]. A type is *sortal* if all of its instances follow the same identity principle. Examples include 'Person' and 'Organization'. A *non-sortal* type aggregates properties that are common to different sortals and, thus, can be instantiated by individuals that follow different identity principles. An example is 'Artifact', which applies to different types of documents, music, and video recordings. As we will see later in the paper, a special type of rigid sortal type called a *kind* plays a fundamental role in existing OntoUML abstraction algorithms [7,8]. For more examples and formalization of these and other notions, one can refer to [31].

The role of conceptual models in general and ODCMs, in particular, is quite precisely specified in the literature. They are intended to enable clients and analysts to understand one another, to communicate successfully with application programmers, and hence "play a fundamental role in different types of critical semantic interoperability tasks" [34]. It has been shown [12], that novice modelers applying the ODCM technique produce models of higher quality when compared to novice modelers using traditional modeling approaches, especially when working on more challenging or advanced facets of a domain or scenario.

However, when dealing with complex domains, often the number of concepts and sub-diagrams of (OD)CMs goes far beyond the cognitive tractability threshold of those people who are supposed to work with them. The problem of making conceptual models (and ODCMs) more comprehensible is addressed in the literature by the proposal of different complexity management techniques, and for quite some time this research area has been under active study. According to Villegas Niño, existing methods can be grouped into the following categories: *(1)* clustering methods, *(2)* relevance methods, and *(3)* summarization methods [3, p. 54]. The first group covers methods in which elements of the CM are divided into groups (clusters). Relevance methods rank CM elements into ordered lists according to their value, while summarization methods produce a reduced version of the original CM.

A number of approaches for complexity management have been proposed precisely for ODCML, e.g., [35,36]. In this paper, we refer to the task of producing a meaningful but reduced version of the original conceptual model by filtering out the details and keeping the most important notions — also known as *summarizing* or *abstracting*.

2.2. Abstracting ontology-driven conceptual models

Traditional algorithms for abstracting CMs mostly depend on syntactic properties of the model, such as closeness or different types of distances between model elements — hierarchical distance, structural-connective distance, or category distance [5]. However, when relying solely on these properties, there is no guarantee that a model element satisfying some topological requirement by necessity belongs to the most important concepts of the model. Moody and Flitman referred to this issue as a *lack of cognitive justification* [37].

One of the most interesting approaches for CMs abstraction was proposed in [6] and is based on pattern detection and replacement rules. Although it was originally designed for UML, the author attempted to define patterns that reflect the intuitive semantics of UML Class Diagrams [38]. These patterns include transitivity of dependence, inheritance, and propagation from parts to wholes. The drawback of the approach is that it requires the users of the model to perform a manual *seeding*, i.e., a pre-selection of certain model elements that need to be preserved in the final abstraction. This, however, imposes on these stakeholders a manual, tedious, time-consuming task but, more importantly, it requires an *a prioristic* and in-depth understanding of the model itself. Since supporting the understanding of a large and complex model is exactly what the algorithm is supposed to be doing, requiring that users are always able to perform a sensible seeding is an unrealistic assumption. In order to circumvent this important limitation, Huang et al. have suggested relying on the structural properties of the graph and selecting seeding using a version of the PageRank algorithm [39].

In contrast, the graph-rewriting rules in the existing OntoUML abstraction algorithms [7,8] leverage the ontological semantics of the language based on the UFO ontology. The first version of an abstraction algorithm was introduced in [7], followed by an enhanced version in [8], which was able to abstract more sophisticated models, i.e., models employing a larger number of formal ontological primitives. For detailed descriptions and justifications of the algorithms, we refer to the previously published papers [7,8], while the final set of rules is provided further. For the scope of this paper, it is also important to highlight that: *(1)* both algorithms bear a remarkable simplicity in the number of rules; *(2)* they are deterministic; *(3)* they do not require human intervention and seeding; and *(4)* they are computationally efficient and scalable, thus, able to process very large models in a timely manner.

The newest version of the algorithm proposed in [8] defines 11 graph-rewriting rules, which are grouped into three categories, namely, rules for abstracting: *(1)* parthood relations (compositions and aggregations); *(2)* different *aspects* of objects — i.e., dependent objectified features of objects, such as *relators*, *qualities*, and *modes* (as defined in UFO); and *(3)* hierarchies of concepts (generalization relations). In line with this algorithm, in this paper (i.e., during discussions, as well as in figures and tables) we refer to models produced by applying rules from these groups as *parthood abstractions*, *aspect abstractions*, and *hierarchy abstractions*, respectively. Also, one should note that the application of a rule does not always imply the complete elimination of the corresponding construct being addressed, e.g., after applying rules from the first group, some of the parthood relations could still be kept in the resulting model. These rules can be applied in a compositional way, so that we can, e.g., abstract both parthood relations and hierarchies. Thus, when compositionally applying these three groups of rules, one can receive eight possible models (including the original model and full abstraction) [8].

In Tables 1–3, the graph-rewriting rules for producing abstractions are in fact patterns for models expressed in OntoUML. Thus, to apply them, one needs to relieve the matching model with the replacement, where the placeholder classes are substituted with the concrete classes of the model.

In this paper, we extend the initial evaluation performed in [7], assessing the compression rate achieved by this algorithm (i.e., how much information is filtered out) on a larger sample of models. Moreover, we investigate the cognitive adequacy (validity) of the produced abstraction results from the modelers' point of view.

Table 1
Graph-rewriting rules for abstraction parthood relations.

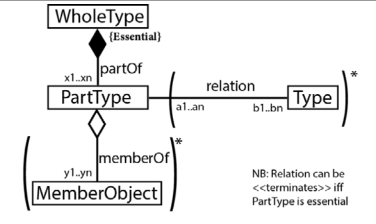
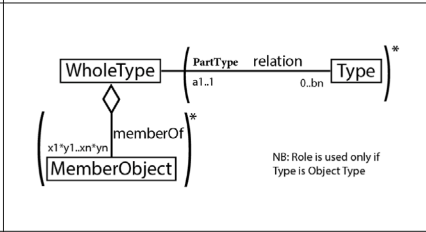
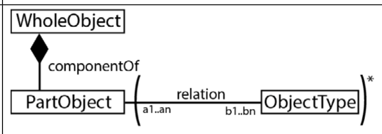
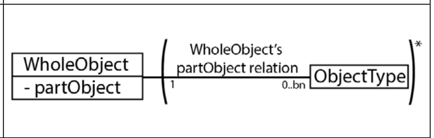
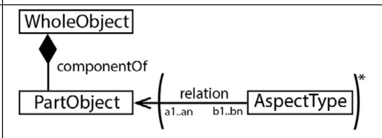
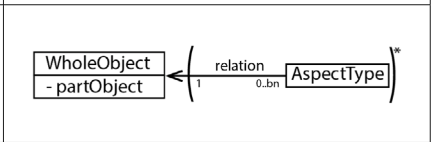
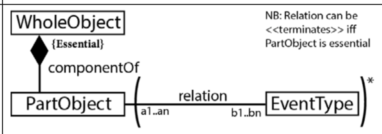
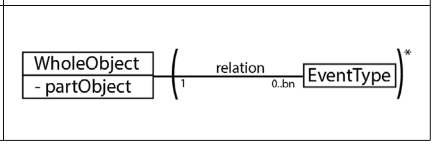
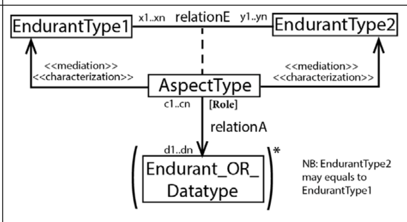
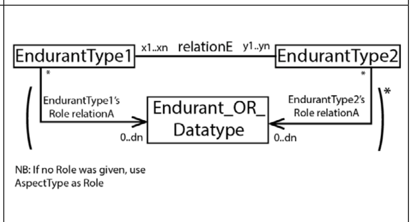
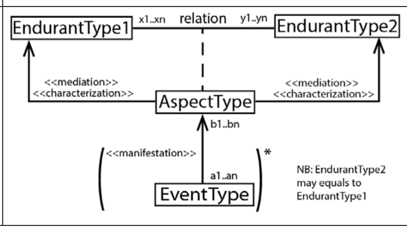
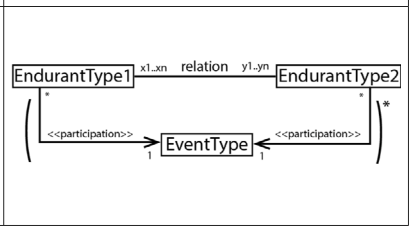
Name	Matching Graph	Replacing Graph
P.1		
P.2		
P.3		
P.4		

Table 2
Graph-rewriting rules for aspects abstraction.

Name	Matching Graph	Replacing Graph
A.1		
A.2		

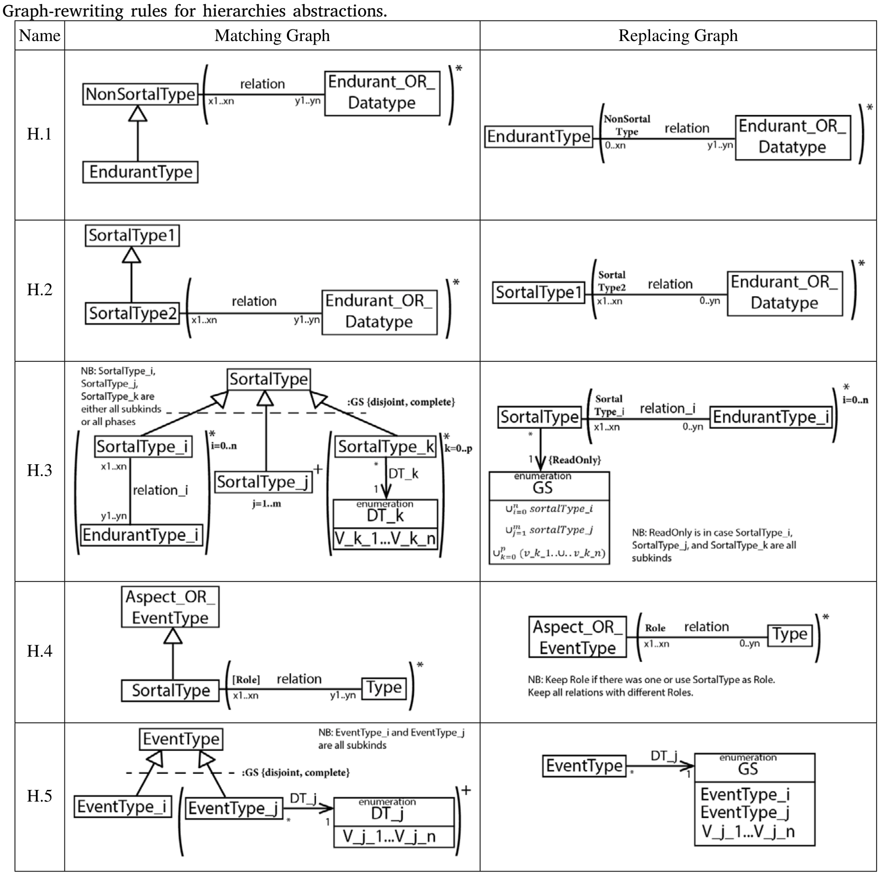
2.3. Quality in conceptual modeling

The notion of *quality* in conceptual modeling is a difficult one. From the beginning of the 90s there were attempts to provide a proper definition, but, when given, those definitions were "vague and complicated" [40]. Thus, researchers preferred simply to list the desired properties of the models. With time, a number of model quality frameworks have been proposed.

