Protocol for a Systematic Study on

Blended Modeling in Commercial and Open-source Tools

Istvan David (VU Amsterdam, Université de Montréal)
Ivano Malavolta (VU Amsterdam)
Federico Ciccozzi (Mälardalen University)
Malvina Latifaj (Mälardalen University)
Jan-Philipp Steghöfer (University of Gothenburg)
Weixing Zhang (University of Gothenburg)

PROTOCOL FOR A SYSTEMATIC STUDY ON

Blended Modeling in Commercial and Open-source Tools

VERSION 1.0, SEPTEMBER 1, 2021

ABSTRACT

This document describes the review protocol of a systematic study on modeling tools supporting multiple notations for software-intensive systems.

KEYWORDS

Systematic Study, Model-Driven Engineering.

Contents

1 Definitions and Unit of Study

This study is centered around the concept of **blended modeling**. Ciccozzi et al. [1] define blended modeling as *the activity of interacting seamlessly with a single model (i.e., abstract syntax) through multiple notations (i.e., concrete syntaxes), allowing a certain degree of temporary inconsistencies*.

In the context of this study, the main implications of this definition are the following:

1. Blended modeling assumes one and only one abstract syntax; supported with multiple concrete syntaxes;
2. Blended modeling is orthogonal to the view/viewpoint paradigm;
3. In blended modeling, notation $\equiv$ concrete syntax.

The core concepts of this study are defined as follows[1]:

- *Modeling language* is defined as an artificial language that is used to express information or knowledge or systems in a structured manner by using a well-defined and consistent set of concepts and following a well-defined and consistent set of rules. Concepts and rules are formalized in a so-called metamodel (or equivalent specification).
- *Abstract syntax* of a language is defined as a well-defined and consistent description of the language's structure as a data type and it is independent of any particular representation or encoding. A representation-agnostic metamodel is considered to be an abstract syntax. In our work, we consider: metamodel $\equiv$ abstract syntax.
- *Concrete syntax* is defined as a well-defined and consistent set of rules (or productions) that define the visual representation of models conforming to a specific abstract syntax. A context-free grammar is a concrete syntax. In our work, we consider: (modelling) notation $\equiv$ concrete syntax.
- *Software-intensive system* is defined as a system where software contributes essential influences to the design, construction, deployment, and evolution of the system as a whole [2].

Unit of study. The goal of this study is to systematically survey the state of the art and state of the practice on modeling tools.

- supporting canonical modeling activities (i.e., beyond simple sketching and drawing; typically: instantiating a more explicit metamodel); and
- their support for the concepts related to blended modeling.

In the remainder of this protocol we refer to such tools as **blended modeling tools**.

2 Research Process

This research is carried out by following the process shown in Figure 1. Our process can be divided into three main phases, which are well-established in systematic literature studies [3, 4]: planning, conducting, and documenting.

In the following, the three phases of the process are detailed.

[1]These definitions should be considered as informal indications for streamlining our research, they are intended for internal use only.

