What Do Users Think About Abstractions of Ontology-Driven Conceptual Models?

Elena Romanenko[1(✉)], Diego Calvanese[1,2], and Giancarlo Guizzardi[3]

[1] Free University of Bozen-Bolzano, 39100 Bolzano, Italy
{eromanenko,diego.calvanese}@unibz.it
[2] Umeå University, 90187 Umeå, Sweden
[3] University of Twente, 7500 Enschede, The Netherlands
g.guizzardi@utwente.nl

Abstract. In a previous paper, we proposed an algorithm for ontology-driven conceptual model abstractions [18]. We have implemented and tested this algorithm over a FAIR Catalog of such models represented in the OntoUML language. This provided evidence for the *correctness* of the algorithm's implementation, i.e., that it correctly implements the model transformation rules prescribed by the algorithm, and its *effectiveness*, i.e., it is able to achieve high compression (summarization) rates over these models. However, in addition to these properties, it is fundamental to test the *validity* of this algorithm, i.e., that it achieves what it is intended to do, namely provide summarizing abstractions over these models whilst preserving the gist of the conceptualization being represented. 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 user studies and reflects on how they can be exploited to improve the existing algorithm.

Keywords: Conceptual Model Abstraction · Ontology-Driven Conceptual Models · User Study

1 Introduction

The complexity of an (ontology-driven) conceptual model highly correlates with the complexity of the domain and software for which it is designed. According to Guttag [14], one way to reduce complexity is through the abstraction process. Speaking of conceptual modeling, Egyed [7] defined abstraction as "a process that transforms lower-level elements into higher-level elements containing fewer details on a larger granularity". The main idea is to provide the user with a bird's-eye view of the model by filtering out some details.

Our previously suggested algorithm for ontology-driven conceptual model abstraction [18] was implemented and tested over the models from a recently created FAIR catalog for ontology-driven conceptual modeling research [4]. Although the algorithm has shown that it is applicable to a wide range of

S. Nurcan et al. (Eds.): RCIS 2023, LNBIP 476, pp. 53–68, 2023.
https://doi.org/10.1007/978-3-031-33080-3_4

models, the question of the quality of the resulting abstractions was still open. In order to answer it, we conducted three user studies. The results of the studies and our suggestions for algorithm improvement are presented in this paper.

The remainder of the paper is organized as follows: Sect. 2 presents our baseline and background; Sect. 3 describes user studies conducted to find out the ways for improvement of the approach; Sect. 4 elaborates on final considerations and future work.

2 Background

By a conceptual model, one could equally mean a UML Class Diagram as well as a Business Process Model. This is because conceptual models are high-level abstractions used to capture information about the domain and both these languages (among many others) are employed for that. Ontology-driven conceptual models (ODCMs) are usually considered a special class of conceptual models that utilize foundational ontologies to ground modeling elements, modeling languages, and tools [22].

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" [11].

However, the complexity of the conceptual model correlates with the complexity of the domain. This sometimes leads to situations where the number of concepts and sub-diagrams goes far beyond the cognitive tractability threshold of those people who are supposed to work with those diagrams. Thus, despite the fact that conceptual models are developed for human communication, one of the most challenging problems is "to understand, comprehend, and work with very large conceptual schemas" [23].

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. For analysis and classification of the existing approaches, one could refer to [23]. Here, it suffices to point out that producing a meaningful but reduced version of the original conceptual model via filtering out the details and keeping the most important notions—also known as *summarizing* or *abstracting*—is one of the most challenging tasks.

Most of the methods for conceptual model summarization are based on classic modeling notations (UML, ER) [23, p.44] and rely on syntactic properties of the model, such as closeness or different types of distances between model elements (see [2]), while in case of ODCMs, there is the possibility to leverage their built-in ontological semantics. The first version of an abstraction algorithm leveraging foundational ontological semantics was introduced in [10], followed by an enhanced version in [18], 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 [10,18]. For the scope of this paper, it is enough to highlight that both algorithms bear a remarkable simplicity in the number of rules, are deterministic and do not require human intervention or *seeding*[1] (as opposed to competing algorithms, e.g., [7]), are computationally efficient and scalable, thus, able to process very large models in a timely manner.