Lindland et al. suggested a framework based on semiotic theory. The authors claimed that one of the main features of their framework is a separation of *goal* from *means* of modeling [41]. The quality dimensions explicitly defined in the framework are organizational quality, social quality, pragmatic quality, semantic (and perceived semantic) quality, empirical quality, and physical quality. We here abstain from discussing the details of this framework, but it is worth mentioning that its main focus is on the *product of conceptual modeling*, i.e., the model itself.

Another approach was taken in the so-called Bunge–Wand–Weber (BWW) framework for analysis and conceptualization of real-world objects (see [42,43]). Two main evaluation criteria in that framework are ontological completeness and ontological

Table 3
Graph-rewriting rules for hierarchies abstractions.



clarity. Nelson et al. emphasized that BWW focuses on the *process of conceptual modeling*. In particular, within this framework users of the system should apply the "direct observation process" in order to create a user's view of the domain.

As an extension and a combination of the above-mentioned frameworks, Nelson et al. have suggested a conceptual modeling quality framework that incorporates 24 quality dimensions distributed into four layers: physical layer, knowledge layer, learning layer, and development layer (see Table 1 in [40]).

The idea of employing semiotic theory in conceptual modeling design was later reused in FRISCO, a framework of information system concepts [44], and within a framework for understanding quality in conceptual modeling — SEQUAL [45]. The latter (which is a more detailed framework) distinguishes between goals and means to achieve these goals in models, and is closely linked to linguistic and semiotic concepts. Since this framework guided the evaluation of the abstractions, it is worth mentioning all quality dimensions that it differentiates.

In total, SEQUAL distinguishes seven quality levels, namely:

– *physical quality:* how the model is presented on paper, could it be persistent and available;
– *empirical quality:* ergonomics of modeling tools, visual impression of the model, including coloring schema;
– *syntactic quality:* syntactical correctness according to the modeling language and vocabulary;
– *semantic and perceived semantic quality:* validity and completeness of the model;
– *pragmatic quality:* comprehension of the model by participants, including content relevance;
– *social quality:* agreement in model interpretation by several users of the model;
– *deontic quality:* how the model achieves the goal.

For each level, there are one or more quality characteristics. For a detailed description of SEQUAL, we refer to [45].

However, in evaluating our abstraction algorithm, we dispense with some of these dimensions. For example, we dispense with the physical and ergonomic quality dimensions, since the layout of the model is kept as it was in the original model, and the coloring schema is fixed by the OntoUML Editor. The same holds for the last two dimensions: deontic quality is hard to assess in the case of abstractions since the goal of abstraction differs from the goal of the original model, while social quality would require a deeper understanding of the domain by all subjects of our study. Thus, in the following, *we assess the resulting abstractions at syntactic, semantic, and pragmatic levels*.

Before coming to the details of the semantics of abstraction, it is worth mentioning that deontic quality is hard to assess not only for the abstraction but for the original model as well. Although a framework for the goals of modeling has been proposed by Guizzardi and Proper [46], a recent empirical study has shown that "modelers only subjectively assess the satisfaction of their modeling goals" [47]. In other words, even when goals are explicitly articulated, model properties that are relevant to their achievement (e.g., correctness, completeness, and confinement) are not explicitly measured.

2.4. On the semantics and dual nature of model abstractions

Before assessing ODCM abstractions and formulating their desired properties, we should elaborate on some of the cognitive aspects of model abstraction. In cognitive sciences, abstraction is considered one of the highest forms of thinking [48]. Speaking of conceptual modeling, Egyed defined abstraction as "a process that transforms lower-level elements into higher-level elements containing fewer details on a larger granularity" [6]. This definition was used while developing algorithms, such as [7,8]. However, it leaves open several fundamental questions: Can we provide different types of abstractions? Is abstraction by definition a lossy transformation? Is the abstraction process finite? In the following, we provide our view on abstraction in the conceptual modeling domain.

Hereinafter, we consider abstraction in relation to the information content. Although information content is only one of the dimensions along which abstraction can be conducted [49], it is the most relevant one in case of conceptual modeling. We also intuitively refer to the abstraction process as *a process of reducing the amount of information* contained in the original model. So far, by the amount of information in the model, we simply mean the number of classes and relations defined in it.

The foundational contributions to the field of semantics of abstraction were done by Hobbs [50], Plaisted [51], Tenenberg [52], and also Nayak and Levy [53]. As outlined by Saitta and Zucker, most existing theories identify abstraction with *a mapping from a ground* (original) *to an abstracted* (intended) *space*, but differ in the nature of spaces and the corresponding type of mapping [49, p. 49].

Saitta and Zucker suggest distinguishing the following categories of representation changes [49, p. 50]: *(1)* perceptive, *(2)* syntactic, *(3)* semantic, and *(4)* axiomatic. The suggested categories are not mutually disjoint and one approach can belong to several groups at the same time. Taking into account this classification, the algorithm suggested in [8] belongs to all of the categories specified above. At the level of perception, the granularity is changed when abstracting, for example, generalization relations. In such cases (see Rules H.2 and H.4 in Table 3), the dependent concept is absorbed by the parent concept. At the syntactic level, some rules (namely, Rule P.2 in Table 1 and Rule A.1 in Table 2) rename the relations when abstracting. At the level of semantics, some of the concepts are eliminated (see Rule H.1 in Table 3), so that the number of competence questions that the model can answer is reduced. Also, when abstracting generalization sets, the original set of axioms is changing as well (see Rules H.3 and H.5 in Table 3).

Giunchiglia and Walsh proposed a more general theory of the abstraction process. They suggested distinguishing between *theorem-decreasing*, *theorem-constant*, and *theorem-increasing* abstractions depending on the changes in the theorems of the formal system [54]. In theorem-constant abstractions, the abstract space has exactly the same theorems as the ground space reformulated in another language, so that all well-formed formulas of the original space map onto well-formed formulas in the abstract space. In theorem-increasing abstractions, the abstract space has more theorems than the ground one, while the opposite happens for theorem-decreasing abstraction. The authors argued that "certain subclasses of theorem-increasing abstractions are the appropriate formalization for abstraction" [54] because they preserve all existing theorems and have intended properties. However, in [55] we have shown that this property cannot always be guaranteed for our ODCM abstraction algorithms [7,8].

It is interesting to note, that authors with a more practical view on abstraction (see [6,56]), intuitively define *abstraction as a process*. The process, that "can be iterated to generate hierarchies of abstract spaces" [54]. At the same time, the developed algorithms were mostly focused on the result of that process and referred to the *abstraction as a new version of the original model*.

In the first version of the algorithm on which our work is based [7], the constraints on rule applications were part of the methodology and the algorithm itself. In other words, the order of the rule application was constant. In the second version of the algorithm, there has been an attempt to specify the order for rule applications (see listings in [8]). However, those order constraints were part of the methodology for rule application, but not part of the semantics of the rules themselves. Also, it was shown [55], that one path in that generated hierarchy could be preferred. This goes in line with the idea of having a transparency path in that hierarchy of abstract spaces (see [57]).

Summarizing all of the above, abstraction in conceptual modeling has a dual nature. On the one hand, it is a process that leads to changes in the models. On the other hand, an abstracted model, which reduces the information contained in the original one, has its own value.

In the close domain of eXplainable AI, Miller has noted that there is a need to distinguish between "explanation" as: *(1)* a cognitive process; *(2)* a product that may come in different forms; and *(3)* a social interaction [58]. Qian and Choi emphasize that, in cognitive domains, abstraction is reached through three cognitive processes: *(1)* filtering irrelevant information, *(2)* locating fundamental similarities, and *(3)* mapping out problem structures [48]. The question is whether in conceptual modeling *abstraction-as-a-process* is as important as the final result. So far we will leave this question open, but when assessing the quality of abstraction it is worth assessing not only the resulting models but the process as well.

In the following, in order to distinguish the terms, we refer to an *abstraction* as an ontology-driven conceptual model with specific characteristics (in contrast to the original model) and to an *abstraction process*, or *abstracting*, as the process of producing those models.

3. Empirical studies on ontology-driven conceptual models abstractions

An initial attempt to compare the abstraction results of the first version of the algorithm (as in [7]) to other existing approaches was made in [38]. The main hypothesis of that experiment was that the first abstraction algorithm produces models capturing the gist of the original model more appropriately than the competing algorithms proposed in [6,39].

The experiment was organized as follows. A group of 50 participants with different modeling backgrounds — from students to professionals with years of modeling experience — were presented with the original conceptual model in the car rental domain and with three abstractions based on Egyed's algorithm. In the first abstraction the seeding was done by experts, in the second — *kinds* were selected as seeding, and in the third — seeding was done using PageRank as suggested by Huang et al. The participants were asked to rate the models according to their view on the quality of the abstraction and justify their choice.

The suggested algorithm from [7] was clearly preferred by practitioners with large modeling experience. However, overall, the experiment did not demonstrate a significant preference for one of the tested abstractions. This result is not negative per se, given that [7] achieves the same results with only four rules (as opposed to 92 rules for the competing algorithms). However, the main informative result of that experiment is to be found in the comments received from the participants, which challenged some of the assumptions of the suggested algorithms.

The original abstraction algorithm and its refined version were developed with two assumptions, whose influence was not fully apparent until the first questionnaire was filled out in [38]. The first assumption was that *aspects* — given that they are existentially dependent entities — could always be safely abstracted from the models without a significant loss of information content. For some of the *aspects* that was found to be true, e.g., it is not that important to mention the color of the car while talking about car rental, but the *relator* 'Car Rental Agreement', although being dependent on other entities, according to participants should always be preserved in the resulting abstractions of the car rental domain.

The second assumption implicitly assumed that *kinds* are the most valuable entities of an ODCM. As mentioned before, *kind* is a type that applies necessarily to its instances and provides them with the identity criteria. Thus, *kinds* uniquely divide the entire space of all objects existing within the domain into non-overlapping groups. However, in some cases, abstraction to the level of *kinds* leads to situations when the result includes more objects than expected. For example, if in our system 'Person' and 'Organization' are only playing the *role* of 'Customer' who can rent a car, and they do not have valuable relations to other objects, does it make sense to keep both of them and duplicate all relations with the corresponding role name, or would it be enough to abstract them up to the 'Customer'?

In order to properly evaluate the second version of the algorithm [8] and to provide recommendations for its improvement, we then conducted several empirical studies driven by the following research questions:

- *RQ1:* Does the algorithm provide syntactically valid models?
- *RQ2:* What is the rationale for choosing the most valuable concepts of the model (seeding)? How does it change depending on the given goal?
- *RQ3:* What is the pragmatic value of the abstractions? Can abstraction serve as an explanation of the original model?
- *RQ4:* What are the characteristics of models that are preferred as intermediate steps during the abstraction process?

Following the SEQUAL framework, each research question was formulated to assess one of the facets of the model's quality (syntactic, semantic and pragmatic quality of the abstractions), and — taking into account the discussion in the previous section — quality of the abstraction process as well.

As mentioned before, in most of the cases the abstraction contains *less* semantic value than the original model, so we cannot measure the semantic quality of the model directly. However, we may have a 'lower bound'. If the model does not contain all the necessary concepts (seeding), it should not be considered valid anyway. Thus, we would like to know what rationale is hidden in the expert's mind when the decision about the importance of a particular concept is made.

Also, with three groups of rules (applied together or separately), we can generate seven abstractions, where six of them could be considered as intermediate steps towards the final one. However, it is not clear if some of those steps would be preferred by the users of the algorithm.

