Systematic Mapping
Formalization of UML Semantics using Temporal Logic

Vinicius Pereira and Marcio E. Delamaro
Instituto de Ciencias Matematicas e Computacao, Universidade de Sao Paulo, Sao Carlos, SP, Brazil

Keywords: Systematic Mapping, UML Semantics, Unified Modeling Language, Temporal Logic, Formal Verification.

Abstract: Despite offering a wide variety of elements for graphical representation of models, the UML does not have a well-defined semantics. Therefore, over the years researches seek to assign some kind of formal semantics for UML. **Objective:** In this context, this paper seeks to bring evidence about the techniques for formalizing the UML semantics available in the literature, particularly those using temporal logic. **Method:** For this purpose, we conducted a systematic mapping study based on searching of major electronic databases. **Results:** We explored 278 studies, of which we claim 13 studies for analysis. In other words, the result shows that the overall picture defined by them is interest, because it shows that the majority of studies deal only with the formalization of one type of UML diagram. **Conclusion:** Summing up, we found out that State Diagram is the more formalized diagram in the studies. It is difficult to find the formalization of three or more UML diagrams, perhaps because of the difficulty in ensuring the overlap between UML elements. Furthermore, the results can provide perception of new research in the UML semantics for investigating and defining new tools/process to assist the software engineers.

1 INTRODUCTION

Formal methods introduces a great rigor in the software development. It aims to improve software structure and maintainability besides making them less prone to errors. Consequently, the formal methods field have researches that aim to creating techniques and methods to help with software formal specification (Bowen and Hinchey, 2006). Furthermore, UML (Unified Modeling Language) (Eriksson et al., 2004) Eriksson04 is a modeling language widely spread and employed in different fields of computer application in industry (Grobelna et al., 2010). Despite the benefits, however, its semantics is not well defined. Moreover, in most cases developers use the semantics of target programming language (e.g. Java). Thus, many researchers seek to assign a clearer semantics for UML elements, e.g. (Micskei and Waeselynck, 2011; Bouabana-Tebiel, 2009; Snook and Butler, 2006). These researches deal with a limited number of UML elements and impose constraints on practical use of the same. Nevertheless, we noted that the proposed approaches do not always provide support tools for users without large experience in formal methods, making it difficult the adoption of formal verification by software developers.

The objective here is to present the planning and conduction of a systematic mapping that aimed to identify studies that have one or more items of interest related to the theme "formalization of the UML semantics using temporal logic", using the systematic review guidelines (Kitchenham, 2004) as base.

Following this introduction, this is our papers structure: in Section 2 we present the planning of systematic mapping, including the strategy adopted to select and use of the search engines, selection of works, among other items; Section 3 shows how we conduct the selection of the studies. Section 4 discusses the results. In Section 5 we present the main findings in this systematic mapping. The Section 6 shows the threats to validity identified in this paper. Finally, Section 7 presents the conclusions of this research.

2 PLANNING

We conducted the systematic mapping based on the protocol model for systematic review presented by Kitchenham (Kitchenham, 2004). The objective is to analyze the techniques and approaches in

Pereira V. and E. Delamaro M..
Systematic Mapping - Formalization of UML Semantics using Temporal Logic.
DOI: 10.5220/0005468704860493
In *Proceedings of the 17th International Conference on Enterprise Information Systems* (ICEIS-2015), pages 486-493
ISBN: 978-989-758-097-0

the literature that use formal verification to assign semantic to UML models. In order to achieve such objective we worked out the following research questions:

- (**RQ1**) *"Which techniques to formalize the semantics of UML models using temporal logic have been proposed until the moment of this research?"*
- (**RQ2**) *"Which UML diagrams are more common to be formalized?"*
- (**RQ3**) *"Which are support tools and how practical they are?"*
- (**RQ4**) *"Which techniques has the results displayed back to the model?"*

According to Kitchenham (Kitchenham, 2004) it is necessary to define the scope of the systematic mapping, which describes the population considered, intervention and expected results. Therefore, we defined the scope as follows: (i) Population: primary studies related to techniques to formalize the semantics of UML models; (ii) Intervention: primary studies that document, describe and present techniques to formalize the semantics of UML models; and (iii) Expected results: overview of the main techniques to assign formal semantics to UML that we investigated in the literature. Based on this result, we intend to highlight the researches that aimed to enhance or introduce new techniques to formalize the semantics of UML models.