The newest version of the algorithm proposed in [18] 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 (*relators*, *qualities*, and *modes* in terms of Unified Foundational Ontology [13])[2], and *(3)* hierarchies of concepts (generalization relations). Also, one should note that application of the 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 be kept. It is possible to apply them in a compositional way, so we can abstract both parthood relations and hierarchies. Thus, with three groups of rules, one can receive eight possible models (including the original model and full abstraction, when all rules are applied) [18].

The defined graph-rewriting rules utilised the ontological semantics of UFO. However, since ODCMs in general are not bound to any specific foundational ontology, one can choose the most appropriate to the task at hand. A recent special issue of Applied Ontology journal [5] describes seven of them—BFO, DOLCE, GFO, GUM, TUpper, UFO, and YAMATO. UFO appeared to be the most fruitful for the abstraction algorithm development because of the existence of the UFO-based OntoUML language and corresponding tools. For an in-depth discussion, philosophical justification, and formal characterization of UFO and OntoUML, we refer to [9,12].

The suggested algorithms needed to be properly evaluated from two angles. Although an initial attempt to calculate the compression rate was done in [10], there was a need to assess how much information is reduced on a larger sample of models. Also, we wanted to investigate whether the suggested algorithm provided reasonably good results from the modelers' point of view. Since the rest of the paper is devoted to the latter problem, let us briefly describe the compression results that were obtained.

The ability to assess the algorithm over models becomes possible with the creation of a FAIR model catalog[3] for ontology-driven conceptual modeling research [4] (hereinafter referred to as the Catalog). The Catalog offers a diverse collection of conceptual models, created by modelers with varying modeling

[1] Seeding is the (typically, manual) pre-selection of certain model elements that need to be maintained in the final abstraction.

[2] A characteristic feature of *aspects* in UFO is that they are existentially dependent from the main entities. For example, *quality* `Colour` cannot exist without the object itself. Also, *relator* `Employment` is not possible without `Employer` and `Employee`.

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

skills, for a wide range of domains, with different purposes, and currently consists of 135 models.

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. From the point of view of the Catalog, the decision to keep models as they were created was reasonable, because one of the purposes was an empirical discovery of modeling (anti-)patterns [4]. Unfortunately, this contradicts the goal to assess the quality of the algorithm, since most of the time the original errors would be propagated to the abstraction.

Taking into account the above-mentioned conditions, we selected 41 models for the purpose of the algorithm evaluation. The selected models satisfied the following criteria: *(1)* they contained only those 16 stereotypes, for which the second version of the algorithm was developed (that left us with 71 models out of the original number), and *(2)* they did not contain syntactical modeling errors that could not be easily fixed. The syntactical correctness of models was checked automatically with the OntoUML plugin[4] for Visual Paradigm[5].

A fuller analysis of the algorithm (e.g., in terms of computational complexity) is out of the scope of this paper, so here we report only 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 [10] for the first version of the algorithm.

As it can be seen from Fig. 1, 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).

Despite achieving large compression rates, these numbers by themselves tell us nothing about the appropriateness of the abstracted models, i.e., to what degree the abstractions are perceived as useful and meaningful by modelers. An initial attempt to compare the abstraction results of the first version of the algorithm to other existing methods was made in [21]. The main hypothesis of the experiment was that the abstraction algorithm (the first version proposed in [10]) produces models capturing the gist of the original model more appropriately than the competing algorithms proposed in [7] and [16].

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 several abstractions. They were asked to rate the models according to their view on the quality of the abstraction and justify their choice.

The suggested algorithm was clearly preferred by practitioners with large modeling experience. However, overall, the experiment did not demonstrate a significant preference for one of the tested abstraction algorithms. This result

[4] https://github.com/OntoUML/ontouml-vp-plugin.

[5] https://www.visual-paradigm.com.

