On the Usage of UML: Initial Results of Analyzing Open UML Models

Philip Langer, Tanja Mayerhofer, Manuel Wimmer, Gerti Kappel

Business Informatics Group
Vienna University of Technology
langer, mayerhofer, wimmer, kappel@big.tuwien.ac.at

Abstract: While UML is recognized as the de-facto standard in modeling software systems, it is at the same time often criticized for being too large and complex. To be able to evolve UML to overcome this criticism, evidence is needed about which parts of UML are actually used. In this respect, a few studies exist that investigate which diagram types of UML are commonly used. However, to the best of our knowledge, in none of these studies, evidence is provided about which modeling concepts of UML are used. Thus, we quantitatively analyze UML models to determine on a fine granularity level the usage frequency of the modeling concepts provided by UML. In this paper, we present initial results of our analysis of 121 open UML models and compare our findings with the results reported in related studies about the usage of UML.

1 Introduction

The first official version of UML (version 0.8) has been proposed already 19 years ago in 1995 [BM98]. Since then, UML underwent several extensive changes leading to the latest version UML 2.4.1 at the time of writing and UML 2.5 is underway. Especially, with the advent of UML 2.0, the language introduces several additional modeling concepts leading to a family of modeling languages usable for different modeling domains [Kob99]. On the one hand, UML is acknowledged as being the de-facto standard in modeling software systems and empirical evidence exists that UML is adopted by the industry (e.g., [HWRK11]). However, on the other hand, UML is often criticized for being too large and too complex mitigating its understandability and adoption [FGDTS06].

To be able to identify a concise core of UML that allows a smooth introduction to the language, evidence is needed about which parts of UML are actually most frequently used. The identification of a concise core of UML might ease the process of learning UML and weaken the barriers to entry for adopting UML in industry. Vendors of UML modeling tools and of UML-based tools, such as code generation frameworks, could focus on increasing the quality of their tools for this core of UML. Based on the knowledge which parts of UML are extensively used, which are scarcely or never used, and which are often extended using UML's language inherent extension mechanism UML profiles, can also help in further evolving UML by discarding unused parts, improving or clarifying the scarcely used parts and enhancing the often extended parts.

Only a few studies investigate the usage of UML [BBB+11] by analyzing the usage frequency of the different diagram types of UML in literature, teaching materials, and tutorials, as well as the frequency in which they are supported by UML tools (e.g., [RLRC13]) or by performing surveys or interviews with practitioners (e.g., [DP06, Pet13]). These studies indicate that certain diagram types are more frequently used than others. However, to the best of our knowledge, in-depth and fine-grained analyses about the usage frequency of UML's modeling concepts, and not only of certain diagram types, are missing so far.

In this paper, we address this lack of a fine-grained study about the usage frequency of UML's modeling concepts. In particular, we quantitatively analyze 121 UML models that are publicly available on the Web and present initial results towards answering the following research questions (RQ):

RQ1: What is the usage frequency of UML's sublanguages? As UML comprises actually a family of languages, we investigate the usage frequency of these sublanguages and also compare our results with those of other existing studies dealing with the usage frequency of the diagram types provided by UML.

RQ2: What is the usage frequency of UML's modeling concepts? Furthermore, we analyze the usage frequency of the modeling concepts provided by UML's sublanguages. To the best of our knowledge, this question has not been investigated by existing studies.

RQ3: What is the usage frequency of UML profiles? In our analysis, we also investigate the usage frequency of UML profiles—UML's language-inherent extension mechanism. Again, to the best of our knowledge, this question has not been investigated before.

The remainder of this paper is structured as follows. In the next section, we provide an overview of related studies on the usage of UML and summarize their findings. In Section 3, we document the model acquisition and analysis process of our study. In Section 4, we present the results of our study and compare them with the results obtained by related studies. Finally, we discuss the threats to the validity of our results in Section 5 and we conclude the paper with an outlook on future work in Section 6.

2 Related Work

A survey on existing studies on the usage of UML in general may be found in [BBB+11]. Please note that we focus on the usage frequency of UML's modeling concepts in this paper and we do not intend to compute metrics about comprehensibility and design quality of UML models as it is done in [NC08, GPC09, MSZJ04]. Thus, we summarize studies related to the usage frequency of UML's modeling concepts and report on language usage studies for domain-specific modeling languages (DSMLs). Finally, we highlight limitations of existing UML usage studies and how we address these limitations in this paper.

Studies on the usage of UML. There are several studies aiming to answer, besides others, the question: what is the usage frequency of the different UML diagram types? Dobing and Parsons [DP06] have been one of the first studying the *how* and *why* of using UML. They focused on questions to which extent a UML diagram type is used and subsequently

relate the results to the complexity of UML. They collected their data between 2003 and 2004 based on an online survey. Grossman *et al.* [GAM05] also investigated the adoption and the usage of UML in the software development community by an online survey. They conclude that there is a wide diversity of opinions regarding UML. Furthermore, a more recent study done by Hutchinson *et al.* [HWRK11] also investigates the question on used modeling languages by using an online survey yielding that 85% of the survey participants use UML as modeling language.

A very recent study on the usage of UML in practice is reported by Petre in [Pet13]. The study is based on interviews with professional software developers and five patterns on UML usage are identified. In the context of this study, the developers have been also asked about the usage of the different diagram types.

A different data acquisition method concerning the sources of information is used by Reggio *et al.* [RLRC13]. Instead of surveys or interviews, different kinds of teaching and training material are investigated, as well as UML tools. Based on this set of different resources, the usage frequency of UML diagram types is determined.

Table 1 summarizes the results of the mentioned studies by stating the three most used UML diagram types as given by the corresponding studies.