In total, we have conducted four experiments. The first one with the FAIR catalog of ontology-driven conceptual models (precisely aimed at answering *RQ1*), and three — following the triangulation approach [59] — with users[2]: *(1)* in-person interviews about the abstraction process (contributed to answering *RQ2* and *RQ3*), *(2)* online structured interviews with modelers (for answering *RQ2*, *RQ3* and *RQ4*), and *(3)* online questionnaire with conceptual model users (again, *RQ2*, *RQ3* and *RQ4*). Detailed descriptions of each experiment are given in the next sections.

3.1. Experiments with the FAIR model catalog

The main goal of the first experiment was to answer *RQ1*, i.e., to make sure that the abstractions generated by the algorithm do not contain syntactic errors. This becomes possible with the creation of a FAIR model catalog of ontology-driven conceptual models [10,11] (hereinafter referred to as the Catalog[3]). The Catalog offers a diverse collection of conceptual models, created by

[2] The study has been reviewed and received approval from the Ethics Research Committee at the Free University of Bozen-Bolzano, Italy (Prot. n. 5/2022 from 28/09/2022).

[3] https://w3id.org/ontouml-models.

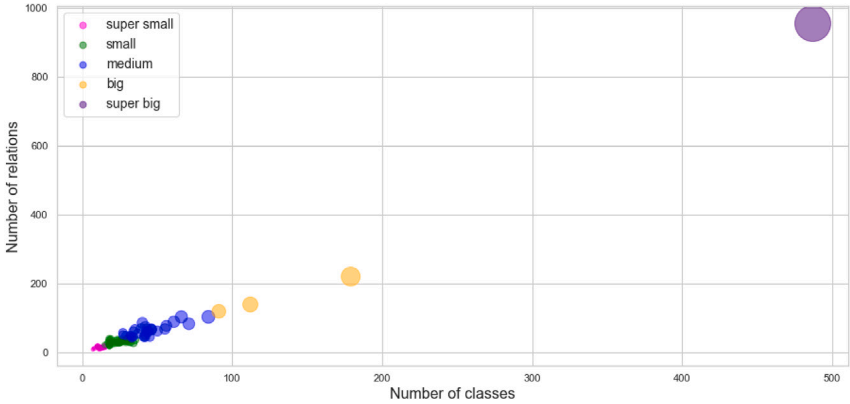


Fig. 1. Diffusion of eligible conceptual models from the Catalog in the number of classes and relations.

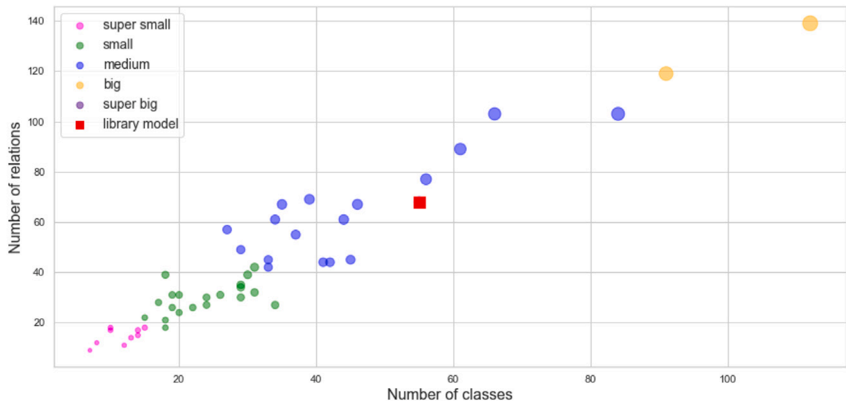


Fig. 2. Diffusion of syntactically valid models from the Catalog used in experiments.

modelers with varying modeling skills, for a wide range of domains, with different purposes, and during the time of experiments consisted of 168 models. To make API requests to the servers that check model syntax and create model abstractions, we were interested in the JSON distribution of the models. And because of the importing/exporting issues with JSON, we considered 159 of the original number.

The problem with the Catalog from the point of view of our research is that it contains all errors that were introduced by the model's authors. Those include not only typos but also modeling mistakes. The decision to keep models as they were created was reasonable for the Catalog's authors because one of the purposes was an empirical discovery of modeling (anti-)patterns [10]. Unfortunately, this contradicts the goal of assessing the quality of the algorithm, since most of the time the original errors would be propagated to their respective abstractions. Therefore, only part of the Catalog was used for the experiments.

We selected models satisfying the following criteria: *(1)* they contained only those 16 stereotypes, for which the second version of the algorithm was developed (that left us with 87 models out of the original number, see Fig. 1), and *(2)* they did not contain syntactical modeling errors that could not be easily fixed. The last statement requires some explanation. First, all models were automatically checked on the syntactical correctness with the OntoUML Editor (OntoUML plugin[4] for Visual Paradigm[5]). Only 41 (out of 87) models did not contain syntactical errors. After a manual review of the rest, 8 models — that contained obvious and easily-fixable errors — were rectified and re-included in the set under consideration. Taking into account the aforementioned conditions, we selected 49 models for the purpose of the algorithm evaluation.

Before creating new models for further experiments, it was also necessary to check how different were the selected models within the Catalog. As for the size of the models, we distinguished five groups of models, from the super small with less than 35 modeling elements to the super big, which was represented by one ODCM (see Fig. 1). On average, the models have about 38 classes and 55 relations, where about half of them are generalization relations. The distribution of the 49 valid models together with the *Library model* that was created for the experiments with users is shown in Fig. 2.

4 https://github.com/OntoUML/ontouml-vp-plugin.
5 https://www.visual-paradigm.com.

For each of these pre-selected models, from one to seven abstractions were generated, giving us, in total, 278 unique models. To one of the models, 'pereira2020ontotrans'[6] none of the abstraction rules were applicable, and this model was excluded from the consideration. The resulting 230 abstractions were checked for syntactical correctness with a script using the API developed for the OntoUML plugin.[7]

3.2. Experiments with users of ontology-driven conceptual models: face-to-face interviews

The results of the questionnaire in [38] leave unexplained the reason why modelers prefer one abstraction over another. Our first experiment with users was then designed to find out what people consider when abstracting a conceptual model, i.e., their (often tacit) rationale for choosing the main concepts to be preserved, and how the final abstraction produced by the algorithm corresponds to that target preferred one. Thus, the main goal of the study was to contribute to answering *RQ2* and *RQ3*, however, we also wanted to observe the abstraction process in order to see the underlying reasons that guide it.

From our point of view, the biggest drawback of the algorithm suggested by Egyed [6] is not in its lack of simplicity (i.e., a large number of rules), but in the necessity for the modeler to perform *seeding* — again, an explicit selection of a list of concepts that are considered as the most valuable in that model. The problem with that approach is that, in order to perform a sensible seeding, one needs to be familiar with the domain and with the conceptual model. But if the conceptual model is large and complex, this requires the modeler to deal with the complexity of the model, thus, risking to defeat of the purpose of an abstraction technique. So, on the one hand, the non-determinism of that approach has an advantage in the ability to generate different abstractions according to one's alternative goals, but on the other hand, it requires an expert and cannot be used for supporting users in getting acquainted with a new domain.

The purpose of this first user-study was two-fold. First, we wanted to understand how conceptual model users abstract from complicated ODCMs and how they perform model seeding. The hypothesis was that by understanding their rationale, we could derive information to (perhaps, partially) automate the seeding process, thus, mitigating the aforementioned problems while preserving some of the advantages of a non-deterministic approach (personalization).

Second, we wanted to preliminary check the hypothesis that the (simpler) abstracted model has a pragmatic value of serving as an explanation of the (complex) original model. We suggested that abstraction could be part of the pragmatic explanation process in the case of ODCMs, in line with what is argued for domain ontologies in [60], as well as with pragmatic aspects to explanations employing ontology-driven conceptual models — *ontological unpacking* [61]. This view of an abstract conceptual model as a type of explanation also fits in with the literature on Design Theory (e.g., [62]). In this community, a conceptual model is taken to be a simplified and useful explanation of how something works from the point of view of an external observer.

Thus, on the one hand, since the abstraction should correspond to the concrete goal, it should be reviewed or even modified in accordance with the given goal. On the other hand, if the already given abstraction contained an error, i.e., a contradiction with the original model, it could pass unnoticed due to over-reliance on the given explanation. In other words, if given, the explanations are interpreted as a signal of competence and are simply accepted regardless of their correctness, especially by non-experts (see [63,64] and experiments in eXplainable AI).

For that, we conducted 5 one hour interviews, and to reflect on this, we used the transcripts of think-aloud and retrospective reports of the participants. We followed the approach suggested in [65], under the assumption that "cognitive processes are not modified by these verbal reports" [65, p. 16].

The experiment was conducted individually and face-to-face with the researcher, using a laptop and a standard well-known UML editor, namely Visual Paradigm. The participants received a pure black-and-white model without any additional notes, also without the OntoUML stereotypes for the classes and associations. The level of expertise was defined as a self-assessment before participation (we also asked our participants some general questions about their experience, familiarity with different modeling languages, etc.), and the study included two experts in ontology-driven conceptual modeling, two experts in conceptual modeling, and one non-expert but an experienced user of conceptual models. A pilot study with one conceptual modeling expert was conducted to ensure the tasks were clear enough and did not raise difficulties.

Each participant was given two tasks to be solved one by one. Both models related to the same domain of a library management system, which was quite general and did not require special knowledge. In the first task, given the ODCM (see Fig. 3), the participant was asked to produce a model abstraction, where the abstraction was defined according to Egyed's algorithm [6]. In order to simplify this process, they were presented with a short narrative telling them why they need to create an abstraction. During the abstracting process, they were asked only to think aloud, without additional comments. After solving the task, they also gave a retrospective reflection on their choices.

In the second task, the participant was asked to change the given abstraction while keeping in mind a concrete goal. The abstraction was produced by the algorithm with some modifications. We introduced a contradiction w.r.t. the original model by making 'Person' and 'Organization' subtypes of 'Client'.[8] In order to make the error even more obvious, we kept some other concepts:

[6] This model (accessible in https://github.com/OntoUML/ontouml-models/tree/master/models/pereira2020ontotrans) has all its classes as *abstract classes*, i.e., classes that cannot be directly instantiated. In particular, all its classes are *role mixins*, i.e., role-like types that can be represented by entities of multiple kinds. As such, it is an atypical conceptual model as it can only be instantiated after being extended with domain-specific notions. In particular, after the kinds of entities playing those role mixins are determined.

[7] http://api.ontouml.org. The documentation: https://github.com/OntoUML/ontouml-server.

[8] For a discussion of why this is an error (in this case, it introduces a logical contradiction in the model), we refer to [30].

