Software and Systems Modeling (2024) 23:57–84
https://doi.org/10.1007/s10270-023-01094-5

REGULAR PAPER

Modelling assistants based on information reuse: a user evaluation for language engineering

Ángel Mora Segura[1] · Juan de Lara[1] · Manuel Wimmer[2]

Received: 17 March 2021 / Revised: 12 September 2022 / Accepted: 13 February 2023 / Published online: 17 April 2023

Abstract

Model-driven engineering (MDE) uses models as first-class artefacts during the software development lifecycle. MDE often relies on domain-specific languages (DSLs) to develop complex systems. The construction of a new DSL implies a deep understanding of a domain, whose relevant knowledge may be scattered in heterogeneous artefacts, like XML documents, (meta-)models, and ontologies, among others. This heterogeneity hampers their reuse during (meta-)modelling processes. Under the hypothesis that reusing heterogeneous knowledge helps in building more accurate models, more efficiently, in previous works we built a (meta-)modelling assistant called EXTREMO. EXTREMO represents heterogeneous information sources with a common data model, supports its uniform querying and reusing information chunks for building (meta-)models. To understand how and whether modelling assistants—like EXTREMO—help in designing a new DSL, we conducted an empirical study, which we report in this paper. In the study, participants had to build a meta-model, and we measured the accuracy of the artefacts, the perceived usability and utility and the time to completion of the task. Interestingly, our results show that using assistance did not lead to faster completion times. However, participants using EXTREMO were more effective and efficient, produced meta-models with higher levels of completeness and correctness, and overall perceived the assistant as useful. The results are not only relevant to EXTREMO, but we discuss their implications for future modelling assistants.

Keywords Modelling · Modelling assistants · Language engineering · Modelling process · Empirical studies

1 Introduction

Model-driven engineering (MDE) promotes the usage of models throughout the complete software development lifecycle [1]. In this context, models are used to analyse, specify, test, simulate, execute, generate, and maintain the software to be built, to name just a few possibilities [2].

It is clear that high-quality models and meta-models are a must for the success of MDE projects. Meta-models capture the most important concepts of a domain and allow describing the features of systems using the vocabulary of highly specialized domains, such as manufacturing, logistics, and finance. Constructing a meta-model generally involves the following two roles: (i) a *domain expert*, who has in-depth knowledge of a particular domain, and (ii), a *meta-modelling expert*, who is experienced in object-oriented design, class-based modelling, and language engineering. However, often the meta-modelling expert is left alone in the construction of a meta-model in a domain that s/he may not be an expert in, and thus, needs to make uninformed decisions due to missing tacit domain knowledge or under-specified language requirements. This may lead to mistakes or omissions compromising the overall quality of the language under construction [3].

Most programming IDEs (e.g. Eclipse) provide assistants to facilitate code completion, code search [4,5] and reuse.

Communicated by Lionel Briand.

✉ Juan de Lara
Juan.deLara@uam.es
http://www.ii.uam.es/~jlara

Ángel Mora Segura
Angel.MoraS@uam.es; angelmoraseg@gmail.com

Manuel Wimmer
manuel.wimmer@jku.at
https://www.se.jku.at/manuel-wimmer

1 Modelling and Software Engineering Research Group, Universidad Autónoma de Madrid , Madrid, Spain

2 Business Informatics - Software Engineering Institute, Johannes Kepler University Linz, Linz, Austria

Modelling assistants are still not the norm within modelling environments (like the Eclipse Modelling Framework [6]), but they are recently gaining much attention by the MDE community [7]. For example, proposing reference architectures for intelligent modelling assistants [8], and assistants targeting different activities like model/meta-model completion [9–15], model/meta-model creation [16,17], model finding and reuse [18–21], and model repair [22–24]. Assistants are being built using a variety of techniques, like constraint solving [24,25], reinforcement learning [22], machine learning [13,21], natural language processing techniques [12,14] and information retrieval methods [9,15,18,19,23].

However, there is currently a lack of understanding of the strong and weak points of different types of assistants—built with different techniques and directed to help in different activities—from the user point of view. This understanding is important for developers of future modelling assistants, given the prominent role that they are expected to have within MDE tools [8,26]. The work we report on this paper is a step to fill this gap.

In previous works, we created a modelling assistant called Extremo [19,20]. Extremo helps in reusing information from heterogeneous artefacts, like models, meta-models, ontologies, or XML documents. The approach is based on representing all these heterogeneous sources within a common data model, which then can be queried uniformly. Extremo is level-agnostic, i.e. it can be used to help creating models at any meta-level, and extensible, i.e. it supports the definition of new types of queries and provides support for new data formats. Extremo was designed to be integrable with external modelling tools within the Eclipse ecosystem, to achieve true information reuse. Our hypothesis when building Extremo was that such an assistant would be useful to obtain more accurate and complete (meta-)models, in a more efficient way.

The purpose of this paper is to test this hypothesis, analysing the benefits that a modelling assistant like Extremo can bring to the (meta-)modelling process. The *research method* is empirical and includes a controlled experiment in an academic environment using students and researchers as *subjects*, assuming the role of meta-modelling experts. The experiment investigates the effects of introducing our tool for the *construction of a meta-model* in the financial domain. We divided the experiment subjects in two groups (one was allowed to use Extremo, and the other was not), and we provided them with heterogeneous resources to enable *domain exploration*, along with the description of the language to be built.

The experiment results indicate that Extremo leads users to obtain more complete and correct meta-models, significantly improving their efficiency and effectiveness. A subjective evaluation—based on the System Usability Scale [27]—showed a good level of tool usability, and a general agreement about the usefulness of the assistant to perform the modelling task. The results of our work are relevant for creators of new modelling assistants, especially for those targeting finding and recommending (meta-)models or fragments [18], and those facilitating (meta-)model reuse [21]. More generally, this work also contributes to increase the body of knowledge about experimentation in MDE, as the lack of experimental evaluations has been frequently recognized as a limiting factor in MDE tools [28].

Key findings and contributions of the paper:

- We report on a user study for a modelling assistant, which can be used as a template for the evaluation of other—current and future—modelling assistants, to understand their strong and weak points.
- Our results show that using assistance did not lead to a faster completion of the modelling task.
- Participants using assistance created meta-models that were more complete and correct than the meta-models built without assistance.
- The use of assistance lead to more *productivity*, measured as the ratio between the number of meta-model elements, and the completion time.

The organization of this paper is as follows. Section 2 introduces related research, identifying gaps and providing motivation for our work. Section 3 gives an overview on our approach to modelling assistance and describes the capabilities of Extremo. Section 4 illustrates the use of the assistant to create a meta-model in the financial domain. Section 5 describes the plan, organization, and execution of the experiment. Section 6 describes the obtained results and their interpretation. Section 7 provides a discussion of the results, including implications for future modelling assistants. Section 8 analyses *threats* to the validity of our experiment. Finally, Sect. 9 draws conclusions and proposes lines for future work.

2 Related work

In this section, we revisit work proposing assistants for programming and modelling (Sect. 2.1), methods to aggregate and reuse heterogeneous information (Sect. 2.2), and on evaluation of modelling tools (Sect. 2.3).

2.1 Assistants for programming and modelling

Assistants are frequently used within programming IDEs [4, 29], helping in tasks like API usage, code completion or quick-fix applications. Several recent recommenders, like Kite[1] or Codota[2], use machine learning and information retrieval techniques to propose sophisticated code completions, or find relevant code examples based on the current context. In most cases, higher productivity is the selling point.

In the modelling world, the use of assistants is not so widespread as in programming. However, we can find assistants to help *creating* (meta-)models from scratch, as well as to *complete*, *repair*, and *reuse* models.

Regarding creation, DSL-maps [16] is an approach to create an initial version of a meta-model given DSL requirements expressed as a mind-map. An assistant suggests the use and combination of meta-modelling patterns to create the initial meta-model draft, on the basis of the requirements and an encoding of the patterns in the form of an ontology. At the model level, UCcheck [17] helps in creating new UML use case diagrams using existing ones as templates.

Some assistants to help in model completion use constraint solving [25], for example using Alloy [30]. In this case, the assistant proposes several completions of a partial model to obtain a syntactically correct model, conforming to the meta-model. Other assistants can be used to obtain more *semantical* completions. In this case, the assistants extract the information from heterogeneous sources, like the case of Extremo [19,20], or from natural language documents, as in the case of DoMoRe [9] or [12]. In both cases, the assistant needs to access the information in a homogeneous way, either by using read adapters [9] or by importing the information into a uniform model [19]. There are also approaches based on graph neural networks (GNNs) [13], pre-trained language models [14], or classical recommendation methods [15]. In all these cases, the recommenders were trained with data-sets of (Ecore) meta-models. Other approaches to model completion employ similarity criteria to recommend similar items to the item being edited. This is the case of the assistant of Elkamel et al. [11], which recommends similar UML classes from a repository of UML class diagrams. SimVMA [21] recommends (Simulink) model completions based on the detection of near-miss model clones, in repositories of the organization, using machine learning techniques. Such near-clone models can also be recommended for reuse.

Other assistants recommend ways to repair faulty models. This has been approached using a variety of techniques [31], for example based on constraints [24,32], graph transformations [23,33] or reinforcement learning [22]. While some assistants propose purely syntactical fixes, ensuring correctness with respect to the meta-model [24,32,33], other approaches take into account previous ways to correct a model [22], or actions perform in previous model histories [23].

[1] https://kite.com/.

[2] https://www.codota.com/.

Some assistants are built for a specific language, like UML use cases [17], class diagrams [11], Ecore models [16], or Simulink models [21]. Instead, others are applicable to arbitrary modelling languages [19,20,22–25,33].

Another important aspect is whether the assistant can be easily plugged into existing modelling tools. Most approaches are closed tools or extend particular tools [11, 21,25]. Instead, assistant architectures enabling extensions of existing modelling tools would enable the reuse of the assistants themselves [8]. This is the principle followed by Extremo, and to some extent by Hermes [10,34,35]. The latter work proposes a generic, extensible architecture that supports plugging-in different recommender strategies. In contrast, the extensibility of Extremo comes from the possibility to be incorporated into existing (Eclipse) tools and to handle additional heterogeneous data sources, which are then uniformly represented to enable its querying.

2.2 Aggregation and querying of heterogeneous information

Extremo is based on querying and reusing heterogeneous information, once it has been represented uniformly and stored in a common data model. In the programming world, storing large code bases in a database, to enable their flexible query, has been proposed [36,37], e.g. as a way to provide insights in an existing project, or retrieve interesting code based on similarity. While the approaches are similar, in our case the artefacts come from different heterogeneous sources (XML, Ontologies, CSV, XMI models, Ecore models), and hence, we need to transform them into the common data model.

In the modelling world, some researchers have proposed architectures to index and represent models uniformly for their reuse. In particular, in [38,39] the authors apply this concept to SysML and Simulink models, which can be queried using model similarity or natural language. Other approaches, like Moogle [40], are also directed to search relevant models within a repository using Google-like queries, while MAR [41] uses query by example. In the three cases, the goal is finding complete relevant models, while our approach is more fine-grained—we aim at finding relevant fragments—and directed for information integration.

Additionally, approaches have been proposed for querying models of specific languages (e.g. process models) [42,43]. While these approaches permit taking into account the language semantics in the search, our approach supports a more general search—because it is done over the common data model—which is independent of the modelling domain and the technical space.

Technically, instead of relying on a uniform, common data model, our design could have been based on adapters, adding a layer of indirection between the query language and the data source [9,44]. This adapter layer enables uniform access to different technologies, without requiring transformation to a common representation. However, as the data sources will be queried heavily, the performance penalty incurred by this import will be amortized by faster queries.

2.3 Evaluating MDE tools

The lack of experimental evaluations has been frequently recognized as a limiting factor in MDE tools [28,45]. One of the reasons is that participants require specialized skills, which are harder to find than, e.g. programming skills. Next, we revisit some representative experiments for evaluating modelling tools, in order to compare with the approach we follow in our experimental setting, motivating our choices.

Some researchers have used interviews, e.g. to investigate issues on MDE tool adoption in industry [46]. An interview-based approach is not possible in our case, since we want to measure the effects of bringing a novel assistant to the modelling process, and hence, we require modellers using the tool.

In [47], the authors presented a user evaluation of the effectiveness, efficiency, and user satisfaction of applications generated with MDE tools. For this purpose, eight participants were asked to complete two tasks with no time limit. Regarding satisfaction, the authors used the Questionnaire for User Interaction Satisfaction (QUIS), an alternative to System Usability Scale (SUS) [27], which is more oriented to aspects of the application user interface. In our case, we opted for the SUS, for being more standard, focus on usability and having a larger body of reported results we can rely on.

FlexiSketch [48] is a flexible, collaborative modelling tool that enables engineers to create models informally, while developing a lightweight meta-model. For its evaluation, the authors designed an experiment consisting in creating a meta-model. They divided the participants (mostly undergraduate students) in two groups, one using FlexiSketch and the other using pen and paper. They compared the experiment results to a reference implementation to measure completeness and correctness. These metrics are appropriate for our goals, but in addition we aim to measure productivity and usability. They also presented another evaluation, consisting of more open-ended, free tasks, where the experiment was recorded, and interaction with the tool studied. This type of experiment is less appropriate in our context, since we want to observe and measure the advantages of using Extremo.

In [49], the authors evaluate three different UML modelling tools by means of a controlled experiment. The study was conducted with undergraduate and graduate students, and the main goal was to compare the productivity achieved when modelling with the given set of UML tools. For this, they measured completeness of the solutions and the effort required, learnability, as well as memory load. Interestingly, they did not measure correctness of the solutions. This is important in our case, since we want to measure whether an assistant improves this aspect. Another work on experimenting with UML modelling tools is presented in [50]. In particular, they compare different modelling tools by collecting the time and number of steps needed to achieve particular tasks as well as the subjective opinions of the participants.

In [51], two collaborative modelling tools (Creately and Socio [52]) were compared regarding efficiency, effectiveness, satisfaction and quality. For this purpose, an experiment based on students was designed using a within-subjects cross-over design of 2 sequences and 2 periods. The experiment involved measuring speed (time to completion), fluency (number of messages exchanged), completeness and precision and error rate of the created models (by comparison with respect to a "gold standard" model created by the researcher), and satisfaction (given by a SUS questionnaire). Our experiment design is similar, but we did not use cross-over to avoid learning effects. In addition, while time to completion is a measure of speed, it does not consider the quality of the output, and so in our experiments we propose metrics for productivity and performance.

Finally, a process to evaluate modelling tools as well as an example instantiation for a particular tool is presented in [53]. The main focus of this work is to determine the knowledge level needed to use a particular modelling tool.

The literature also reports some evaluations of industrial modelling approaches and tools [54–58]. Typically, these focus on productivity gains [55,56,58], but there are also interview-based studies on the use of MDE techniques and subjective usefulness [57]. In our case, productivity gain is a factor of interest, but we could not perform an experiment in an industrial setting. Instead, we used master and PhD students, similar to all mentioned experiments from academic researchers [47–51].

In addition to tool usability, another element subject to usability studies is the DSL itself. Such DSL usability evaluation can be done using frameworks like the cognitive dimension of notations [59], or the physics of notations [60]. Tools have been proposed to facilitate both the development of DSLs that are usable [61], and to evaluate their usability [62]. Sometimes, observations made using such theoretical frameworks are validated using user experiments. For example, in [63] the WebML notation was studied using the physics of notations, and some recommendations for improvement suggested. These improvements were then confirmed by a user study. In case of modelling assistants, a reference architecture for their design was proposed in [8]. However, there is currently no underlying theory or framework that can be used for the design of a user study for such assistants.