While the mentioned studies use surveys, interviews, or analyses of teaching materials, training materials, and UML tools to collect the data for concluding about the usage of UML, only a few studies exist that analyze models for this purpose. In [OC13], Osman and Chaudron analyze ten open source repositories for determining the usage of UML diagrams in addition to other questions, such as model and code co-evolution. They conclude that UML class diagrams are used in the studied open source projects, but other diagrams are scarcely used.

Concerning the usage of UML profiles, Pardillo [Par10] used a manual approach analyzing existing literature on UML profiles to identify the current practices in defining them. He concludes that the majority of profiles is defined for class diagrams and only a few profiles are defined for the behavioral part of UML.

Studies on the usage of DSMLs. Two studies about language usage are presented by Tairas and Cabot in [TC13]. In particular, two DSMLs are analyzed by collecting different usage statistics, such as the instantiation frequency of metaclasses. A similar approach is followed by Kusel *et al.* [KSW+13] where the subject of investigation is the application frequency of reuse mechanisms of the ATLAS Transformation Language (ATL). Finally, Williams *et al.* [WZM+13] discuss through a corpus-based analysis of metamodels defined in Ecore how metamodels look like by computing several different metrics.

Rank	Dobing & Parsons [DP06]	Grossman *et al.* [GAM05]	Reggio *et al.* [RLRC13]	Petre [Pet13]	Hutchinson *et al.* [HWRK11]
1	Class Diagram	Use Case Diagram	Class Diagram	Class Diagram	Class Diagram
2	Use Case Diagram	Class Diagram	Activity Diagram	Sequence Diagram	Activity Diagram
3	Sequence Diagram	Sequence Diagram	Sequence Diagram	Activity Diagram	Use Case Diagram

Table 1: Most used UML diagram types reported in literature.

Going beyond the state-of-the-art. Several papers aim to uncover the how and why UML is used by utilizing different data acquisition methods and analysis approaches. However, there is currently a lack of fine-grained quantitative studies investigating a larger corpus of UML models. Such studies are needed in order to answer more detailed research questions concerning how UML is used from a language perspective. Currently, the usage of UML is discussed on diagram level granularity, only. But, for instance, the usage frequency of modeling concepts has not been explored to the best of our knowledge. Thus, we aim in this paper for a fine-grained quantitative study in the spirit of [LKR05, KSW+13, TC13, WZM+13] based on a corpus of open UML models to answer additional research questions on the usage of UML.

3 Study Design

As an infrastructure for our analysis, we chose to use the UML modeling tool Enterprise Architect[1] (Version 9.0, Ultimate Edition) of the company Sparx Systems—which is one of the most popular UML modeling tools[2] and supporting UML 2.4.1. The main reason for this choice is our long-standing research collaboration with Sparx System's global partner SparxSystems Software GmbH Central Europe who supports us in this study.

The corpus of analyzed models consists of open UML models, which are publicly available on the Web and which have been created with Enterprise Architect. However, in ongoing work we are currently also analyzing—by the same means as described in the following—open UML models that have been created with other modeling tools.

In the remainder of this section we first describe how the analyzed UML models have been retrieved from the Web and provide general figures about these models, and we second explain the model analysis process used in our study[3].

3.1 Data Set

For retrieving publicly accessible UML models from the Web that have been created with Enterprise Architect, we used the file type search provided by the search engine of Google, which enables to search for files with specific file extensions. Thus, for searching Enterprise Architect models, we used the search string `filetype:eap`. We carried out this search twice, once in December 2012 and once in April 2013 resulting in 151 UML models. By investigating the URLs of the websites hosting these 151 models and reviewing the models' content, we identified twelve duplicates and 17 revisions of retrieved UML models, as well as one empty model. After filtering these models, we analyzed the remaining 121 UML models regarding their usage of UML's modeling concepts.

[1] `http://www.sparxsystems.com/products/ea/index.html`

[2] `http://list.ly/list/2io-popular-uml-modeling-tools`

[3] Please refer to `http://www.modelexecution.org/?page_id=982` for additional information about the design of our study.

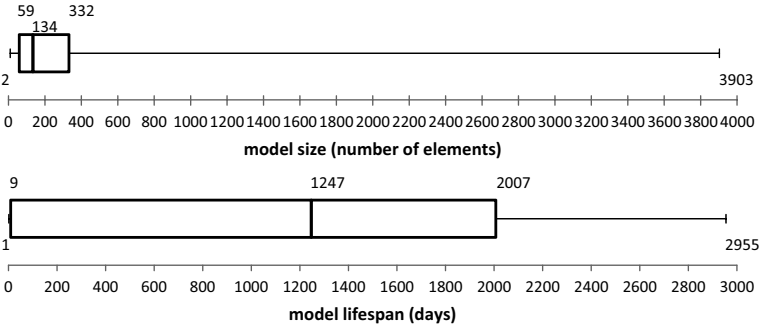

Figure 1: Size and lifespan of the analyzed UML models.

About half of the models (54%) have been retrieved from open source software repositories, namely *google code*, *assembla*, and *github*. About a quarter of the models (28%) has been retrieved from project websites which use the software project management system *trac*, and 18% have been retrieved from other sources. By manually reviewing the models' content, we found out that they are also mainly concerned with software-related aspects of the modeled systems.

To characterize the retrieved models, we determined their size in terms of the number of elements contained by the models, as well as the lifespan of the models in terms of days between the creation date and the last modification date of the models (cf. Figure 1).