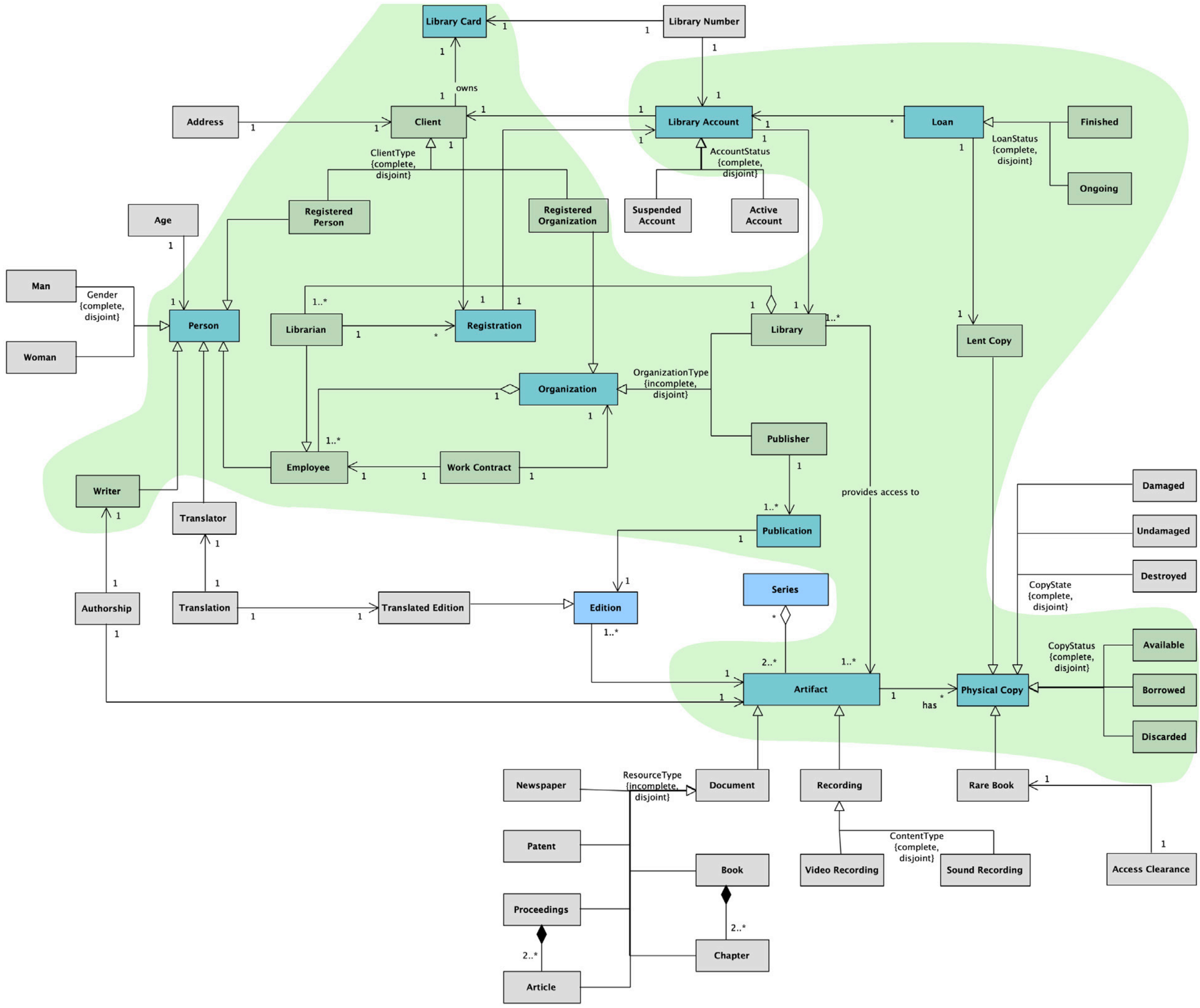


Fig. 3. Library management system model for the first task.

'Librarian', 'Employee', and 'Library' (see Fig. 4 where the error is shown in red). In particular, in this modified abstraction, every 'Librarian' is a 'Client' of the 'Library', which was not true in the original model. According to the narrative, this abstraction was produced by one of the participant's colleagues.

3.3. Experiments with users of ontology-driven conceptual models: structured interviews

After the first study (the results of which are discussed in Section 4) we introduced a threshold for the minimum number of relationships that *aspects* should have in order to stay in the abstraction. This small modification allowed us to check if new models would receive positive feedback (thus, having a better semantic quality), or if we would be suggested to remove some additional concepts. We also wanted to check if some ideas that we received from the participants for further simplifications, e.g., removing cardinality constraints, would be accepted more widely.

Out of the valid pre-selected models (see Section 3.1), we removed those that were anonymous (that left us with 23 models) and those that were too small for abstraction. All authors from the final list of 10 models, namely 26 ontology-driven conceptual modeling experts, received links to the abstractions of their own models and invitations for online structured interviews. The interviews were conducted anonymously, and the questions did not specify which model was being referred to.

Hence, after reviewing an abstraction of the original model that they had published, the authors answered up to 18 questions from the following groups (some examples of questions are provided):

1. Questions about the satisfaction with the abstraction:
   - I understand the abstraction of the original model.

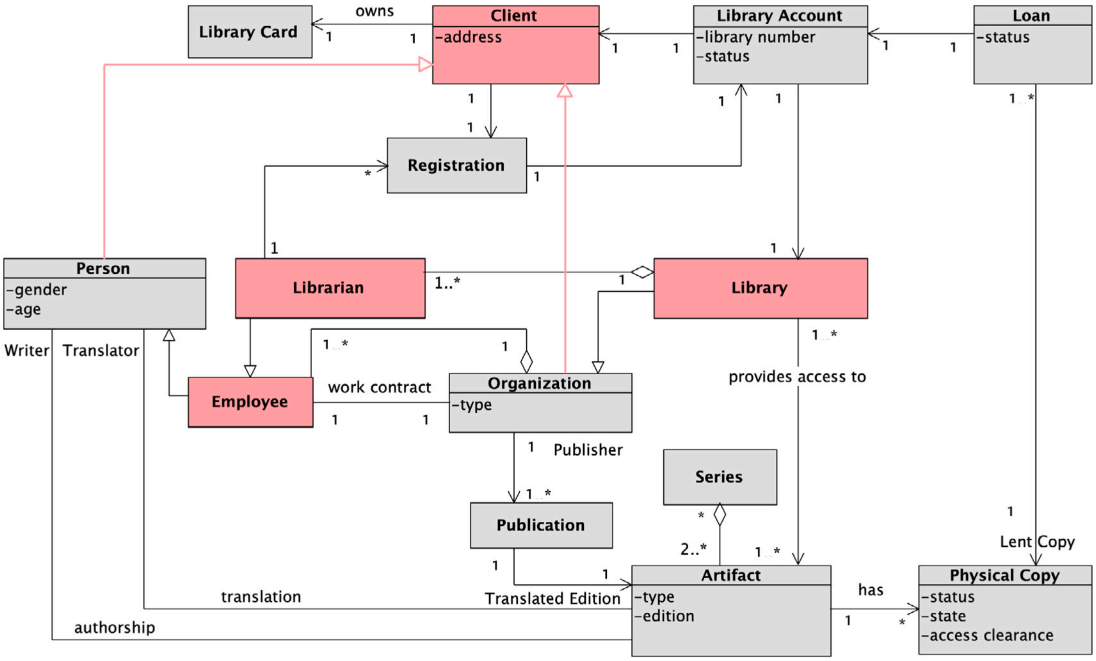


Fig. 4. Corrupted library management system model for the second task.

- The abstraction of the original model has sufficient details.
- The abstraction tells me enough about the domain.
- The abstraction could be useful for a specific goal.
- The abstraction of the original model contains irrelevant details.
- The abstraction could be used for the acquaintance with the domain.
- Why do you think the abstraction could be useful?

2. Questions about the correctness of the abstraction:
   - The abstraction introduced wrong concepts that did not exist in the domain.
   - The abstraction did not introduce any semantically wrong relations.
   - The abstraction reveals some errors that were unintentionally introduced in the original model.
3. Questions about algorithm improvements:
   - Removing cardinalities in the abstraction will make the model clearer.
   - Removing role names in the abstraction will make the model clearer.
4. Questions regarding the abstraction process:
   - I clearly see how the original model was abstracted.
   - Which of the following models you would prefer to see as an intermediate step?

Some questions used a 5-point Likert scale, others were left open. In total, we conducted 7 interviews with an average time of about 40 min (including the time for the abstractions' review).

3.4. Experiments with users of ontology-driven conceptual models: questionnaire

An aspect that became apparent in the second study is that domain experts tend to be biased against the removal of information from their own models, exactly because they know the reason behind each modeling element, they run the risk of amplifying their importance. In a limit case, they would see all modeling elements as essential, forgetting that, as a *lossy* (as opposed to lossless) technique, abstraction necessarily implies the removal of information content from the model. One of the respondents from the previous study formulated this in the following way:

> *"The idea of conceptual models is to represent the complexity of the entities of a domain and their relationships. When applying an algorithm to generate simpler representations, there is a risk of generating an interpretation bias in the reader. Complex problems often require complex solutions."*

However, the model authors are not the only users of the models they create. In fact, if a conceptual model is successful, the model creator will be just one among a multitude of users. Our hypothesis is that other users of the model would have a different attitude towards abstraction in this respect. With that in mind, we developed a questionnaire for a more general audience. The

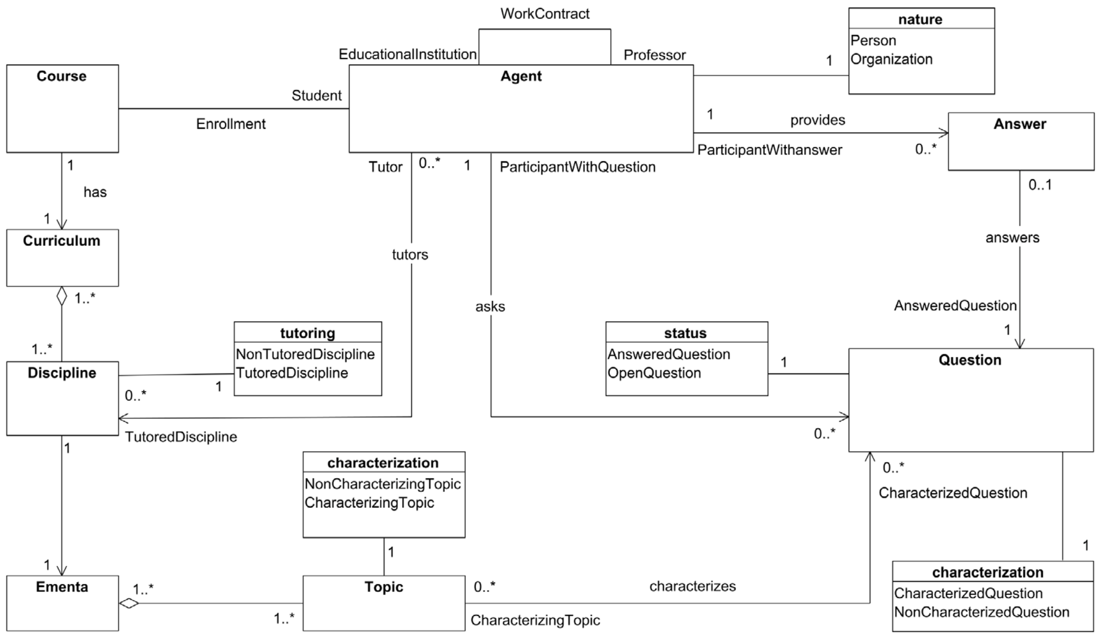


Fig. 5. One of the abstractions used in the questionnaire.

questionnaire was developed based on two anonymous models from the Catalog and did not require any special knowledge except familiarity with UML Class Diagrams. One of the abstractions, which corresponds to the 'online-mentoring' model from the Catalog[9] and was presented in the questionnaire, is shown in Fig. 5.

The questionnaire consisted of 20 questions, which were correlated with the questions of the structured online interviews, and it was partially based on the assessment suggested in [66]. In total, we received 24 responses with an average completion time of 15 min.

3.5. Evaluation of validity threats

During the first experiment with the Catalog, the major threat is in the construct validity. In order to generalize the results of the experiment, it should be properly organized, and, first, we would like to be sure that all of the rules specified in Tables 1–3 were evaluated. However, we selected as many valid models, as we were able to find. Thus, if the rule has never been applied to the given models, the question arises about the expediency of its existence. Probably, the situation it specifies is rare enough, so that it could be removed from the algorithm for the sake of the algorithm's simplicity. Second, sometimes rule application can eliminate the error that already existed in the model. Hence, we would like to test the syntactical correctness after the application of each single rule.