Regarding assistants, applicable recommendation and assistance methods are being actively researched by the MDE community, but we believe that it is equally important to understand the user acceptance of these methods, and to clarify their benefits from a user perspective. Some approaches mentioned in Sect. 2.1 provide no evaluation, or evaluations based on off-line experiments [22,24] (i.e. with no users). Instead, others provide qualitative information only [9,17], which is not enough to show the real benefits of the assistants, or do not consider the users subjective perception [11]. Hence, more work is needed in this respect, able to answer questions, like (for assistants based on information reuse) *is the benefit to obtain more efficient (i.e. faster) modelling, to achieve more effectiveness (i.e. more complete and correct models), or both?*. *Would the same benefits be obtained by using different types of model completion assistants (i.e. more syntactical vs. more semantical)?*. *Would the users be willing to promote the use of those types of assistants for their modelling tasks?*. This paper is a step in this direction. While we report on a set of experiments conducted over Extremo, the results can be useful for developers of modelling assistants, especially for those based on information reuse, as we discuss in Sect. 7.3.

3 Extremo: an assistant for modelling and meta-modelling

This section first revisits our approach for modelling assistance [19,20] (Sect. 3.1) and then, summarizes the main functionality of Extremo, the tool implementing the approach (Sect. 3.2).

3.1 Approach and architecture

Modelling often requires checking information about a domain, which may be scattered in heterogeneous formats and locations. To alleviate this problem, we developed a modelling assistance approach directed to facilitate reuse of disperse, heterogeneous information [19]. Its scheme is shown in Fig. 1.

The main idea of our approach is to gather heterogeneous information sources (label ① in Fig. 1) into a common data model (label ② in Fig. 1). Once the heterogeneous data are stored under a common representation, it can be queried and visualized uniformly, facilitating domain exploration. Moreover, the query results can be directly reused (copied) into the model being built (label ③ in Fig. 1).

Our approach represents heterogeneous information as a model, conforming to the meta-model of Fig. 2. The meta-model organizes the different information resources (class `Resource`) into repositories (class `Repository`), representing uniformly both models and meta-models, classes and objects, and attributes and slots, leading to simplicity and generality [64]. In addition, the instantiation relation (`describes/descriptors`) is reified, so that the data model is not limited to a fixed number of meta-levels. This way, the approach becomes *level agnostic* and the data model can store information of both types and instances, which can then be reused to build models and meta-models.

Please note that, in addition to the structural information, the data model can also store defined constraints (e.g. OCL in case the resource is an Ecore meta-model, or XML-specific constraints in case of XSD schemas), as well as format-specific information (e.g. whether an `ObjectProperty` represents a composition in an Ecore meta-model) using key-value pairs (class `MetaData`). We refer to [19] for more details on the approach.

3.2 Tool support

We have realized the previous approach in an Eclipse plugin called Extremo. The tool has an extensible architecture, which profits from Eclipse extension points. This way, new data formats, types of queries, persistence mechanisms for the data model, and connections with Eclipse (meta-)modelling tools can be externally defined. It must be stressed that Extremo is not an assistant for a specific modelling tool, but it can be connected with any modelling tool within the Eclipse ecosystem using extension points. Figure 3 shows the tool being used to create a meta-model in the financial domain, which will actually be used in the experiment we present in Sect. 4.1.

Extremo offers three main views, labelled as ①, ② and ④ in Fig. 3. The *repository view* (label ①) displays the information gathered from heterogeneous data sources into the common data model. In Fig. 3, the repository view shows the content of three repositories within the data model, indicating the original technological space of each contained resource (Ontologies, Ecore). The structural contents of each resource can be visualized in this view, and in addition, in the *resource explorer* (label ②) using a graph-based representation.

Resources and repositories can be queried through a wizard (the dialog window with label ③). Extremo offers a catalogue of predefined queries, and the wizard permits selecting a query, providing values for the required input parameters, and selecting additional services (e.g. inexact match or synonym search using WordNet [65]) to make the query more flexible.

The *results view* (label ④) is responsible for displaying the query results. This view organizes the results by query types, supports browsing the results, and incorporating them into the (meta-)model being built. The latter action is done through a contextual menu, shown with label ⑤. The three first items on the menu depict three integrations of Extremo:

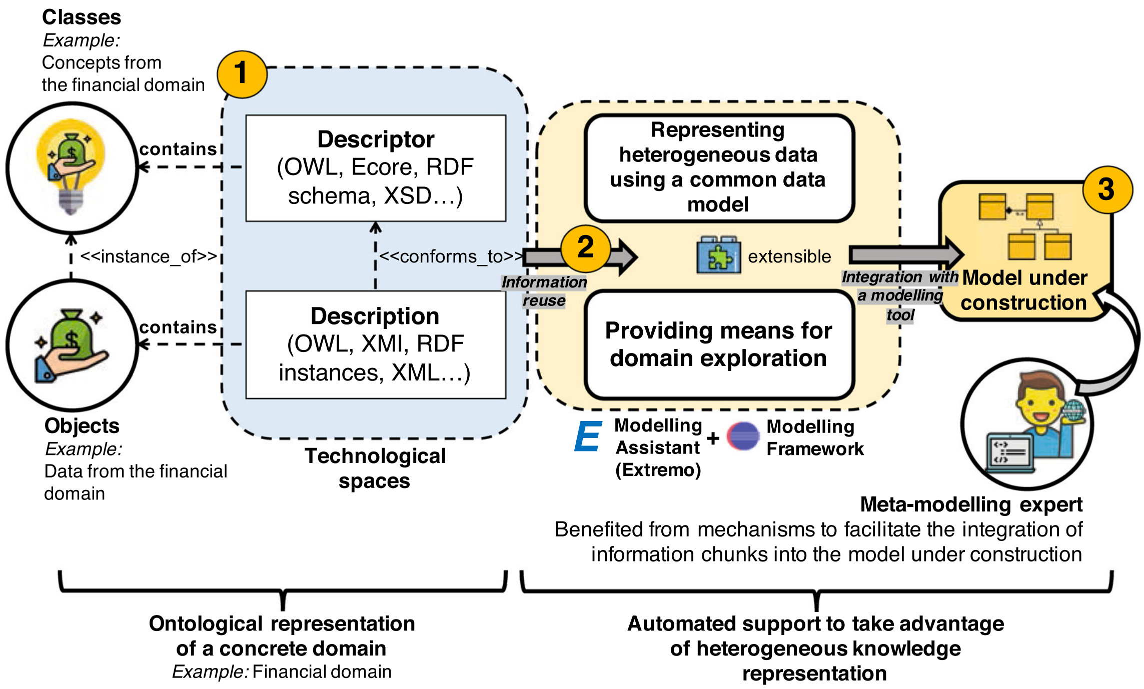


Fig. 1 Our approach to reuse heterogeneous information for (meta-)modelling

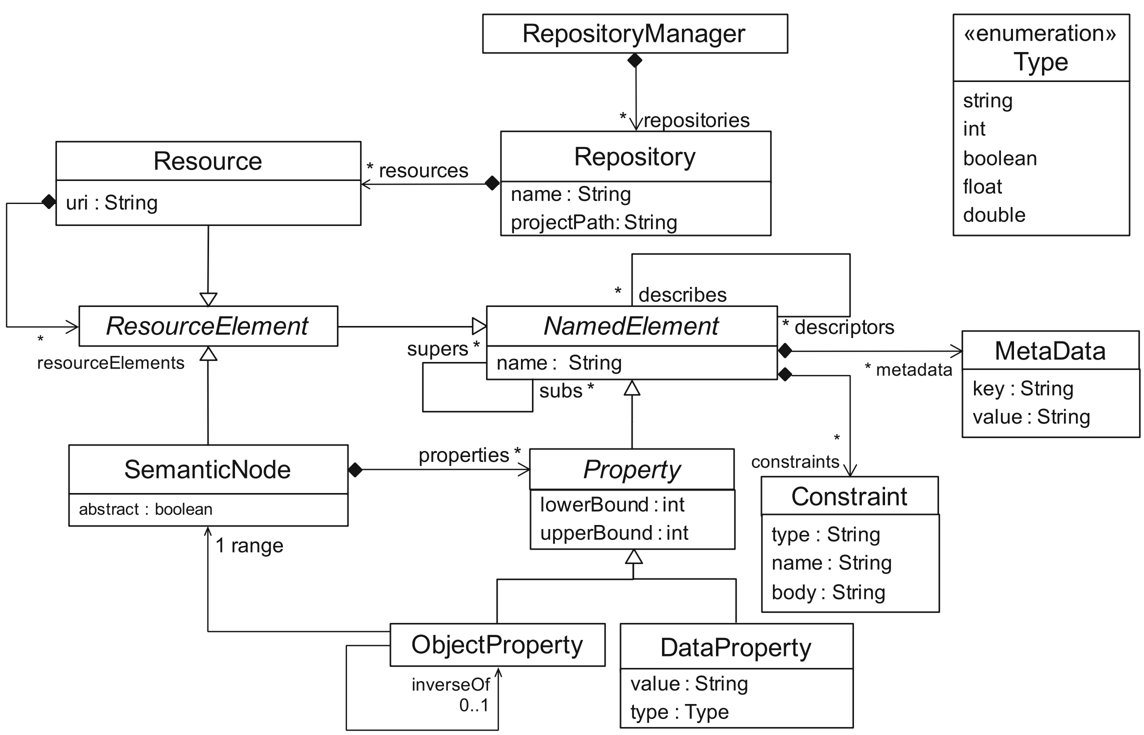


Fig. 2 The common data model (extracted from [19])

with a UML model editor,[3] with the standard Ecore editor, and with Exeed, an enhanced version of the EMF tree-based editor.[4] The menu further comprises commands to locate the query results in the repository and to navigate to the node type, resource type, and repository of a node. Figure 3 (label

[3] UML2-MDT, www.eclipse.org/modeling/mdt.

[4] Epsilon Exeed, http://www.eclipse.org/epsilon/.

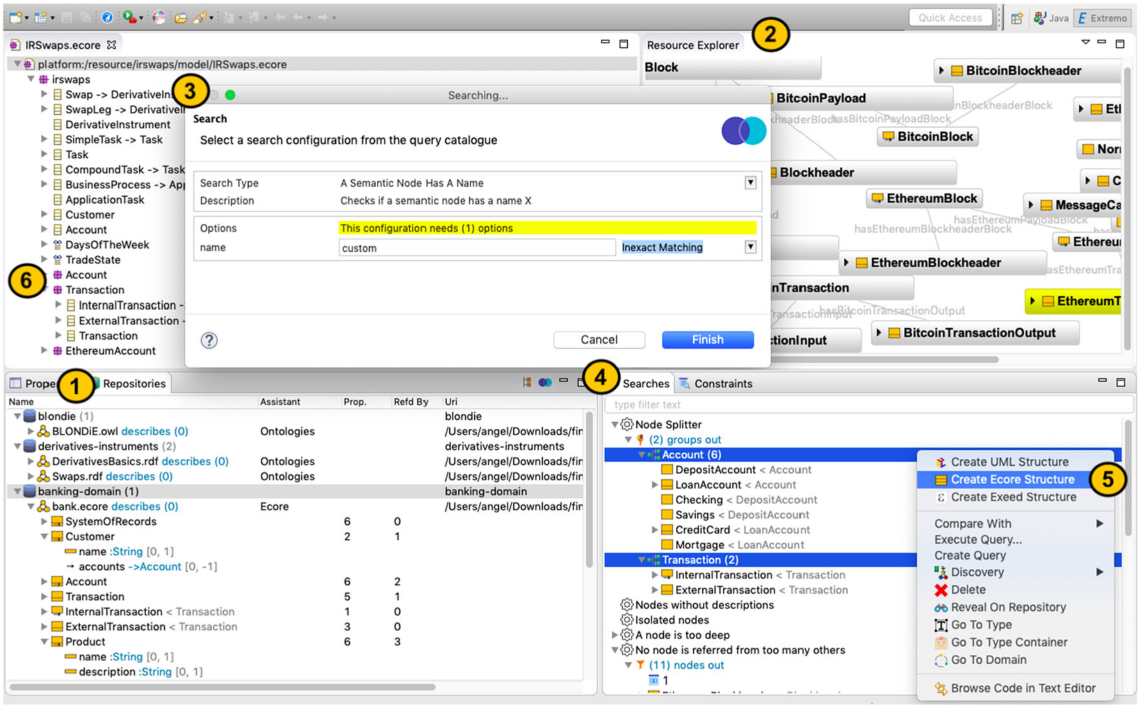


Fig. 3 EXTREMO in action (extracted from [20])

⑥) illustrates EXTREMO for creating a meta-model by using the standard Ecore tree-based editor. EXTREMO is available at http://angel539.github.io/extremo/ as open-source software.

4 Using EXTREMO in practice

We illustrate the use of EXTREMO to create a domain-specific language (DSL) from the point of view of a language engineer. For simplicity, we use as example the same task as used in the experiment.

In the following, Sect. 4.1 details the task to be performed in the experiment, Sect. 4.2 describes how the task would typically be performed by a language engineer without assistance and Sect. 4.3 explains how the task would be performed with the help of EXTREMO.

4.1 Task to perform: a DSL for plain vanilla swaps

Our DSL is inspired by a case study in the financial domain proposed by the EDM Council,[5] briefly introduced in [20]. We chose this case study for being representative of a complex, highly specialized domain, for which a meta-modelling expert may lack knowledge and need assistance in its creation. Since the lack of domain knowledge may impact the quality of meta-models created by non-specialists (leading commonly to higher levels of incompleteness or incorrectness), the purpose of our approach is not to substitute a domain-expert but to guide the meta-modelling expert in building models with higher quality in an alien domain.

This way, the scope of the DSL is the definition of *Plain Vanilla Swaps*, the simplest version of *Interest Rate Swaps (IR Swaps)* between companies. Swaps are often used when a company wants to borrow money at fixed interest rate, but the lender prefers to offer a floating-rate loan. In this case, the company borrows at the floating rate and makes a separate deal (with another bank, a facilitator firm or another company) to obtain the fixed rate. The parties involved in a swap trade *over-the-counter* (OTC), i.e. they operate outside the conditions offered by the trading markets. Hence, each party needs to specify its operating conditions, and to guarantee the operation safety, the swap process is executed over a blockchain-based ecosystem with an external entity acting as a validator.

The DSL should include elements showing the variety of financial entities involved, concepts from the financial domain and the information required to describe the exchange operation. The description of the task given to the participants is included in "Appendix A.3".

[5] https://edmcouncil.org/.

4.2 Building the DSL without assistance

Figure 4 depicts two typical scenarios that may arise when building a meta-model for a DSL like the one we have just described in Sect. 4.1.

The left of the figure shows the ideal scenario. In this case, a client defines the problem requirements for building the DSL (label ①), which might include her vision about the system, the goal of the DSL and what she wishes for the project. Thus, the *description of the problem* typically is defined in conformity with those requirements. Next, an engineer—probably with lack of knowledge in the financial domain—would work collaboratively with a domain expert to build the meta-model of the DSL (label ②), while they both periodically validate the partial results with the client. In this step, the meta-modelling and domain experts exchange ideas about the *description of the problem* in order to understand deeply which concepts, primitives and vocabulary are more adequate for inclusion in the meta-model (label ③). Once both roles agree upon a part of the meta-model, the engineer uses a tool, such as the Eclipse Ecore model editor, for its implementation. Finally, the process ends when the DSL is validated with the client, checking that it fulfils her expectations.

However, sometimes, this ideal situation is not encountered, but instead the workflow follows the scheme in the right part of the figure. In that situation, there is a lack of involvement of the domain-expert or the client (who may be expert in the domain as well). Hence, the meta-modelling expert (label ④) has the burden to understand the description of the problem alone or following indications loosely described, making decisions about what concepts define better a domain that s/he might not understand well (label ⑤). Thus, a meta-modelling expert follows a process of domain exploration (label ⑥) checking relevant references and sources. This way, it is difficult for the engineer to guarantee that the meta-model under construction is complete and correct with respect to the description of the problem. Hence, this process may lead to mistakes or omissions in the meta-model under construction, and therefore result in disagreements with the client.

