

# Extensions of UML to Model Aspect-oriented Software Systems

Milena Guessi, Lucas Bueno Ruas Oliveira, and Elisa Yumi Nakagawa

Dept. of Computer Systems

University of São Paulo - USP

PO Box 668, 13560-970, São Carlos, SP, Brazil

*{milena, buenolro, elisa}@icmc.usp.br*

## Abstract

Aspect-Oriented Programming (AOP) has arisen as a new technology to support a better SoC (Separation of Concerns), intending to contribute to the development of reusable, maintainable, and evolvable software systems. Aspects have been also explored in the early life cycle phases, aiming at contributing to a more adequate development of aspect-oriented software systems. In this perspective, in order to better represent these systems, a diversity of extensions of the UML (Unified Modeling Language) has been proposed; however, there is a lack of a complete panorama that identify all these extensions. This paper presents an overview about which are possibly all extensions of the UML to represent aspect-oriented software systems. For this, we have used a technique proposed by Evidence-Based Software Engineering (EBSE): the Systematic Mapping. As achieved results, we can observe a diversity of work; however, there is not a consensus about which are the more adequate or more used extensions. Based on this overview, interesting and important perspectives for future research can also be found.

**Keywords:** UML, Aspect-orientation, Systematic Mapping.

## 1. INTRODUCTION

Aspect-Oriented Programming (AOP) has arisen as an approach that supports a better SoC (Separation of Concerns) and more adequately reflects the way developers think about the system [14]. Essentially, AOP introduces a unit of modular implementation – the aspect – which has been typically used to encapsulate crosscutting concerns in software systems (i.e., concerns that are spread across or tangled with other concerns). Modularity, maintainability, and facility to write software can be achieved with AOP [17]. Aspects have also been explored in the early life cycle phases.

According to Baniassad et al. [2], aspects in early phases support easier identification and analysis of aspects during later activities and, as a consequence, a better SoC in the aspect-oriented software systems, i.e., systems that contain aspects. In order to adequately model these systems, it is observed a diversity of initiatives [5, 19]. In particular, UML (Unified Modelling Language) [20] has been investigated as a language to represent aspect-oriented software systems, since it is already widely used as in the academy as in the industry.

In another perspective, Evidence-Based Software Engineering (EBSE) [9] has recently arisen, inspired in the medicine area, and has given considerable contribution to the Software Engineering area. EBSE provides means by which the best current research evidence can be coupled with practical experience and human values into the decision making process regarding software development and maintenance [15]. Thus, EBSE can contribute to software engineering researchers and professionals to make decisions concerning the adoption of new technologies, including processes, techniques, methods, and so on. In this context, systematic mapping is a technique that provides a systematic way in order to comprehend a topic

of research [4, 22]. Considering its relevance, systematic mapping for different domains can be found and, specifically for Software Engineering area, mapping for software testing [1] and for requirement specification [6] are two examples. In this perspective, the conduction of systematic mapping in order to identify all work that propose UML extensions to represent aspect-oriented systems seems also to be interesting.

The main objective of this paper is then to present a panorama containing possibly all research work that have investigated and/or extended UML to represent aspect-oriented software systems. For this, we have adopted and conducted a systematic mapping to investigate this topic of research. As main achieved results, we have observed that there are many research groups publishing different initiatives to the aspect-oriented modeling using UML. However, there is not a consensus about a unified solution for that modeling. It is worth highlighting that this work corresponds to a reconduction of the systematic mapping previously presented in [11], containing therefore updated results.

This paper is organized as follows. In Section 2 we present a brief overview on aspect-orientation, UML, and systematic mapping technique. In Section 3 we present the conducted systematic mapping. In Section 4 we discuss about achieved results. Finally, in Section 5 we summarize our contributions and discuss perspectives for further work.