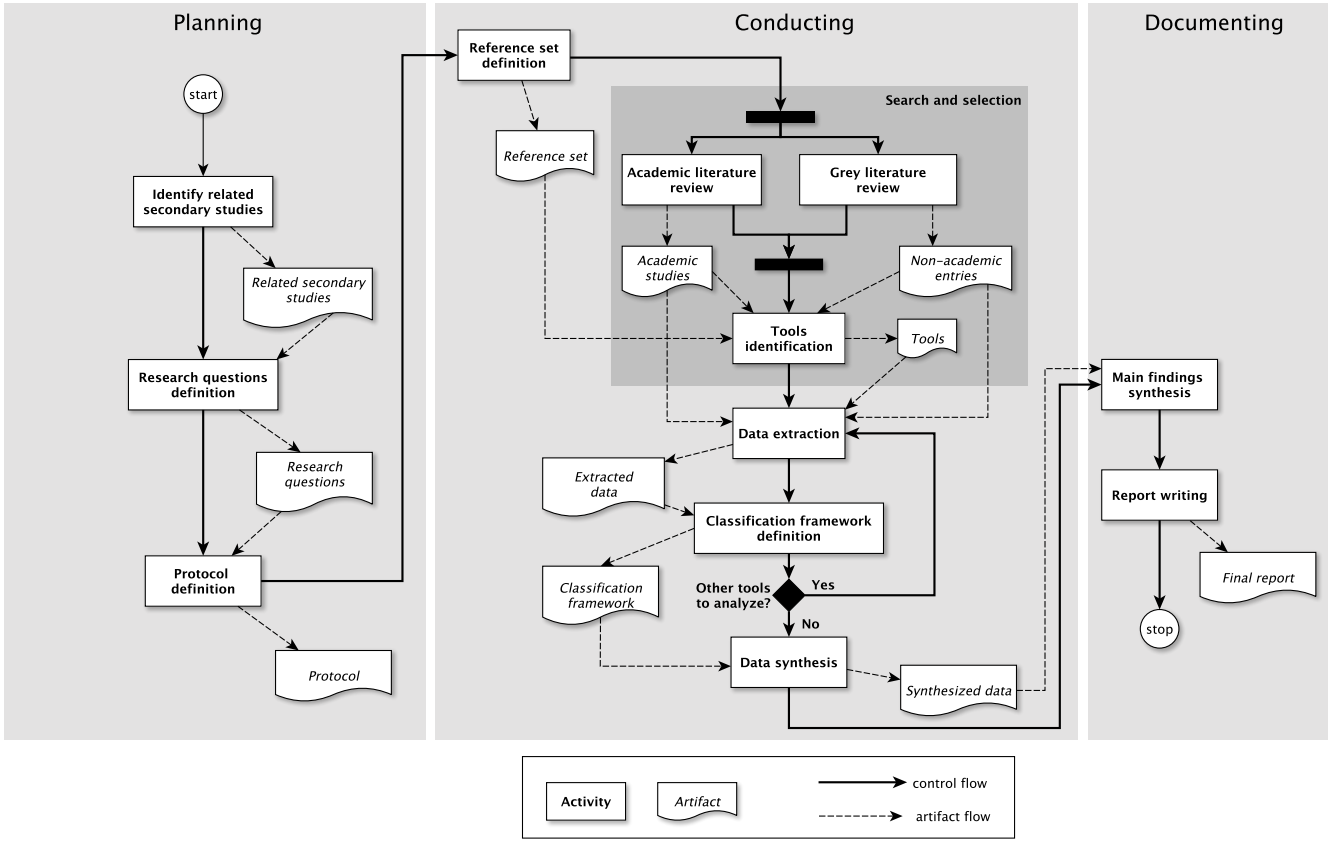

Figure 1: Overview of the whole review process

2.1 Planning

This phase aims at producing a research protocol (i.e., this document), which is followed in the *Conducting* and *Documenting* phases. More specifically, we firstly identified *related secondary studies*, i.e., surveys and literature reviews with a scope similar to the current review's scope (see Section 3. Subsequently, the *research questions* are identified, and the *protocol is compiled*. In order to mitigate potential threats to validity and possible biases, the research protocol is circulated to one external expert for independent review. The expert is contacted for getting feedback about the proposed review protocol, possible unidentified threats to validity, possible problems in the overall construction of the review, and about whether the proposed research protocol and final reports can be effective with respect to the object of our mapping study (i.e., blended modeling tools).

2.2 Conducting

In this phase, the mapping study is performed, by following all the steps previously defined. More specifically, the following activities are carried out.

- *Reference set identification*. In the first step, the *domain knowledge of the authors* is made explicit, resulting is the *reference set* for this research. The reference set is a collection of modeling tools, which we know *a priori*, that should be part of the final set of modeling tools considered in this study. The results of the academic and grey literature review are eventually compared to the reference set in order to verify the comprehensiveness of their results. Section 5 provides more details on this activity.
- *Search and selection*. Two parallel activities are carried out: the *review of the academic literature*, and the *review of the grey literature*. Each of these activities yields in a set of

papers, describing the *tools* of interest.

In both search activities, we perform a combination of automated search, manual search, and backward-forward snowballing for identifying the set of academic studies on modeling tools relevant to our review. Section 6 describes in details the search and selection process.

- *Tools identification*. Based on the output of the previous phases, we define the final set of modeling tools that are analysed in this study. Section 6.2 provides more details on this activity.
- *Data extraction*. We go into the details of each modeling tool, and document our findings in a corresponding data extraction form. Filled forms are collected and aggregated in order to be ready to be analyzed. Section 7 provides more details on this activity.
- *Classification framework definition*. We define the set of parameters which are used to compare the modeling tools. So, the classification framework provides the structure for classifying the modeling tools (i.e., categories, parameters, possible values, etc.). Differently, the data extraction is about populating the classification framework with the data extracted from each modeling tool. There is a loop between the classification framework definition and data extraction since new parameters and values can be included in the classification framework while extracting data. Section 8 provides more details on this activity.
- *Data synthesis*. We perform a comprehensive analysis and summary of the extracted data. The goal of this activity is to elaborate on the extracted data in order to answer each research question (Section 4). The activity involves both quantitative and qualitative analysis of the extracted data. Section 9 provides further details on this activity.

2.3 Documenting

The main activities performed in this phase are: (i) a thorough elaboration of the data extracted in the previous phase with the main aim of discovering the main findings of the study and setting the obtained findings in their context, (ii) the discussion of possible threats to validity, specially to the ones identified during the definition of the review protocol (in this activity new threats to validity may emerge too), and (iii) the writing of a final report describing the performed study. First, the produced report is evaluated by a set of experts. Then, it is submitted to an academic journal, thus undergoing a peer reviewed evaluation by the scientific community too.

2.4 Team

Ten researchers are involved in the planning, conducting and reporting of this study, each of them with a specific role within the research team.

Principal researchers (5)

Malvina Latifaj (PhD student, Mälardalen University), Weixing Zhang (PhD student, University of Gothenburg), Jakob Pietron (research assistant, Ulm University), Istvan David (postdoctoral researcher, VU Amsterdam, Université de Montréal), Alexander Raschke (research associate, Ulm University)

They are part of all the activities, i.e., planning the study, conducting it, and reporting.