4.3 Building the DSL with assistance

Extremo was built to help developers in creating high-quality (meta-)models, by facilitating domain exploration and enabling information reuse. Its aim is not to substitute domain experts, but to help the engineer in the modelling process. This help is especially useful when scenarios like the one shown to the right of Fig. 4 arise.

Figure 5 depicts the process of using Extremo to solve the problem described in Sect. 4.1. In a first step (label ①), resources with relevant information about the domain are identified. For the example, we selected resources from the Finance Domain Task Force of the OMG,[6] specifically the FIBO collection, which includes information about the financial domain in OWL and RDF formats; *Blondie*, the Blockchain Ontology with Dynamic Extensibility,[7] which contains descriptions of the exchange operation; Ecore standard meta-models available in the OMG repository, like BMM, BPEL or BPMN; and finally, Ecore meta-models available in open repositories like GitHub and the ATL Ecore zoo[8] with concepts from the banking domain.

Next, the available resources are imported into Extremo's common data model (label ②). Once imported, they are accessible through Extremo's repository view and are ready to be explored, queried, and reused (labels ③ and ④). For example, the engineer might want to obtain a high-level view of the repository content, for which she can use a query like Nodes Splitter, to split the entities in the resources into inheritance hierarchies [20]. Additionally, the resource explorer can be used to visualize the resource contents as a graph, or search existing entities related to, e.g. *Swaps* using synonym-based search. The exploration of the domain, which may depend on the level of expertise of the user, should result in useful entities to reuse in the meta-model under construction (Fig. 3, labels ③ to ⑥). When a set of entities are added to the meta-model under construction, Extremo automatically takes care of the mapping between the common data model and the specific modelling tool (the Ecore Model Editor in the example). Please note that steps ③ and ④ in Fig. 5 are iterative.

Figure 6 depicts a fragment of the meta-model which was mostly constructed by reusing information from the chosen *resource collection*. The meta-model has concepts like Swap or SwapLeg reused from the FIBO ontologies, elements such as Transaction or Account reused from different Ecore meta-models from the banking domain, elements taken from Ecore meta-models from the process modelling domain (CompoundTask and Task) and elements (such as EthereumAccount) from the Blondie ontology. Overall, about *90% of the meta-model classes* were obtained by reuse, *most generalization relationships* were reused from the information chunks returned by the queries, and *some of the associations* by the combination of semantic nodes. The rest of the meta-model elements were added manually as usually done with modelling editors.

[6] https://www.omg.org/fdtf/projects.htm. The OMG is the standardization body behind many modelling standards such as UML, SysML, MOF or BPMN.

[7] https://hedugaro.github.io/Linked-Blockchain-Data/.

[8] http://web.imt-atlantique.fr/x-info/atlanmod/index.php?title=Ecore.

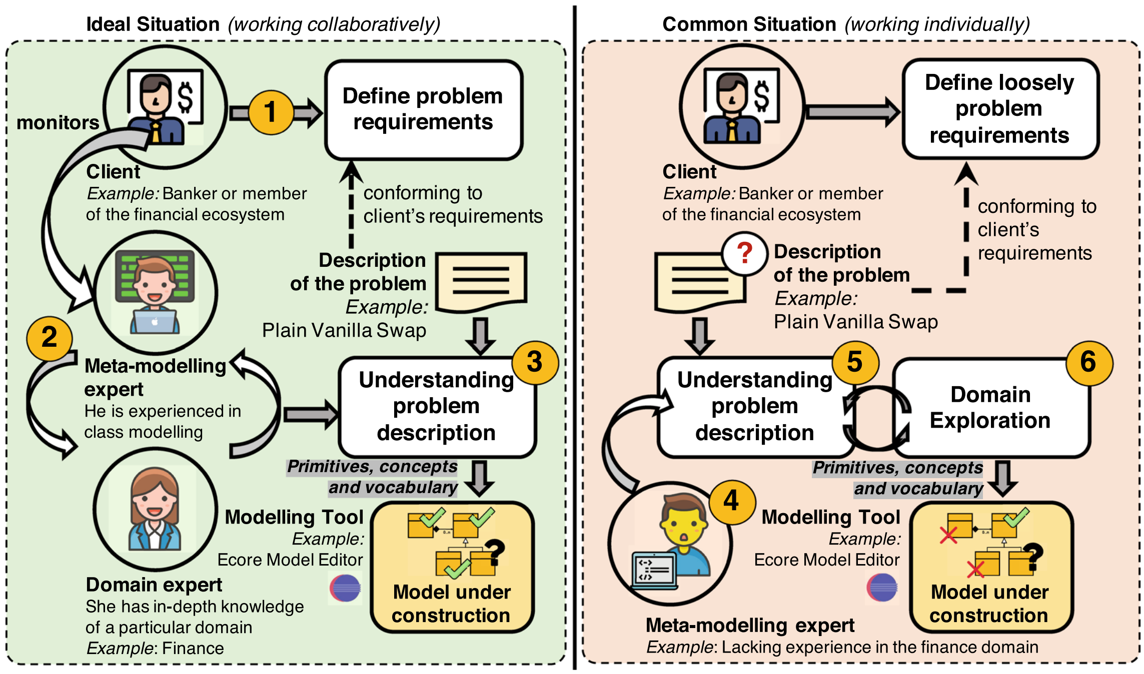


Fig. 4 Possible scenarios arising when building meta-models for specialized domains

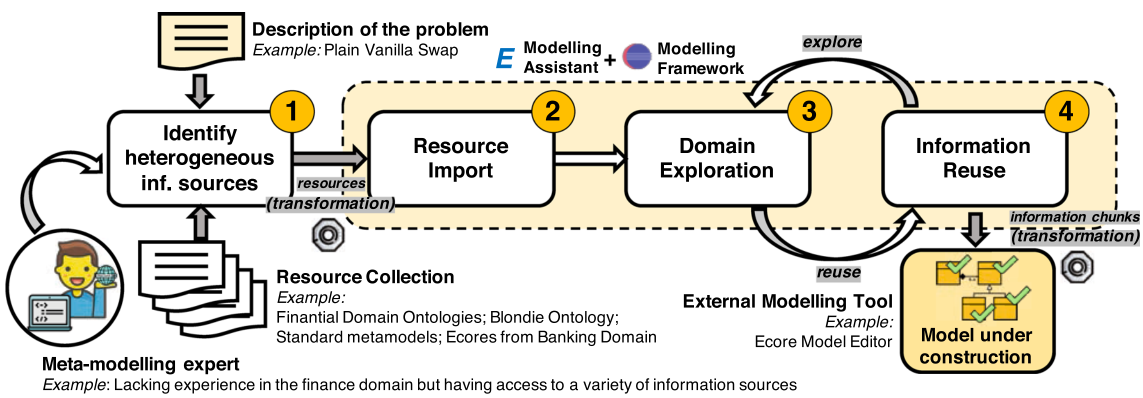


Fig. 5 Process for building a meta-model with EXTREMO

5 Experiment setup

We conducted a user study to evaluate our modelling assistant. The *purpose* of the experiment is to evaluate if the productivity of the engineer and the accuracy of the meta-models improve by using our tool. We also aim at gathering insights into what can be expected when a meta-modelling expert is left alone in the construction of a domain that s/he might not understand well. We will measure the usability of EXTREMO and determine what features are perceived by the participants as more useful or helpful. Moreover, we will evaluate the completeness and correctness of the artefacts produced with and without assistance, compared to an expected solution reached by consensus between the authors. Finally, we will evaluate if the productivity of the engineer (measured as the number of correct and complete meta-model elements created per minute) improve by using our tool.

Thus, in the rest of this section we detail the *context* of the experiment and the selection of the subjects in Sect. 5.1, the *research questions* in Sect. 5.2, the formulation of the *hypothesis* and experiment *variables* in Sect. 5.3, the *experiment design* and its *execution* in Sect. 5.4, and its replicability in Sect. 5.5.

5.1 Context and selection of the subjects

The *context* of the experiment is to build a meta-model for the problem described in Sect. 4.1. The setup of the experiment was controlled and instructed by the first author of this paper in an academic environment. The planning includes

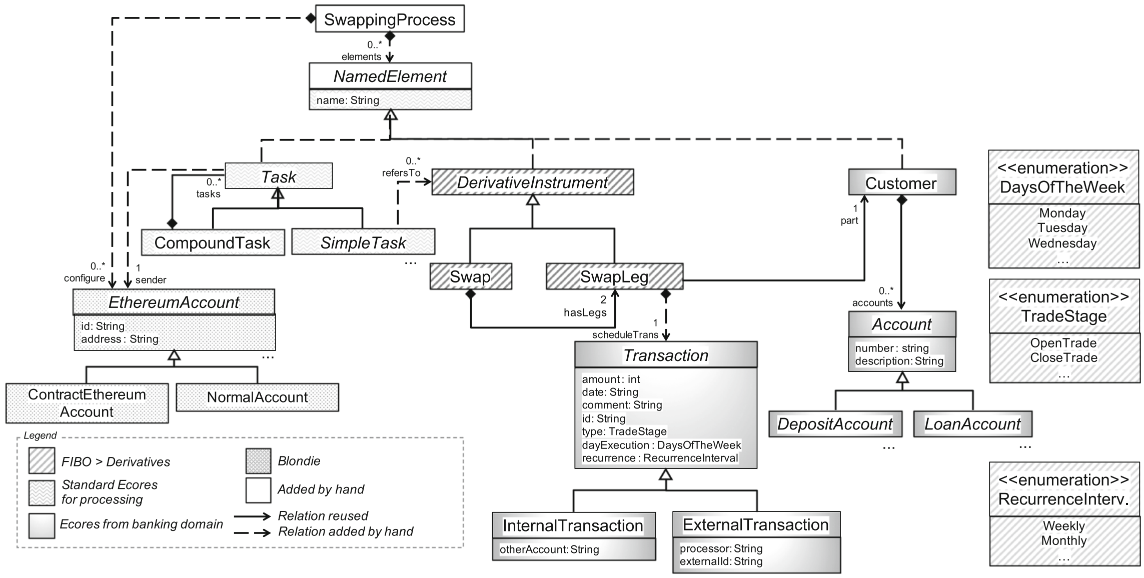


Fig. 6 Excerpt of the resulting Plain Vanilla Swap process meta-model (extracted from [20])

one independent variable with two conditions [66]: a *control group* (C) that uses an Eclipse general distribution to build the meta-model, and an *experimental group* (E) that uses the same distribution plus Extremo to solve the same task.

We invited 23 people with different backgrounds and ages to participate in the study. The control group consisted of 10 participants, students coming from two computer science master's degrees of the Autónoma and Complutense Universities in Spain, and members of the Modelling & Software Engineering (MISO) research group in Madrid. The experimental group had 13 participants, students coming from two master's degrees in business informatics and software engineering of TU Wien, and some members of the CDL-MINT research laboratory at TU Wien. The timing of the experiments avoided affecting the students' normal course, being performed using extra hours from their respective lectures. During the study, in the experimental group, 3 subjects decided to withdraw making a total of 20 subjects involved.

Participants in both groups had comparable backgrounds, since all students had taken one course on MDE, which included concepts of meta-modelling, graphical and textual concrete syntaxes, model transformations, OCL and code generation, as well as technological platforms like Eclipse/EMF. As this is the platform used in the experiment, the participants were already familiar with it. We chose two different places (Madrid and Vienna) to perform the experiments with the control and the experimental groups to maintain the separation of interests and not to contaminate both groups among each other.

The use of students as subjects in controlled experiments is an ongoing discussion in the software engineering community [67–70]. This point will be discussed further in Sect. 8 (*threats* to validity), but we argue on their validity for our experiment, on the basis of the following reasons:

Characterizing the subjects of the experiment:

- Students were chosen to minimize the cost of the experiment, considering also that they should serve as proxies of professional modellers. Most participants (75%) were Master's students, while the rest (25%) were PhD students.
- Choosing Master's and PhD students was planned before conducting the experiment, and considering that participants in both groups had comparable backgrounds as well as the knowledge needed to serve as subjects.
- All subjects presented experience for the technologies used during the experiment, measured beforehand of presenting the task. Also, 60% of the *experimental group* were working in *Industry*, while in the *control group*, most of them (70%) were working in *Academia*.

This way, we consider the *subjects* chosen as suitable to perform the experiment in the laboratory context used. The background and experience showed by the subjects of the experiment were appropriate in the specific circumstances of the experiment, showing relevant and recent experience for the purpose of the study and the technologies involved.

5.2 Research questions

By conducting this evaluation, we aim at answering the following research questions:

RQ1 What *problems* typically faces a meta-modelling expert when s/he is left alone in the construction of a language for a new (unfamiliar) domain?
RQ2 How *useful* is our approach for modelling assistance?
RQ3 What is the impact of using EXTREMO on the *completeness* and *correctness* of the solutions developed?
RQ4 Does EXTREMO improve *productivity*?

5.3 Variables and hypotheses

Variables The variables used in the study are summarized in Table 1. In our experiment, we measure the effects of using EXTREMO (and hence, this is the independent variable). Thus, we divided the participants into two groups: control (C) and experimental (E) group. The purpose of the experiment is to measure the quality of the artefacts built (MM), the *general usability* (U) and the *specific usefulness* (S) of EXTREMO's features. To assess the meta-models built, quality metrics (i.e. accuracy, error rate, precision, recall and F-measures) are computed. Finally, the time to completion (TTC) is measured in order to provide an indicator of *modeller productivity* ($p_{\#MM}$).

Hypotheses When building EXTREMO, our hypothesis was that reusing heterogeneous knowledge helps in building more accurate models, in a more efficient way. Hence, the use of the assistant was expected to have a positive impact on the accuracy and quality of the solutions, lead to shorter completion times and be perceived as useful. A null hypothesis, H_0, states that there are no real consequences of using EXTREMO when performing the experiment. Therefore, it is the hypothesis to reject with as high significance as possible [72]. The objective of our study is to empirically test the following hypotheses, each directed to answer the research questions presented in Sect. 5.2:

- $H_{0,MM}$. We will assume that our approach does not lead to models considered more complete and correct than those built in the *control group*.
- $H_{0,U}$. We will assume that our approach has a usability perceived as *not acceptable* [73] by the subjects, i.e. the scores given by the users are lower than 50 in a range from 1 to 100.

  $H_{1,U}$. We will assume that our approach has a usability perceived as *marginal* by the subjects, i.e. the scores are in a range from 50 to 70.

  $H_{2,U}$. We will assume that our approach has a usability perceived as *acceptable* by the subjects, i.e. the scores are in a range from 70 to 100.
- $H_{0,S}$. We will assume that the features of our tool have a specific usefulness perceived as *not acceptable* [73] by the subjects, i.e. the scores are lower than 50 in a range from 1 to 100.

  $H_{1,S}$. We will assume that the tool has a specific usefulness perceived as *marginal* by the subjects, i.e. the scores are in a range from 50 to 70.
  $H_{2,S}$. We will assume that the tool has a specific usefulness perceived as *acceptable* by the subjects, i.e. the scores are in a range from 70 to 100.
- $H_{0,TTC}$. In relation with variables MM and TTC, we will assume that our approach is less effective and less efficient for the same modelling task, i.e. the *effectiveness* and the *efficiency* for building the meta-model are worse in the *experimental group* than in the *control group*. For that purpose, we will compute the *modeller productivity* as the number of meta-model elements created per minute.

5.4 Experiment design and execution

To test these hypotheses and answer the stated research questions, we prepared a set of documents available at "Appendix A" and the project Wiki.[9] To assess the complexity of the task, the experiment design and to identify possible mistakes, we carried out a *pilot experiment* with 6 members of the CDL-MINT group (none of which participated in the real experiment). The questionnaires and the description of the task were corrected according to the feedback provided by the pilot participants. The material prepared for the evaluation includes the following five documents:

– *Informed Consent*. The *Informed Consent* (cf. Section A.1 in "Appendix A") had to be signed by all the participants in order to cover the basic ethical aspects [74, 75] considered by the project and authorize the data treatment. The subjects were not aware of the specific aspects to be studied, or the hypothesis we wanted to validate, but they were informed that the researchers wanted to study the outcomes obtained in both groups comparing the results. All participants were guaranteed anonymity, and they were informed about their right to withdraw from the experiment at any moment. The condition of the domain being outside of the subjects' expertise was presented to the subjects as follows: *"If you work or you know directly someone who works in the topic of the case*

[9] https://github.com/angel539/extremo/wiki/Instructions.