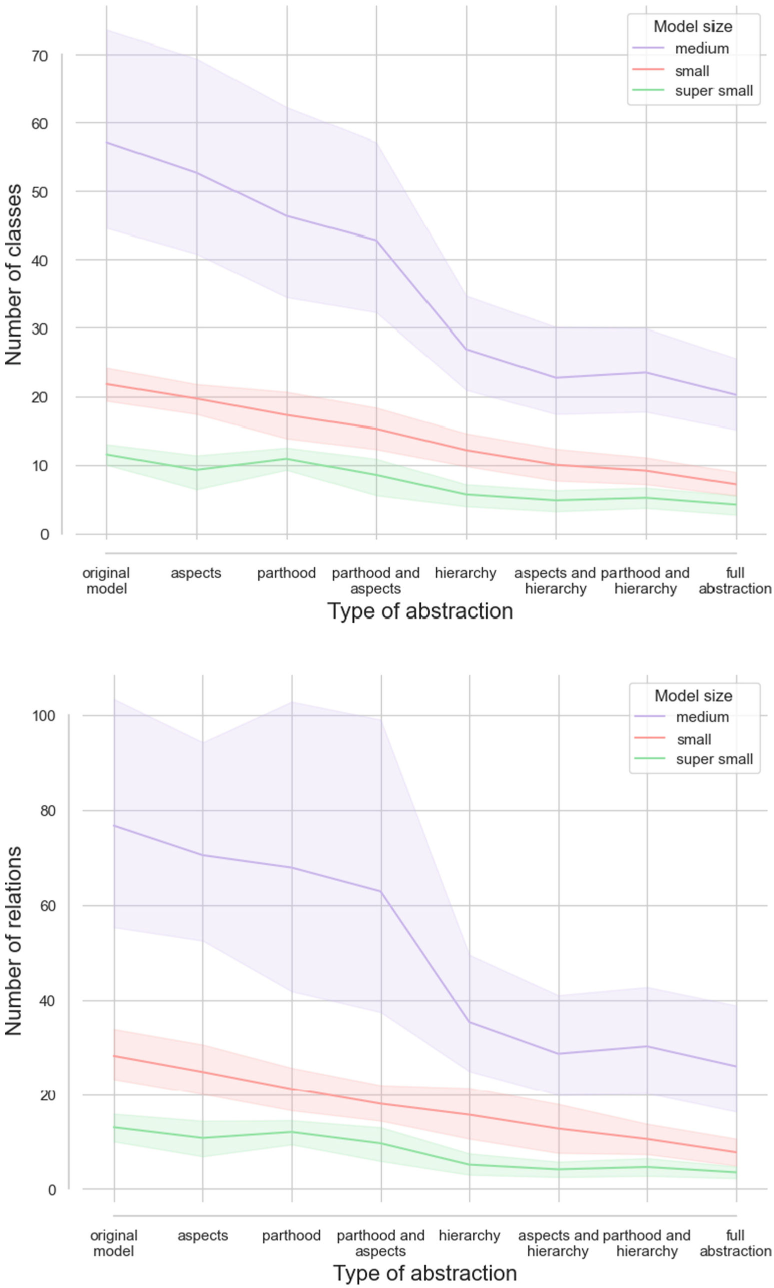

Fig. 1. Compression of the abstracted models.

is not negative per se, given that [10] 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. These assumptions are discussed in the next section.

3 Empirical Studies in Conceptual Modeling Abstraction

As mentioned before, the original abstraction algorithm and its refined version were developed with two assumptions, which influence was not fully apparent until the first questionnaire was filled out in [21]. 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 colour of the car while talking about car rental, but the *relator* `Car Rental Agreement`, although being dependent on other entities, 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. According to UFO [13], *kind* is a type that applies necessarily to its instances and provides them with the identity criteria. Examples typically include `Person`, `Organization`. 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 it 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 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 improve the existing algorithm, following the triangulation approach [1], we conducted the following experiments: *(1)* in-person interviews about the abstraction process, *(2)* online structured interviews with modelers, and *(3)* online questionnaire with conceptual model users. The purpose of the first study was to improve the existing algorithm taking into account the rationale behind the abstraction process when done manually. The following studies were aimed to check those preliminary ideas on two groups of users: models' authors and general users of ODCMs.