## 2. BACKGROUND AND RELATED WORK

In the context of Aspect-Oriented Software Development (AOSD), identification of aspects during early phases of the software life cycle — for instance, requirement specification and architecture design — has been a recurrent issue in the literature. According to Baniassad [2], aspects in early phases support an easier identification and analysis of aspects during later activities, such as design and coding phases. In the last years, conferences, such as Aspect-Oriented Software Development (AOSD) Conference, Early Aspects workshops series and Workshop on Aspects in Architectural Description, have broadly discussed the use of aspects in these early phases. In order to deal with aspects in early phases related to modeling of aspect-oriented software systems, there is a diversity of initiatives [18, 21, 23]. In special, extensions to the UML can be also found [7, 12].

Currently, UML has been widely used in order to mainly facilitate development of object-oriented software systems. This standardized general-purpose modeling language includes a set of graphical notation techniques (also UML techniques or UML diagrams) to create visual models of software-intensive systems [20]. Its newest version (version 2.2) presents 14 techniques (such as Class Diagram, Sequence Diagram, and Component Diagram) divided into two categories (structural and behavioral) [20]. The diversity of UML techniques has therefore been basis for modeling of software systems with diverse characteristics, including aspect-oriented software systems.

In another perspective, it is noticed that when a research area is mature, there is almost always an increase in the number of reports and results made available. During the study of a new knowledge area, researchers usually conduct a bibliographical review (almost always an informal review) to identify publications related to a specific subject. For instance, Khan [13] conducted recently a survey on UML extensions to design aspect-oriented systems. Their study considered as relevant only eight works. The selection criteria used to select these studies was based mainly on previous background of the authors on the subject and was restricted to well-known events or those specific of aspect-orientation. The authors present an evaluation of each work considering, for instance, its coverage of the main constructs of aspect-oriented programming languages, which were these constructs, the UML technique used and if the work presents a case study. However, this kind of review do not use a systematic approach and do not offer any support to avoid bias during the selection of the publications that will be analyzed. Thus, it is important to have mechanisms to summarize and provide an overview about an area or topic of interest [22]. For this, EBSE has investigated and proposed the use of systematic mapping technique [4, 22]. In this context, an individual evidence (for instance, a case study or an experimental study divulged in a publication/paper) which contributes to a systematic mapping is called *primary study*, while the result of a systematic mapping is a *secondary study*. Systematic mapping aims at providing an overview of a research area to assess the quantity and type of primary studies existing on a topic of interest [22]. In short, systematic mapping is conducted by planning, conduction of search and screening of primary studies using inclusion and exclusion criteria [4]. Besides that, systematic mapping also conducts data extraction and analysis through the identification of categories and classification of the primary studies in these categories. As a result, this technique provides maps (for instance, in table or graphical format) containing condensed information about a research area.

### 3. SYSTEMATIC MAPPING APPLICATION

Our systematic mapping aims at identifying all primary studies that propose extension to the UML in order to represent aspect-oriented software systems. This systematic mapping considered studies published until February 5, 2011 and involved three people (a researcher in software engineering, a specialist in systematic mapping, and an undergraduate student). In order to conduct our systematic mapping, we have used the process presented in Figure 1. In short, it is composed by four steps: (i) systematic mapping planning; (ii) conduction of the search; (iii) selection of the primary studies; and (iv) analysis, classification, and map building. These steps are explained in more details during presentation of our systematic mapping. Following, we detail each step:

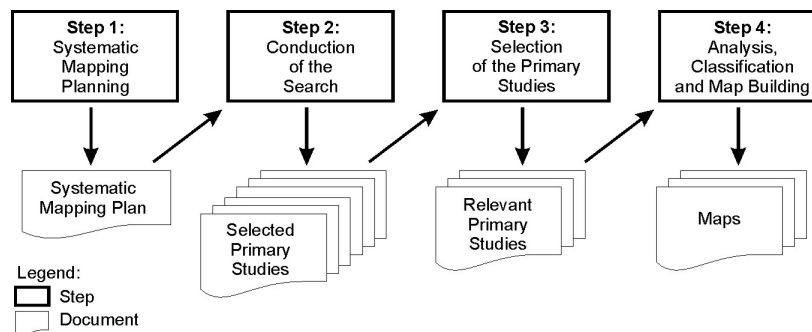


Figure 1: Systematic mapping process (Adapted from [22])

#### 3.1 Step 1: Systematic Mapping Planning

In this step, it is established the systematic mapping plan that will be used as basis to conduct the systematic mapping. This plan consists in the formulation of the research questions, selection of the sources of primary studies, and establishment of the inclusion and exclusion criteria.