Table 1 Variables used in the experiment

Name	Value	Description
Independent Variables		
$GROUP$	{C, E}	The subjects are divided in two groups: control (do not use EXTREMO) and experimental (use EXTREMO)
Dependent Variables		
MM	File	Meta-model. Ecore file containing the solution to the task described in Sect. 4.1. We will analyse the accuracy of the solutions provided by the subjects (variable MM) in terms of various quality metrics, i.e. precision, recall, and F-measures
U	[1 . . . 100]	*Usability*. We will collect a score from 1 to 100, based on the System Usability Scale, and map it into acceptability ranges from *not acceptable* to *acceptable*
S	[1 . . . 100]	*Specific usefulness*. We will collect a score from 1 to 100 based on a 5-item bipolar Likert Scale [71] asking about specific features of EXTREMO and their usefulness
TTC	Integer (minutes)	*Time to completion*. Time spent by each subject in performing the task. We will use it to compute the number of meta-model elements created per minute ($p_{\#MM} = \#elem.MM/TTC$) as an indicator of *modeller productivity*

study, please inform the research staff about this situation". None of the subjects declared any issue with the domain chosen for the task.

- *Demographic Questionnaire*. The *Demographic Questionnaire* (cf. Section A.2 in "Appendix A") was distributed at the beginning of the experiment to both groups. It contained eleven questions about the background and experience of the participants. We used the collected data to characterize the participants' profile.
- *Description of the task*. The *Description of the task* (cf. Section A.3 in "Appendix A") contained explanations of the meta-model domain, similar to the one presented in Sect. 4.1. Participants had 1 h maximum to perform the task, but they were allowed to finish when they would consider their solution was complete. The instructor was present during the execution of the experiment to solve possible tooling issues. Also, in the experimental group, the participants received a training session with EXTREMO focused on its capabilities during the introductory talk [19]. The training material used is available at the project Wiki.[10] Upon completion, participants submitted their solutions in ".ecore" format.
- *Control Group Questionnaire*. To evaluate the problems presented during the construction of a language for a new domain, we asked all subjects in the *control group* to answer an *Opinion Questionnaire* (cf. Section A.4 in "Appendix A"). This questionnaire measured the opinion of the participants about the information and resources provided to perform the task, and the functionalities they considered an ideal modelling assistant should have.
- *Experimental Group Questionnaires*. We asked all subjects in the *experimental group* to answer a *General* and a *Specific Questionnaire* (cf. Section A.5 in "Appendix A"). The former measured the usability of the tool using the System Usability Scale (SUS) [27]. The latter measured the usefulness of the specific tool features, and the participants general opinion.

We prepared two Eclipse distributions with EMF tooling, for the control and the experimental groups. In the former, we provided access to useful resources mentioned in Sect. 4.3 to build the meta-model (ontologies and ecore meta-models). In the latter, such resources were replaced by EXTREMO.

5.5 Replicability of the experiment

To allow easy replication and verification of our experiment, a complete replication package is publicly available to interested researchers. The replication package includes the data, code for analysis and artefacts collected. Also, it includes the material described above. The package has been made publicly available via the Open Science Framework and can

10 https://github.com/angel539/extremo/wiki/Training-Material.

be accessed at https://osf.io/r3cs5/. All the information left in that repository is under an Attribution-NonCommercial-ShareAlike 4.0 International Licence (CC BY-NC-SA 4.0).

6 Experiment results

This section reports on the results of our user study. First, Sect. 6.1 presents an overview summary of the results, Sect. 6.2 summarizes the participants profile, and the following sections (6.3–6.6) describe the results obtained to answer the four research questions.

6.1 Summary

Overall, the experiment results in the control group indicate that language engineers would need assistance to create a meta-model in an unfamiliar domain. A lack of tools to search information uniformly, to facilitate the modelling process, may lead to (meta-)models with many omissions (Sect. 6.3). In contrast, the participants of the experimental group found our approach for modelling assistance useful, highlighting the usefulness of most EXTREMO's features (Sect. 6.4). Generally, reusing information can help to create models with higher level of precision or *correctness* and recall or *completeness* compared to a prototype solution (Sect. 6.5), yielding a higher ratio of classes per minute and a significant improvement in terms of *effectiveness* and *efficiency* of the modelling task (Sect. 6.6).

6.2 Participants profile

The experience and background of the participants was measured through a *Demographic Questionnaire* (Sect. A.2 in "Appendix A"). The average age of the participants was 29.83. Most participants (75%) were Master's students. In the *control group*, the distribution between Master's and PhD students was more balanced than in the *experimental group* (60%–30%). Overall, 16 out of 20 participants claimed that their English Level was at least *B2* according to the Common European Framework of Reference for Languages.[11] 60% of the *experimental group* rated themselves as to be working in *Industry*, while in the *control group*, most of them were working in *Academia*. Regarding their technical background, most subjects had less than 4 years of experience in software development (for being master's students), with some subjects having more than 6 years of experience (for being working already). Concerning modelling languages, most subjects (13, counting both groups) stated that they had less than 2 years of experience.

In general, while both groups are not homogeneous, the *control group* indicated a higher experience in modelling, containing also more members that were already Master's students in the technologies involved. Conversely, in the *experimental group*, they manifested to be more experienced in software development practices (excluding modelling) and their working experience was predominantly in industrial environments. Hence, we can state that good results in the *experimental group* can not be attributed to higher experience as might be expected by other general questions such as the previous knowledge within software development environments. Further details can be found in the Project's website.[12]

Notation. In the rest of this section, we use the notation $SubXX$ to refer to participant XX. Hence, $Sub01$ to $Sub10$ are the participants of the *control group*, while $Sub11$ to $Sub23$ are the participants of the *experimental group*. Subjects $Sub11$, $Sub20$ and $Sub21$ decided to withdraw from the experiment. Hence, they were discarded from the experiment results and the demographic profiling.

6.3 (RQ1) Meta-modelling problems reported by the *control group*

In order to collect the opinion from the *control group* and detect what *problems* a meta-modelling expert typically faces when building a DSL for a new domain, we prepared an opinion questionnaire [76] (cf. Section A.4 in "Appendix A"). This was handed out to the control group after the task was finished and included 5 yes–no questions and 1 open-ended question. In addition, for each yes–no question the subjects could provide a rationale for their answers.

Figure 7 shows the results obtained, and next, we extract the main findings.

Problem 1. Most participants would need assistance to perform the task. 100% of the participants stated that they think a tool to assist them during the meta-model construction phase would be useful ($Q5$). Most participants missed some tool to query the provided resources uniformly (60% answered *"Yes"* to $Q4$). This lead to omissions in their solutions as it will be discussed in Sect. 6.5. Additionally, the questionnaire explicitly asked about the functionalities a modelling assistant could have ($Q6$). Some subjects suggested features for organizing the model repositories and improve domain understanding ($Sub02$, $Sub08$ and $Sub09$). Others proposed features for searching or grouping the information ($Sub02$ and $Sub03$).

Problem 2. The lack of uniform search tools increases the risk of not using the available resources. As seen in Fig. 7, only 50% of the participants checked the resources provided with the description of the task (*Q1*), even if they could have reused fragments from the resources provided. Actually, when they

[11] https://www.coe.int/en/web/common-european-framework-reference-languages.

[12] https://github.com/angel539/extremo/wiki/Demographic-Study.

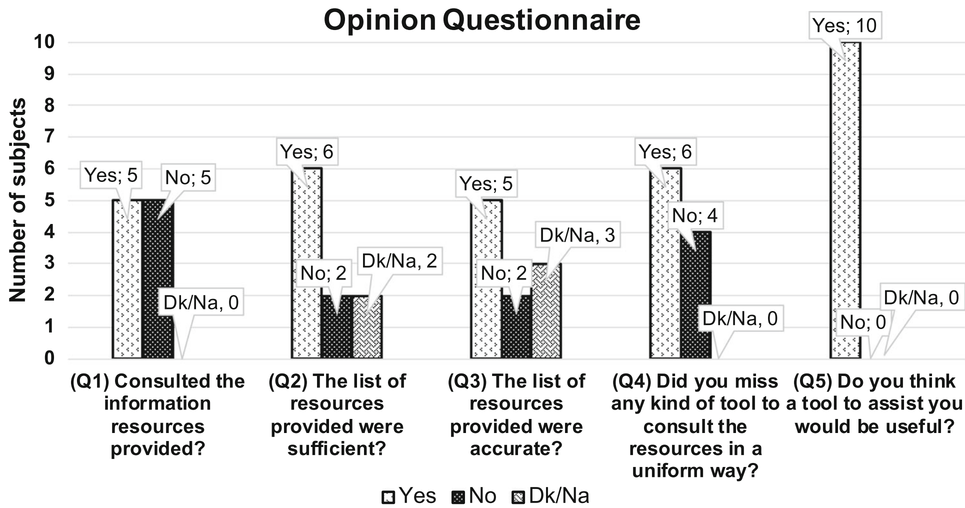


Fig. 7 Results obtained for the *Opinion Questionnaire* (cf. Section A.4 in "Appendix A"), with Dk/Na "Do not know, no answer"

were asked if they thought that the list of resources provided was sufficient and accurate to perform the task, 6 of them answered that *"Yes, it was sufficient"* ($Q2$) while 5 indicated that the resources provided were accurate ($Q3$). We found some inconsistencies in the answers as, e.g. some subjects that did not check the resources, indicated they were accurate. Some rationales included for their answers included:

- *"Not enough time. I was focused on the problem, not minor details."* (collected from $Sub03$ in $Q1$)
- *"Too much information for the time we have. I spent a lot of time to find what I wanted."* (collected from $Sub10$ in $Q1$)

Thus, we can conclude that even though the resources provided were generally considered *accurate* and *sufficient*, the lack of a uniform search tool increases the risk of not using them. This may increase the uncertainty about the correctness of the solution built, as expressed by $Sub01$ in the rationale extracted from $Q2$: *"Maybe it is enough. I am not sure if I did the Ecore right."* Additional details about the results obtained in the *control group* can be found in the Project's website.[13]

6.4 (RQ2) Usability and usefulness of EXTREMO's features

In the *experimental group*, participants used our approach for modelling assistance to build the meta-model for the problem described in the Sect. 4.1. Next, we present the usability of our approach and the usefulness of EXTREMO's features perceived by the subjects of this group. Figure 8 shows the results obtained for the SUS score (U variable, *usability*) and the Specific SUS (S variable, specific usability or *usefulness* of EXTREMO's features) collected through the *General* and *Specific Questionnaires* (cf. Section A.4 in "Appendix A"). Both variables were measured using a 5-item bipolar Likert scale [71] and computed as indicated by the cited authors [27].

The box plot shown in Fig. 8a shows the maximum, minimum, average (marked with an X), median (marked with a black line) and the first and third quartile for each variable. The scatter plot (b) displays the values obtained per subject and grouped by similarities found. Its horizontal axis represents the SUS score (U), while the vertical axis is the Specific SUS (S). As shown in the scatter plot, the subjects can be split in 4 groups: (①) those who gave the assistant the lowest score for the U variable but gave the assistant an excellent score for the S variable, (②) those who gave the assistant an excellent score in general above average, (③) those who gave the assistant an average score in both categories and those (④) who gave the assistant a very good score in general, but rated S below average.

In detail, Table 2 shows the scores (average and median) for the main functionalities of EXTREMO, where for all the questions posed to the participants with a positive tone in a 5-item Likert scale (from 1, meaning *strongly disagree*, to 5, meaning *strongly agree*), the assistant obtained a score between 4.50 and 5.00 for the importer, the query and the integration mechanisms, and for the general perceived utility. From all the questions posed with a negative tone (where 1 is good and 5 is bad), only the importer mechanism obtained a 2.60 in average, but in median it obtained a 2.00 out of 5, and the rest of the negatives questions were ranked between 1.00 and 2.00.

Table 3 summarizes the results obtained for the set of hypotheses presented in Sect. 5.3, where in average, the $H_{2,U}$ and the $H_{2,S}$ were *accepted*. In average for the U variable, the assistant obtained a score of 70 out of 100 (*acceptable* or *good* in terms of adjective ratings [73]) and for the S variable

[13] https://github.com/angel539/extremo/wiki/Control-Group.

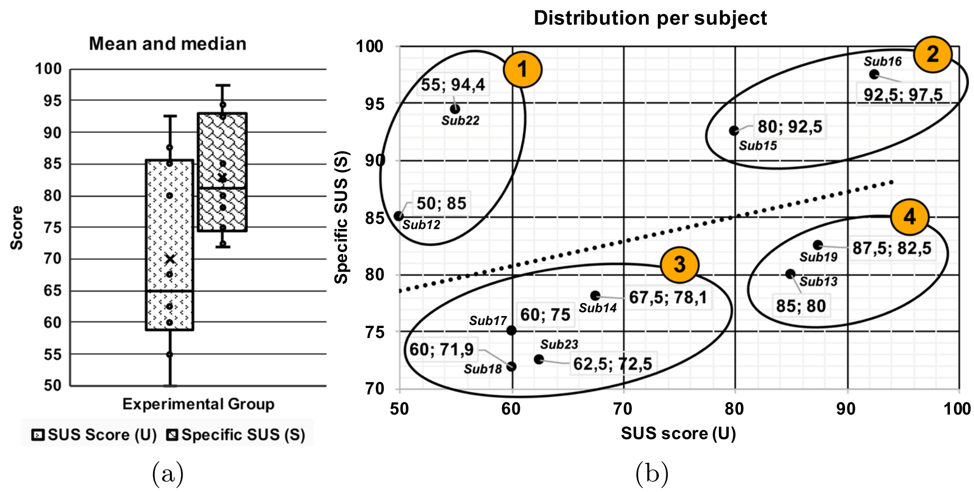


Fig. 8 **a** Box plots with the results obtained for the SUS score (U) and the Specific SUS (S). **b** Distribution per subject

Table 2 Scores obtained for the main features used during the experiment

Feature	Question	Tone	Average	Median
Importer	Useful for the task	Positive	4.90	5.00
	Problems	Negative	2.60	2.00
Query mechanism	Useful for the task	Positive	4.00	4.50
	Mismatches found	Negative	1.60	1.00
Integration	Useful for the task	Positive	4.50	4.50
	Discrepancies found	Negative	1.40	1.00
General utility	General opinion	Positive	4.80	5.00
	Prefer Editor without assistant	Negative	1.80	2.00

Table 3 Results obtained for the SUS Score (U) and the Specific SUS (S)

SUS Score (U)			Specific SUS (S)		
Metric	Result	H_U testing	Metric	Result	H_S testing
Average	70.0	$H_{2,U}$ *accepted*	Average	82.9	$H_{2,S}$ *accepted*
Median	65.0	$H_{1,U}$ *accepted*	Median	81.3	$H_{2,S}$ *accepted*
Stand. Dev	15		Stand. Dev	9.22	

the assistant obtained a score of 82.9 out of 100 (*acceptable* or *excellent* in terms of adjective ratings).