As can be seen in the boxplot of the sizes of the analyzed models, depicted at the top of Figure 1, the smallest model contained two elements, while the largest model contained 3903 elements. The average number of elements (arithmetic mean) contained by the analyzed UML models is 385 and the median number is 134, the first quartile is 59 and the third quartile is 332. Thus, our data set does not contain very huge models (companies report on models comprising tens of thousands of elements [KPP08]), but they are on average of reasonable size for being considered useful for analyzing the usage frequency of UML's modeling concepts.

The boxplot of the lifespans of the analyzed models, depicted at the bottom of Figure 1, shows, that the minimal lifespan of the analyzed models is one day while the maximal lifespan is 2955 days (i.e., about eight years). The average lifespan (arithmetic mean) of the analyzed models is 1083 days (i.e., about three years) and the median value for the lifespan is 1247 days (i.e., 3.4 years), the first quartile is 9 days and the third quartile is 2007 days (i.e., 5.5 years). This data shows that on average the models have been created and maintained for a considerable period of time.

3.2 Data Analysis

Enterprise Architect provides an API, as well as a scripting environment that can be used to directly access the content of UML models. Using the API and the scripting environment,

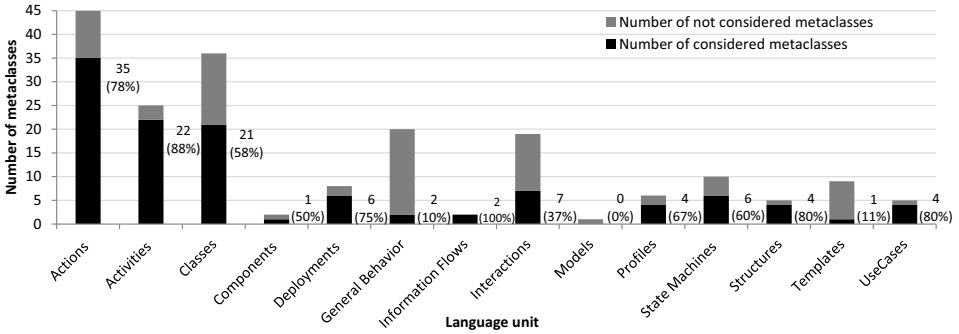

Figure 2: Number of metaclasses (considered / not considered in the analysis) per language unit.

we developed a script that iterates over all elements contained by a model and determines for each element its type, as well as whether a stereotype is applied on the element. The type of a model element corresponds to the instantiated UML metaclass. We regard each non-abstract metaclass defined in the metamodel of UML as a modeling concept of UML. We assigned each of these metaclasses to exactly one UML language unit, that is the language unit to which the package containing the respective metaclass is assigned to by the UML standard [Obj11, page 7–8]. Please note that we could not include all UML metaclasses in our analysis because they are either not explicitly represented in Enterprise Architect (e.g., the metaclass LiteralBoolean is represented as simple String), or they are not available at all in Enterprise Architect (e.g., the metaclass ReduceAction). Figure 2 depicts the number of metaclasses assigned to the respective language units, as well as the number of metaclasses which have been considered in the analysis and the number of metaclasses which have not been considered. In summary, 115 of 193 non-abstract metaclasses defined in the metamodel of UML are considered in our analysis.

As output of the script, we obtain an XML file that contains the information how often each modeling concept is used by the analyzed UML model and how often each modeling concept is extended by stereotype applications. Based on this data, it is possible to determine for each analyzed UML model the size of the model in terms of contained model elements, the usage frequency of UML's language units, the usage frequency of UML's modeling concepts, and the usage frequency of UML profiles. We calculate the respective figures using an additional Java program which aggregates the data captured in the XML files obtained for the analyzed models.

4 Analysis Results

In this section, we present the analysis results based on our corpus of open UML models structured according to the research questions RQ1-3[4].

[4]More detailed results can be found at `http://www.modelexecution.org/?page_id=982`.

4.1 RQ1: UML Sublanguages

The modeling concepts of UML are organized in 14 *language units* which represent *sublanguages* of UML. Each language unit consists of modeling concepts that provide the means for modeling a certain aspect of a system under study according to a particular paradigm. For instance, the language unit *Activities* defines modeling concepts enabling to model the behavior of a system based on a workflow-like paradigm.

To get a first indicator about which parts of UML are used, we determine the usage frequency of UML's language units in the analyzed UML models. In particular, we compute (i) the number of language units used by the analyzed models, (ii) the number of models using a particular language unit, and (iii) the number of models using distinct combinations of language units.

(i) **Number of language units used per model.** We regard a model to use a distinct language unit if it contains at least one instance of at least one modeling concept assigned to this language unit. As depicted in Figure 3, 34% of the analyzed models use modeling concepts of one language unit, 17% use modeling concepts of two language units, and 22% use modeling concepts of three language units. Thus, three-quarter of all models (73%) use modeling concepts of up to three UML language units. The average number (arithmetic mean) of used language units is 2.72 and the median number is two.

(ii) **Usage frequency of language units.** Figure 4 depicts the frequency in which the distinct UML language units are used by the analyzed models. The three most frequently used UML language units are *Classes*, *Use Cases*, and *Interactions*. All analyzed models use the language unit *Classes*, 47% use the language unit *Use Cases*, and 39% use the language unit *Interactions*. This result is in line with the findings of Dobing and Parsons [DP06] who report that 73% of their survey respondents use class diagram, 51% use case diagrams, and 50% sequence diagrams in two-thirds or more of their projects. Compared to the findings of Grossmann *et al.* [GAM05], who report that use case diagrams, class diagrams, and sequence diagrams are used by about 90% of their survey respondents, our analysis results indicate that modeling concepts defined by the language unit *Classes* are significantly more frequently used than modeling concepts defined by the language units *Use Cases* and *Interactions*. The studies of Reggio *et al.* [RLRC13] and Petre [Pet13] also identified class diagrams and sequence diagrams among the most frequently used three UML diagram types. However, in their studies as well as in the study of Hutchinson *et*

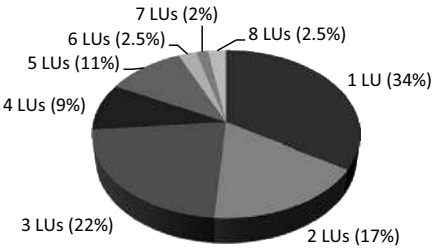

Figure 3: Number of language units (LU) used per model.