2.1 Search Strategy

To establish the search strategy of the primary studies, we built the search string and selected the electronic databases. Thus, we defined the search string based on a set of keywords and acronyms. We split these keywords and acronyms into some categories, such as:

- Related to Techniques: *Technique, Approach, Framework, Environment* and *Tool*;
- Related to UML: *Unified Modeling Language, Unified Modelling Language* (variation between American and British English) and *UML*;
- Related to UML Semantics: *UML Semantics* and *Semantics of UML* (these two cases are together with the previous one to avoid an AND clause, which could constrain the results);
- Related to formal semantics, formalization and other cases: *Formal Semantics, Formalization, Formalisation* (variation between American and British English), *Formal Verification* (e.g. "formal verification of UML models") and *Formal Model* (e.g. "UML...verification of its formal models") and
- Related to temporal logic: *Temporal Logic*.

Based on these keywords and acronyms we created the following search string:

(("Technique" OR "Approach" OR "Framework" OR "Environment" OR "Tool") AND ("Unified Modeling Language" OR "Unified Modelling Language" OR "UML" OR "UML Semantics" OR "Semantics of UML") AND ("Formal Semantics" OR "Formalization" OR "Formalisation" OR "Formal Verification" OR "Formal Model") AND ("Temporal Logic")).

The next step involved choosing which databases we should use to conduce the search of primary studies. We chose the following: ACM Digital Library, IEEE Xplore, Scopus and Springer. These search databases were chosen because, according to Dyba et al (Dyba et al., 2007), they are among the most relevant in the context of software engineering. In addition, we conducted a manual search to verify if it was possible to find a study that we missed by using the search string.

2.2 Inclusion/Exclusion Criteria

The inclusion and exclusion criteria support the selection of relevant studies and appropriately assist in clarifying the research question proposed. In this context, studies were examined according to the following criteria:

- **(Criterion 1)** Studies that deal with the formal semantics of UML.
- **(Criterion 2)** Studies that show the formalization through the use of temporal logic.
- **(Criterion 3)** Complete studies in English or Portuguese.
- **(Criterion 4)** Studies with full text available on the Web.
- **(Criterion 5)** If there is duplicate studies, one should select the most recent study and/or more complete. Exclusion criteria are the negation of the criteria presented above. For example, if a primary study is not written in English or Portuguese, Criterion 3 will exclude it.

The selection of primary studies was performed by the authors of this paper and other three students in two stages: a preliminary and a final selection. After this, it was applied quality criteria for the studies selected, in order to improve the level of primary studies chosen. Regarding the preliminary selection stage, the process followed was: **(i)** Identify relevant studies; **(ii)** Exclusion of studies based on reading the title and abstract; and **(iii)** Validation of the selected studies. The final selection had the following procedure: **(i)** Selection based on a full reading of the papers; **(ii)** Validation of the results; and **(iii)** Conduct of Quality Evaluation of the selected studies.

Finally, to guide the Quality Evaluation, we stipulated some quality criteria (QC), applied in the primary studies remaining after the final selection. They are:

- **(QC1)** Is there a clear description of the technique/approach used to formalize the semantics of UML models?
- **(QC2)** Is there a presentation about the used temporal logic?
- **(QC3)** Is there a description of the characteristics of the support tool?
- **(QC4)** Was the tool evaluated?
- **(QC5)** Are the results reported clearly?. For each question, the studies were scored with 1.0 if the answer is "Yes", with 0.5 if the answer is "Partially" and 0.0 if the answer is "No". Adding the points of the five criteria, we excluded studies with a score equal or less than 2.5 (half of the total points) from the systematic mapping because it does not have satisfactory quality.

3 CONDUCTION

The systematic mapping was conducted for a period of three months (January/2014 to March/2014). A resume of the whole process can be seen in Figure 1. In total, 439 primary studies were identified. Of this amount, 161 were duplicated and excluded according to the Criterion 5. The total number of studies for the preliminary selection was reduced to 278 studies (see Search Results in Figure 1).