In summary, **the participants found our approach for modelling assistance useful**, and usefulness of EXTREMO's features were positively perceived. Additional details about the results obtained in the *experimental group* can be found in the Project's website.[14]

6.5 (RQ3) Completeness and correctness of the artefacts collected

Next, we compare the artefacts collected from both groups (variable MM presented in Sect. 5.3) using various quality metrics, i.e. precision as an indicator of *correctness*, recall as an indicator of *completeness*, and F-measures [77]. Also, we indicate the overall accuracy and error rate achieved in both groups. To compute those metrics, we compared the meta-model *contents* with a reference solution that was reached by consensus between the authors (presented in Sect. 4.3 and denoted as $Sub00$). Our solution ($Sub00$) was composed of 68 elements in total, including among them 22 classes, 23 attributes and 8 references. This way, we classify an element in a participant meta-model as true positive (TP) if it also belongs to our reference meta-model. An element is classified as false positive (FP), if it belongs to the participant meta-model, but not to our reference meta-model. Finally, an element is a false negative (FN) if it is expected by the reference meta-model, but missing in a participant meta-model. No negatives conditions were established a priori, and so we do not have true negatives (TNs).

The accuracy (ACC) and the error rate (ERR) are computed following Eqs. 1 and 2.

$$\mathrm{ACC} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FN} + \mathrm{FP}} \tag{1}$$

$$\mathrm{ERR} = \frac{\mathrm{FP} + \mathrm{FN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FN} + \mathrm{FP}} \tag{2}$$

[14] https://github.com/angel539/extremo/wiki/Experimental-Group.

Precision (PREC) is calculated using Eq. 3 and gives the level of *correctness*, while recall (REC) is calculated using Eq. 4 and gives the level of *completeness*.

$$\text{PREC} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{3}$$

$$\text{REC} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{4}$$

Finally, because a meta-model with high precision might have a bad recall and vice versa, usually the harmonic-mean (called $F_1\!-\!Score$) is computed following Eq. 5 with $\beta = 1$.

$$F_\beta = \frac{(\beta^2 + 1) * \text{PREC} * \text{REC}}{(\beta^2 * \text{PREC}) + \text{REC}} \text{ where } (0 \leq \beta \leq +\infty) \tag{5}$$

In particular, the matching method used to compare the meta-model *contents* presented on this paper is based on EMFCompare,[15] using names as identifiers and distinguishing meta-model elements by their type (i.e. we distinguished if elements were classes, attributes, references or other kind of meta-model element). We also explored other alternatives based on matching methods less strict. For example, comparing flattened collections of the classes obtained, or providing means for *inexact matching* based on the comparison of words using their lexical roots. For the sake of brevity, all the alternatives explored for meta-model comparison are available at the project's repository[16] and the replication package.

Table 4 shows the results obtained in both groups for the set of quality metrics described above. In average, the experimental group presented higher levels of precision (or *correctness*) and higher levels of recall (or *completeness*) compared to the control group. Also, the subjects in the experimental group committed less errors (measured with the error rate). In general, the number of TPs or elements found in a participant meta-model and belonging to our reference meta-model, are increased using our approach. Finally, subjects in the experimental group have a high $F_1\!-\!Score$ (0.5114 in average), meaning high combined *correctness* and *completeness*.

Graphically, Fig. 9a shows the Precision-Recall Curve (PRC) [78] of our experiment indicating the metrics obtained per subject. With thresholds situated in $Q2$ (or median) and $Q3$ for both values, we can determine that most meta-models of the control group fell below the median of recall ($Q2$ (REC) = 0.1176) and below the median of precision ($Q2$ (PREC) = 0.2750). This region is called *"low accuracy"* region (written *low-ACC region* in Figure). Only the meta-model provided by $Sub01$ is above the median of recall and precision in the control group.

Table 4 Metrics obtained in both groups compared to our reference solution ($Sub00$)

Metric	Average	Median
Control Group (C)		
TP	3.7	3.0
TN	0.0	0.0
FN	64.3	65.0
FP	21.2	22.5
N	89.2	90.5
ACC	0.0417	0.0339
ERR	0.9583	0.9661
PREC	0.1499	0.1282
REC	0.0544	0.0441
F_1	0.0790	0.0656
Experimental Group (E)		
TP	32.7	28.0
TN	0.0	0.0
FN	35.3	40.0
FP	21.8	16.5
N	89.8	84.5
ACC	0.3839	0.3538
ERR	0.6161	0.6462
PREC	0.5995	0.6955
REC	0.4809	0.4118
F_1	0.5114	0.5222

TP true positives, FN false negatives, TN true negatives, FP false positives, N number of elements in total, ACC accuracy, ERR error rate, $PREC$ precision, REC recall, F_1 $F_1\!-\!Score$

Most meta-models of the experimental group are above the median of recall and precision, with four of them situated above the $Q3$ for both values ($Q3$ (REC) = 0.4118 and $Q3$ (PREC) = 0.6770), in a region called *"high accuracy"* (written *high-ACC region* in the figure). Only one meta-model from the experimental group ($Sub23$) fell in the region of low accuracy. The four meta-models of the *experimental group* that fell in the region of *"high accuracy"* are those built by $Sub12$, $Sub13$, $Sub16$ and $Sub19$.

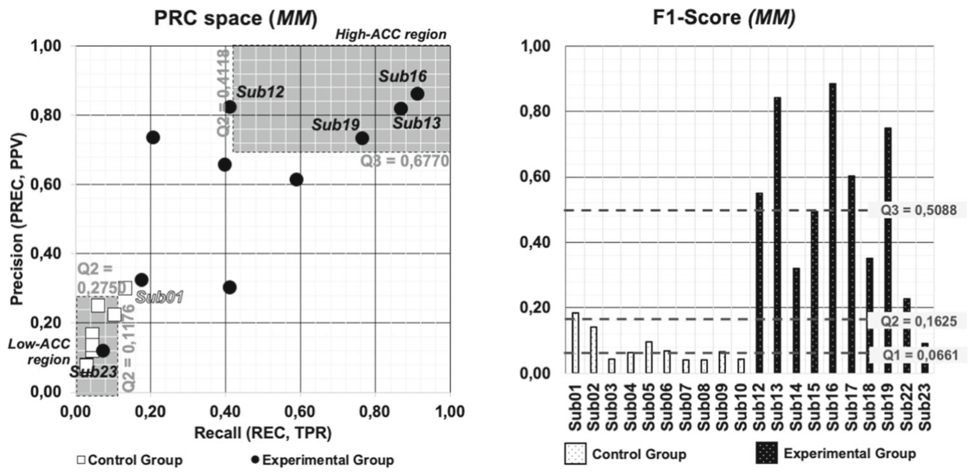
Figure 9b represents the $F1\!-\!Scores$ of each meta-model. It includes marks for $Q1$ ($Q1(F_1\!-\!Score)$ = 0.0661), $Q2$ ($Q2(F_1\!-\!Score)$ = 0.1625) and $Q3$ ($Q3(F_1\!-\!Score)$ = 0.5088). Half of the meta-models collected in the *control group* fell below the area marked by $Q1$ and four of them under $Q2$. Only the meta-model of $Sub01$ is above the median in the *control group*.

In summary, we can conclude that **our approach led to more accurate meta-models**, more complete and correct compared to the expected and reference solution. Hence,

[15] https://www.eclipse.org/emf/compare/.

[16] https://github.com/angel539/extremo/tree/comparator.

Fig. 9 PRC Space for the *participant* meta-models (**a**). $F_1-Scores$ (**b**)



(a) (b)

Table 5 Time to completion (TTC) to perform the modelling task and *modeller productivity* ($p_{\#MM}$)

	Min	Max	Avg	Q2
Control Group (C)				
$p_{\#MM}$	0.367	1.364	0.729	0.685
$\#MM$	18	30		
TTC	0 : 49	0 : 22		
$Subject$	$Sub06$	$Sub01$		
Experimental Group (E)				
$p_{\#MM}$	0.322	1.559	0.924	0.907
$\#MM$	19	92		
TTC^*	0 : 59	0 : 59		
$Subject$	$Sub14$	$Sub18$		

min (minimum productivity), max (maximum productivity), avg (average productivity) and $Q2$ (median value)

Table 6 F_β scores obtained for both groups. *Average* and *Geo. Mean* (computed as the mean of all folds [80]) and *Median*

	$\beta = 0.5$	$\beta = 1$	$\beta = 2$
Control Group (C)			
$Average$	0.1095	0.0790	0.0872
$Median$	0.0962	0.0656	0.0777
Experimental Group (E)			
$Average$	0.5507	0.5114	0.4946
$Median$	0.5958	0.5222	0.4559

$H_{0,MM}$ can be rejected. Additional details can be found in the Project's website.[17]

6.6 (RQ4) Time to completion, productivity, efficiency, and effectiveness

Next, we evaluate the time to completion (TTC) in both groups. The time allowed per subject to perform the task was 1 h, but the participants could finish earlier if they considered that their solutions were complete. Table 5 presents the results obtained as an indicator of *modeller productivity* and computed as the ratio between the number of elements per meta-model and the time needed to complete the task ($p_{\#MM} = \#elements\ MM/TTC$).

In the *control group*, the fastest subject was $Sub01$, spending 22 minutes in the task and had the highest *modeller productivity* (1.36 elements per minute). In contrast, $Sub06$ had the lowest *modeller productivity* with 0.37 elements per minute. In the *experimental group*, $Sub18$ produced the most elements per minute (1.56) and $Sub14$ had the lowest *modeller productivity* (0.322). In average, the *experimental group* had higher *modeller productivity* ($avg(p_E) = 0.924$ against $avg(p_C) = 0.729$), even though the time required to produce their meta-models was higher.

In Sect. 6.5, we already discussed that the level of accuracy in the *experimental group* was higher than in the *control group*. Now, we will analyse the relation between the level of accuracy produced per group and the time to perform the task. For that purpose, we will use different values for β in Eq. 5. In practice, the origin of the F_β measures comes from the concept of *effectiveness* [79]. Depending on the value given to the parameter β, it assigns β times as much importance to *recall* than to *precision*, by balancing them. The most common values for β are 0.5 (favouring *precision*), 1 (the harmonic-mean) and 2 (favouring *recall*). If $F_{\beta=0} = PREC$ and if $F_{\beta\to+\infty} = REC$.

As seen in Table 6, in average the *experimental group* meta-models had 6.472 times a better $F_1-Score$ than in the *control group*, and 5.673 times a better $F_2-Score$, i.e.

[17] https://github.com/angel539/extremo/wiki/Artifacts-Evaluation.

the meta-models had a significant higher level of *recall* or *completeness*. In terms of $F_{0.5}-Score$, where the *precision* or *correctness* is more valued, the difference was slightly less significant (in average, 5.031 times better in the *experimental group*).

Graphically, the *modeller performance* can be represented as the relation between *productivity* ($p_{\#MM}$, doing the meta-modelling task in a specific amount of time) and *effectiveness* (F_β, doing the meta-modelling task in the right way). Hence, the scatter plot in Fig. 10 displays *precision* ($F_{\beta=0}$) and *recall* ($F_{\beta\rightarrow+\infty}$) in Cartesian coordinates with thresholds situated in $Q2$ (the median) to determine the region of *"high effectiveness and high efficiency"*.

Figure 10a compares the *productivity* ($p_{\#MM}$) with *precision* (PREC or $F_{\beta=0}$) and Fig. 10b with *recall* (REC or $F_{\beta\rightarrow+\infty}$). Overall, five meta-models of the *experimental group* reached the region with the highest performance for both cases ($Sub13$, $Sub16$, $Sub17$, $Sub18$ and $Sub19$), while only one from the *control group* is included in that region ($Sub01$). In addition, the meta-models of the *experimental group* were more *precise* or more *correct* in general, as they were chiefly above $Q2(PREC)$ with a higher distance from the threshold.

In summary, we can conclude that **our approach may improve modeller performance significantly** in terms of *effectiveness* and *efficiency*, and the $H_{0,TTC}$ can be *rejected*. Additional details can be found in the Project's website.[18]

7 Discussion

Next, we present some points of discussion derived from the results presented in Sect. 6. Thus, we discuss on the *factors* that influenced the participants the most in the perceived usability (Sect. 7.1); on how large is the effect of using our modelling assistant in the *effectiveness* of the artefacts and the *performance* of the *experimental group*'s participants (Sect. 7.2); and on the implications of the results and our findings for future modelling assistants (Sect. 7.3).

7.1 Most influencing psychometric factors in usability

Looking back at our analysis of usability, the subjects grouped in label ① in Fig. 8b have a dissenting opinion from the rest in terms of usability (U), while the subjects in group ③ make the average decrease. Thus, we will discuss a *psychometric evaluation* of the SUS for these subjects to analyse which *factors* influenced them the most in the actual usability perceived [81–84].

The scoring of the SUS implies two steps [27]: ($step$ 1) we subtract one point from the scores given to odd questions, and scores for even questions must be subtracted from 5; ($step$ 2) the result (up to 40) must be scaled to 100, multiplying the obtained score by 2.5. Hence, if we want to take a SUS's partial score for a particular subject, we can follow the recommended guidelines [81] and change the *factor of scale* based on convenience depending on the number of questions involved in the partial SUS or $step$ 1.

This way, Table 7 shows the results obtained in terms of *psychometric factors* along with the questions used to calculate the concrete factor (column *Qs. involved*).

As it can be noted by the scores in Table 7, the subjects have a dissenting opinion specially in terms of the Learnable factor as defined in [81] (involving questions $Q4$ and $Q10$ of the SUS) and [83] (questions $Q4$, $Q6$, $Q8$ and $Q10$).

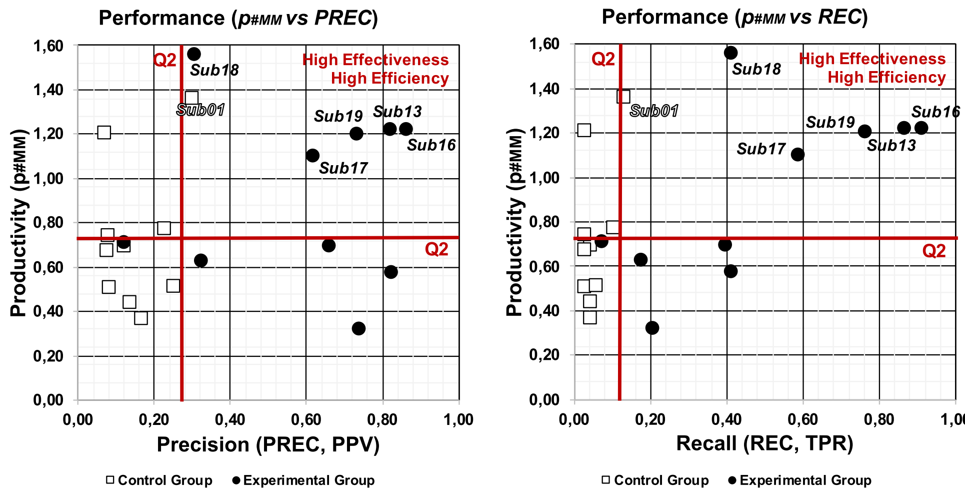
Table 8 focusses in the questions involved in the Learnable factor [81,83] (all of them have a negative tone, so a high value is bad). As it can be noted, $Q4$ has the strongest influence in this factor for all the subjects evaluated in Table 7. Overall, these subjects claimed that *"I think that I would need assistance to be able to use this tool"*. In contrast, as positive aspect, they stated that there was not too much inconsistency in the tool ($Q6$).

Next, we list some of the opinions of those subjects expressed in the open-ended questions of the *Specific Questionnaire*:

- Positive aspects:
  - For $Sub12$: *"(i) Import of resources, (ii) query and (iii) adding EClasses to my own ecore model."*
  - For $Sub14$: *"Good for bootstrapping a model. Simple workflow (for basics)"*
  - For $Sub22$: *"(i) Easy to navigate, (ii) Easy to import, (iii) Easy to integrate with model."*
- Negative aspects:
  - For $Sub12$: *"I found the given task too complicated for the given time. A very simple task with very simple repositories would be better."*
  - For $Sub17$: *"I am confused where to look for my results. Fix some bugs. Documentation!!!!"*
  - For $Sub22$: *"I would need more guidelines on how to use the tool."*

This way, for those subjects the tool was *"Easy to use"*, *"Easy to navigate"*, *"Good for bootstrapping a model"*, but the description of the task (belonging to the financial domain) biased the score provided by the $Sub12$, the documentation biased the score provided by the $Sub17$ and the guidelines provided biased the score provided by the $Sub22$. In contrast, when they were asked for their general opinion about the

[18] https://github.com/angel539/extremo/wiki/Artifacts-Evaluation.