Research methodologists (4)

Ivano Malavolta (assistant professor, VU Amsterdam), Federico Ciccozzi (associate professor, Mälardalen University), Jan-Philipp Steghöfer (associate professor, University of Gothenburg), Regina Hebig (associate professor, University of Gothenburg)

They are mainly involved in (i) the planning phase of the study, and (ii) supporting the principal researchers during the whole study, e.g., by reviewing the data extraction form, selected studies, extracted data, produced reports, etc.

Advisors (1)

Matthias Tichy (professor, Ulm University)

Takes final decisions on conflicts and methodological options to facilitate conclusions and "avoid endless discussions" [5]; supports the other researchers during data and findings synthesis activities.

3 Related Secondary Studies

In this section, we discuss other existing secondary studies related to our work. To the best of our knowledge and after a manual search on Google Scholar, no other systematic study has considered this topic. The studies presented in the following are related to our research, but differ in terms of motivation and objectives and are generally limited to a more narrow scope by providing partial information related to our research.

Torres et al. [6] conducted a systematic literature review with the aim of identifying a list of available tools to support model management, and providing a categorization of these tools into (i) tools that can provide consistency checking on models of different domains, (ii) tools that can provide consistency checking on models of the same domain, and (iii) tools that do not provide any consistency checking. Furthermore, the authors identify the inconsistency types, the strategies to keep the consistency between models of different domains, and the challenges to manage models of different domains. The information retrieved from the primary studies is also complemented with additional data sources (e.g., tool official website). Our study differs from Torres et al. [6] because, in addition to model consistency checking, we consider other features for the comparison of these tools such as notations, generation of notation editors, collaboration mechanisms, and more.

Iung et al. [7] conducted a systematic mapping study with the aim of identifying tools, language workbenches, or frameworks for DSL development. The authors identify 59 tools, and they use the feature model proposed by Erdweg et al. [8], for their comparison. The study focuses on the technologies/tools used for DSL development, their license types, the application domains, and the features of the DSL creation process that these tools support. 48 tools support only one notation (graphical OR textual), seven tools support two notations (graphical AND textual), two tools support three notations, and two tools support four notations. Our study differs from Iung et al. [7] as it aims to have a broader scope, by extending the set of features on which the comparison will be based, with features such as synchronization mechanisms, collaborative features, or conformance relaxation. Furthermore, our search process considers studies published in any year up to 2020, while Iung et al. [7] limit their search to the time frame of 2012-2019.

Franzago et al. [9] conducted a systematic mapping study with the aim of identifying and classifying collaborative MDSE approaches based on different categories such as characteristics of collaborative model editing environments, model versioning mechanisms, model repositories, support for communication and decision making, and more. Additionally, the authors identify limitations and challenges with respect to the state of the art in collaborative MDSE approaches. Regarding model management, they provide a taxonomy for the management support of collaborative MDSE approaches, collaboration support, and communication support. This study covers some of the aspects that we will cover in our systematic mapping study (e.g., conflict detection). However, while this study is mostly focused on the characteristics of the collaborative approaches, we aim towards a classification of tools based on a broader set of features such as synchronization mechanisms and their generation, or conformance relaxation.

Granada et al. [10] conducted a systematic mapping study with the aim of collecting model-based tools with which to generate editors for visual DSLs and pointing out their features and function-

alities. The authors collected eight model-based tools for the generation of editors for visual DSLs and the features taken into consideration for their analysis were as follows: scope, framework, the distinction between abstract and concrete syntax, abstract syntax, concrete syntax, editing capabilities, use of models, automation, usability, and methodological basis. The conclusions point out that the most complete commercial tools are MetaEdit+ and ObeoDesigner, while the most complete open-source tools are Eugenia, GMF, Graffiti, and Sirius. Our study differs from Granada et al. [10], as we focus on tools that provide multiple notations, not only tools that can be used to develop editors for visual DSLs.

do Nascimento et al. [11] conducted a systematic mapping study with the aim of providing the most popular application domains where DSLs have been applied, different tools for handling DSLs, and techniques, methods, and/or processes for dealing with DSLs. The tools are categorized into (i) tools for using DSLs, (ii) tools for DSL creation, and (iii) language workbenches. Our study differs from do Nascimento et al. [11], as we focus on DSL tool comparison, while they only provide a categorization of DSL tools, and do not go into the details of conducting a comparison of the technical features of these tools.

Furthermore, there are a few studies related to our research that are not systematic studies but do conduct a comparison of language workbenches and modeling tools. Negm et al. [12] conducted a survey comparing 14 language workbenches based on: (i) structure (grammar-driven or model-driven), (ii) editor (parser-based or projectional), (iii) language notations (textual, tabular, symbols, or graphical), (iv) semantics (translational or interpretive), and (v) composability language aspects. However, this study is limitied to language workbenches, and does not cover aspects such as synchronization mechanisms and their generation, or collaborative features.

Erdweg et al. [8] conducted a comparison study of 10 language workbenches participating in Language Workbench Challenge 13'. The comparison of the language workbenches was based on a feature model that included: notation, semantics, editor support, validation, testing, and composability, where some of them support multiple notations (fully or partially). The conclusions state that no language workbench realizes all features. However, this study is limited to language workbenches presented in Language Workbench Challenge 13'.

Merkle [13] conducted a comparison study of textual language workbenches categorizing them in pure text-based and projectional-based with a textual projection. The LWs compared in this study are as follows: Xtext, TEF, TC, EMFText, and MPS. The language workbenches were compared based on workflow, abstract/concrete syntax, and editor. However, this study is limited to textual language workbenches, while our focus is on tools that provide multiple notations.