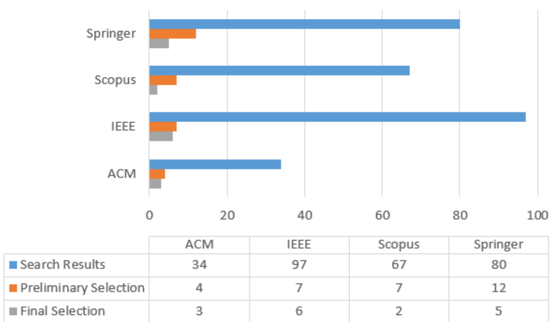

Figure 1: Papers in each search database.

Initially, 7 studies were excluded because we can not find them on the web with at least their abstract (criterion 5). Later, we excluded other 241 primary studies, based on the Criterion 1 and the Criterion 2. Thus, at the end of the preliminary selection, a total of 30 studies were included for the next stage (see Preliminary Selection in Figure 1).

At the beginning of the final selection one study was removed, due to unavailability for complete reading on the Web (Criterion 4). After this, we followed the procedure set out in the Planning phase. The full reading of the primary studies was made, with the application of the inclusion and exclusion criteria. Thirteen studies were excluded based on reading the full text, by non deal about formal semantics of UML models (Criterion 1) or do not present the formalization through the use of temporal logic (Criterion 2). Summing up at the end of this stage we included a total of 16 studies (see Final Selection in Figure 1 and Table 1).

Table 1: Selected Studies.

ID	Paper	year
1	(Baresi et al., 2012)	2012
2	(Mayerhofer et al., 2012)	2012
3	(Bauer and Jurjens, 2010)	2010
4	(Diethers and Huhn, 2004)	2004
5	(Dong et al., 2001)	2001
6	(Forster et al., 2007)	2007
7	(Graw et al., 2000)	2000
8	(Grobelna et al., 2010)	2010
9	(Kaliappan and Konig, 2012)	2012
10	(Kaliappan et al., 2010)	2010
11	(Konrad et al., 2004)	2004
12	(Lavazza et al., 2001)	2001
13	(Motta, 2012)	2012
14	(Rossi et al., 2004)	2004
15	(Zhang et al., 2009)	2009
16	(Zhao et al., 2005)	2005

Of this total of 16 primary studies, after apply the quality criteria, 3 of them were excluded for having obtained final score equal or less than 2.5, namely: (Bauer and Jurjens, 2010) (ID 3), (Dong et al., 2001) (ID 5), and (Zhang et al., 2009) (ID 15). The notes of these excluded studies were 2.5, 2.0 and 2.5, respectively. Table 2 presents the score for the quality of all 16 primary studies with some data about them.

Therefore, after applying the quality criteria in selected studies of this stage, a total of 13 studies were reviewed for data extraction. The Table 3 shows the 13 studies and their score in each Quality Criterion.

By using the selected studies were possible to extract some information. The distribution of the search databases for the 13 selected primary studies

Table 2: Studies after Final Selection.

ID	Criteria	Quality	Excl. Quality
1	I1, I2	3,5	-
2	I1, I2	4,0	-
3	I1, I2	2,5	X
4	I1, I2	3,5	-
5	I1, I2	2,0	X
6	I1, I2	3,5	-
7	I1, I2	3,0	-
8	I1, I2	3,0	-
9	I1, I2	4,0	-
10	I1, I2	3,0	-
11	I1, I2	4,0	-
12	I1, I2	3,5	-
13	I1, I2	3,0	-
14	I1, I2	3,0	-
15	I1, I2	2,5	X
16	I1, I2	4,0	-

Table 3: Quality Score of each study.

ID	QC1	QC2	QC3	QC4	QC5	Total
1	1.0	1.0	0.5	0.0	1.0	3.5
2	1.0	0.5	1.0	0.5	1.0	4.0
4	1.0	0.5	1.0	0.5	0.5	3.5
6	0.5	1.0	1.0	0.5	0.5	3.5
7	1.0	1.0	0.0	0.0	1.0	3.0
8	1.0	0.5	0.5	0.0	1.0	3.0
9	1.0	1.0	1.0	0.5	0.5	4.0
10	1.0	1.0	0.0	0.0	1.0	3.0
11	1.0	0.5	1.0	0.5	1.0	4.0
12	1.0	1.0	0.5	0.0	1.0	3.5
13	1.0	0.5	1.0	0.0	0.5	3.0
14	1.0	1.0	0.0	0.0	1.0	3.0
16	1.0	0.5	1.0	0.5	1.0	4.0

is: ACM with 23% of the total (3 studies); IEEE Xplore with 31% (4 studies); Scopus with 8% (1 study) and Springer with 38% (5 studies). The distribution among types of publication is: book chapters (LNCS) correspond to 38% of the total (5 studies); both journals and conferences correspond to 31% each (4 studies each one). The Table 4 summarizes these informations.