(a) (b)

Fig. 10 Performance evaluation of the subjects: **a** Productivity ($p_{\#MM}$) versus PREC (*precision*, $F_{\beta=0}$) and **b** REC (*recall*, $F_{\beta\rightarrow+\infty}$)

Table 7 Psychometric evaluation of the SUS for the subjects in groups ① and ③ in Fig. 8

Psychometric Evaluation of the outliers							
Metric	Qs. involved	*Sub*12	*Sub*14	*Sub*17	*Sub*18	*Sub*22	*Sub*23
SUS Score (U)	1-10	50.00	67.50	60.00	60.00	55.00	62.50
Usable [81]	1,2,3,5,6,7,8,9	56.25	65.63	68.75	59.38	62.50	65.63
Learnable [81]	4,10	25.00	75.00	25.00	62.50	25.00	50.00
Positive [82]	1,3,5,7,9	55.00	60.00	70.00	55.00	60.00	60.00
Negative [82]	2,4,6,8,10	45.00	75.00	50.00	65.00	50.00	65.00
Usable [83]	1,2,3,5,7,9	58.38	66.72	70.89	54.21	58.38	62.55
Learnable [83]	4,6,8,10	37.50	68.75	43.75	68.75	50.00	62.50
Confidence [84]	1,9	50.00	50.00	75.00	37.50	75.00	62.50
Consistency [84]	2,3,4,5,6,7,8,10	50.00	71.80	56.25	65.63	50.00	62.50

Table 8 Learnable factor for the subjects grouped in groups ① and ③ in Fig. 8

Subject	(Q4) *I would need assistance*	(Q6) Too much inconsistency	(Q8) *It is very awkward to use*	(Q10) *I need to learn a lot of things*
*Sub*12	4	3	3	4
*Sub*14	3	1	4	1
*Sub*17	4	3	2	4
*Sub*18	3	1	3	2
*Sub*22	5	1	3	3
*Sub*23	4	2	2	2

Table 9 Scorecard of $Q10$ in the *Specific Questionnaire* for the subjects in groups ① and ③ in Fig. 8

I would prefer to use the model editor without assistant						
	*Sub*12	*Sub*14	*Sub*17	*Sub*18	*Sub*22	*Sub*23
Score	2	3	1	2	1	2

assistant ($Q10$, in a 5-item Likert scale [71]) in the *Specific Questionnaire*, they stated that they actually prefer to use the model editor with the assistant, as Table 9 shows.

Therefore, we can conclude that most of subjects were mainly influenced by the fact of having to learn a new tool in a limited time frame (1 h) while they also were performing the task.

7.2 Effect of using our approach in the artefacts collected

Next, we discuss the *effect size* of using our modelling assistant in the artefacts collected. For that purpose, we use Cohen's d and Hedges's g to compute the size of the effect. Both metrics are dependent between each other and range from 0 to $+\infty$, where 0 means no effect at all and starting in 0.8 for Cohen's d a *large effect* [85]. For small samples, it is common to refer also to Cohen's d_s as an standardized mean difference between two groups of independent observations. In addition, Hedges's g_s is considered to be an unbiased version of Cohen's d_s.

Table 10 shows the results of both groups in terms of *effectiveness* or F_β measures: the effect sizes (d and g) or difference between means, the confidence intervals (CI, a metric to quantify the *uncertainty* about our estimates) and the p-value or the possibility of accepting the H_0 in the future [86,87]. We chose the geometric mean as a general representation of the mean.

Table 10 Effect Size of using our modelling assistant in the artefacts collected: $H_{0,MM}$ rejected with a power $\approx$ 97%

F_1-Score CI and Effect Size Independent Samples				95% CI $\bar{x}_{diff}$		Effect Size	
	$\bar{x}$	SD	n	Low	0.1815	Cohen's d_s	1.8906
Experimental Group	0.4296	0.2655	10	High	0.5401	Cohen's d	1.9928
Control Group	0.0687	0.0483	10	df	18	Hedges's g_s	1.1807
Difference ($\bar{x}_{diff}$)	0.3608			t	4.22	CL effect size	0.9094
				p	0.0005	power	$\approx$ 97%

In the *experimental group*, the *effectiveness* was clearly higher ($\bar{x} = 0.4295$, $SD = 0.2655$) compared to the *control group* ($\bar{x} = 0.0687$, $SD = 0.0482$), $t(18) = 4.22$, $p = .0005$, 95% CI [0.1815, 0.5401], Hedges's $g_s = 1.8107$. The Common Language (CL) effect size [88] yields a probability of 90.94% that for a randomly selected pair of participants the *effectiveness* of the one of the *experimental group* is higher than the *effectiveness* of the one of the *control group*. In addition, by computing the *power* of our rejection [89] we can reject our $H_{0,MM}$ with $\approx$ 97%.

When the *effectiveness* tends to reach peak values for β or when we take into account the *correctness* (PREC) and the *completeness* (REC) in isolation, we also found that the effect size increased in case of *precision* (Hedges's $g_s(PREC) = 2.0147$) and decreased in case of *recall* (Hedges's $g_s(REC) = 1.5267$). This is because the meta-models of the *experimental group* are in general more *precise* and *correct* than those of the *control group*, as we already discussed in Sect. 6.6.

In Sect. 6.6, we measured *performance* as the relation between *productivity* ($p_{\#MM}$, doing the meta-modelling task in a specific amount of time) and *effectiveness* (F_β, doing the meta-modelling task in the right way) in order to reject $H_{0,EFF}$. In that section, we said that the *productivity* did not increase so much from the *experimental group* to the *control group*. By performing the same evaluation as that one presented in Table 10, we detected that our modelling assistant has a low effect in the productivity for the same modelling task ($g_s = 0.4513$). In contrast, when we established the relation between productivity and effectiveness by computing the ratio between both measures ($performance = F_1 - Score/p_{\#MM}$) we detected that in the *experimental group* the general *performance* was again clearly higher compared with the *control group* ($p = .0003$, $g_s = 1.8862$). In that case, the CL effect size also indicated that the probability that for randomly selected pair of participants the *performance* detected for one subject from the *experimental group* is higher than the *performance* of a subject from the *control group* is 91.81%.

Hence, we can conclude that introducing our modelling assistant during the modelling task has a large effect in the creation of the new meta-model.

7.3 Implications for future modelling assistants

Next, we discuss the implications of the results of our study for future modelling assistants.

– *Increased (meta-)model quality*. The findings in Sect. 6.5 indicate that using EXTREMO may lead to more detailed and more accurate meta-models. The main feature of EXTREMO is the possibility to reuse information from heterogeneous sources. Hence, we may expect similar advantages from assistants targeting modelling completion, based on information reuse. Our experiment consisted in creating a meta-model, and we may expect similar results for other types of structural models, like class diagrams, entity-relationship diagrams, or mind maps. Nonetheless, this type of assistants might have different results for other types of models, like behavioural ones.
  Please note that modelling assistants based on syntactical means (e.g. analysing the meta-model structure and integrity constraints, and proposing model completions) may not achieve the goal of increased level of model detail and accuracy, since they focus instead on linguistic correctness of the model being built.
– *Quality of data sources*. An assistant can only be useful if the data sources the recommendations rely on have enough quality. In the case of our experiment, we manually selected and filtered appropriate data sources, from trusted origin (mostly from the OMG). The recent trend of using machine learning techniques in MDE [13,90] is also triggering the development of techniques to facilitate the collection and cleaning of large datasets of models [91]. However, these techniques would need to be extended to heterogeneous sources (to serve assistants like EXTREMO) and include ways to deal with inconsistency and contradictions in the data.
– *Repeatability of modelling*. Repeatability is a desired quality of engineering processes, and it would be valuable if assistants would increase repeatability in modelling. Assistants based on information reuse may offer assistance leading to more repeatable processes and increased quality models. However, this may not be applicable to

assistance based on syntactical means, even though uniformity can be achieved here by recommending model completions based on quality criteria (e.g. modelling patterns).

– *General acceptance of assistance, but learnability is a strong factor on usability*. Even though we found (in Sect. 6.4) that the participants found our approach for modelling assistance useful, we also found that learnability has a strong effect on the perceived usability (Sect. 7.1). This way, builders of future modelling assistants may pay special attention to designing an easy-to-learn tool, over adding more sophisticated features requiring more specialized knowledge from the users.
– *Increased productivity and performance*. Our experiment (cf. Section 6.6) shows that our assistant may improve modeller productivity (elements/completion time), and performance (which also considers the quality of the output). We would expect other assistants based on information reuse to have similar effects on structural models. Faster coding is often the major selling point of coding assistants, like Kite or Github Copilot.[19] Interestingly, our approach did not lead to faster completion times. Instead, we may hypothesize that syntactical modelling assistants may actually help in obtaining faster completion times, but this would need to be assessed in user experiments.
– *Need for user studies of modelling assistants*. According to a recent survey [7], most modelling assistants are either not evaluated, or evaluated off-line (i.e. with no users involved). We claim that user studies are needed to understand the benefits of modelling assistants for solving practical tasks.
– *Basis for future evaluations*. As we provide all artefacts needed to execute the experiment and to analyse the results, we hope to provide a good basis for performing future evaluations of reuse-based modelling assistants. Currently, the only precondition to reuse the provided artefacts is to have EMF-based models. However, this dependency is only on the technical level and the general evaluation methodology applied should be also reusable for models coming from other technological spaces. To further facilitate the design and execution of evaluations of modelling assistants, we also propose the construction of tool support to automate the different experimental phases: experiment design, group formation, task allocation, artefact collection, evaluation, and data analysis and presentation.

[19] https://github.com/features/copilot/.

8 Threats to validity

Next, we analyse the threats to the validity of our study. We report threats for four types of validity [92]: *conclusion*, *internal*, *external* and *construct*.
Conclusion validity is concerned with the conclusions about the relationship between the treatment and the outcome. This way, it concerns the data and statistical analysis performed to the results and the composition of subjects (how large is the sample size). The main threat detected was the final number of subjects we had in each group (10 subjects per group). In relation to this threat, several investigations have highlighted how a low number of subjects may lead to misinterpretation of the p-values and the *power* associated with the hypotheses rejection [93] or that the *likelyhood* of detecting a false alarm or Type I Error increases if the data set collected is not reliable enough [94]. For this reason, and under the premise that our sample size was chosen based on a *feasibility criteria*, i.e. we gathered all the subjects we could in a determined slot of time, we decided to perform a data analysis using information retrieval techniques to validate whether the introduction of our modelling assistant presented any kind of effect in the artefacts collected in both groups and, measure the size of that effect. In that sense, we think that by using standard metrics for measuring the *correctness* and the *completeness* of the artefacts collected we have empirically proven that effectively our assistant has a strong effect in the results obtained in the *experimental group* not only in the sense of the *accuracy* of those artefacts but also in the *performance* of the participants of that group. Also, we have analysed in different ways how *effectiveness* has been significantly improved in the *experimental group* compared to the *control group* (cf. Sections 6.6 and 7.2).
Internal validity concerns confounding factors which might affect the results of our study. Hence, the main threats may be related to the instrumentation and the selection of the subjects to guarantee a high level of reliability in the experiment set-up. Concerning the instruments used, we decided to rely on the SUS for being considered a simple and reliable tool for usability evaluations [95]. Additionally, the questionnaire for measuring the *utility* or S variable was prepared ad hoc, but to avoid any possible bias, it was based on a bipolar 5-item Likert scale in order to refute every question with a double check. Such questionnaire was refined using the pilot study described in Sect. 5.4. Moreover, by performing a Cronbach's α [96] statistical test we can determine that the answers provided by the participants in the *Specific Questionnaire* are highly correlated and the set of items defined in that questionnaire show to be an accurate estimate for its designed (Cronbach's $\alpha = 0.8725$).

For measuring accuracy (cf. Section 6.5), we compared the meta-models created by the subjects with a meta-model created by us. To ensure a reasonable baseline for comparison, this meta-model was built by consensus, based on solutions built by this paper's authors using EXTREMO itself. Two of the authors of the paper have an academic background on business, and one of them also in finance, and was thus acting as a domain expert. Please note that similar approaches can be found in the literature to create comparison baselines for modelling experiments [51]. Finally, we assigned 1 h to the experiment. One may argue that having the same amount of time would be detrimental to the participants in the control group. However, since all subjects of the control group finished well before time (cf. Table 5), we can conclude that it did not have a meaningful effect.

Regarding the subjects of the experiment, there are two main threats. The first one is related to the level of expertise of both groups, more specifically on whether in the *experimental group* the subjects may present a higher experience level than in the *control group*. We asked all the subjects for their experience in the technologies involved (questions Q7 to Q11 of the Sect. A.2 in "Appendix A") in different ways (with open-ended questions and rating themselves). Actually, in average the *control group* seemed to be more experienced for the main technologies involved in the experiment (EMF, modelling and meta-modelling) than the *experimental group* (cf. Section 6.2), while they presented worst results in the *completeness* of the artefacts collected and even worse results in their *correctness* (Sect. 6.5). In the sense of whether our results could be extrapolated to other environments, in the *experimental group* actually, 60% of the participants stated that they were working in *Industry*. Since all participants had a BSc degree, the sample could be representative of some teams in industry.

The second threat detected concerns whether having non-native English speakers as subjects could invalidate some of the answers, but in that sense no one expressed problems in understanding the documentation prepared for the experiment (Sect. 5.4). Also, most of the participants rated themselves to be at least at a *B2* English level (Sect. 6.2). Finally, we avoided any kind of interaction between groups by separating both groups, experimental and control, even though we chose subjects with a similar background and experience (Sect. 5.1).

External validity refers to which extent the presented results can be generalized beyond the presented study. The main threat in that sense is having used mainly students of a Master's degree as subjects of the experiment. The use of students is common in user studies within our field, as we have discussed in Sect. 2. Also, the use of students as subjects in controlled experiments is an ongoing discussion in the software engineering community [67–70]. In Sect. 5, we characterized the *subjects* of the experiment in terms of their attributes considering also that they should serve as proxies of professional modellers. In the laboratory context chosen, subjects presented relevant and recent experience for the purpose of the study and the technologies involved. In fact, most of the subjects of the *experimental group* were also working in *Industry*, while in the *control group*, most of them were also working in *Academia*. Hence, we consider that the subjects may be a reasonable sample for the purpose of our study. However, replicating the experiment in an industrial setting would help strengthening our conclusions.

Running more experiments consisting on building meta-models in other domains would possibly bring more confidence in the results obtained. Our rationale for the choice of a meta-model in the financial domain was to use a domain where subjects would have little or no experience, to avoid any bias that this experience could bring. Other settings—where subjects have more knowledge about the domain—are also worth studying to assess the advantages of an assistant in those cases. Our intuition is that the usefulness of the assistant would be higher the less knowledge the modeller has on that domain, and the higher the quality of the information sources used by the assistant (as mentioned in Sect. 7.3).

While the experiment consisted in the creation of a meta-model, we believe the results could be extrapolated to structural models. However, investigating generalization to behavioural models would require additional experiments. As discussed in Sect. 7.3, we believe the results and findings are applicable to other assistants based on information reuse, but for assistants based on syntactical completions, experiments would need to be performed.

Construct validity is concerned with the relationship between the theory of the experiment and what is observed. The major threat we found is whether the domain chosen for the description of the task (belonging to the financial domain) was too complicated for the purpose of the study. However, we chose a highly specialized field as being a representative of a foreign domain for an engineer, reflecting a typical situation with the design of DSLs. This allowed us to extract more results and reinterpret our data analysis by learning how participants used our tool (see Sect. 7.1). As mentioned in Sect. 4.1, in the informed consent (cf. Section A.1 in "Appendix A") we assessed that the domain was actually outside of each subject's experience.