In the first experiment with users, the number of participants was quite small, and the influence of natural variations in human perception and task understanding could be significant. For that reason, the interviewer was always present for clarification during the time of the experiment. Also, a small preliminary example was introduced in order to make sure the interviewees did not have any doubts about the task or the system. Furthermore, since it was a qualitative study, such variation in participants' understanding is inevitable [67], and no statistical conclusions were drawn from this experiment.

The second experiment with users revealed a strong social threat to construct validity. We expected, that since interviewees — as authors of the models — see the importance of abstraction generation in conceptual modeling, they could have been much more favorably disposed towards the algorithm's results. However, authors of the models were biased towards keeping as much information in the model as possible (see Section 3.3). Thus, in order to mitigate this threat the last experiment was introduced.

Last but not least, we acknowledge the internal validity threat in our questionnaire-based evaluation with 24 participants. We took a few active steps to counter this threat. First, we grounded the questionnaire on two different models and randomized our interviewees among them. Second, we also left some questions intentionally open and suppose the results are likely to hold for modelers in general since there is no reason to think that people with different background in modeling would assess the process or the pragmatic quality of abstractions differently.

9 https://github.com/OntoUML/ontouml-models/tree/master/models/online-mentoring.

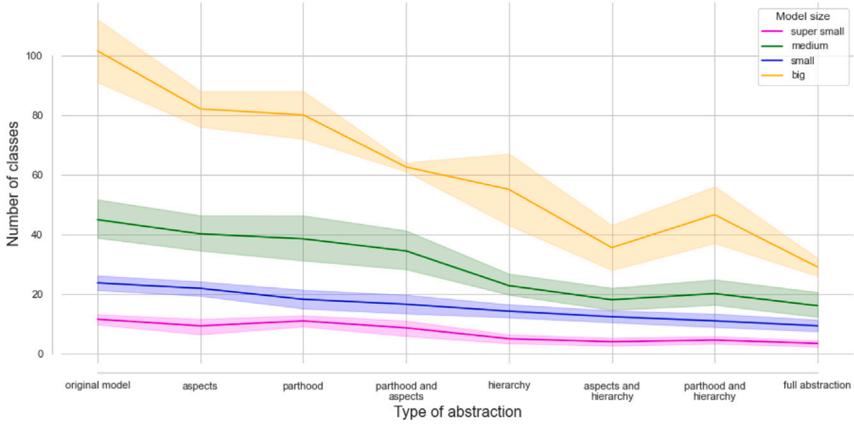


Fig. 6. Compression of the abstracted models: classes.

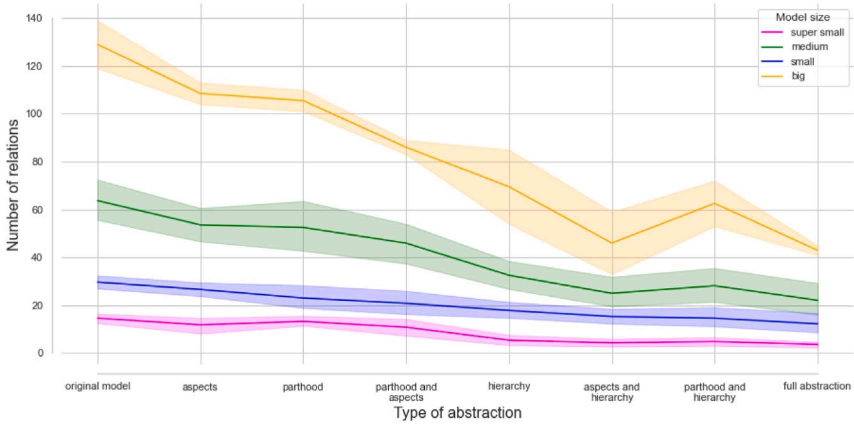


Fig. 7. Compression of the abstracted models: relations.

4. Discussions

4.1. Evaluation of model abstraction: syntactical quality

The first facet of the quality of the abstractions — the correspondence to the OntoUML syntax — was checked automatically with a script. Those 230 models that were generated before, were verified via API on a syntactical validity. None of the abstractions contained any error.

We also report on the results of evaluating the compression rates produced by the algorithm against the set of selected models. The interested reader may compare these with the results published in [7] for the first version of the algorithm.

As it can be seen from Fig. 6 and 7, the algorithm leads to a reduction of the number of concepts as well as the number of relations of about three times for the medium-size models in case of applying all of the proposed rules (the so-called *full abstraction*). The maximum reduction rate happens after removing generalizations relations (abstracting hierarchies of concepts).

Moreover, we checked an assumption, that during the abstraction process — because of the reduction in concepts and relations — some of the syntactical errors could be eliminated. For that, we used the rest of the models that contained only 16 stereotypes that the algorithm could process but had modeling errors. Those 38 ODCMs also were abstracted and checked. The results are presented in Table 4. One can see that the average number of errors is reduced when even one type of abstraction is applied. As expected (due to the high percentage of generalization relations in the models), the most impactful is the reduction from 11.29 in the average number of errors for original models to 3.5 after abstracting hierarchies.

Hence, since the application of several rules may lead to a valid model even if one of the rules introduces an error, for the original 48 models we also tested the syntactical correctness after application of each single rule. In total, 783 models were tested, and none of the abstractions contained any syntactical error.

However, during this process, we realized that, unfortunately, Rules A.2 and H.5 have never been applied, since the selection of the models did not contain many models with *events*. Nevertheless, we decided to keep those rules in the algorithm for future research.

Table 4
Statistics on reduction of syntactical errors after abstracting.

	Original model	Aspects	Parthood	Parthood & aspects	Hierarchy	Aspects & hierarchy	Parthood & hierarchy	Full abstraction
Minimum	1	0	0	0	0	0	0	0
Maximum	72	72	67	67	20	20	20	20
Median	8.00	6.00	6.00	4.50	2.00	2.00	2.00	2.00
Mean	11.29	8.89	9.29	8.05	3.50	3.50	3.47	3.47
Mode	8	2	4	0	0	0	0	0
Standard Deviation	13.01	12.13	11.43	11.56	4.66	4.66	4.67	4.67

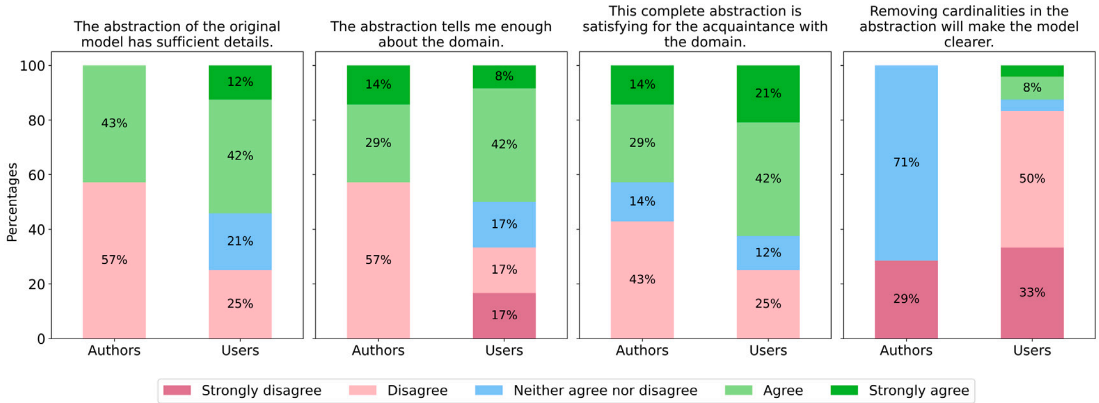


Fig. 8. Comparison of the answers for different groups of users (percentages less than 8 are not specified).

4.2. Evaluation of models abstraction: semantic quality

As mentioned, evaluating the semantic quality of the model is a hard task, since expert disagreement is a norm and users may have different opinions on the same model. An additional complication with abstractions is that these techniques are lossy transformations. Although it was shown that in theory, existing rules do not always lead to theorem-increasing abstractions [55], in practice — mostly because the rules are applied together — they do. Thus, since authors of the models know the rationale behind each element of the model, they are against abstractions that they perceive as 'oversimplifying' the original model.

Our experiments with users were mostly aimed at finding out the hidden rationale behind selecting the most valuable model's concepts. During the first experiment with users, we noticed that the idea of a correlation between the number of relations for a given concept and its significance for the domain is surprisingly well-regarded. Participants preferred highly connected models with a bounded number of concepts, and four interviewees reduced the size of the original model with 52 concepts to less than 20. Those concepts that were selected by most participants are shown in green in Fig. 3. In the same figure, we show in blue the concepts that would have been selected by the algorithm proposed in [8]. Contrasting the latter with the former, we can observe that, on the one hand, 80% of concepts selected by the algorithm are also selected by the aggregated judgement of the experts. On the other hand, the experts selected 3 times more the number of concepts selected by the algorithm, i.e., the latter is much more restrictive than the former.

During the structured interviews, most of the respondents were able to understand the abstraction of the model quite easily. However, opinions about whether the abstraction contains enough details were divided — 3 agreed and 4 disagreed (see Fig. 8). This is even more interesting, taking into account that only two of the subjects claimed that the abstraction contained irrelevant details. In other words, subjects would prefer to have less restrictive abstractions and were unsatisfied with their conciseness — in line with the judgement of experts, as we have seen before.

As for the correctness of the abstraction, we were glad to see an agreement of opinion that the abstraction did not introduce any semantically wrong concepts. The most heated debate happened with the question of whether the abstraction introduced any semantically wrong relations. Five authors did not notice anything wrong, while two others state that semantically wrong relations have been introduced. In one case the author refused to specify which relation is wrong, but the second case requires additional comments. It involves abstracting parthood.

The algorithm in [8] abstracts a parthood relation by transferring certain properties (including relational ones) from the part to the whole. For example, if the Faculty of Computer Science of UNIBZ has a project with the Government, then UNIBZ has a project with the Government or, more precisely, UNIBZ has the property of "having a faculty that has a project with the Government". However, our study showed that when the part-whole relation is established between two parts of the same whole, the result can seem incorrect to model creators (see the pattern in Fig. 9(a) and the abstracted result in Fig. 9(b)). This pattern requires special attention and should be addressed explicitly in the further development of the algorithm.

At the same time, 6 out of 7 authors agreed that abstraction may be used for getting acquainted with the domain.

(a) Original model. (b) Abstracted model.

Fig. 9. One of the abstraction patterns for the parthood relation.

As for the last experiment, again, most of the users (92%) were able to understand the abstraction with ease. For 75% of the respondents, the abstraction had sufficient details (compare this result to the result of the previous study with models authors, see Fig. 8), about 62% agreed that the abstraction provided a good overview of the domain, and 29% agreed to reduce an abstraction even more.

When assessing the quality of the abstractions, these conceptual model users were more tepid: about 55% of them were convinced that the abstractions were correct and that they did not introduce any wrong relations and concepts; about 21% suggested that the abstraction could be wrong.

Unexpected consent was reached for the question of whether we should remove cardinality constraints and role names. Those ideas (suggested during the first interviews) were not accepted. Authors either preferred to stay neutral or disagreed with the removal. Most of the users (83% in both questions) prefer them to be kept in the abstractions as well.