4 Goal and Research questions

This study aims at characterizing the current state of the art of modeling tools supporting multiple notations. We formulate the goal of this study by using the Goal-Question-Metric perspectives [14], shown in Table 1.

Purpose	Identify, classify, and analyse
Issue	the user-oriented and realization-oriented characteristics, of
Object	existing modeling tools supporting multiple notations
Context	for software-intensive systems
Viewpoint	from a researcher's and practitioner's point of view. (both tools' users and developers)

Table 1: The goal of this research.

This abstract goal is refined into the following research questions. For each research question, we also provide the rationale for it being part of this study.

RQ1: *What are the **user-oriented characteristics** of blended modeling tools?*

Rationale. Modeling tools are designed and developed in order to be adopted by specific users, application domains, and usage scenarios.
By answering this research question, we are aiming at identifying the external characteristics of modeling tools, pertaining to their adoption and usage [1]. Notably, parameters such as application domains, addressed user groups, supported (types of) notations, human-computer interfaces, and licensing models will be investigated.
Practitioners can benefit from the answer to this research question by understanding how the specific concepts and tools of the state of the art address their problems, what are their limitations from this aspect, and how can be adopted.

RQ2: *What are the **realization-oriented characteristics** of blended modeling tools?*

Rationale. With the advent of model-based approaches, and domain-specific modeling in particular, several modeling tools are being developed to support multiple notations, formalisms, and semantics. Moreover, until the recent spread of mainstream language workbenches (e.g., Xtext, Sirius, MPS, etc.), the development of such modeling tools had been relatively ad-hoc.
By answering this research question, we are aiming at identifying the internal characteristics of modeling tools supporting multiple notations, and that, in terms of (i) their features, and (ii) the techniques employed to implement those features. Notably, parameters such as implementation platforms, consistency mechanisms, internal architectures, and the linguistic level of model-to-model correspondence will be investigated.
Researchers can benefit from the answer to this research question by understanding the state-of-the-practice on the techniques of blended modeling tools, including the gaps to contribute to.

The identified research questions will drive the whole study, with a special influence on (i) search and selection of academic studies, (ii) data extraction, and (iii) data analysis.

5 Reference Set Definition

This activity aims to externalize (i) the modeling tools mentioned in the related secondary studies (see Section 3), (ii) the authors' experiences with blended modeling tools, (iii) specific searches for querying generic web search engines with the search string defined in Section 6.1.1, and (iv) knowledge garnered from existing networks of experts, e.g., by accessing forums, mailing lists.
The reference set is composed of the following tools:

- NoMagic MagicDraw, `https://www.nomagic.com/products/magicdraw`
- Eclipse Papyrus/Moka, `https://www.eclipse.org/papyrus/`
- MetaEdit+, `https://www.metacase.com/products.html`
- Umple, `https://cruise.umple.org/umple/`
- Open Source AADL Tool Environment (OSATE), `https://osate.org/`

Further tools that merit consideration, but eventually fell outside of the scope of this study:

- Whole Language Workbench, `https://marketplace.eclipse.org/content/whole-language-workbench` – Language workbench, and as such, it is outside of our scope.
- FXDiagram, `https://jankoehnlein.github.io/FXDiagram/` – Only visualization, no support for real modeling via multiple syntaxes.

- Mermaid, `https://mermaid-js.github.io/mermaid/#/` – Only visualization, no support for real modeling via multiple syntaxes.

6 Search and Selection

The goal of our *search and selection* activities is to retrieve a comprehensive set of modeling tools supporting multiple modeling notations in a blended fashion. Given the interest of both scientific and industrial communities on the subject, we firstly perform a *systematic review* of both the academic (e.g., scientific articles published at peer-reviewed academic venues) and grey literature (e.g., websites, on-line blogs, etc.), see Section 6.1. The output of those two steps (i.e., academic studies and non-academic entries) is then further analysed in order to identify the modeling tools either considered, mentioned, or discussed in them (see Section 6.2).

6.1 Systematic Reviews

We follow the same overall process when reviewing the academic and grey literature. In this phase it is fundamental to achieve a good trade-off between the coverage of existing results on the considered topic, and to have a manageable number of studies to be analysed. In order to achieve the above mentioned trade-off, our search and selection process has been designed as a multi-stage process; this gives us full control on the number and characteristics of the entries being either selected or excluded during the various stages. In the following, we present each step of our systematic review process[2].

The systematic review is divided into three subsequent and complementary steps. The first step is carried out by automatically inspecting all the results returned from a query execution on (i) Google Scholar for academic studies and (ii) the Google Search engine for the grey literature (see Section 6.1.1). In the second step, the identified potentially relevant studies undergo a rigorous filtering based on the application of a set of selection criteria (see Section 6). In the third step, we complement the preliminary set of academic studies by applying the snowballing process [15] (see Section 6.1.3).