9 Conclusions and future work

In this paper, we have presented an empirical evaluation of EXTREMO, a (meta-)modelling assistant supporting information reuse from heterogeneous sources. The experiment was directed to check if reusing heterogeneous knowledge helps in building more complete and correct meta-models, in a more efficient way; and to understand the perceived usability and utility of the tool.

The experiment results indicate that developers would need assistance to create a meta-model in a domain they

Table 11 Opinion Questionnaire

Id	Question	Answer
Q1	Have you consulted the information resources provided with the description of the task? (mark one)	(Yes / No)
Q2	Do you think the list of resources provided with the description of the task were sufficient to perform the task? (mark one)	(Yes / No)
Q3	Do you think the list of resources provided with the description of the task were accurate to perform the task? (mark one)	(Yes / No)
Q4	Did you miss any kind of tool to consult the resources in a uniform way during the meta-model construction phase? (mark one)	(Yes / No)
Q5	Do you think a tool to assist you during the meta-model construction phase would be useful? (mark one)	(Yes / No)
Q6	Please, indicate which functionalities do you think this tool could have	(Open answer)

are not expert in, and such lack of knowledge may result in a low-quality meta-model with omissions. A modelling assistant may help in obtaining more *correct* and *complete* meta-models, which can potentially be produced in more repeatable, standardized ways, by reusing existing information. Moreover, even though an assistant may not lead to *faster* completion times, overall engineer *performance* (the relation between productivity and effectiveness) is improved. Regarding usability, participants generally perceived EXTREMO as a useful tool and were willing to promote its use in new modelling projects. In terms of utility, the most common terms that appear in the participants' opinions demonstrated that our approach was easy to use, highlighting the query mechanism. While the experiment was specific to EXTREMO, we discussed on the implications of our main findings for future modelling assistants.

As future work, we would like to explore pro-active ways of assistance, where the system proposes content related to the model being created. For that purpose we may use information retrieval techniques. We would also like to incorporate support for the discovery of relevant information sources, which then can be selected, cleaned and incorporated into EXTREMO's common data model. We will also extend this common model with the possibility to annotate data sources with weights regarding their quality levels, to influence the recommendation. In addition, we are currently benefiting from the comments of the participants to improve the tool. Finally, while this paper has reported on the evaluation of EXTREMO, a challenge for the MDE community is the comparison—based on user studies—of the different recommenders currently proposed [9,11–13,13–15], to better understand the strong and weak points of each other and the complementarity of their techniques. In the future, we aim at establishing a forum enabling benchmarking and comparison of recommenders, especially focusing on user evaluations. In addition, the study presented here could serve as a basis for replication studies with other recommenders to compare results. To further facilitate the design and execution of evaluations of modelling assistants, we aim at developing tool support to automate the different experimental phases: experiment design, group formation, task allocation, artefact collection, evaluation, and result data analysis and presentation.

A Appendix: Evaluation material

This appendix contains the documents provided to the participants in our evaluation. Section A.1 shows a condensed version of the *Informed Consent* to participate in the user study. Section A.2 contains the *Demographic Questionnaire*

Table 12 General Questionnaire (based on the SUS Questionnaire [27])

Id	Question	Answer
Q1	I think that I would like to use this tool frequently	(1–5)
Q2	I found this tool unnecessarily complex	(1–5)
Q3	I thought this tool was easy to use	(1–5)
Q4	I think that I would need assistance to be able to use this tool	(1–5)
Q5	I found the various functions in this tool were well integrated	(1–5)
Q6	I thought there was too much inconsistency in this tool	(1–5)
Q7	I would imagine that most people would learn to use this tool very quickly	(1–5)
Q8	I found this tool very cumbersome/awkward to use	(1–5)
Q9	I felt very confident using this tool	(1–5)
Q10	I need to learn a lot of things before I could get going with this tool	(1–5)

1 = Strongly disagree, 2 = Disagree, 3 = Neutral, 4 = Agree, 5 = Strongly agree

Table 13 Specific Questionnaire

Id	Question	Answer
Q1	I think that the importer mechanism is useful for the task	(1–5)
Q2	I found problems in the importer mechanism using the assistant	(1–5)
Q3	I think that the query mechanism is useful for the task	(1–5)
Q4	I found mismatches in the query results using the assistant	(1–5)
Q5	I think that the integration with the set of modelling is useful to perform the task	(1–5)
Q6	I found discrepancies in the fragments created by the integration system	(1–5)
Q7	I think that the constraint evaluation functionality is useful for the task	(1–5)
Q8	I found mismatches in the constraint evaluation results using the assistant	(1–5)
Q9	I think that the use of this kind of assistants together with a modelling tool improves the modelling task	(1–5)
Q10	I would prefer to use the model editor without the assistant	(1–5)
OC1	Please describe the kind of queries you have used (if any)	Open-ended
OC2	Please indicate three good aspects you want to highlight about the tool	Open-ended
OC3	Please indicate three suggestions for improvement	Open-ended

1 = Strongly disagree, 2 = Disagree, 3 = Neutral, 4 = Agree, 5 = Strongly agree

used in the survey. Section A.3 contains a condensed version of the *Description of the Task* provided to the subjects of the evaluation. Section A.4 contains the *Opinion Questionnaire* used in the control group. Finally, Sect. A.5 contains the following documents provided to the participants in the evaluation group: the General Questionnaire (according to the System Usability Scale) shown in Table 12 and the Specific Questionnaire shown in Table 13.

A.1 Informed consent

The *Informed Consent* had to be signed by all the participants in order to cover the basic ethical aspects [74,75] considered by the project.

> You are being asked to take part in a research study to evaluate the usefulness of a modelling assistant to perform a modelling task. During the study, you have to build a meta-model of a language to describe the process involved in the Interest Rate Swaps (IR Swaps) between companies that are customers of a bank. If you work or you know directly someone who works in the topic of the case study, please inform the research staff about this situation. Your participation in the study is completely voluntary. At the end of the experiment, you will be informed about the full purpose of the study, and we will provide you an answer that we consider is valid for the task. The experiment will require about 1 hour of your time. You have the right to withdraw from the study at any time without penalty. During the experiment, you will be asked to fulfill a set of questionnaires. Some questions are about your personal background (e.g., level of education) for statistical purposes. All the collected data will be guarded for research purposes, and anonymity is guaranteed. In the case you decide to follow with the experiment, you are giving your *informed and voluntary consent* to take part in this research study and we will maintain a copy of this consent and the results obtained for our records.

A.2 Demographic questionnaire

The *Demographic Questionnaire* contained 11-item, and it was handed out at the beginning of the experiment to both groups.

1. Specify your age__________
2. Specify your gender (mark one) ☐ Female ☐ Male ☐ Not listed: ______ ☐ Prefer Not to Answer
3. Specify your level of education (mark one)

Title Suppressed Due to Excessive Length

☐ Undergraduate Student
☐ Master Student
☐ PhD Student
☐ PhD
☐ Other ______

4. Do you have studies related to computer science? (mark one) ☐ Yes ☐ No
5. What level of English do you currently have? (mark one)

☐ Beginner (A1)
☐ Elementary (A2)
☐ Intermediate (B1)
☐ Upper Intermediate (B2)
☐ Advanced (C1)
☐ Proficient (C2)

6. You currently work in...

☐ Academia
☐ Industry
☐ Research center
☐ Student
☐ Other ______

7. State your experience with SW development______
8. State your experience with mod. languages______

Rate your experience in the following technologies:

	expert → ← novice				
9. Eclipse Modelling Framework	1	2	3	4	5
10. Modelling	1	2	3	4	5
11. Meta-modelling	1	2	3	4	5

A.3 Description of the task

It has some common statements presented to both groups:

> You have to build a language for the modelling of the process involved in the *Interest Rate Swaps (IR Swaps)* between companies that are customers of a bank. An interest rate swap is a derivative contract between two or more parties (also called *legs*) according to their desired specifications. Swaps are often used if a company can borrow money from a bank at one type of interest rate (for example, fixed) but prefers to pay the loan at a different type (for example, floating). Then, the company decides to swap its conditions with another entity making periodic payments based on an agreed amount. Because the parties involved in a swap trade over the counter (OTC), which means, they operate outside the conditions offered by the trading markets, each part needs to specify its conditions to operate. To guarantee the operation, the swapping process have to be executed over a blockchain-based ecosystem with an external entity acting as a validator. The language have to include elements showing the variety of financial entities involved, concepts from the financial domain and the information required to describe the interchange operation. To help you, you will receive a set of resources taken from: (*i*) the Finance Domain Task Force of the OMG[a], which includes information on financial entities in OWL and RDF formats; (*ii*) the *Blondie* Ontology, which contains information describing the interchange operation; (*iii*) Ecore files available on the OMG repository[b] with standard meta-models, such as BMM, BPEL or BPMN; and (*iv*) Ecore files available in open repositories with concepts in the banking domain.
>
> [a] https://www.omg.org/fdtf/projects.htm
> [b] The OMG is the standardization body behind many modelling standards such as UML, SysML, MOF or BPMN. (http://www.omg.org/spec/)

Some statements presented to the subjects of the *control group*:

> You will receive the 4.5.2 (Mars.2) version of the Eclipse IDE[a] with the Eclipse Modelling Tools package that has been installed in a virtual machine running Windows 7. You can open the resources using a text editor. In addition, you can use a web navigator to look for information and get familiar with the concepts of the domain. You have to build a meta-model describing the domain using the Ecore Model Editor. The meta-model is expected to be as complete as possible regarding the types of entities involved. Submit your solution sending an email to *Angel.MoraS@uam.es* with the file in ".ecore" format.
>
> [a] http://www.eclipse.org/downloads/packages/release/Mars/2

Some statements presented to the subjects of the *experimental group*:

> You will receive the 4.5.2 (Mars.2) version of the Eclipse IDE[a] with the Eclipse Modelling Tools package that has been installed in a virtual machine running Windows 7. In addition, we have installed EXTREMO, a tool for modelling and meta-modelling assistance. EXTREMO gathers heterogeneous information sources and represents them uniformly in a common repository. This enables their uniform querying and constraint evaluation. In addition, you can use a web navigator to look for information and get familiar with the concepts of the domain. You have to build a meta-model describing the domain using the Ecore Model Editor. The meta-model is expected to be as complete as possible regarding the types of entities involved. Submit your solution sending an email to *Angel.MoraS@uam.es* with the file in ".ecore" format.
>
> [a] http://www.eclipse.org/downloads/packages/release/Mars/2

A.4 Control group questionnaire

The *Opinion Questionnaire* contained 6-item, and it was handed out to the *control group* after having performed the task. In addition, subjects had the option to provide a rationale to each answer. For the sake of brevity, we omitted the space for the rationale in this appendix.

A.5 Experimental group questionnaires

The General Questionnaire and the Specific Questionnaire were answered by the *experimental group* in order to measure the usability perceived by language engineers about using the assistant during the modelling task, and the usefulness of EXTREMO's features. In both cases, all the questions were mandatory.

Acknowledgements We would like to thank the reviewers for their valuable comments. This work was supported by the Ministry of Education of Spain (FPU Grant FPU13/02698 and stay EST17/00803); the Spanish Ministry of Science and Innovation (PID2021-122270OB-I00); the R&D programme of the Madrid Region (P2018/TCS-4314); and the Austrian Federal Ministry for Digital and Economic Affairs and the National Foundation for Research, Technology and Development (CDG).

References

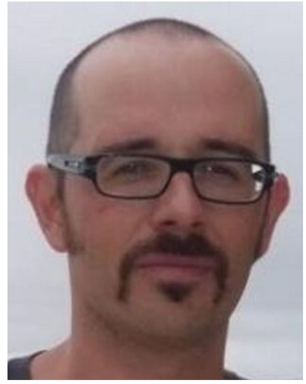
1. Brambilla, M., Cabot, J., Wimmer, M.: Model-Driven Software Engineering in Practice, 2nd edn. Morgan & Claypool, San Rafael (2017)
2. Schmidt, D.C.: Guest editor's introduction: model-driven engineering. Computer **39**(2), 25–31 (2006)
3. Kelly, S., Pohjonen, R.: Worst practices for domain-specific modeling. IEEE Softw. **26**(4), 22–29 (2009)
4. Eclipse. Eclipse Code Recommenders. https://marketplace.eclipse.org/content/eclipse-code-recommenders (2020)
5. Mens, K., Lozano, A.: Source code-based recommendation systems. In: Recommendation Systems in Software Engineering, pp. 93–130. Springer (2014)
6. Steinberg, D., Budinsky, F., Paternostro, M., Merks, E.: EMF: Eclipse Modeling Framework. Addison-Wesley, Boston (2008)
7. Almonte, L., Guerra, E., Cantador, I., de Lara, J.: Recommender systems in model-driven engineering. Softw. Syst. Model. **21**(1), 249–280 (2022)
8. Mussbacher, G., Combemale, B., Kienzle, J., Abrahão, S., Ali, H., Bencomo, N., Búr, M., Burgueño, L., Engels, G., Jeanjean, P., Jézéquel, J.-M., Kühne, T., Mosser, S., Sahraoui, H.A., Syriani, E., Varró, D., Weyssow, M.: Opportunities in intelligent modeling assistance. Softw. Syst. Model. **19**(5), 1045–1053 (2020)
9. Agt-Rickauer, H., Kutsche, R.-D., Sack, H.: Automated recommendation of related model elements for domain models. In: 6th International Conference on Model-Driven Engineering and Software Development (MODELSWARD), Revised Selected Papers, volume 991 of CCIS, pp. 134–158. Springer (2018)
10. Dyck, A., Ganser, A., Lichter, H.: A framework for model recommenders—requirements, architecture and tool support. In: MODELSWARD, pp. 282–290 (2014)
11. Elkamel, A., Gzara, M., Ben-Abdallah, H.: An UML class recommender system for software design. In: 13th IEEE/ACS International Conference of Computer Systems and Applications (AICCSA), pp. 1–8. IEEE Computer Society (2016)
12. Burgueño, L., Clarisó, R., Gérard, S., Li, S., Cabot, J.: An NLP-Based Architecture for the Autocompletion of Partial Domain Models. In: Advanced Information Systems Engineering—33rd