Finally, the Figure 2 shows the distribution of selected primary studies over the years, varying between 2000 and 2012, with the majority of studies in recent years.

4 DISCUSSION

The selected primary studies deal with the formal semantics of UML models, making use of concepts of temporal logic. The following are present some information about these studies and helps to respond **RQ1**.

Table 4: Database and Publication info.

ID	Search DB	Publication Type
1	ACM	Journal
2	Springer	Chapter
4	Springer	Conference
6	IEEE	Conference
7	IEEE	Conference
8	IEEE	Conference
9	Springer	Chapter
10	Springer	Chapter
11	IEEE	Journal
12	ACM	Journal
13	ACM	Conference
14	Springer	Journal
16	Scopus	Chapter

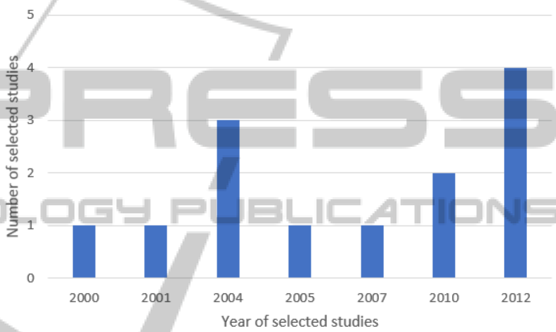

Figure 2: Distribution over the years.

The study (Baresi et al., 2012) presents the MADES UML, a subset of UML comprising some of the major UML diagrams, such as: Class Diagram, Sequence Diagram, State Diagram and Interaction Diagram. The MADES UML uses TRIO temporal formalism and the model checker Zot to analise its models. The paper cites a prototype tool that translates UML elements to the input recognized by Zot. As example, a car collision avoidance system is used, showing their respective diagrams, properties the system should satisfy and the counterexample (if available).

The authors in (Mayerhofer et al., 2012) show how they created an extension of the fUML and both transformations (model to formal and formal to model) through the use of a dedicate trace. Despite being interesting, the process presented in their paper can be applied only with Statecharts.

In (Diethers and Huhn, 2004) the authors present a tool to be used in the formal verification of UML models through State Diagrams. This tool uses the model checker UPPAAL as a base. The study explores just some concepts of temporal logic, quoting only that it is the same logic used by UPPAAL. The presented results are few due to scalability issues with State Diagrams, mostly with non-deterministic models.

The study (Forster et al., 2007) presents a formalization of the Activity Diagram semantics based on stereotypes of PPSL (Process Pattern Specification Language). To execute the semantics of the business processes, it is used the framework Dynamic Meta Modeling (DMM).Within the DMM approach, the authors chose to produce the results as a labeled system, making it possible visualize the defects found. Finally, formal verification is performed by the model checker NuSMV.

This other study (Graw et al., 2000) uses temporal logic based on cTLA (compositional Temporal Logic of Actions semantics), which is adequately explained. It integrates the Class, Sequence, State and Activity Diagrams. Although the authors present an extensive section on formal proofs, they are strictly conceptual. No tools are cited in the study, but the authors indicate the UPPAAL tool for future works.

In (Grobelna et al., 2010) the authors present a transformation of Activity Diagrams (AD) to Petri nets. The AD are presented as well the equivalent Petri nets and a table of inputs and outputs of the system. The temporal logic used by the authors to analyze this problem and create the properties is the LTL, however without specifying which type of LTL. Despite showing the input code of NuSMV, the authors do not explain how the two transformations were made (AD to Petri nets and then to the input of NuSMV).