6.1.1 Automatic search

Academic Literature In this initial stage we perform an automatic search by executing a search query on two search engines: one to identify potentially relevant studies in the academic body of literature and one to identify potentially relevant studies in the gray body of literature. For the academic literature, we target *Google Scholar*. We use Google Scholar as data source for the following main reasons: (i) it is one of the largest and most complete databases and indexing systems for scientific literature, (ii) as reported in [15], the adoption of this data source has proved to be a sound choice to identify the initial set of literature studies for the snowballing process (see Section 6.1.3), (iii) the query results can be automatically processed via state-of-the-art tools.

Below we report the search string used in this study. In order to cover as many potentially relevant studies as possible, we defined the search string so that it includes academic studies on blended modeling. Indeed, the search string can be divided into three main components: the first component captures the model-driven paradigm, the second one captures the focus on multiple entities and blended of the targeted studies, and the third one is used for ensuring that this study focuses on software aspects. The search string has been tested by executing pilot searches on Google Scholar[3]. In order to keep the results of this initial search as focused as possible, the query has been applied to the title of the targeted studies.

[2]For the sake of simplicity, in the remainder of this document we will refer to both academic studies and non-academic entries as *primary studies*, and will use different terms only when strictly needed

[3]At the time of writing, Google Scholar produces a total of 280 hits when searching with the reported search string

```
(intitle:"modeling" OR intitle:"modelling" OR intitle:"model based" OR intitle:"model driven")
AND
(intitle:"multi*" OR intitle:"blended")
AND
(intitle:"notation*" OR intitle:"syntax*" OR "intitle:editor" OR "intitle:tool" OR "intitle:software")
```

Grey Literature For the grey literature, we target the regular *Google Search Engine*. The search engine is selected in accordance to the recommendations for including grey literature in software engineering multi-vocal reviews [16]. The search string used for the academic literature yields mostly academic results even in a general web search. We have therefore adapted our search strategy to find non-academic sources. In particular, we identified a number of relevant hits through manual searches early on. These manual hits could be classified as either *lists* (e.g., Wikipedia's "List of Unified Modeling Language tools" or *tool-specific pages* (e.g., tool vendor pages or blog posts about how specific tools are used).

We experimented with several search strings to ensure that we find all relevant hits. In particular, we tried to combine different modelling languages and diagram types into one large all-encompassing search string to simplify our search and make it easier to extract results. However, on prototyping this approach, we realised that the "OR" clauses that we used did not have the desired effect and we did not find the tools we expected and in particular not the lists that we were expecting. In comparison, a search string such as "(MARTE) AND (tool OR editor OR notation OR modelling)" yields 162 results on Google, whereas our combined search string that included MARTE and many other languages only yielded 150 results.

We therefore decided to do independent searches for different, popular modelling languages. This meant that we ran different searches independently and merged the results later on. To address the large number of hits we would get this way, we limit the search results for each search to the first 50 hits.

We selected the relevant modelling languages using a mixture of expert knowledge, browsing the web pages of well-known modelling tools from the reference set (e.g., NoMagic MagicDraw, Eclipse Papyrus, Umple) and beyond (e.g., Eclipse Capella and Enterprise Architect), using lists such as Wikipedia's page on "Modelling Languages". We narrowed down the resulting list of more than 50 potential modelling languages by searching for "(`Language Name`) AND (tool OR editor OR notation OR modelling)" and analysing the first ten non-academic hits, i.e., search results that are not academic papers. If these ten hits contained a link to a modelling tool for the language, we included it in our search, otherwise, we disregarded it.

We are aware that this approach can introduce some bias. For instance, the Extended Enterprise Modelling Language (EEML) was excluded this way. However, the fact that no tool-related hits were found for EEML indicates that there is no wide-spread tool support for this language and it is therefore not relevant for our purposes.

Search Execution The automatic searches for both academic and non-academic literature are executed in November 2020. In order to reduce potential threats to validity and for the sake of reliability, all the cookies of the browser from which the query was executed were cleaned. In addition, incognito mode was used. The browsers were set to English as the primary language. Searches were conducted using IP addresses assigned to Swedish ISP respectively. It is worth noting that the number of results shown by Google Web Search (often in the millions) is not an indication of the actual number of results that are obtained by going through the result pages manually until Google does not provide additional hits.

6.1.2 Application of Selection Criteria

Following the guidelines for systematic literature review for software engineering [3], we define the set of inclusion and exclusion criteria *a priori* in order to reduce the likelihood of bias. The potentially relevant entries are rigorously examined by adopting multiple selection rounds in an adaptive reading depth fashion [17]. Specifically, in the first round, the title of the entry is examined. This first step enables us to discard all those papers or web pages that clearly do not fall in the scope of this study. In the second exclusion round, the introductions and the conclusions are inspected (if present). Finally, the entries are further inspected by considering their full text in order to ensure that only the ones relevant for answering the research questions will be selected. When going through the full text of a paper/web page we also keep track of all the mentioned modeling tools and will consider them in the tools identification phase (see Section 6.2).

In the following, we detail the set of inclusion and exclusion and criteria that will guide the selection of the academic and non-academic entries for our systematic review. A potentially relevant entry is selected if it (i) satisfies *all* inclusion criteria and (ii) does not satisfy *any* of the exclusion criteria. The selection inclusion and exclusion criteria are divided into three categories, namely: *generic* (i.e., applying for both academic and non-academic studies), *academic-specific*, and *grey-specific*. The decision of adopting three categories of criteria originates from the different nature of the sources of primary studies considered (i.e., Google Scholar and the Google Search Engine). By defining three different sets it is possible to design selection criteria specifically tailored to the specific characteristics of academic and non-academic entries, and hence improve the overall quality of the selection process.