4.3. Evaluation of models abstraction: pragmatic quality

The second task in the first experiment with users was aimed at investigating if the abstraction may serve as an explanation for the original model. We assumed, that this may be reflected in the user's comprehension of the model and, thus, should suffer from the same problems as explanations do, including over-reliance.

Hence, the participants were given an abstraction with the goal of modifying it according to the task to "develop a personal account page for users" (this model is in Fig. 4). Although the participants had an opportunity to see the original model during the whole experiment on paper as well, only one of the interviewees noticed the inconsistency with the original model. All others accepted their "colleague's work", i.e., an abstraction from a sufficiently trusted source, as it was provided, but all made different modifications according to their understanding of the current goal.

In other words, we have noticed the same effect of over-reliance, which was first observed with explanations (see [63,64]). Thus, it is of great importance to have high-quality abstractions — if not done properly, they can lead to comprehension problems with the original models.

We also received interesting feedback on the question of how the participants envisage the abstraction to be useful. Although the question was intentionally left open, some of the answers were recurrent, among them "better communication between stakeholders", and "understanding the original model".

This indicates that ODCM's abstraction could serve as an overview of the original model and could be used for the acquaintance with the domain during the explanation process.

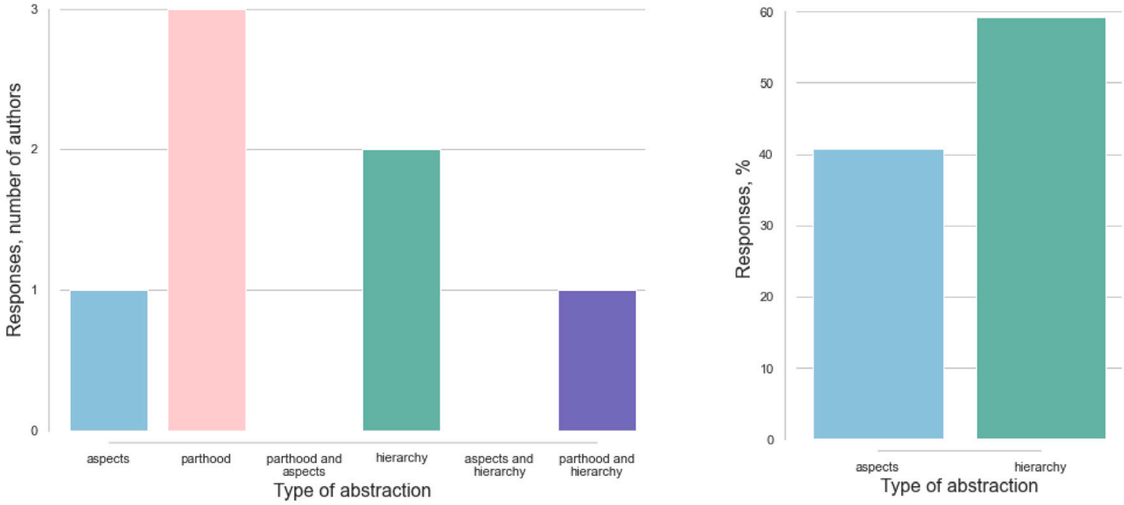
4.4. Evaluation of abstraction process

As discussed before, the evaluation of the abstraction process did not receive much attention in the literature, because of the focus on the final result — the abstracted model. However, the results of the protocol analysis of our first user study were quite interesting.

Four out of five interviewees were regularly distracted by the layout of the model (when during the modifications it was deemed "ugly" and "annoying"). Moreover, during the retrospective reports they reflected on "chopping the things that are unconnected to anything else" (from an ontology-driven conceptual modeling expert), and the need to remove all the cardinality constraints ("Of course they are interesting, but if we talk about simplifying and abstracting, I would do that" — from an expert in conceptual modeling). The last idea, however, did not receive much support in follow-up experiments (see Section 4.2).

In other words, languages for (ontology-driven) conceptual modeling are typically visual languages as well. And because of that, it is impossible to completely isolate the assessment of that model from the assessment of the layout of that model. The idea to remove cardinality constraints also comes from the desire to have a visually simple model without information that is not needed at that precise moment. Thus, we agree that "the notion of simplicity is essential to characterize abstract representations" [49, p. 63].

Although the idea of "chopping" the concepts that have fewer relations was pronounced aloud only once, all other participants were doing the same during the abstraction process. So the idea of Huang et al. to use the PageRank algorithm to determine the seeding has merit because people consider the less connected concepts as less important for the model. Of course, that does not work all the time but can give an approximation of seeding.



(a) In the second experiment. (b) In the third experiment.

Fig. 10. Preferences in intermediate abstractions.

We were also interested if there is any preference in the path that leads to the final abstraction in a broad sense. In other words, if there is any preference from the point of view of users in applying one group of rules before another one. Seven authors who participated in the second user study gave a little preference to models with abstracted parthood relations as an intermediate step (see Fig. 10(a)) However, the number of parthood relations in models is not that big, less than 10% of all relations in ODCMs that were pre-selected. Thus, as intermediate steps in the final experiment with users, only two abstractions were compared: aspects and hierarchy. The results are shown in Fig. 10(b). For example, a preferred intermediate abstraction for the already mentioned 'online-mentoring' model[10] (see Fig. 5) is shown in Fig. 11.

In general, typical users also prefer to abstract hierarchies instead of aspects as intermediate steps of the abstraction process. Our interpretation of these results is that people have a tendency to keep the information contained in the model as long as possible. Due to the nature of the abstraction rules, where generalization relations, when abstracting, are partially transformed in enumerations and parthood relations are sometimes substituted by class properties, parthood and hierarchy abstractions are preferred as intermediate views on the model.

5. Final considerations

The abstraction algorithm suggested in [8] lacked a proper evaluation. We conducted several studies with the purpose of gaining an understanding of what can be further improved and whether the resulting models are of good quality.

The first problem that became visible thanks to the algorithm evaluation over the Catalog is the problem of *excessive compression* (see Fig. 6 and 7 for "super small" models). The algorithm was developed in such a way that it stops only when no rule is applicable anymore. However, for some models, this approach leads to full abstraction with 3 concepts and 2 relations, or even only one concept. To avoid such situations one could, for example, consider a parameter for the algorithm that determines the minimal number of entities (classes) that should be left in the abstraction.

During our experiments with the Catalog we have shown, that our abstraction algorithm [8] provides syntactically valid models. We also have shown that the number of elements in the model correlates with the number of errors in it, so abstracting sometimes eliminates the errors introduced by the modelers.

Huang et al. [39] suggested using the PageRank algorithm as a way to automate the seeding of concepts in the algorithm proposed by Egyed [6]. This has the advantage of dispensing the involvement of an expert in the abstraction task. Using PageRank for this purpose implies selecting highly connected classes as seeds. During the interviews, it became apparent that the idea of preserving classes involved in many relations is indeed a common practice: interviewees tended to remove the classes that were isolated or connected to only one other class more often.

In future work, we intend to investigate the use of topological metrics (e.g., the degree of connectivity of a class) in combination with ontological semantics to improve the algorithm in [8]. For example, classes that would otherwise be eliminated by the algorithm could instead be preserved if they are connected over a certain (absolute or relative) threshold. We observed this, especially in the

[10] Again, the original model can be found in the Catalog, https://github.com/OntoUML/ontouml-models/blob/master/models/online-mentoring/new-diagrams/online-mentoring.png.

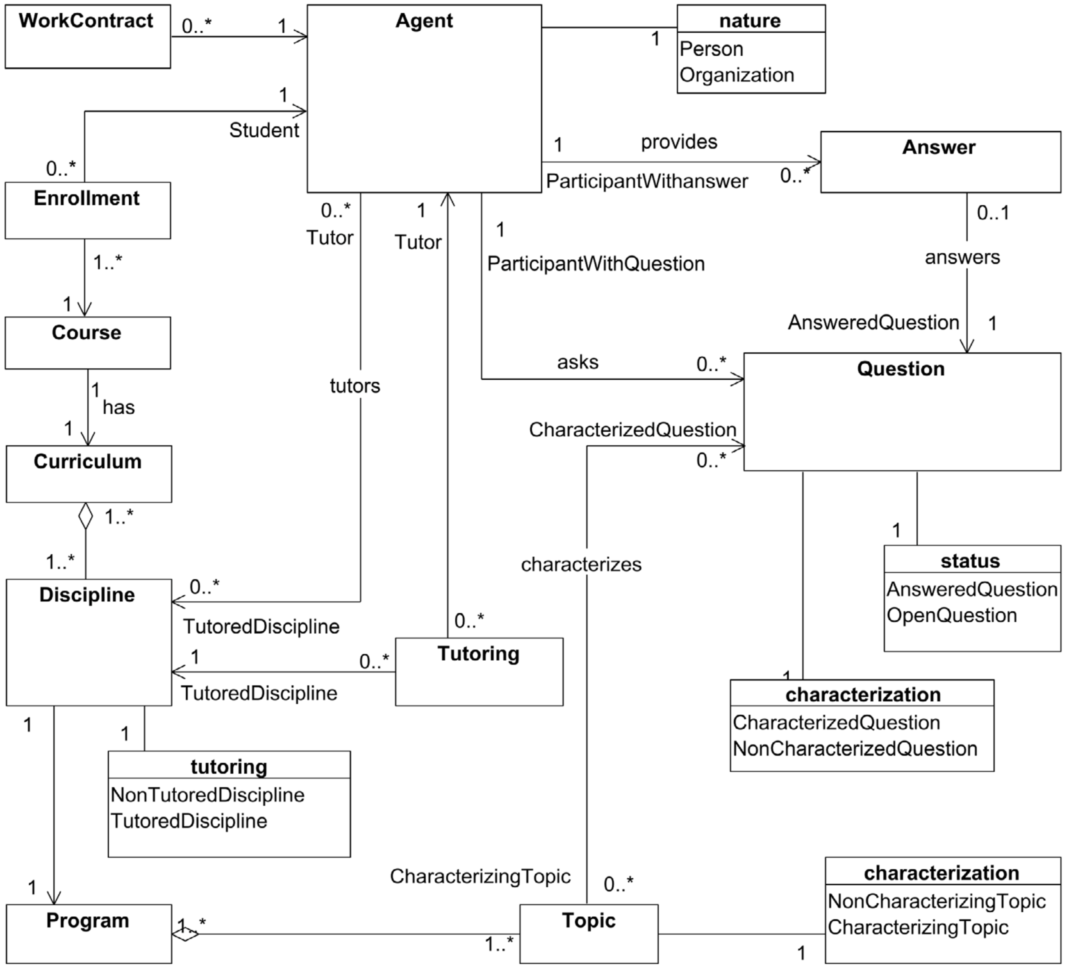


Fig. 11. An intermediate abstraction without hierarchies that was preferred by users.

case of *aspects*, in general, and *relators*, in particular, where the removal of some of these elements led to the model being perceived as incomplete by participants.

Formally, the original transformation rules as proposed in [7] only prescribed the abstractions into material relations of those *relators* that were connected to at most two mediation relations. In both [8,38], this idea was extrapolated to cover *relators* participating in more than two mediation relations. This experiment confirms a tacit rationale behind the original rule: *relators* participating in more than two mediation relations are exactly those cases that would lead to relation reification in traditional conceptual modeling, i.e., one of those cases in which modelers want to perform model expansion — the exact opposite of abstraction.

We hypothesize that the creators of ODCMs and typical users of the same models have different views on the usefulness of the abstraction, and those views must be taken into account when developing an abstraction system. A future study with a larger cohort of subjects would be needed to properly investigate this point.