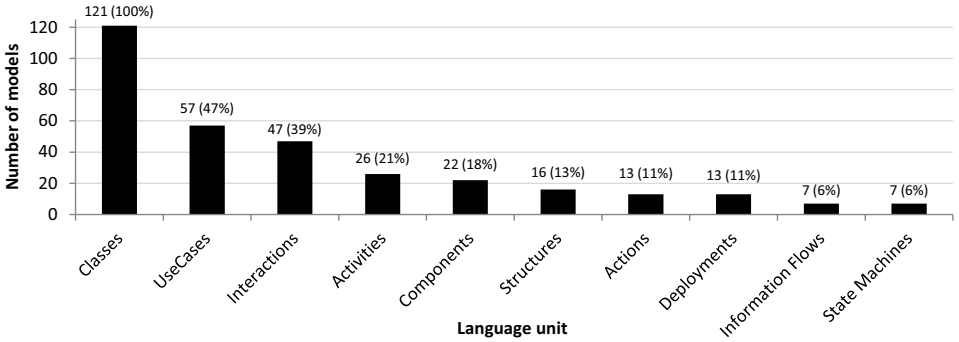

Figure 4: Number of models using modeling concepts of a particular language unit.

al. [HWRK11], activity diagrams are among the top three used diagram types, whereas the language unit *Activities* is on fourth position in our ranking. The language unit *State Machines* is according to our analysis results the least used language unit, however, the other studies did not rank state machine diagrams on the last position of used diagram types. The language units *General Behavior*, *Profiles*, and *Templates* are not used at all by the analyzed models. However, it has to be noted that for the language units *General Behavior* and *Templates* only a low number of metaclasses has been considered in the study.

(iii) **Usage frequency of language unit combinations.** As depicted in Figure 3, 66% of the analyzed models use modeling concepts defined by two or more UML language units. Figure 5 shows on the left-hand side that 71% of these models use modeling concepts defined in the language units *Classes* and *Use Cases*. Other frequently used combinations of two language units are *Classes* and *Interactions* (59%), *Use Cases* and *Interactions* (50%), as well as *Classes* and *Activities* (33%).

We also determined which combinations of three language units are frequently used. 49% of the analyzed models use modeling concepts of three or more UML language units (cf. Figure 3). 68% of these models use modeling concepts of each of the language units *Classes*, *Use Cases*, and *Interactions*, which are the most frequently used language units. Other less frequently used combinations of three language units are *Classes*, *Use Cases*, and *Activities* (31%), *Classes*, *Activities*, and *Interactions* (25%), *Classes*, *Use Cases*, and *Components* (25%), as well as *Classes*, *Structures*, and *Components* (24%).

For the usage of language unit combinations we can conclude that the most frequently used language units are also the ones that are most frequently used in combination.

4.2 RQ2: UML Modeling Concepts

Each language unit in UML consists of a number of modeling concepts, which are represented by metaclasses. For instance, the language unit *Activity* contains the metaclass InitialNode, which is a control node that initiates the control flow in an invoked Activity.

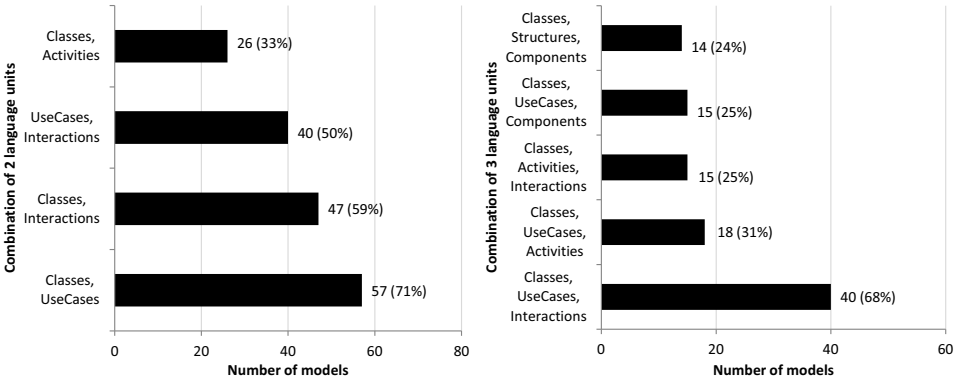

Figure 5: Number of models using a particular combination of language units.

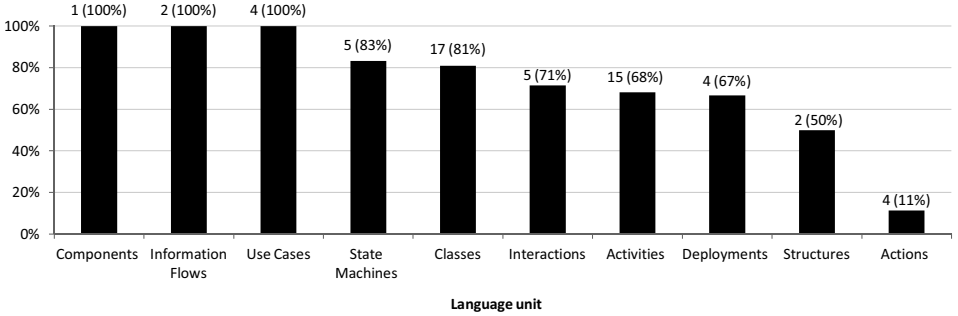

Figure 6: Proportion of used metaclasses among all considered metaclasses per language unit.