3.1 What Do People Think When Abstracting a Conceptual Model?

The results of the questionnaire in [21] leave unexplained the reason modelers prefer one abstraction over another. Our first experiment 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 goal of the study was to answer the following questions: *(1)* What is the rationale for choosing the most valuable concepts of the model (seeding)? *(2)* How does it change depending on the given goal? *(3)* Can abstraction serve as an explanation of the original model?

From our point of view, the major drawback of the algorithm suggested by Egyed [7] 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 determine those concepts, 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 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 preliminarily check the hypothesis that the (simpler) abstracted model is perceived 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 [19][6]. 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 overreliance 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 [3,6] 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[7].

[6] This view of an abstract conceptual model as a type of explanation is in line with the literature on Design Theory (e.g., [17]). 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. We come back to this idea of an abstracted model as a sort of pragmatic explanation and its grounds in Sect. 4.

[7] 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).

We followed the approach suggested in [8], under the assumption that "cognitive processes are not modified by these verbal reports" [8, 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, 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 assure 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. 2), the participant was asked to produce a model abstraction, where the abstraction was defined according to Egyed's algorithm [7]. 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: `Librarian`, `Employee`, and `Library` (see modifications in pink in Fig. 2). 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.

The results of the protocol analysis 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).

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

[8] For a discussion for why this is an error (in this case, it introduces a logical contradiction in the model), we refer to [9].