Before reusing an existing ODCM one needs to understand it. However, since the number of concepts and diagrams may be large, typical users may face problems in familiarizing themselves with an ODCM. We claim that abstraction could be part of the pragmatic explanation process of an ODCM (as well as sub-models for domain ontologies in [60]). In other words, an abstracted model in some cases may serve as an explanation of the original more complex model, and comments from the users received in this study implicitly support this idea.

Also, abstractions, when playing the role of explanations, struggle with the same problems as the latter. They are not taken critically (see results from the first study) and should correspond to the current goal. This means that the results of the deterministic algorithm should be used for the first acquaintance with the domain, but for the further explanation process, the most valuable concepts should be perhaps selected explicitly by the user (with some automated support as discussed here). Also, it is very important to generate a proper abstraction. Otherwise, due to the gap between the abstracted and the original model, a user exposed to the former could have difficulties in understanding the latter.

Finally, recent studies, e.g., [68], announced using large language models in the conceptual modeling domain. While some authors report on "enormous potential" for supporting modeling tasks [68], others remain more skeptical mostly because of the quality of the generated models (see findings in [69]). Moreover, some authors suggest incorporating "structured semantics" to improve the factual correctness of the summarization produced by the large language model [70]. In line with the idea of using abstractions in the explanation process, we also believe that large language models could be used for changing the form of the explanation rather than generating it.

CRediT authorship contribution statement

Elena Romanenko: Writing – review & editing, Writing – original draft, Validation, Software, Data curation. **Diego Calvanese:** Writing – review & editing, Writing – original draft, Validation, Supervision. **Giancarlo Guizzardi:** Writing – review & editing, Writing – original draft, Validation, Supervision.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Diego Calvanese reports financial support was provided by Italian Basic Research. Diego Calvanese reports financial support was provided by Autonomous Province of Bolzano - South Tyrol. Diego Calvanese reports financial support was provided by Knut and Alice Wallenberg Foundation. Elena Romanenko reports a relationship with Free University of Bozen-Bolzano that includes: funding grants. Diego Calvanese reports a relationship with Free University of Bozen-Bolzano that includes: employment. Diego Calvanese reports a relationship with Umeå University that includes: funding grants. Giancarlo Guizzardi reports a relationship with University of Twente that includes: employment. Giancarlo Guizzardi reports a relationship with Stockholm University that includes: employment. co-author is an associate editor for the Data & Knowledge Engineering Journal - G.G. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The data of experiments with users are confidential. The data for experiments with the Catalog are available in the corresponding folder of the project, https://w3id.org/ExpO/github.

Acknowledgments

Empirical studies and user experiments are never possible without the generous voluntary participation of several individuals. The authors would like to express their great appreciation for all the people who spent their time answering the questionnaires and participating in the interviews. The authors also would like to thank Maya Daneva for her valuable comments during the preparation of this manuscript.

This research has been partially supported by the Province of Bolzano and DFG through the project D2G2 (DFG grant n. 500249124), by the HEU project CyclOps (grant agreement n. 101135513), and by the Wallenberg AI, Autonomous Systems and Software Program (WASP), funded by the Knut and Alice Wallenberg Foundation.

References

[1] M. Verdonck, F. Gailly, R. Pergl, G. Guizzardi, B. Franco Martins Souza, O. Pastor, Comparing traditional conceptual modeling with ontology-driven conceptual modeling: An empirical study, Inf. Syst. 81 (2019) 92–103, http://dx.doi.org/10.1016/j.is.2018.11.009.
[2] D. Bork, Conceptual modeling and artificial intelligence: Challenges and opportunities for enterprise engineering, in: Advances in Enterprise Engineering XV, Springer, 2022, pp. 3–9, http://dx.doi.org/10.1007/978-3-031-11520-2_1.
[3] A. Villegas Niño, A Filtering Engine for Large Conceptual Schemas (Ph.D. thesis), Universitat Politècnica de Catalunya, 2013.
[4] J. Guttag, Abstract data types and the development of data structures, Commun. ACM 20 (6) (1977) 396–404.
[5] J. Akoka, I. Comyn-Wattiau, Entity-relationship and object-oriented model automatic clustering, Data Knowl. Eng. 20 (2) (1996) 87–117, http://dx.doi.org/10.1016/S0169-023X(96)00007-9.
[6] A. Egyed, Automated abstraction of class diagrams, ACM Trans. Softw. Eng. Methodol. 11 (4) (2002) 449–491, http://dx.doi.org/10.1145/606612.606616.
[7] G. Guizzardi, G. Figueiredo, M.M. Hedblom, G. Poels, Ontology-based model abstraction, in: Proc. of the 13th Int. Conf. on Research Challenges in Information Science, RCIS, IEEE, 2019, pp. 1–13, http://dx.doi.org/10.1109/RCIS.2019.8876971.
[8] E. Romanenko, D. Calvanese, G. Guizzardi, Abstracting ontology-driven conceptual models: Objects, aspects, events, and their parts, in: Proc. of the 16th Int. Conf. on Research Challenges in Information Science, RCIS, in: LNBIP, vol. 446, Springer, 2022, pp. 372–388, http://dx.doi.org/10.1007/978-3-031-05760-1_22.
[9] E. Romanenko, D. Calvanese, G. Guizzardi, What do users think about abstractions of ontology-driven conceptual models? in: S. Nurcan, A.L. Opdahl, H. Mouratidis, A. Tsohou (Eds.), Research Challenges in Information Science: Information Science and the Connected World, Springer Nature Switzerland, Cham, 2023, pp. 53–68, http://dx.doi.org/10.1007/978-3-031-33080-3_4.
[10] P.P.F. Barcelos, T. Prince Sales, M. Fumagalli, C.M. Fonseca, I. Valle Sousa, E. Romanenko, J. Kritz, G. Guizzardi, A FAIR model catalog for ontology-driven conceptual modeling research, in: Proc. of the 41st Int. Conf. on Conceptual Modeling, ER, in: Lecture Notes in Computer Science, vol. 13607, Springer, 2022, pp. 3–17, http://dx.doi.org/10.1007/978-3-031-17995-2_1.
[11] T.P. Sales, P.P.F. Barcelos, C.M. Fonseca, I.V. Souza, E. Romanenko, C.H. Bernabé, L.O. Bonino da Silva Santos, M. Fumagalli, J. Kritz, J.P.A. Almeida, G. Guizzardi, A FAIR catalog of ontology-driven conceptual models, Data Knowl. Eng. 147 (2023) 102210, http://dx.doi.org/10.1016/j.datak.2023.102210.
[12] M. Verdonck, F. Gailly, Insights on the use and application of ontology and conceptual modeling languages in ontology-driven conceptual modeling, in: Conceptual Modeling, Springer, 2016, pp. 83–97, http://dx.doi.org/10.1007/978-3-319-46397-1_7.
[13] G. Guizzardi, T. Halpin, Ontological foundations for conceptual modelling, Appl. Ontol. 3 (1–2) (2008) 1–12.

[14] D. Calvanese, M. Lenzerini, D. Nardi, Unifying class-based representation formalisms, J. Artificial Intelligence Res. 11 (1999) 199–240, http://dx.doi.org/10.1613/jair.548.

[15] D. Berardi, D. Calvanese, G. De Giacomo, Reasoning on UML class diagrams, Artif. Intell. 168 (1–2) (2005) 70–118.

[16] C.M. Keet, A. Artale, Representing and reasoning over a taxonomy of part–whole relations, Appl. Ontol. 3 (1–2) (2008) 91–110.

[17] S. Cranefield, M.K. Purvis, UML as an ontology modelling language, in: Proc. of the IJCAI 1999 Workshop on Intelligent Information Integration, in: CEUR Workshop Proceedings, vol. 23, CEUR-WS.org, 1999, URL https://ceur-ws.org/Vol-23/cranefield-ijcai99-iii.pdf.

[18] J. Evermann, Y. Wand, Towards ontologically based semantics for UML constructs, in: International Conference on Conceptual Modeling, Springer, 2001, pp. 354–367.

[19] S. Milton, E. Kazmierczak, C. Keen, On the study of data modelling languages using Chisholm's ontology, in: Proc. of Information Modelling and Knowledge Bases XIII, 2002.

[20] G. Guizzardi, On ontology, ontologies, conceptualizations, modeling languages, and (meta) models, in: Selected Papers from the 7th Int. Baltic Conf. on Databases and Information Systems, DB&IS, vol. 155, IOS Press, 2006, pp. 18–39.

[21] G. Guizzardi, Ontology-based evaluation and design of visual conceptual modeling languages, in: Domain Engineering: Product Lines, Languages, and Conceptual Models, Springer, 2013, pp. 317–347.

[22] S. Borgo, A. Galton, O. Kutz, Foundational ontologies in action, Appl. Ontol. 17 (2022) 1–16, http://dx.doi.org/10.3233/AO-220265.

[23] J.N. Otte, J. Beverley, A. Ruttenberg, BFO: Basic formal Ontology, Appl. Ontol. 17 (1) (2022) 17–43, http://dx.doi.org/10.3233/ao-220262.

[24] S. Borgo, R. Ferrario, A. Gangemi, N. Guarino, C. Masolo, D. Porello, E.M. Sanfilippo, L. Vieu, DOLCE: A descriptive ontology for linguistic and cognitive engineering, Appl. Ontol. 17 (1) (2022) 45–69, http://dx.doi.org/10.3233/ao-210259.

[25] F. Loebe, P. Burek, H. Herre, GFO: The general formal ontology, Appl. Ontol. 17 (1) (2022) 71–106, http://dx.doi.org/10.3233/ao-220264.

[26] J.A. Bateman, GUM: The generalized upper model, Appl. Ontol. 17 (1) (2022) 107–141, http://dx.doi.org/10.3233/ao-210258.

[27] M. Grüninger, Y. Ru, J. Thai, TUpper: A top level ontology within standards, Appl. Ontol. 17 (1) (2022) 143–165, http://dx.doi.org/10.3233/ao-220263.

[28] G. Guizzardi, A. Botti Benevides, C.M. Fonseca, D. Porello, J.P.A. Almeida, T. Prince Sales, UFO: Unified foundational ontology, Appl. Ontol. 17 (1) (2022) 167–210, http://dx.doi.org/10.3233/AO-210256.

[29] R. Mizoguchi, S. Borgo, YAMATO: Yet-another more advanced top-level ontology, Appl. Ontol. 17 (1) (2022) 211–232, http://dx.doi.org/10.3233/ao-210257.

[30] G. Guizzardi, Ontological Foundations for Structural Conceptual Models (CITIT PhD.-thesis series 05-74 Telematica Instituut fundamental research series 015), Centre for Telematics and Information Technology, Enschede, 2005.

[31] G. Guizzardi, C.M. Fonseca, J.P.A. Almeida, T.P. Sales, A.B. Benevides, D. Porello, Types and taxonomic structures in conceptual modeling: A novel ontological theory and engineering support, Data Knowl. Eng. 134 (2021) 101891, http://dx.doi.org/10.1016/j.datak.2021.101891.

[32] G. Guizzardi, G. Wagner, J.P.A. Almeida, R.S. Guizzardi, Towards ontological foundations for conceptual modeling: The unified foundational ontology (UFO) story, Appl. Ontol. 10 (3–4) (2015) 259–271.

[33] G. Guizzardi, Logical, ontological and cognitive aspects of object types and cross-world identity with applications to the theory of conceptual spaces, in: Applications of Conceptual Spaces: the Case for Geometric Knowledge Representation, Springer, 2015, pp. 165–186.