When creating a UML model, these metaclasses are instantiated and populated with property values. The number of metaclasses in the language units vary significantly ranging from one metaclass in the language unit *Models* up to 45 metaclasses in *Actions*.

In this section, we analyze the usage frequency of these metaclasses. In particular, (i) we identify the proportion of metaclasses that are actually used among all considered metaclasses. To also consider how often they are used, we further analyze (ii) in how many models each metaclass is used, and the number of instances of each metaclass in comparison to the number of all model elements of the respective language unit. Here, we focus on the most frequently used language units *Classes*, *Interactions*, and *Use Cases*.

(i) **Used metaclasses versus unused metaclasses.** Figure 6 depicts the proportion of metaclasses that are actually used (i.e., at least one instance exists) among all considered metaclasses in the respective language unit. According to this data, the language units *Components*, *Information Flows*, and *Use Cases* seem to be the most concise ones, as *all* metaclasses introduced in these language units are actually used in the considered models. For the language unit *State Machines*, we also observe a high proportion of used metaclasses among all metaclasses: 83% of the metaclasses are instantiated at least once. We may also highlight the language unit *Classes*. This language unit contains significantly

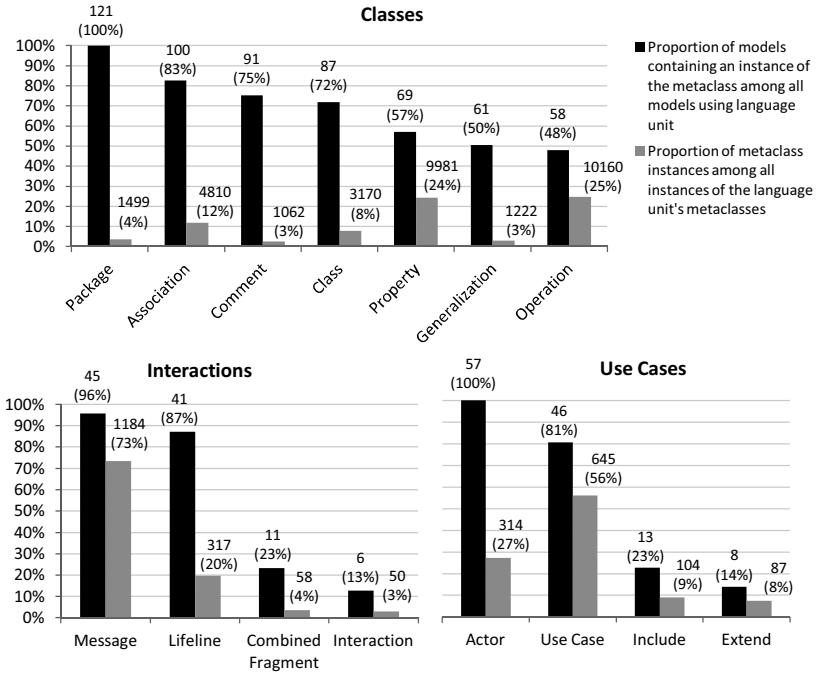

Figure 7: Usage frequency of UML metaclasses.

more metaclasses than *Components*, *Information Flows*, *Use Cases*, and *State Machines*, and still 81% of all available metaclasses are instantiated in the analyzed models. This indicates that the comparatively large number of metaclasses of the language unit *Classes* seems to be reasonable, largely required, and well understood. On the contrary, the high number of available metaclasses in the language unit *Actions* is not instantiated to a high extent in the analyzed models, as only 11% of the metaclasses are used.

(ii) **Usage frequency of metaclasses.** In Figure 7, the usage frequencies of the most frequently used metaclasses of the language units *Classes*, *Interactions*, and *Use Cases* are depicted. In particular, we show the usage frequency in terms of the absolute and the relative number of models that contain at least one instance of the respective metaclass, as well as the absolute number of instances of the respective metaclass and their proportion to the overall number of all model elements of the respective language unit. For instance, the metaclass Association of the language unit *Classes* is instantiated in 100 models (i.e., 83% of all models that use the language unit *Classes*) and 12% of all instances of metaclasses defined in the language unit *Classes* are instances of Association (i.e., 4810 instances of Association among 40993 instances of any metaclass of the language unit *Classes*).

Classes. Considering the most frequently instantiated metaclasses of the language unit *Classes*, it is not surprising that Package is used by all models, because a model created with Enterprise Architect must define at least one Package containing all other model elements. The metaclass Class is used by 72% of the models that use the language unit

Classes. The reason why not all of these models use the modeling concept Class is that the language unit *Classes* contains also metaclasses that are used in combination with metaclasses of other language units. For instance, the modeling concept Generalization may not only be used among instances of Class, but also among instances of Actor (belonging to the language unit *Use Cases*). Similarly, as the metaclass Association is, for instance, also used to associate an Actor with a UseCase, more models contain instances of Association than models containing instances of Class. Interestingly, there are more models that contain instances of Class than models containing instances of either Operation or Property. This indicates that in some of the models Class instances exist without owned operations or properties. It is also worth noting that 75% of the models that use the language unit *Classes* also contain at least one instance of Comment, whereas the number of comments is rather small (3% of the model elements). Modeling concepts that are not used at all in the analyzed models are Abstraction, PackageImport, PackageMerge, and PrimitiveType. The remaining 10 metaclasses of the language unit *Classes* considered in this study are used in between 2% and 39% of the analyzed models using this language unit.