Generic inclusion criteria

GEN-I1) Studies or web pages on modeling tools, i.e., where models are used as first-class entities and used as a substantial abstraction from the problem domain (e.g., OSATE for modeling hardware/software systems according to the AADL modeling language).

GEN-I2) Studies or web pages discussing at least two different concrete syntaxes (possibly for the same abstract syntax). The notations can be of the same type (e.g., both textual).

Generic exclusion criteria

GEN-E1) Studies on non-modeling tools. For example, articles on IDEs, programming tools, drawing tools, etc.)

GEN-E2) Studies that have not been published in English.

GEN-E3) Duplicates of already included studies.

GEN-E4) Studies that are not available, and hence not analyzable.

Academic-specific exclusion criteria

AC-E1) Studies in the form of full proceedings, books, etc. since they are too broad for being thoroughly analysed in this phase of the study.

AC-E2) Studies that have not been peer-reviewed[4].

[4]Peer-reviewing is the *de facto* standard of quality assurance for scientific literature.

Grey-specific exclusion criteria

GR-E1) Web pages reporting exclusively the basic principles of modeling techniques.

GR-E2) Web pages reporting exclusively abstract best practices while applying modeling techniques.

GR-E3) Web pages reporting an implementation without a discussion of its benefits and/or drawbacks.

GR-E4) Academic literature, since such type of studies is considered by a different process in our protocol.

GR-E5) Videos, webinars, books, etc. since they are too time-consuming to be considered for this study.

6.1.3 Snowballing

In order to mitigate a potential bias with respect to the construct validity of the study, backward and forward snowballing is used to complement the automatic search of the academic literature [18]. In particular, this process is carried out by considering the scientific publications selected in the initial automatic search, and subsequently selecting relevant studies among those cited by one of the initially selected ones (backward snowballing). Then, we also perform forward snowballing, i.e., selecting relevant studies among those citing one of the initially selected academic studies [19]. In this context, the *Google Scholar*[5] bibliographic database is adopted to retrieve the studies citing the ones selected through the initial search phase. The final decision about the inclusion of the newly considered publications in the study is based on the application of the selection criteria presented in Section 6.

6.2 Tool identification

In the tool identification step, each primary study is manually analysed and the mentioned modeling tools are identified. This is achieved by investigating the full text of each primary study and collecting every modeling tool mentioned in it. In this phase, for each tool we collect the following information: name, link/reference to official documentation, organization(s) implementing, maintaining, and supporting the tool, and tracing information towards all primary studies mentioning the tool.

Then, the set of identified modeling tools is filtered for duplicates, which are subsequently merged, regardless of whether the tool originates from an academic or from a non-academic study. In order to ensure that the identified tools will support us in answering the research questions of this study, we will further filter the list of all modeling tools according to the following selection criteria.

Inclusion criteria:

TI-1 – The tool allows its users to edit the same model in multiple notations. The user can switch between these notations easily and without an extra processing step (i.e., the tool supports some level of blended modelling). The tool allows a certain degree of temporary inconsistencies. Notations like an overview tree for navigation purposes or any textual representation used for file persistency purposes only are not considered. Examples: DOT representation is fine, XMI not. Examples: DOT representation is fine, XMI not.

TI-2 – The tool is publicly available (either as an open-source or commercial product).

[5] `https://scholar.google.it/`

TI-3 – The documentation of the tool is publicly available.
Examples: short description on website is not enough; missing technical description is still fine.

Exclusion criteria:

TE-1 – The tool is not available for download as a binary that can be run on current operating systems from an official website or from an affiliated platform supporting it.

TE-2 – The documentation of the tool is not in English.

A modeling tool is included in the final list of modeling tools for our study if it satisfies *all* inclusion criteria and it is discarded if it satisfies *any* exclusion criterion.

Finally, after the current set of modeling tools is identified, we check if it includes all tools in the reference set (see Section 5). If all tools in the reference set are included in the current set of modeling tools, then we continue with the subsequent phases of the protocol (i.e., data extraction), otherwise a dedicated meeting with at least one of the research methodologists is setup and a refinement of the systematic review process is designed and conducted again.

In order to minimize bias, this step is performed by five researchers and organized as follows. Firstly, the set of potentially relevant tools is divided into two random subsets, then two principal researchers are assigned to each of the two subsets and independently apply the tool selection criteria. The inter-researcher agreement among the two pairs of resarchers assigned to each subset of potentially relevant tools is measured using the Cohen Kappa statistic; the obtained Cohen Kappa statistics are reported as a quality assessment of this stage in the final report. Finally, emerging conflicts are discussed and resolved with the intervention of a research methodologist.

7 Data extraction

The main goal of this activity is to extract relevant data for answering the research questions for each modeling tool.

Inputs. The inputs to this activity are (i) the set of tools we have previously identified; and (ii) the textual contents of the studies referring the tool, and the tool's official documentation (when publicly available). Moreover, it might happen that we will not be able to collect all relevant data for some specific aspects of a tool (e.g., the internal consistency mechanisms of a proprietary tool); in those cases we will perform a series of ad-hoc Web searches and will contact the support team of the tool for collecting the missing data. For the sake of external verifiability, full tracing information will be kept between the extracted data and the considered data sources and it will be included in the replication package of the study.