The authors in (Kaliappan and Konig, 2012) explain the developed tool for converting Activity Diagrams for temporal logic cTLA. The basic concepts about cTLA are explained, with an example showing the difference between TLA and cTLA. It is detailed how was developed the transformation process created, showing the process of defining the semantics used, through a simple example of communication between processes. The formal verification step, which is the next step, is not discussed, only briefly mentioned how it works.

The study (Kaliappan et al., 2010) is prior to the above, showing the theoretical conversion between Activity Diagrams and cTLA temporal logic. The authors detail how the conversion is performed, showing the protocol established for this purpose. This protocol shows the use of Sequence Diagrams for modeling communication and Activity Diagrams for modeling the expected behavior, but only the Activity Diagrams are converted to cTLA.

In (Konrad et al., 2004) is presented a case study to illustrate the use of the approach proposed by the authors. The approach assigns a semantics to Class and State Diagrams, through a temporal formalism developed by the authors. This approach translates these diagrams into a formal specification that can be analyzed using SPIN. Finally, the authors show how the formal elements are equivalent to Class Diagrams before they are analyzed by SPIN.

The authors in (Lavazza et al., 2001) present a technique for formalize the semantics of Class and States Diagrams. For this, they use the TRIO formalism and developed a tool that automates the translation process of models into the TRIO axioms. To analyze these axioms, the authors used a tool called TRIOMatic and the counter-examples generated are converted into OCL (Object Constraint Language) and displayed on modeling environment. It is not specified if this transformation for OCL is included in the tool created by the authors.

In (Motta, 2012) the author presents a support tool developed for MADES UML. Although still call it a prototype, it is best detailed when compared with other studies on MADES UML. It is displayed the workflow of the tool, going through the modeling process with UML to the transformation step. After that, the transformed model is used as input by Zot tool.

(Rossi et al., 2004) present in the study an approach to formalize the semantics of States Diagrams, through a temporal logic developed by them, called LNint-e. The study explains in detail the formalism of LNint-e, providing examples to assist in understanding. Unfortunately, the authors do not indicate how the desired properties can be analyzed, if through another tool or using the LNint-e itself.

In (Zhao et al., 2005) is present a combination of techniques for models transformation with tools to perform formal verification. It is used an extension of the State Diagrams (RT-UML - Real-Time UML) and, regarding the temporal formalism, an extension of the OCL (RT-OCL - Real-Time OCL). The transformation of the properties written in RT-OCL for axioms is done directly. But the same does not happen with the RT-UML, according to the authors. In this case, it is necessary use AsmL (Abstract State Machine Language). Once with AsmL and the RT-OCL axioms ready, a set of tools for AsmL models is used to validate the model.

5 MAIN FINDINGS

Many of the 278 papers found using the search string fit the research question, because they deal about the semantic formalization of the UML. However, when analyzing them about the use of temporal logic, we note that the "UML" union with "temporal logic" is a less comprehensive topic within this context. Based on this systematic mapping, it is noticed that there is a concern about creating more intuitive ways (with different levels of abstraction) to assign semantic to UML. This shows that the fact of UML is widely known and used in the industry does not go unnoticed by researchers working on formal verification.

About the **RQ2**, some of the studies are focused only on the use of State Diagrams (Diethers and Huhn, 2004; Rossi et al., 2004; Zhao et al., 2005; Mayerhofer et al., 2012). Others only in Activity Diagrams (Forster et al., 2007; Grobelna et al., 2010; Kaliappan et al., 2010; Kaliappan and Konig, 2012). The other studies in the use of more than one UML diagram (Graw et al., 2000; Lavazza et al., 2001; Konrad et al., 2004; Baresi et al., 2012; Motta, 2012).

An important thing to be noted is that the last two cited studies (Baresi et al., 2012; Motta, 2012) deals with the MADES UML, a subset of the UML which attempts to assign semantics to a greater variety of diagrams. It is believed that by increasing the number of formalized UML diagrams, it creates a list of options for the software developer to work on different types of fields. This may contribute to greater adoption of formal verification in industrial applications.