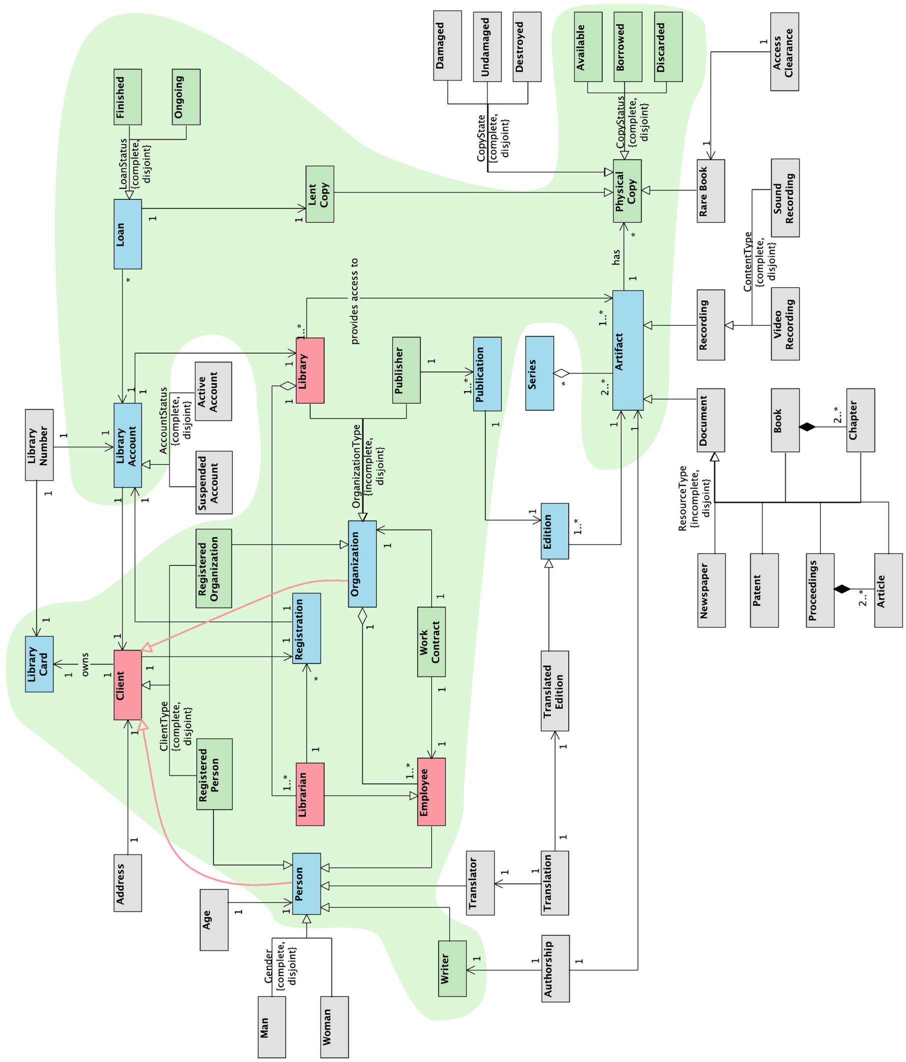

Fig. 2. Library management system model for the first user study. The sub-model selected as the abstraction by most of the participants is in green. The concepts selected by the abstraction algorithm are in blue. In pink, we have an error that was introduced for the second part of the experiment. (Color figure online)

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" [20, p.63].

We also noticed that the idea of a correlation between the number of relations for a given concept and its significance for the domain is surprisingly wellregarded. 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. 2. In the same figure, we show in blue the concepts that would have been selected by the algorithm proposed in [18]. 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. We will come back to this point later in the paper.

As for the second part of the experiment, the participants were given an abstraction with the goal to modify it according to the task to "develop a personal account page for users" (this model is shown in blue and pink in Fig. 2). Although the participants had an opportunity to have a look at the original model during the whole experiment on paper as well, only one of the interviewees did notice 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.

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.

3.2 Interviews with Models' Authors

After the first study, 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, 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 41 originally pre-selected models (see Sect. 2), 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.

After reviewing an abstraction of the original model that was published by them, the authors answered up to 18 questions from three groups (some examples of questions are provided):

1. Questions about the satisfaction with the abstraction:
   - I understand the abstraction of the original model.
   - The abstraction of the original model has sufficient details.
   - The abstraction tells me enough about the domain.
   - The abstraction of the original model contains irrelevant details.
   - The abstraction could be used for the acquaintance with the domain.

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.

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).

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. 3). This is even more interesting, taking into account that only two of the authors claimed that the abstraction contained irrelevant details. In other words, authors 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.

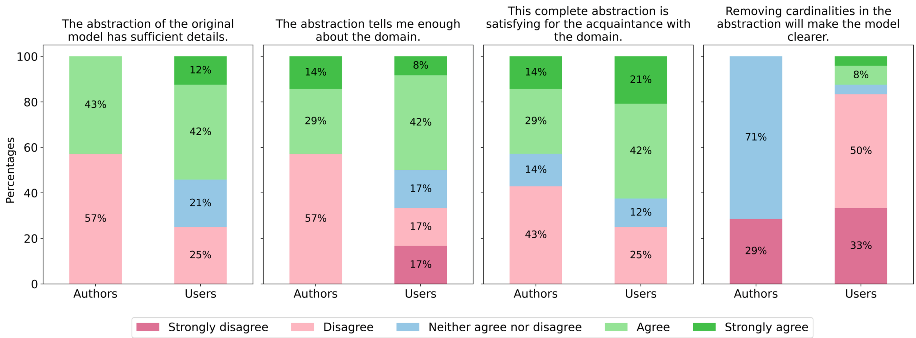

Fig. 3. Comparison of the answers for different groups of users. Percentages less than 8 are not specified.

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. 5 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 [18] 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. 4a and the abstracted result in Fig. 4b. This pattern requires special attention and should be addressed explicitly in the further development of the algorithm.

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

Fig. 4. One of the abstraction patterns for the parthood relation.

Ideas to remove cardinality constraints or role names were not accepted. Authors either preferred to stay neutral or disagreed with the removal. At the same time, 6 out of 7 authors agreed that abstraction may be used for getting acquainted with the domain.

3.3 Questionnaire for Conceptual Models' Users

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 creators 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 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 `bank-model` from the Catalog[9] and was presented in the questionnaire, is shown in Fig. 5.