In order to carry out a rigorous data extraction process, and to ease the control and the subsequent analysis of the extracted data, a predefined data extraction form is designed prior the data extraction process. The structure is composed of the various categories of the classification framework (see Section 8). For each tool, the principal researchers collect in a spreadsheet a record with the extracted information for subsequent analysis: the spreadsheet columns are the parameters of the classification framework, while each spreadsheet row represents the data of each modeling tool.

As suggested in [4], the principal researchers pilot the data extraction form independently over the first 3 modeling tools to be analysed. In order to validate our data extraction strategy, the principal researchers pilot the data extraction form independently and their results are checked with respect to whether they are consistent independently from the researcher performing the analysis. Specifically, each principal researcher gets a random sample of 5 modeling tools and analyzes them independently by filling the data extraction form. Then, they assess their level of

agreement and each disagreement is discussed and resolved with the intervention of the research methodologist, if needed.

When going through the material related to a tool in detail for extracting information, researchers can agree that the currently analysed tool may be semantically out of the scope of this research, and so it can be excluded.

8 Classification Framework Definition

The main goal of this activity is to create a classification framework for multi-notation modeling tools. In our study, the classification framework is composed of two distinct facets, each of which addresses one of the research questions (Section 4):

1. *Technical characteristics of multi-notation modeling tools*, addressing RQ1, see Section 8.1;
2. *Support for adoption*, addressing RQ2, see Section 8.2;

In this study, we partially reuse the results of previous work related to blended modeling for defining the initial version of the classification framework. Specifically, we reuse the dimensions of blended modeling defined in [1]. Then, the customization of the classification framework is performed as follows: (i) firstly we select a random sample of 10 pilot modeling tools, (ii) then two researchers independently extract the data from the 10 pilot modeling tools by using the initial version of our classification framework, (iii) the two researchers then discuss the results of the data extraction with the research methodologists, with a special focus on too generic/abstract parameters, parameters which did not fully fit with the characteristics of the primary studies, parameters with redundant values, and recurrent missing concepts, (iv) the classification framework is customized according to the discussion, and lastly (iv) the final version of the classification framework is applied to all modeling tools.

In the following sections, we describe the two facets of the initial version of the classification framework.

8.1 User-oriented characteristics (RQ1)

1. *Number of concrete syntaxes,* as blended modeling requires at least two concrete syntaxes to be present. Range: 2–N.
2. *Degree of language flexibility* to deviate temporarily from the rules of the language, potentially in a switchable fashion. Range: full correspondence between the concrete syntax and the abstract syntax; partial correspondence between the concrete syntax and the abstract syntax.
3. *Degree of overlap,* i.e., the percentage of language constructs expressible in multiple concrete syntaxes. Range: from partial to complete.

*Clarification.*The rest of the parameters are TBD and should be decided after the data extraction phase.

8.2 Realization-oriented characteristics (RQ2)

1. *Change propagation,* i.e., how changes are propagated between concrete syntax and abstract syntax as well as when and how multiple concurrent changes are merged.
2. *Inconsistency management approach.* Range: allow-detect-resolve, prevention.
3. *Inconsistency detection* between the multiple concrete syntax representations. Range: On-the-fly, on-demand, time-triggered, condition-triggered.

4. *Inconsistency resolution,* i.e., how the system supports resolution of inconsistencies. Range: from manual to automatic.

5. *Inconsistency tolerance.* Range: parameter, spatial, temporal – weak, strong, eventual, strong eventual.

6. *Number of abstract syntaxes required for multiple concrete syntax,* i.e. the cardinality of the technical mapping between abstract and concrete syntaxes.

7. *Mapping process between abstract and concrete syntaxes*, i.e., how elements of the abstract syntax are technically kept consistent with the representational elements.

*Clarification.*The rest of the parameters are TBD and should be decided after the data extraction phase.

9 Data synthesis

The data synthesis activity involves collating and summarising the data extracted from the studies [20, § 6.5] with the main goal of understanding, analysing, and classifying the state of the art of modeling tools supporting multiple notations.

In this phase, we have a fully populated spreadsheet with all the information coming from the data extraction form of each modeling tool. According to this, our data synthesis is divided into two main phases: vertical analysis and horizontal analysis. In both cases, we perform a combination of content analysis [21] (mainly for categorizing and coding approaches under broad thematic categories) and narrative synthesis [22] (mainly for detailed explanation and interpretation of the findings coming from the content analysis). When performing *vertical analysis*, we analyze the extracted data to find trends and collect information about *each parameter* of each category of our classification framework. When performing *horizontal analysis*, we analyse the extracted data to explore possible relations *across different parameters* of our classification framework.

Vertical analysis. Depending on the parameters of the classification framework (see Section 7), in this research we apply both quantitative and qualitative synthesis methods, separately. When considering quantitative data, depending on the specific data to be analysed, we apply descriptive statistics for better understanding the data. When considering qualitative data, we apply the *line of argument* synthesis [4], that is: firstly we analyse each tool individually in order to document it and tabulate its main features with respect to each specific parameter of the classification framework, then we analyse the set of studies as a whole, in order to reason on potential patterns and trends. When both quantitative and qualitative analyses are completed, we integrate their results in order to explain quantitative results by using qualitative results [20, § 6.5].