[34] G. Guizzardi, T.P. Sales, J.P.A. Almeida, G. Poels, Automated conceptual model clustering: A relator-centric approach, Softw. Syst. Model. 21 (2022) 1363–1387, http://dx.doi.org/10.1007/s10270-021-00919-5.

[35] G. Figueiredo, A. Duchardt, M.M. Hedblom, G. Guizzardi, Breaking into pieces: An ontological approach to conceptual model complexity management, in: Proc. of the 12th Int. Conf. on Research Challenges in Information Science, RCIS, 2018, pp. 1–10, http://dx.doi.org/10.1109/RCIS.2018.8406642.

[36] J. Lozano, J. Carbonera, M. Abel, M. Pimenta, Ontology view extraction: An approach based on ontological meta-properties, in: Proc. of the IEEE 26th Int. Conf. on Tools with Artificial Intelligence, 2014, pp. 122–129, http://dx.doi.org/10.1109/ICTAI.2014.28.

[37] D.L. Moody, A. Flitman, A methodology for clustering entity relationship models — A human information processing approach, in: Proc. of the 18th Int. Conf. on Conceptual Modeling, ER, Springer, 1999, pp. 114–130, http://dx.doi.org/10.1007/3-540-47866-3_8.

[38] G.V.d. Figueiredo, Ontology-Based Complexity Management in Conceptual Modeling (Ph.D. thesis), Federal University of Espírito Santo, 2022.

[39] L. Huang, Y. Duan, Z. Zhou, L. Shao, X. Sun, P.C.K. Hung, Enhancing UML class diagram abstraction with page rank algorithm and relationship abstraction rules, in: Service-Oriented Computing – ICSOC 2016 Workshops: ASOCA, ISyCC, BSCI, and Satellite Events, Revised Selected Papers, in: LNPSE, vol. 10380, Springer, 2016, pp. 103–116, http://dx.doi.org/10.1007/978-3-319-68136-8_10.

[40] H.J. Nelson, G. Poels, M. Genero, M. Piattini, A conceptual modeling quality framework, Softw. Qual. J. 20 (1) (2011) 201–228, http://dx.doi.org/10.1007/s11219-011-9136-9.

[41] O. Lindland, G. Sindre, A. Solvberg, Understanding quality in conceptual modeling, IEEE Softw. 11 (2) (1994) 42–49, http://dx.doi.org/10.1109/52.268955.

[42] Y. Wand, R. Weber, On the ontological expressiveness of information systems analysis and design grammars, Inf. Syst. 3 (4) (1993) 217–237, http://dx.doi.org/10.1111/j.1365-2575.1993.tb00127.x.

[43] Y. Wand, R.Y. Wang, Anchoring data quality dimensions in ontological foundations, Commun. ACM 39 (11) (1996) 86–95, http://dx.doi.org/10.1145/240455.240479.

[44] E. Falkenberg, W. Hesse, P. Lindgreen, B. Nilsson, J. Han Oei, C. Rolland, R. Stamper, F. van Assche, A. Verrijn-Stuart, K. Voss, FRISCO: A Framework of Information System Concepts : The FRISCO report (WEB edition), International Federation for Information Processing (IFIP), 1998, URL https://research.utwente.nl/files/5157230/frisco-full.pdf.

[45] J. Krogstie, Quality of models, in: Model-Based Development and Evolution of Information Systems, Springer, London, 2012, pp. 205–247, http://dx.doi.org/10.1007/978-1-4471-2936-3_4.

[46] G. Guizzardi, H.A. Proper, On understanding the value of domain modeling, in: Proc. of the Int. Workshop on Value Modelling and Business Ontologies, in: CEUR Workshop Proceedings, vol. 2835, CEUR-WS.org, 2021, pp. 51–62, URL https://ceur-ws.org/Vol-2835/paper6.pdf.

[47] I.V. Sousa, T.P. Sales, E. Guerra, L.O.B. da Silva Santos, G. Guizzardi, What do I get from modeling? – An empirical study on using structural conceptual models, in: Proc. of the 27th Int. Conf. on Enterprise Design, Operations, and Computing, EDOC, in: Lecture Notes in Computer Science, vol. 14367, Springer, 2023, pp. 21–38, http://dx.doi.org/10.1007/978-3-031-46587-1_2.

[48] Y. Qian, I. Choi, Tracing the essence: Ways to develop abstraction in computational thinking, Educ. Technol. Res. Develop. 71 (3) (2022) 1055–1078, http://dx.doi.org/10.1007/s11423-022-10182-0.

[49] L. Saitta, J.-D. Zucker, Abstraction in Artificial Intelligence and Complex Systems, Springer, 2013, http://dx.doi.org/10.1007/978-1-4614-7052-6.

[50] J.R. Hobbs, Granularity, in: Proc. of the 9th Int. Joint Conf. on Artificial Intelligence, IJCAI, vol. 1, Morgan Kaufmann Publishers Inc., 1985, pp. 432–435.

[51] D.A. Plaisted, Theorem proving with abstraction, Artif. Intell. 16 (1) (1981) 47–108, http://dx.doi.org/10.1016/0004-3702(81)90015-1.

[52] J.D. Tenenberg, Preserving consistency across abstraction mappings, in: Proc. of the 10th Int. Joint Conf. on Artificial Intelligence, IJCAI, 1987, pp. 1011–1014, URL http://ijcai.org/Proceedings/87-2/Papers/090.pdf.

[53] P.P. Nayak, A.Y. Levy, A semantic theory of abstractions, in: Proc. of the 14th Int. Joint Conf. on Artificial Intelligence, IJCAI, vol. 1, Morgan Kaufmann Publishers Inc., 1995, pp. 196–202.

[54] F. Giunchiglia, T. Walsh, A theory of abstraction, Artif. Intell. 57 (2) (1992) 323–389.

[55] E. Romanenko, O. Kutz, D. Calvanese, G. Guizzardi, Towards semantics for abstractions in ontology-driven conceptual modeling, in: Proc. of the ER 2023 Workshops – 9th Int. Workshop on Ontologies and Conceptual Modeling, OntoCom, in: Lecture Notes in Computer Science, vol. 14319, Springer, 2023, pp. 199–209, http://dx.doi.org/10.1007/978-3-031-47112-4_19.

[56] C. Ghidini, F. Giunchiglia, A Semantics for Abstraction, Technical Report DIT-03-082, University of Trento, 2003.

[57] F. Giunchiglia, S. Kleanthous, J. Otterbacher, T. Draws, Transparency paths - documenting the diversity of user perceptions, in: Adjunct Proceedings of the 29th ACM Conference on User Modeling, Adaptation and Personalization, Association for Computing Machinery, 2021, pp. 415–420, http://dx.doi.org/10.1145/3450614.3463292.

[58] T. Miller, Explanation in artificial intelligence: Insights from the social sciences, Artificial Intelligence 267 (2019) 1–38, http://dx.doi.org/10.1016/j.artint.2018.07.007.

[59] A. Adams, A.L. Cox, Questionnaires, in-depth interviews and focus groups, in: Research Methods for Human-Computer Interaction, Cambridge University Press, 2008, pp. 17–34, http://dx.doi.org/10.1017/CBO9780511814570.003.

[60] E. Romanenko, D. Calvanese, G. Guizzardi, Towards pragmatic explanations for domain ontologies, in: Proc. of the 23rd Int. Conf. on Knowledge Engineering and Knowledge Management, EKAW, in: LNAI, vol. 13514, Springer, Cham, 2022, pp. 201–208, http://dx.doi.org/10.1007/978-3-031-17105-5_15.

[61] G. Guizzardi, N. Guarino, Explanation, semantics, and ontology, Data Knowl. Eng. (2024) 102325, http://dx.doi.org/10.1016/j.datak.2024.102325.

[62] D. Norman, The Design of Everyday Things: Revised and Expanded Edition, Basic Books, 2013.

[63] G. Bansal, T. Wu, J. Zhou, R. Fok, B. Nushi, E. Kamar, M.T. Ribeiro, D. Weld, Does the whole exceed its parts? The effect of AI explanations on complementary team performance, in: Proc. of the 2021 CHI Conf. on Human Factors in Computing Systems, CHI, Association for Computing Machinery, 2021, http://dx.doi.org/10.1145/3411764.3445717.

[64] Z. Buçinca, M.B. Malaya, K.Z. Gajos, To trust or to think: Cognitive forcing functions can reduce overreliance on AI in AI-assisted decision-making, Proc. ACM Hum.-Comput. Interact. 5 (CSCW1) (2021) http://dx.doi.org/10.1145/3449287.

[65] K.A. Ericsson, H.A. Simon, Protocol Analysis, The MIT Press, 1993.

[66] R.R. Hoffman, S.T. Mueller, G. Klein, J. Litman, Metrics for Explainable AI: Challenges and Prospects, CoRR Technical Report, arXiv.org e-Print archive, 2018, arXiv:1812.04608.

[67] N. King, C. Horrocks, J. Brooks, Interviews in Qualitative Research, second ed., Sage Publications Ltd, United Kingdom, 2019.

[68] H.-G. Fill, P. Fettke, J. Köpke, Conceptual modeling and large language models: Impressions from first experiments with ChatGPT, Enterp. Model. Inform. Syst. Archit. (EMISAJ) 18 (2023) http://dx.doi.org/10.18417/EMISA.18.3.

[69] J. Cámara, J. Troya, L. Burgueño, A. Vallecillo, On the assessment of generative AI in modeling tasks: an experience report with ChatGPT and UML, Softw. Syst. Model. 22 (3) (2023) 781–793, http://dx.doi.org/10.1007/s10270-023-01105-5.

[70] T. Chen, X. Wang, T. Yue, X. Bai, C.X. Le, W. Wang, Enhancing abstractive summarization with extracted knowledge graphs and multi-source transformers, Appl. Sci. 13 (13) (2023) 7753, http://dx.doi.org/10.3390/app13137753.

Elena Romanenko is a Ph.D. candidate at Free University of Bozen-Bolzano, Italy, and a member of the Research Centre for Knowledge and Data (KRDB). She holds a joint master's degree in Computational Science from Perm State University, Russia, and the University of Reading, the UK. Her current research interests lie in pragmatic explanations of ODCMs and ontologies.

Diego Calvanese is a Full Professor at the Research Centre for Knowledge and Data (KRDB) at the Faculty of Engineering of Free University of Bozen-Bolzano (Italy), and Wallenberg guest professor in AI for Data Management at Umeå University (Sweden). His research interests include knowledge representation and reasoning, virtual knowledge graphs, ontology languages, description logics, conceptual data modeling, and data integration. He is one of the editors of the Description Logic Handbook. He is a fellow of EurAI and a fellow of ACM. He is the originator and a co-founder of Ontopic, a startup whose mission is to bring the VKG technology to industry.

Giancarlo Guizzardi is a Full Professor of Software Science and Evolution as well as Chair and Department Head of Semantics, Cybersecurity & Services (SCS) at the University of Twente, The Netherlands He is also an Affiliated Full Professor at the Department of Computer and Systems Sciences (DSV) at Stockholm University, in Sweden. He has been active for more than two decades in the areas of Applied Ontology, Conceptual Modeling, Business Informatics, and Information Systems Engineering, working with a multi-disciplinary approach that combines results from Formal Ontology, Cognitive Science, Logics and Linguistics. He is currently an associate editor for the Applied Ontology journal and for Data & Knowledge Engineering, a co-editor of the Lecture Notes in Business Information Processing series, and a member of a number of international journal editorial boards. Finally, he is a member of the Steering Committees of ER, EDOC, and IEEE CBI, and of the Advisory Board of the International Association for Ontology and its Applications (IAOA).