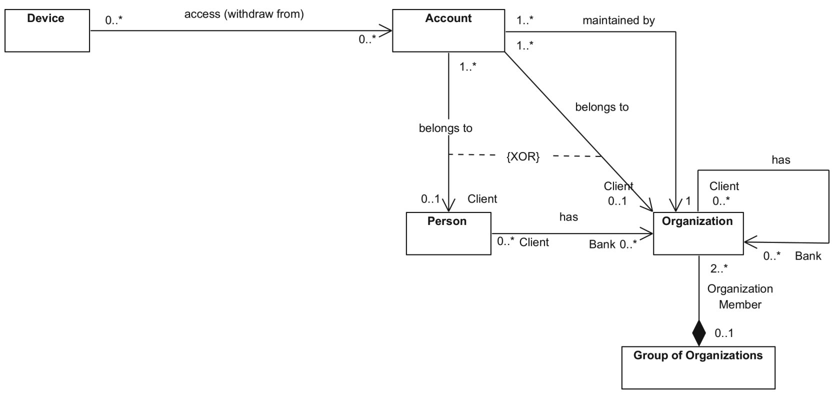

Fig. 5. One of the abstractions used in the questionnaire.

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 [15]. In total, we received 24 responses with an average completion time of 15 min.

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, see Fig. 3), 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: most of the users (83% in both questions) prefer them to be kept in the abstraction.

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

[9] https://github.com/OntoUML/ontouml-models/tree/master/models/bank-model.

4 Final Considerations

The abstraction algorithm suggested in [18] lacked a proper evaluation. We conducted several studies with the purpose to gain an understanding of what can be further improved and whether the resulting models are able to keep the semantics of the original model.

The first problem that became visible thanks to the algorithm evaluation over the Catalog is the problem of *excessive compression* (see Fig. 1 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. For avoiding 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.

Huang et al. [16] suggested using the PageRank algorithm as a way to automate the seeding of concepts in the algorithm proposed by Egyed [7]. 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 [18]. 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 case of *aspects*, in general, and *relators*, in particular (see Sect. 3.1), where the removal of some of these elements led to the model being perceived as incomplete by participants[10].

Unfortunately, the number of participants in the last studies was not very large. However, we suppose that creators of the models and typical users of the same ODCMs have different views on the usefulness of the abstraction, and those views must be taken into account when developing the final system.

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

[10] Formally, the original transformation rules as proposed in [10] only prescribed the abstractions into material relations of those *relators* that were connected to at most two mediation relations. In both [21] and [18], 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.

for domain ontologies in [19]). 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 the last study implicitly support this idea.

Moreover, abstractions, when playing the role of explanations, struggle with the same problems. 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. 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 with understanding the latter.

Acknowledgements. 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. This research has been partially supported by the Italian Basic Research (PRIN) project HOPE, and by the Province of Bolzano through the project MENS. Diego Calvanese is also supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP), funded by the Knut and Alice Wallenberg Foundation.

References

1. Adams, A., Cox, A.L.: Questionnaires, in-depth interviews and focus groups, pp. 17–34. Cambridge University Press, Cambridge (2008). https://doi.org/10.1017/CBO9780511814570.003
2. Akoka, J., Comyn-Wattiau, I.: Entity-relationship and object-oriented model automatic clustering. Data Knowl. Eng. **20**(2), 87–117 (1996). https://doi.org/10.1016/S0169-023X(96)00007-9
3. Bansal, G., et al.: Does the whole exceed its parts? The effect of AI explanations on complementary team performance. In: Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems (CHI). Association for Computing Machinery (2021). https://doi.org/10.1145/3411764.3445717
4. Barcelos, P.P.F., et al.: A FAIR model catalog for ontology-driven conceptual modeling research. In: Ralyté, J., Chakravarthy, S., Mohania, M., Jeusfeld, M.A., Karlapalem, K. (eds.) ER 2022. LNCS, vol. 13607, pp. 3–17. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-17995-2_1
5. Borgo, S., Galton, A., Kutz, O.: Foundational ontologies in action. Appl. Ontol. **17**, 1–16 (2022). https://doi.org/10.3233/AO-220265
6. Buçinca, Z., Malaya, M.B., Gajos, K.Z.: 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). https://doi.org/10.1145/3449287
7. Egyed, A.: Automated abstraction of class diagrams. ACM Trans. Softw. Eng. Methodol. **11**(4), 449–491 (2002). https://doi.org/10.1145/606612.606616
8. Ericsson, K.A., Simon, H.A.: Protocol Analysis. The MIT Press, Cambridge (1993)
9. Guizzardi, G.: Ontological foundations for structural conceptual models. CTIT PhD.-thesis series 05-74 Telematica Instituut fundamental research series 015, Centre for Telematics and Information Technology, Enschede (2005)