For **RQ3**, the existence of a tool to support this process of semantically formalize the UML, despite (Graw et al., 2000) presents an approach that encompasses the largest number of UML diagrams, no tool is cited that supports its use. In fact, only the studies of (Diethers and Huhn, 2004; Kaliappan and Konig, 2012; Konrad et al., 2004; Lavazza et al., 2001; Zhao et al., 2005) and the ones related to MADES UML (in particular the work of (Motta, 2012)) feature a support tool that assist the developer to use their approaches.

Thus, despite efforts to develop new techniques to formalize the semantics of UML, the fact of having few tools hinders the practical application of the approaches, especially in industry. Moreover, these tools require a high level of knowledge outside the domain of the user (usually, the software developer has no knowledge of formal verification and their topics), which is another factor that causes difficulties in the practical use of them.

Therefore, it is necessary to develop techniques that allow the use of simple, functional tools for developers. Thus, formal verification could be more easily adopted by industry.

Finally, about **RQ4**, only one study ((Mayerhofer et al., 2012)) present a how to show the formal results in a way that a non-expert user can understand. The lack of this in other papers may be a problem because it make difficult to use formal verification with a users that do not have knowlegde in this area. Furthermore, create a tool with a formal verification engine "transparent to user" became unfeasible or even one where user does not need to be a expert in formal verification to understand the results provided by a model checker.

6 THREATS TO VALIDITY

Missing Important Primary Studies: We conducted the mapping in several search engines, even though it is rather possible we missed some primary studies. We tried to mitigate this threat by selecting search engines which have been regarded as the most relevant scientific sources (Kitchenham et al., 2009).

Search String Problems: Another threat is related to the search string, since there may be other synonyms for "formal semantics" and with them the search string could collected more evidences. To try to avoid this, we wrote the search string with the most common synonyms for each word.

Search Engines Format: One difficulty in conducting this research was in relation to the search engines, as each one requires a different format for the search string. To avoid this problem, we adapted the search string following the default format of each search engine, but sometimes it appeared to be not the correct one when executed.

7 CONCLUSION

Research in the area of formal verification and UML semantics can result in significant advancement in development of models and therefore software that has a higher level of accuracy and lesser probability of error. To get an overall view of the current research in this area, we defined a few research questions and launched a systematic mapping study. To respond to our **RQ1**, we found 13 publications that possess maximum relevance to fulfill objectives of our study. The selected papers have appeared between 2000 and 2012. Possibly this result came about due to the fact that among the techniques found, we considered only those which have used temporal logic as the basis of formalization.

Regarding **RQ2**, the tendency observed was that in general only one or two UML diagrams has formalized its semantics and in most cases is the State Diagram or the Activity Diagram. A technique that formalized more UML diagrams can be found, but is not much common. Regardless of the number of diagrams formalized, there is a lack of support tools to assist the software developer in the stage of formal verification. Even the tools existing at the moment of this review do not have a practical use, which complicates its adoption in the industry. About **RQ3** the few tools deal in general with the process of translating UML models for the inputs of formal verification tools, ignoring the fact that the results should be presented in a more common language for software developers. Finally, about **RQ4** only one study shows how to put the formal results in a diagram to make it more readable for the user.

Therefore, this paper seeks to show such evidences so that future works in the area of formal verification and UML models have focus on both the formalization of UML semantics as creating tools that can be used by developers without much knowledge in formal semantics and temporal logic which deal with the process of transforming the models into formal inputs and of showing the formal results in non-formal environments.

ACKNOWLEDGEMENTS

The Brazilian funding agency CAPES (Grant. No.: DS-7902801/D) supports this research.

REFERENCES

Baresi, L., Morzenti, A., Motta, A., and Rossi, M. (2012). A logic-based semantics for the verification of multi-diagram UML models. *SIGSOFT Softw. Eng. Notes*, 37(4):1 – 8.

Bauer, A. and Jurjens, J. (2010). Runtime verification of cryptographic protocols. *Computers and Security*, 29(3):315 – 330.

Bouabana-Tebiel, T. (2009). Semantics of the interaction overview diagram. In *Proceedings of the 10th IEEE International Conference on Information Reuse & Integration (IRI'09)*, pages 278 – 283. IEEE Press.

Bowen, J. P. and Hinchey, M. G. (2006). Ten commandments of formal methods... ten years later. *IEEE Comput.*, 39(1):40 – 48.