- (i) **Research Questions (RQ):** These questions are structured corresponding to the objective that is intended with the systematic mapping that, in our case, it is the identification of a panorama involving UML extensions to model aspect-oriented software systems. The research questions of our systematic mapping are:

**RQ1:** Which UML techniques have been applied to model aspect-oriented software systems?

**RQ2:** Which are the terms (word or set of words) that are often associated with work that investigate UML extension to represent aspect-oriented software systems?

**RQ3:** Which vehicles have more published work on UML extensions to model aspect-oriented software systems?

**RQ4:** Which are the research groups that have investigated UML extensions to represent aspect-oriented software systems?

- (ii) **Selection of Sources:**

In addition to the research questions, we established which search sources (i.e., publication databases) would be used to find the primary studies. The criteria used to select the sources were: **content update** (publications are regularly updated); **availability** (full text of the papers are available); **quality of results** (accuracy of the results returned by the search); and **versatility export** (since much information are returned through the search, a mechanism to export the results is required). These criteria are also discussed in [8]. Thus, the selected databases to our systematic mapping are shown in Table 1. According to Brereton et al. [3, 10] and Dybå et al. [10], these databases are efficient to conduct systematic mapping in the context of software engineering. Furthermore, Scopus was added, since it is considered the largest database of abstracts and citations [16].

**Table 1:** Selected Databases

Database	Location
ACM Digital Library	www.portal.acm.org
IEEE Xplore	www.ieeexplore.ieee.org
ScienceDirect	www.sciencedirect.com
Scopus	www.scopus.com
ISI Web of Science	www.isiknowledge.com

- (iii) **Establishment of Selection Criteria:** Another important element of the systematic mapping planning is to define the Inclusion Criteria (IC) and Exclusion Criteria (EC). These criteria make possible to include primary studies that are relevant to answer the research questions and exclude studies that do not answer them. Thus, the inclusion criteria of our systematic mapping are shown in Table 2 and the exclusion criteria are shown in Table 3.

**Table 2:** Inclusion criteria established for the systematic mapping

Id	Criteria
IC1	The primary study proposes UML extensions to represent aspect-oriented software systems
IC2	The primary study proposes the use of UML techniques to represent aspect-oriented software systems
IC3	The primary study proposes a tool to support UML techniques to represent aspect-oriented software systems

**Table 3:** Exclusion criteria established for the systematic mapping

Id	Criteria
EC1	The primary study does not address modeling of aspect-oriented software systems
EC2	The primary study is not related to UML
EC3	The primary study proposes a new language to model aspect-oriented software systems
EC4	The primary study does not present any abstract or it is not available for further reading
EC5	The primary study is written in a different language than English
EC6	The primary study presents an abstract and an introductory section that seem related to UML extensions to model aspect-oriented software systems; however, the rest of the text is not in fact related to
EC7	The primary study is directly related to another primary study of the same author
EC8	The primary study consists of a compilation of work, for instance, from a conference or workshop

### 3.2 Step 2: Conduction of the Search

In this step, the search by primary studies is conducted according to previously established plan. This search is done by looking for all primary studies that match with the search string in the search sources. This can be automatically conducted if these sources provide an efficient search engine. For this, we established the keywords and the search strings.