10. Guizzardi, G., Figueiredo, G., Hedblom, M.M., Poels, G.: Ontology-based model abstraction. In: Proceedings of the 13th International Conference on Research Challenges in Information Science (RCIS), pp. 1–13. IEEE (2019). https://doi.org/10.1109/RCIS.2019.8876971
11. Guizzardi, G., Sales, T.P., Almeida, J.P.A., Poels, G.: Automated conceptual model clustering: a relator-centric approach. Softw. Syst. Model. 1–25 (2021). https://doi.org/10.1007/s10270-021-00919-5
12. Guizzardi, G., et al.: Types and taxonomic structures in conceptual modeling: a novel ontological theory and engineering support. Data Knowl. Eng. **134**, 101891 (2021). https://doi.org/10.1016/j.datak.2021.101891
13. Guizzardi, G., et al.: UFO: unified foundational ontology. Appl. Ontol. **17**(1), 167–210 (2022). https://doi.org/10.3233/AO-210256
14. Guttag, J.: Abstract data types and the development of data structures. Commun. ACM **20**(6), 396–404 (1977)
15. Hoffman, R.R., Mueller, S.T., Klein, G., Litman, J.: Metrics for Explainable AI: Challenges and prospects. arXiv abs/1812.04608 (2018)
16. Huang, L., Duan, Y., Zhou, Z., Shao, L., Sun, X., Hung, P.C.K.: Enhancing UML class diagram abstraction with page rank algorithm and relationship abstraction rules. In: Drira, K., et al. (eds.) ICSOC 2016. LNCS, vol. 10380, pp. 103–116. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-68136-8_10
17. Norman, D.: The Design of Everyday Things: Revised and Expanded Edition. Basic Books (2013)
18. Romanenko, E., Calvanese, D., Guizzardi, G.: Abstracting ontology-driven conceptual models: objects, aspects, events, and their parts. In: Guizzardi, R., Ralyté, J., Franch, X. (eds.) RCIS 2022. LNBIP, vol. 446, pp. 372–388. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-05760-1_22
19. Romanenko, E., Calvanese, D., Guizzardi, G.: Towards pragmatic explanations for domain ontologies. In: Corcho, O., Hollink, L., Kutz, O., Troquard, N., Ekaputra, F.J. (eds.) EKAW 2022. LNAI, vol. 13514, pp. 201–208. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-17105-5_15
20. Saitta, L., Zucker, J.D.: Abstraction in Artificial Intelligence and Complex Systems. Springer, New York (2013). https://doi.org/10.1007/978-1-4614-7052-6
21. de Souza Figueiredo, G.V.: Ontology-based complexity management in conceptual modeling. Ph.D. thesis, Federal University of Espírito Santo (2022)
22. Verdonck, M., Gailly, F.: Insights on the use and application of ontology and conceptual modeling languages in ontology-driven conceptual modeling. In: Comyn-Wattiau, I., Tanaka, K., Song, I.-Y., Yamamoto, S., Saeki, M. (eds.) ER 2016. LNCS, vol. 9974, pp. 83–97. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46397-1_7
23. Villegas Niño, A.: A filtering engine for large conceptual schemas. Ph.D. thesis, Universitat Politècnica de Catalunya (2013)