Interactions. Nearly all models that use metaclasses of the language unit *Interactions* contain at least one instance of Message (96%) and Lifeline (87%). The proportion of instances of these metaclasses together account for 93% among all instances of metaclasses defined in this language unit. The metaclasses CombinedFragment and Interaction are only used by 23% and 13% of the models, respectively. The low number of models using the metaclass Interaction can be justified by the fact that Enterprise Architect does not implement the restriction, defined in the UML metamodel, that instances of the metaclasses Lifeline and Message have to be contained by an instance of Interaction. If a user creates, for instance, a new sequence diagram in Enterprise Architect, no Interaction instance is created automatically, but the user can still add Lifeline instances and Message instances to the diagram. The metaclass Gate is only used in two models which contain one instance each, whereas the metaclasses Continuation and StateInvariant are not used at all.

Use Cases. The most frequently used metaclasses of the language unit *Use Cases* are Actor and UseCase, which are instantiated in 100% and 81% of the models using the language unit *Use Cases*, respectively. This also means that 19% of the models that contain an instance of Actor do not contain an instance of UseCase. This finding can be justified by two reasons. First, some of the analyzed models contain only Actor instances without associating them to UseCase instances. Second, in Enterprise Architect actors can be used equivalently to lifelines for defining interactions. In the analyzed models, instances of Actor and UseCase together account for 83% of all UML elements from the language unit *Use Cases*, whereas there are on average around two use cases per actor in the analyzed models. Interestingly, the metaclasses Extend and Include are only scarcely used in the analyzed models: only 23% and 14% of the models use them at all and their instances account together for only 17% of all UML elements of the language unit *Use Cases*.

Other language units. Besides the language units discussed above, we also highlight some interesting findings in the language units *Activities*, *Actions*, and *Deployments*. However, we omit a dedicated figure due to space limitations.

Among all models that contain instances of metaclasses of the language unit *Activities*, the most frequently used metaclasses are ControlFlow (used in 88% of the respective models),

Activity (77%), InitialNode (73%), as well as ActivityFinalNode and DecisionNode (65% each). Modeling concepts of the language unit *Activities*, that are not used in any of the analyzed models, are ActivityParameter, CentralBufferNode, ConditionalNode, ExceptionHandler, ExpansionNode, SequenceNode, and StructuredActivityNode.

Unfortunately, only 13 models contain at least one instance of a metaclass containted by the language unit *Actions* (cf. Figure 4), which mitigates the validity of any general conclusions that we may draw from this data. However, it is interesting to note that in these models, only the metaclass OpaqueAction has been used frequently (in twelve out of 13 models). CallOperationAction, Pin, and WriteVariableAction are only used in three, two, and one of 13 models, respectively, whereas all other action types are not used at all.

Also modeling concepts of the language unit *Deployments* are only used in 13 models (cf. Figure 4). From this language unit, mainly Node and Device are adopted frequently: Eleven models use Node and eight models use Device. The metaclasses ExecutionEnvironment and DeploymentSpecification are only used in three and two models, respectively, and Artifact, as well as Manifestation, are not used at all.

4.3 RQ3: UML Profiles

We explore in this subsection the usage of UML profiles in our corpus of UML models. In particular, we identify (i) the ratio of models containing profile applications compared to those not having profile applications. To consider to which extent the models are profiled, we further analyze (ii) the ratio of stereotyped elements in models, i.e., elements that have at least one stereotype applied, compared to non-stereotyped elements, i.e., elements that have no stereotype applied. Finally, we determine (iii) the most frequently used stereotypes and the most frequently extended metaclasses.

(i) **Ratio of profiled versus non-profiled models.** We identified the models that have at least one stereotype applied on one of their model elements resulting in a profiling rate of 59% for the analyzed models. In this context, we had to consider also the realization peculiarities of UML. In particular, we did not count the application of UML standard stereotypes and keywords. Thus, more stereotype applications may exist, which are, however, considered as an alternative way to represent standard UML concepts.

(ii) **Ratio of stereotyped versus non-stereotyped elements.** Some models are heavily extended by profile applications. One model even contains nearly exclusively stereotyped elements which is a strong indicator for the usage of UML profiles providing mandatory stereotypes. On average (arithmetic mean), 22% of the elements contained by a profiled model are stereotyped.

(iii) **Most frequently extended metaclasses / most frequently used stereotypes.** In Figure 8, we show the ten most frequently extended metaclasses according to the absolute number of the metaclasses' instances with stereotype applications. The most frequently extended metaclasses are Property, Operation, Class, and Association. Stereotype applications that are attached to instances of these four metaclasses account for 88% of all stereotype applications for the given model population.