- (i) **Keywords:** To keep the scope of the systematic mapping, the keywords must be simple and well chosen. Thus, they must be simple enough to bring many results and also rigorous enough to cover only the desired primary studies. The keywords chosen to our systematic mapping are: UML, aspect based, aspect-oriented, and aspect-oriented modeling.
- (ii) **Search Strings:** Based on the keywords, the search string is built (through connections AND/OR) to incorporate the semantic expected to the search. The search string used in our systematic mapping was:

(UML AND ("aspect-based" OR "aspect oriented" OR "aspect-oriented" OR  
AOM OR "aspect oriented modeling" OR "aspect-oriented modeling"))

We have also built the strings and/or settings for each search engine, since each engine accepts a specific formatted. In Table 4, it is presented these strings and settings.

**Table 4:** Search Strings by Source

Source	Search String
ACM Digital Library	(Title:(uml AND (aspect-based OR aspect-oriented OR “aspect oriented” OR “aspect oriented modeling” OR aom OR “aspect-oriented modeling”)) OR Keywords:(uml AND (aspect-based OR aspect-oriented OR “aspect oriented” OR “aspect oriented modeling” OR aom OR “aspect-oriented modeling”)) OR Abstract:(uml AND (aspect-based OR aspect-oriented OR “aspect oriented” OR “aspect oriented modeling” OR aom OR “aspect-oriented modeling”)))
IEEE Xplore	((uml <and> (aspect-based <or> aspect-oriented <or> ‘aspect oriented’ <or> ‘aspect oriented modeling’ <or> aom <or> ‘aspect-oriented modeling’)) <in> ti) <or> ((uml <and>(aspect-based <or> aspect-oriented <or> ‘aspect oriented’ <or> ‘aspect oriented modeling’ <or> aom <or> ‘aspect-oriented modeling’)) <in>ab)
ScienceDirect	TITLE-ABSTR-KEY((uml AND (aspect-based OR aspect-oriented OR aspect oriented OR aspect oriented modeling OR aom OR aspect-oriented modeling)))
Scopus	TITLE-ABS-KEY(uml AND (aspect-based OR aspect-oriented OR “aspect oriented” OR “aspect oriented modeling” OR aom OR “aspect-oriented modeling”)) AND (LIMITTO(SUBJAREA, “COMP”) OR LIMIT-TO(SUBJAREA, “MULT”))
ISI Web of Science	TI=((uml AND (aspect-based OR aspect-oriented OR “aspect oriented” OR “aspect oriented modeling” OR aom OR “aspect-oriented modeling”)) OR TS = ((uml AND (aspect-based OR aspect-oriented OR “aspect oriented” OR “aspect oriented modeling” OR aom OR “aspect-oriented modeling”)))

As result of this step, we obtained a total of 593 primary studies: 118 from ACM Digital Library, 69 from IEEE Xplore, 67 from ScienceDirect, 119 from ISI Web of Science and 220 from Scopus. Figure 2 shows the distribution of these studies into the web search engines selected.

### 3.3 Step 3: Selection of the Primary Studies

In this step, the selection criteria (i.e., inclusion and exclusion criteria) are applied to select the relevant primary studies. To support the organization and manipulation of the primary studies, we used JabRef<sup>1</sup>, an open source reference manager system. It made possible to store information on the primary studies (including, for instance, title, authors, and abstract), as well as the inclusion/exclusion criteria applied to select each primary study.

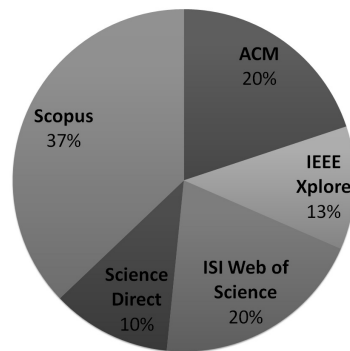
From 593 primary studies, 268 were repeated. Thus, we found 325 different studies. Applying our selection criteria, we included then 84 primary studies<sup>2</sup>. Table 5 summarizes the total of primary studies obtained in each database, the number of studies included (or selected), the precision rate<sup>3</sup>, the rate index<sup>4</sup>, and the search date. For instance, from ACM Digital Library, we obtained a total of 118 studies; using the selection criteria, 50 studies were selected; precision rate is 42% (i.e.,

<sup>1</sup><http://jabref.sourceforge.net/>

<sup>2</sup>The complete reference of each primary study included can be found in the Appendix A.