International Conference, CAiSE, volume 12751 of LNCS, pp. 91–106. Springer (2021)
13. Di Rocco, J., Di Sipio, C., Di Ruscio, D., Nguyen, P.T.: A GNN-based recommender system to assist the specification of metamodels and models. In: ACM/IEEE 24th International Conference on Model Driven Engineering Languages and Systems (MODELS), pp. 70–81 (2021)
14. Weyssow, M., Sahraoui, H., Syriani, E.: Recommending metamodel concepts during modeling activities with pre-trained language models. Softw. Syst. Model. (2022)
15. Almonte, L., Pérez-Soler, S., Guerra, E., Cantador, I., de Lara, J.: Automating the synthesis of recommender systems for modelling languages. In: SLE'21: 14th ACM SIGPLAN International Conference on Software Language Engineering, pp. 22–35. ACM (2021)
16. Pescador, A., de Lara, J.: DSL-maps: from requirements to design of domain-specific languages. In: Proceedings of ASE, pp. 438–443. ACM (2016)
17. Aquino, E.R., de Saqui-Sannes, P., Vingerhoeds, R.A.: A methodological assistant for use case diagrams. In: 8th International Conference on Model-Driven Engineering and Software Development (MODELSWARD), pp. 227–236. SciTePress (2020)
18. Cerqueira, T., Ramalho, F., Marinho, L.B.: A content-based approach for recommending UML sequence diagrams. In: 28th International Conference on Software Engineering and Knowledge Engineering (SEKE), pp. 644–649 (2016)
19. Segura, A.M., de Lara, J., Neubauer, P., Wimmer, M.: Automated modelling assistance by integrating heterogeneous information sources. Comput. Lang. Syst. Struct. **53**, 90–120 (2018)
20. Segura, Á.M., de Lara, J.: Extremo: an Eclipse plugin for modelling and meta-modelling assistance. Sci. Comput. Program. **180**, 71–80 (2019)
21. Stephan, M.: Towards a cognizant virtual software modeling assistant using model clones. In: 41st International Conference on Software Engineering: New Ideas and Emerging Results (NIER@ICSE), pp. 21–24. IEEE/ACM (2019)
22. Iovino, L., Barriga, A., Rutle, A., Heldal, R.: Model repair with quality-based reinforcement learning. J. Obj. Technol. **19**(2), 17:1-17:21 (2020)
23. Ohrndorf, M., Pietsch, C., Kelter, U., Kehrer, T.: ReVision: a tool for history-based model repair recommendations. In: 40th International Conference on Software Engineering (ICSE), Companion Proceeedings, pp. 105–108. ACM (2018)
24. Steimann, F., Ulke, B.: Generic model assist. In: Proceedings of MODELS, volume 8107 of Lecture Notes in Computer Science, pp. 18–34. Springer (2013)
25. Sen, S., Baudry, B., Vangheluwe, H.: Towards domain-specific model editors with automatic model completion. Simulation **86**(2), 109–126 (2010)
26. Mussbacher, G., Combemale, B., Abrahão, S., Bencomo, N., Burgueño, L., Engels, G., Kienzle, J., Kühne, T., Mosser, S., Sahraoui, H.A., Weyssow, M.: Towards an assessment grid for intelligent modeling assistance. In: Proceedings of MODELS Companion, pp. 48:1–48:10. ACM (2020)
27. Brooke, J., et al.: SUS-a quick and dirty usability scale. Usability Evaluat. Ind. **189**(194), 4–7 (1996)
28. Abrahão, S., Bourdeleau, F., Cheng, B.H.C., Kokaly, S., Paige, R.F., Störrle, H., Whittle, J.: User experience for model-driven engineering: Challenges and future directions. In: MODELS, pp. 229–236. IEEE Computer Society (2017)
29. Robillard, M.P., Walker, R.J., Zimmermann, T.: Recommendation systems for software engineering. IEEE Softw. **27**(4), 80–86 (2010)
30. Jackson, D.: Software Abstractions-Logic, Language, and Analysis. MIT Press, Cambridge (2006)
31. Macedo, N., Tiago, J., Cunha, A.: A feature-based classification of model repair approaches. IEEE Trans. Software Eng. **43**(7), 615–640 (2017)
32. Reder, A., Egyed, A.: Computing repair trees for resolving inconsistencies in design models. In: IEEE/ACM ASE, pp. 220–229. ACM (2012)
33. Habel, A., Sandmann, C.: Graph repair by graph programs. In: STAF Workshops, volume 11176 of Lecture Notes in Computer Science, pp. 431–446. Springer (2018)
34. Dyck, A., Ganser, A., Lichter, H.: Enabling model recommenders for command-enabled editors. In: MDEBE, pp. 12–21 (2013)
35. Dyck, A., Ganser, A., Lichter, H.: On designing recommenders for graphical domain modeling environments. In: MODELSWARD, pp. 291–299 (2014)
36. Hajiyev, E., Verbaere, M., de Moor, O., De Volder, K.: Codequest: querying source code with datalog. In: Proceedings of OOPSLA 2005, pp. 102–103. ACM (2005)
37. Linstead, E., Bajracharya, S.K., Ngo, T.C., Rigor, P., Videira Lopes, C., Baldi, P.: Sourcerer: mining and searching internet-scale software repositories. Data Min. Knowl. Discov. **18**(2), 300–336 (2009)
38. Mendieta, R., de la Vara, J.L., Llorens, J., Álvarez-Rodríguez, J.: Towards effective SysML model reuse. In: Proceedings of MODELSWARD, pp. 536–541. SCITEPRESS (2017)
39. Álvarez Rodríguez, J.M., Mendieta, R., de la Vara, J.L., Fraga, A., Morillo, J.L.: Enabling system artefact exchange and selection through a linked data layer. J. UCS **24**(11), 1536–1560 (2018)
40. Lucrédio, D., de Mattos Fortes, R.P., Whittle, J.: MOOGLE: a metamodel-based model search engine. Softw. Syst. Model. **11**(2), 183–208 (2012)
41. Hernández López, J.A., Cuadrado, J. S.: An efficient and scalable search engine for models. Softw. Syst. Model. (2021)
42. Bislimovska, B., Bozzon, A., Brambilla, M., Fraternali, P.: Textual and content-based search in repositories of web application models. TWEB **8**(2), 1–11 (2014)
43. Dijkman, R.M., Dumas, M., van Dongen, B.F., Käärik, R., Mendling, J.: Similarity of business process models: metrics and evaluation. Inf. Syst. **36**(2), 498–516 (2011)
44. Paige, R.F., Kolovos, D.S., Rose, L.M., Drivalos, N., Polack, F.A.C.: The design of a conceptual framework and technical infrastructure for model management language engineering. In: ICECCS, pp. 162–171. IEEE Computer Society (2009)
45. Carver, J.C., Syriani, E., Gray, J. Assessing the frequency of empirical evaluation in software modeling research. In: 1st Workshop on Experiences and Empirical Studies in Software Modelling (2011)
46. Whittle, J., Hutchinson, J.E., Rouncefield, M., Burden, H., Heldal, R.: A taxonomy of tool-related issues affecting the adoption of model-driven engineering. Software Syst. Model. **16**(2), 313–331 (2017)
47. Abrahão, S., Iborra, E., Vanderdonckt, J.: Usability evaluation of user interfaces generated with a model-driven architecture tool. In: Maturing Usability—Quality in Software, Interaction and Value, HCI Series, pp. 3–32. Springer (2008)
48. Wüest, D., Seyff, N., Glinz, M.: Flexisketch: a lightweight sketching and metamodeling approach for end-users. Softw. Syst. Model. **18**(2), 1513–1541 (2019)
49. Safdar, S.A., Iqbal, M.Z., Khan, M.U.: Empirical evaluation of UML modeling tools-a controlled experiment. In: 11th European Conference on Modelling Foundations and Applications (ECMFA), pp. 33–44 (2015)
50. Bobkowska, A., Reszke, K.: Usability of UML modeling tools. In: Conference on Software Engineering: Evolution and Emerging Technologies, pp. 75–86 (2005)
51. Ren, R., Castro, J.W., Santos, A., Pérez-Soler, S., Acuña, S.T., de Lara, J.: Collaborative modelling: Chatbots or on-line tools? An experimental study. In: Proceedings of EASE, pp. 260–269. ACM (2020)

52. Pérez-Soler, S., Guerra, E., de Lara, J.: Collaborative modeling and group decision making using chatbots in social networks. IEEE Softw. **35**(6), 48–54 (2018)
53. Condori-Fernández, N., Panach, J.I., Baars, A.I., Vos, T.E.J., Pastor, O.: An empirical approach for evaluating the usability of model-driven tools. Sci. Comput. Program. **78**(11), 2245–2258 (2013)
54. Tolvanen, J.-P., Kelly, S.: Model-driven development challenges and solutions—experiences with domain-specific modelling in industry. In: Proceedings of MODELSWARD, pp. 711–719. SciTePress (2016)
55. Karna, J., Tolvanen, J.-P., Kelly, S. Evaluating the use of domain-specific modeling in practice. In: Proceedings of the 9th OOPSLA Workshop on Domain-Specific Modeling (2009)
56. Buezas, N., Guerra, E., de Lara, J., Martín, J., Monforte, M., Mori, F., Ogallar, E., Pérez, O., Cuadrado, J.S.: Umbra designer: Graphical modelling for telephony services. In: Proceedings of ECMFA, volume 7949 of Lecture Notes in Computer Science, pp. 179–191. Springer (2013)
57. Hutchinson, J., Whittle, J., Rouncefield, M., Kristoffersen, S.: Empirical Assessment of MDE in Industry. In: ICSE, pp. 471–480. ACM (2011)
58. Cuadrado, J.S., Cánovas Izquierdo, J.L., Molina, J.G.: Applying model-driven engineering in small software enterprises. Sci. Comput. Program. **89**, 176–198 (2014)
59. Green, T.R.G.: Cognitive dimensions of notations. In: People and Computers V, pp. 443–460. Cambridge University Press, Cambridge (1989)
60. Moody, D.L.: The physics of notations: Toward a scientific basis for constructing visual notations in software engineering. IEEE Trans. Software Eng. **35**(6), 756–779 (2009)
61. Granada, D., Vara, J.M., Bollati, V.A., Marcos, E.: Enabling the development of cognitive effective visual DSLs. In: MODELS, volume 8767 of Lecture Notes in Computer Science, pp. 535–551. Springer (2014)
62. Barisic, A., Amaral, V., Goulão, M.: Usability driven DSL development with USE-ME. Comput. Lang. Syst. Struct. **51**, 118–157 (2018)
63. Granada, D., Vara, J.M., Brambilla, M., Bollati, V.A., Marcos, E.: Analysing the cognitive effectiveness of the WebML visual notation. Softw. Syst. Model. **16**(1), 195–227 (2017)
64. Atkinson, C., Kennel, B., Goß, B.: The level-agnostic modeling language. In: 3rd International Conference on Software Language Engineering (SLE), pp. 266–275. Springer (2011)
65. Miller, G.A.: Wordnet: a lexical database for English. Commun. ACM **38**(11), 39–41 (1995)
66. Ko, A.J., LaToza, T.D., Burnett, M.M.: A practical guide to controlled experiments of software engineering tools with human participants. Empir. Softw. Eng. **20**(1), 110–141 (2015)
67. Salman, I., Misirli, A.T., Juristo, N.: Are students representatives of professionals in software engineering experiments? In Proceedings of the 37th International Conference on Software Engineering—Volume 1, ICSE '15, pp. 666–676. IEEE (2015)
68. Falessi, D., Juristo, N., Wohlin, C., Turhan, B., Münch, J., Jedlitschka, A., Oivo, M.: Empirical software engineering experts on the use of students and professionals in experiments. Empirical Softw. Eng. **23**(1), 452–489 (2018)
69. Feldt, R., Zimmermann, T., Bergersen, G.R., Falessi, D., Jedlitschka, A., Juristo, N., Münch, J., Oivo, M., Runeson, P., Shepperd, M., SjØberg, D.I., Turhan, B.: Four commentaries on the use of students and professionals in empirical software engineering experiments. Empirical Softw. Eng. **23**(6), 3801–3820 (2018)
70. Feitelson, D.G.: Using students as experimental subjects in software engineering research—a review and discussion of the evidence (2015). https://arxiv.org/abs/1512.08409
71. Likert, R.A.: A technique for measurement of attitudes. Arch. Psychol. **22** (1932)
72. Wohlin, C., Runeson, P., Ohlsson, M.C., Regnell, B.: Experimentation in Software Engineering. Springer, Martin Höst (2012)
73. Bangor, A., Kortum, P., Miller, J.: Determining what individual sus scores mean: adding an adjective rating scale. J. Usability Stud. **4**(3), 114–123 (2009)
74. Singer, J., Vinson, N.G.: Ethical issues in empirical studies of software engineering. IEEE Trans. Softw. Eng. **28**(12), 1171–1180 (2002)
75. Vinson, N.G., Singer, J.: A Practical Guide to Ethical Research Involving Humans, pp. 229–256. Springer, Berlin (2008)
76. Kitchenham, B.A., Pfleeger, S.L.: Personal Opinion Surveys, pp. 63–92. Springer, Berlin (2008)
77. Powers, D.: Evaluation: From precision, recall and fmeasure to ROC, informedness, markedness and correlation. J. Mach. Learn. Technol. **2**, 37–63 (2007)
78. Saito, T., Rehmsmeier, M.: The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. PLOS ONE **10**(3), 1–21 (2015)
79. Van Rijsbergen, C.J.: Information Retrieval, 2nd edn. Butterworth-Heinemann, Oxford (1979)
80. Forman, G., Scholz, M.: Apples-to-apples in cross-validation studies: Pitfalls in classifier performance measurement. SIGKDD Explor. Newsl. **12**(1), 49–57 (2010)
81. Lewis, J.R., Sauro, J.: The factor structure of the system usability scale. In: 1st International Conference on Human Centered Design, HCD, pp. 94–103. Springer, Berlin (2009)
82. Lewis, J.R., Brown, J., Mayes, D.K.: Psychometric evaluation of the EMO and the SUS in the context of a large-sample unmoderated usability study. Int. J. Human-Comput. Interact. **31**(8), 545–553 (2015)
83. Sauro, J., Lewis, J.R.: When designing usability questionnaires, does it hurt to be positive? In: SIGCHI Conference on Human Factors in Computing Systems, CHI, pp. 2215–2224. ACM (2011)
84. Lewis, J.R., Utesch, B.S., Maher, D.E.: Measuring perceived usability: the SUS, UMUX-LITE, and AltUsability. Int. J. Human-Computer Interact. **31**(8), 496–505 (2015)
85. Cohen, J.: Statistical Power Analysis for the Behavioral Sciences. Routledge, London (2013)
86. Banerjee, A., Chitnis, U.B., Jadhav, S.L., Bhawalkar, J.S., Chaudhury, S.: Hypothesis testing, type I and type II errors. Ind. Psychiatry J. **18**, 127–31 (2009)
87. Nickerson, R.: Null hypothesis significance testing: a review of an old and continuing controversy. Psychol. Methods **5**, 241–301 (2000)
88. McGraw, K., Wong, S.C.P.: A common language effect size measure. Psychol. Bull. **111**, 361–365 (1992)
89. Cohen, J.: Statistical Power Analysis for the Behavioral Sciences, 2nd edn. Lawrence Erlbaum Associates, Hillsdale (1988)
90. Burgueño, L., Cabot, J., Li, S., Gérard, S.: A generic LSTM neural network architecture to infer heterogeneous model transformations. Softw. Syst. Model. **21**(1), 139–156 (2022)
91. Hernández López, J.A., Cánovas Izquierdo, J.L., Cuadrado, J.S.: Modelset: a dataset for machine learning in model-driven engineering. Softw. Syst. Model. (2022)
92. Wohlin, C., Runeson, P., Höst, M., Ohlsson, M.C., Regnell, B., Wesslén, A.: Experimentation in Software Engineering: An Introduction. Kluwer Academic Publishers, Norwell (2000)
93. Colquhoun, D.: An investigation of the false discovery rate and the misinterpretation of p-values. R. Soc. Open Sci. **1**(3), 140216 (2014)
94. Forstmeier, W., Wagenmakers, E.-J., Parker, T.H.: Detecting and avoiding likely false-positive findings—a practical guide. Biol. Rev. Camb. Philos. Soc. **92**(4), 1941–1968 (2017)
95. Brooke, J.: SUS: a retrospective. J. Usability Stud. **8**(2), 29–40 (2013)

96. Cronbach, L.J.: Coefficient alpha and the internal structure of tests. Psychometrika **16**(3), 297–334 (1951)

Ángel Mora Segura is a Ph.D. candidate in Computer Science and FPU grant holder (FPU13/02698) at the Universidad Autónoma de Madrid. He is a past member of the modelling and software engineering research group (http://miso.es/). He is a Computer Engineer and holds a Master's degree in International Business Administration by the University of Almería. Currently employed as a Data Architect and Machine Learning Engineer in the private sector.



Juan de Lara is Full professor at the computer science department of the Universidad Autónoma de Madrid. Together with Esther Guerra, he leads the modelling and software engineering research group (http://miso.es/). His research interests are in model-driven engineering and automated software development.

Manuel Wimmer is full professor and head of the Institute of Business Informatics—Software Engineering at the Johannes Kepler University Linz. He is also the head of the Christian Doppler Laboratory CDL-MINT. His research interests comprise foundations of model-driven engineering techniques as well as their application in domains such as tool interoperability, legacy software modernization, model versioning and evolution, and industrial engineering.