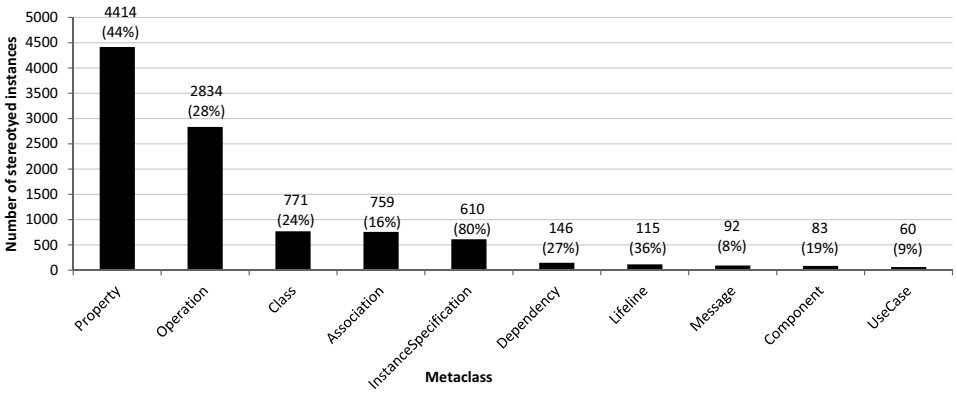

Figure 8: Number of stereotyped model elements per metaclass.

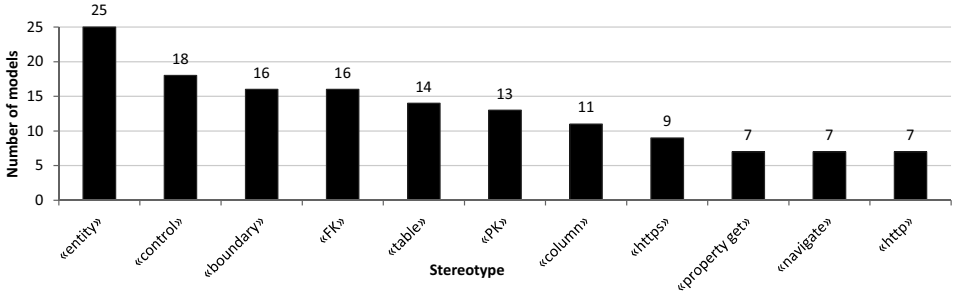

Figure 9: Number of models using a particular stereotype.

When considering the frequency of stereotype applications per metaclass, i.e., the ratio of stereotyped instances versus non-stereotyped instances, instances of InstanceSpecification are most frequently stereotyped, i.e., 80% of all instances of InstanceSpecification contained be the analyzed models are stereotyped. Instances of Property, Lifeline, Operation, Dependency, and Class are also frequently stereotyped (24%–44%).

Figure 9 shows the stereotypes that are used in at least seven distinct models. When considering the domains for which profiles are used in the analyzed models, we observe that stereotypes for defining robustness diagrams (≪entity≫, ≪control≫, and ≪boundary≫) for modeling model-view-controller applications are most frequently used. Besides, we encountered stereotypes for expressing database concepts (≪FK≫, ≪table≫, ≪PK≫, and ≪column≫) and Web application concepts (≪https≫, ≪navigate≫, and ≪http≫) quite often.

Finally, we relate our analysis results with the results obtained by the literature study of Pardillo [Par10]. One of the main findings of Pardillo is that the majority of profiles are defined for the metaclasses Class, Association, and Property. Our results confirm this finding as Property, Operation, Class, and Association are the most frequently stereotyped metaclasses in terms of absolute numbers of stereotyped instances.

5 Threats to Validity

Internal threats. We identified the following two factors that might affect the validity of our analysis results for the analyzed corpus of UML models.

We were not able to analyze the usage of all modeling concepts provided by UML, because not all of them are explicitly supported by Enterprise Architect. From the 193 modeling concepts of UML, we could only consider 115 of them in the analysis (cf. Figure 2). For determining which modeling concepts are explicitly supported, we reviewed the Enterprise Architect user guide, as well as the tool itself.

In the analysis, we chose to classify all modeling concepts provided by UML into sublanguages according to their assignment to language units defined by the UML standard. However, this sublanguage categorization does not take into account that certain modeling concepts can be used also when applying other sublanguages.

External threats. The following characteristics of the analyzed UML models restrict the extent at which it is possible to generalize our findings. The presented analysis considers open models that are publicly available on the Web and that have been created with Enterprise Architect. Further, the analyzed sample is with 121 UML models fairly small. The analyzed set of models does not contain very huge models. Furthermore, most of the models are obtained from open source repositories and thus may largely concern the domain of software systems. We did not take into account the purpose of the models, such as documentation, specification, code generation, or reverse engineering, which might have an impact on which modeling concepts of UML are used. Due to these characteristics of the analyzed data set, the obtained results are only valid for this set of analyzed UML models and they cannot be generalized for closed UML models (i.e., models that are not publicly available), UML models that have been created with UML modeling tools other than Enterprise Architect, huge models, or models that have been used in other application domains than software systems. Further, it has to be investigated whether the purpose of the models has an impact on which modeling concepts of UML are used.

Despite these limitations concerning the generalisability of our results, the analysis method presented in this paper can be applied to analyze arbitrary UML models to compute the usage frequency of UML's sublanguages, UML's modeling concepts, and UML profiles.

6 Summary and Outlook

We presented the results of analyzing 121 open UML models concerning the usage frequency of the sublanguages and modeling concepts of UML, as well as of UML profiles. Our stated research questions have been answered for this set of UML models as follows.

RQ1: What is the usage frequency of UML's sublanguages? The language units that are most frequently used in the analyzed models are *Classes*, *Use Cases*, and *Interactions* (100%, 47%, and 39% of the models, respectively).

RQ2: What is the usage frequency of UML's modeling concepts? We may conclude

that models using the language unit *Classes* use several modeling concepts quite frequently, such as Class, Property, Operation, Generalization, and Association, whereas in the language units *Interactions* and *Use Cases* mainly two modeling concepts each account for the largest proportion among model elements of the respective language unit: 93% of all model elements of *Interactions* metaclasses are either instances of Message or Lifeline and 83% of all model elements of *Use Cases* are either instances of Actor or Use Case.