Diethers, K. and Huhn, M. (2004). Vooduu: Verification of object-oriented designs using UPPAAL. In *Tools and Algorithms for the Construction and Analysis of Systems (TACAS'04)*, volume 2988 of *Lecture Notes in Computer Science*, pages 139 – 143. Springer.

Dong, W., Wang, J., Qi, X., and Qi, Z.-C. (2001). Model checking UML statecharts. In *8th Asia-Pacific Software Engineering Conference (APSEC)*, pages 363 – 370.

Dyba, T., Dingsoyr, T., and Hanssen, G. K. (2007). Applying systematic reviews to diverse study types: An experience report. In *Proceedings of the First International Symposium on Empirical Software Engineering and Measurement (ESEM'07)*, pages 225 – 234. IEEE Computer Society.

Eriksson, H. E., Penker, M., and Lyons, B. (2004). *UML 2 Toolkit*, volume 1 of *OMG Series*. Wiley Pub.

Forster, A., Engels, G., Schattkowsky, T., and Straeten, R. V. D. (2007). Verification of business process quality constraints based on visual process patterns. In *First Joint IEEE/IFIP Symposium on Theoretical Aspects of Software Engineering (TASE)*, pages 197 – 208.

Graw, G., Herrmann, P., and Krumm, H. (2000). Verification of UML-based real-time system designs by means of cTLA. In *3rd IEEE International Symposium on Object-Oriented Real-Time Distributed Computing (ISORC)*, pages 86 – 95.

Grobelna, I., Grobelny, M., and Adamski, M. (2010). Petri nets and activity diagrams in logic controller specification - transformation and verification. In *17th International Conference on Mixed Design of Integrated Circuits and Systems (MIXDES)*, pages 607 – 612.

Kaliappan, P. and Konig, H. (2012). On the formalization of UML activities for component-based protocol design specifications. In *SOFSEM 2012: Theory and Practice of Computer Science*, volume 7147 of *Lecture Notes in Computer Science*, pages 479 – 491.

Kaliappan, P., Konig, H., and Schmerl, S. (2010). Model-driven protocol design based on component oriented modeling. In *Formal Methods and Software Engineering*, volume 6447 of *Lecture Notes in Computer Science*, pages 613 – 629.

Kitchenham, B. (2004). Procedures for performing systematic reviews. Technical report tr/se-0401, Keele University and NICTA.

Kitchenham, B., Brereton, O. P., Budgen, D., Turner, M., Bailey, J., and Linkman, S. (2009). Systematic literature reviews in software engineering - a systematic literature review. *Information and Software Technology*, 51:7 – 15.

Konrad, S., Cheng, B. H. C., and Campbell, L. (2004). Object analysis patterns for embedded systems. *IEEE Transactions on Software Engineering*, 30(12):970 – 992.

Lavazza, L., Quaroni, G., and Venturelli, M. (2001). Combining UML and formal notations for modelling real-time systems. *SIGSOFT Softw. Eng. Notes*, 26(5):196 – 206.

Mayerhofer, T., Langer, P., and Kappel, G. (2012). A runtime model for fUML. In *Proceedings of the 7th Workshop on Models@run.time (MRT'12)*, pages 53 – 58.

Micskei, Z. and Waeselynck, H. (2011). The many meanings of UML 2 sequence diagrams: a survey. *Software and Systems Modeling*, 10:489 – 514.

Motta, A. (2012). Towards the verification of multi-diagram UML models. In *International Conference on Software Engineering (ICSE)*, pages 1531 – 1534.

Rossi, C., Enciso, M., and de Guzman, I. P. (2004). Formalization of UML state machines using temporal logic. *Software and Systems Modeling*, 3(1):31 – 54.

Snook, C. and Butler, M. (2006). UML-B: Formal modeling and desing aided by UML. *ACM Transactions on Software Engineering and Methodology*, 15:92 – 122.

Zhang, P., Duan, Z., and Tian, C. (2009). UML statecharts' PTL formal semantics. In *3rd International Symposium on Intelligent Information Technology Application (IITA)*, volume 1, pages 381 – 385.

Zhao, Y., Kardos, M., Oberthur, S., and Rammig, F. J. (2005). *Comprehensive verification framework for dependability of self-optimizing systems*, volume 3707. LNCS.