Horizontal analysis. We cross-tabulate and group the data, and make comparisons between two or more nominal variables. The main goal of the horizontal analysis is to (i) investigate on the existence of possible interesting relations between data pertaining to different parameters of the comparison framework. We use contingency tables for evaluating the actual existence of those relations and we identify perspectives of interest.

References

[1] F. Ciccozzi, M. Tichy, H. Vangheluwe, D. Weyns, Blended modelling-what, why and how, in: 2019 ACM/IEEE 22nd International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C), IEEE, 2019, pp. 425–430.

[2] Ieee recommended practice for architectural description for software-intensive systems, IEEE Std 1471-2000 (2000) 1–30doi:10.1109/IEEESTD.2000.91944.

[3] B. Kitchenham, P. Brereton, A systematic review of systematic review process research in software engineering, Information and software technology 55 (12) (2013) 2049–2075.

[4] C. Wohlin, P. Runeson, M. Höst, M. Ohlsson, B. Regnell, A. Wesslén, Experimentation in Software Engineering, Computer Science, Springer, 2012.

[5] H. Zhang, M. A. Babar, Systematic reviews in software engineering: An empirical investigation, Information and Software Technology 55 (7) (2013) 1341–1354.

[6] W. Torres, M. G. Van den Brand, A. Serebrenik, A systematic literature review of cross-domain model consistency checking by model management tools, Software and Systems Modeling (2020) 1–20.

[7] A. Iung, J. Carbonell, L. Marchezan, E. Rodrigues, M. Bernardino, F. P. Basso, B. Medeiros, Systematic mapping study on domain-specific language development tools, Empirical Software Engineering 25 (5) (2020) 4205–4249.

[8] S. Erdweg, T. Van Der Storm, M. Völter, L. Tratt, R. Bosman, W. R. Cook, A. Gerritsen, A. Hulshout, S. Kelly, A. Loh, et al., Evaluating and comparing language workbenches: Existing results and benchmarks for the future, Computer Languages, Systems & Structures 44 (2015) 24–47.

[9] M. Franzago, D. Di Ruscio, I. Malavolta, H. Muccini, Collaborative model-driven software engineering: a classification framework and a research map, IEEE Transactions on Software Engineering 44 (12) (2017) 1146–1175.

[10] D. Granada, J. M. Vara, F. P. Blanco, E. Marcos, Model-based tool support for the development of visual editors-a systematic mapping study., in: ICSOFT, 2017, pp. 330–337.

[11] L. M. do Nascimento, D. L. Viana, P. Neto, D. Martins, V. C. Garcia, S. Meira, A systematic mapping study on domain-specific languages, in: The Seventh International Conference on Software Engineering Advances (ICSEA 2012), 2012, pp. 179–187.

[12] E. Negm, S. Makady, A. Salah, Survey on domain specific languages implementation aspects.

[13] B. Merkle, Textual modeling tools: overview and comparison of language workbenches, in: Proceedings of the ACM international conference companion on Object oriented programming systems languages and applications companion, 2010, pp. 139–148.

[14] V. R. Basili, G. Caldiera, H. D. Rombach, The Goal Question Metric Approach, in: Encyclopedia of Software Engineering, Vol. 2, Wiley, 1994, pp. 528–532.

[15] C. Wohlin, Guidelines for snowballing in systematic literature studies and a replication in software engineering, in: Proceedings of the 18th international conference on evaluation and assessment in software engineering, ACM, 2014, p. 38.

[16] V. Garousi, M. Felderer, M. V. Mäntylä, Guidelines for including grey literature and conducting multivocal literature reviews in software engineering, Information and Software Technology (2018).

[17] K. Petersen, R. Feldt, S. Mujtaba, M. Mattsson, Systematic mapping studies in software engineering, in: Proceedings of the 12th International Conference on Evaluation and Assessment in Software Engineering, EASE'08, British Computer Society, Swinton, UK, UK, 2008, pp. 68–77.
URL http://dl.acm.org/citation.cfm?id=2227115.2227123

[18] T. Greenhalgh, R. Peacock, Effectiveness and efficiency of search methods in systematic reviews of complex evidence: audit of primary sources, BMJ 331 (7524) (2005) 1064–1065.

[19] C. Wohlin, Guidelines for snowballing in systematic literature studies and a replication in software engineering, in: Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering, EASE '14, ACM, New York, NY, USA, 2014, pp. 38:1–38:10.

[20] B. A. Kitchenham, S. Charters, Guidelines for performing systematic literature reviews in software engineering, Tech. Rep. EBSE-2007-01, Keele University and University of Durham (2007).

[21] R. Franzosi, Quantitative narrative analysis, no. 162, Sage, 2010.

[22] M. Rodgers, A. Sowden, M. Petticrew, L. Arai, H. Roberts, N. Britten, J. Popay, Testing methodological guidance on the conduct of narrative synthesis in systematic reviews: effectiveness of interventions to promote smoke alarm ownership and function, Evaluation 15 (1) (2009) 49–73.