RQ3: What is the usage frequency of UML profiles? From the observations made in this study, we may conclude that profiles are frequently used for defining robustness diagrams and database schemas. The core concepts of the language unit *Classes*, which are the most frequently used modeling concepts, are also most frequently stereotyped.

These results provide a first indication of which modeling concepts could be contained in a concise core of UML. In future work, we plan to enlarge the corpus of analyzed UML models with closed UML models from our collaborator SparxSystems Software GmbH Central Europe, as well as with models created using other UML modeling tools, to counteract the external threats to validity and to enable drawing more general conclusions. Furthermore, we plan to relate the usage frequency of UML's modeling concepts with different characteristics of the analyzed models, such as their size, lifespan, number of authors, purpose, and domain.

Acknowledgments. We thank Alexander Bohn and Tobias Fink for their contributions to the presented study and SparxSystems Software GmbH Central Europe for their support. This work is partly funded by the European Commission under the ICT Policy Support Programme grant no. 317859 and by the Austrian Federal Ministry of Transport, Innovation and Technology (BMVIT) under the FFG BRIDGE program grant no. 832160.

References

[BBB+11] David Budgen, Andy J. Burn, O. Pearl Brereton, Barbara A. Kitchenham, and Rialette Pretorius. Empirical evidence about the UML: a systematic literature review. *Softw., Pract. Exper.*, 41(4):363–392, 2011.

[BM98] Jean Bézivin and Pierre-Alain Muller. UML: The Birth and Rise of a Standard Modeling Notation. In *First International Workshop on the Unified Modeling Language (UML)*, pages 1–8, 1998.

[DP06] Brian Dobing and Jeffrey Parsons. How UML is used. *CACM*, 49(5):109–113, 2006.

[FGDTS06] Robert B. France, Sudipto Ghosh, Trung T. Dinh-Trong, and Arnor Solberg. Model-Driven Development Using UML 2.0: Promises and Pitfalls. *IEEE Computer*, 39(2):59–66, 2006.

[GAM05] Martin Grossman, Jay E. Aronson, and Richard V. McCarthy. Does UML make the grade? Insights from the software development community. *Information & Software Technology*, 47(6):383–397, 2005.

[GPC09] Marcela Genero, Mario Piattini, and Michel R. V. Chaudron. Quality of UML models. *Information & Software Technology*, 51(12):1629–1630, 2009.

[HWRK11] John Hutchinson, Jon Whittle, Mark Rouncefield, and Steinar Kristoffersen. Empirical assessment of MDE in industry. In *33rd International Conference on Software Engineering (ICSE)*, pages 471–480, 2011.

[Kob99] Cris Kobryn. UML 2001: A Standardization Odyssey. *CACM*, 42(10):29–37, 1999.

[KPP08] Dimitrios S. Kolovos, Richard F. Paige, and Fiona Polack. The Grand Challenge of Scalability for Model Driven Engineering. In *Reports and Revised Selected Papers of Workshops and Symposia at MODELS'08*, pages 48–53, 2008.

[KSW+13] Angelika Kusel, Johannes Schönböck, Manuel Wimmer, Werner Retschitzegger, Wieland Schwinger, and Gerti Kappel. Reality Check for Model Transformation Reuse: The ATL Transformation Zoo Case Study. In *International Workshop on Analysis of Model Transformations (AMT) @ MODELS*, 2013.

[LKR05] Ralf Lämmel, Stan Kitsis, and Dave Remy. Analysis of XML Schema Usage. In *XML Conference*, 2005.

[MSZJ04] Haohai Ma, Weizhong Shao, Lu Zhang, and Yanbing Jiang. Applying OO Metrics to Assess UML Meta-models. In *7th International Conference on the Unified Modelling Language (UML)*, pages 12–26, 2004.

[NC08] Ariadi Nugroho and Michel R. V. Chaudron. A survey into the rigor of UML use and its perceived impact on quality and productivity. In *2nd International Symposium on Empirical Software Engineering and Measurement (ESEM)*, pages 90–99, 2008.

[Obj11] Object Management Group. OMG Unified Modeling Language (OMG UML), Superstructure, Version 2.4.1, August 2011. Available at: http://www.omg.org/spec/UML/2.4.1.

[OC13] Mohd Hafeez Osman and Michel Chaudron. UML usage in Open Source Software Development : A Field Study. In *International Workshop on Experiences and Empirical Studies in Software Modelling (EESSMOD) @ MODELS*, pages 23–32, 2013.

[Par10] Jesús Pardillo. A Systematic Review on the Definition of UML Profiles. In *13th International Conference on Model Driven Engineering Languages and Systems (MODELS)*, pages 407–422, 2010.

[Pet13] Marian Petre. UML in practice. In *35th International Conference on Software Engineering (ICSE)*, pages 722–731, 2013.

[RLRC13] Gianna Reggio, Maurizio Leotta, Filippo Ricca, and Diego Clerissi. What are the used UML diagrams? A Preliminary Survey. In *International Workshop on Experiences and Empirical Studies in Software Modelling (EESSMOD) @ MODELS*, pages 3–12, 2013.

[TC13] Robert Tairas and Jordi Cabot. Corpus-based analysis of domain-specific languages. *Software & Systems Modeling*, pages 1–16, 2013.

[WZM+13] James Williams, Athanasios Zolotas, Nicholas Matragkas, Louis Rose, Dimitris Kolovos, Richard Paige, and Fiona Polack. What do metamodels really look like? In *International Workshop on Experiences and Empirical Studies in Software Modelling (EESSMOD) @ MODELS*, pages 55–60, 2013.