<sup>3</sup>Ratio between the total of included studies of a database and the total of primary studies obtained by this database.

<sup>4</sup>Ratio between the included studies of a database and the total of primary studies included in the systematic review.



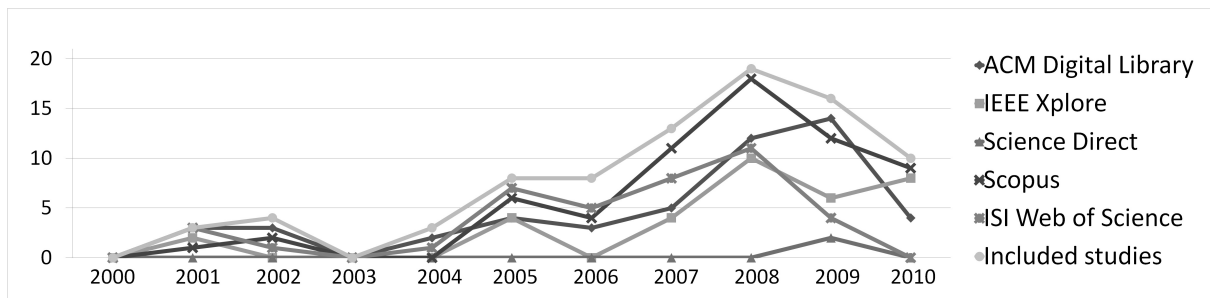
**Figure 2:** Distribution of the primary studies obtained

50/118) and rate index is 60% (i.e., 50/84). It is important to observe that Scopus was the most efficient source, since 75% of all included papers were obtained in this source. Otherwise, ScienceDirect contributed with only 2% of primary studies which were also indexed by other sources. Therefore, we can point out that if there is an interest in searching work that involve the research topic of this work (i.e., UML extensions to model aspect-oriented software systems), searches using Scopus and ACM Digital Library could cover almost all work of this topic. In other words, Scopus obtained 75% and ACM Digital Library obtained 60%. Since Scopus indexes also studies of other databases, such as IEEE Xplore and ACM Digital Library, if the search was concentrated only in these two database (Scopus and ACM Digital Library), we would find 94% of the primary studies that we have selected.

**Table 5:** Search sources, obtained and included primary studies

Database	Obtained	Included	Precision Rate	Rate Index	Date
ACM Digital Library	118	50	42%	60%	Feb 5,2011
IEEE Xplore	69	34	49%	40%	Feb 5,2011
ISI Web of Science	119	40	34%	48%	Feb 5,2011
ScienceDirect	67	2	3%	2%	Feb 5,2011
Scopus	220	63	29%	75%	Feb 5,2011

Figure 3 presents the amount of primary studies included as relevant and separated by years. It is observed a strong growth of publications in the last years, mainly in Scopus and ACM Digital Library. Other sources (IEEE Xplore and ISI Web of Science) have what might be called an expected growth, considering the other two most numerous sources (Scopus and ACM Digital Library).



**Figure 3:** Amount of included primary studies

### 3.4 Step 4: Analysis, Classification and Map Building

While in previous section we present a more quantitative analysis of our systematic mapping, in this section, we present the categorization of the primary studies. For this, two tasks were conducted: (i) search by keywords; and (ii) grouping of the primary studies into categories. Thus, primary studies were firstly carefully read (i.e., title, keywords, and abstract). Next task was the creation of categories. For this, research questions were considered. In the case of our systematic mapping, we were interested in identifying ways of applying UML techniques to model aspect-oriented systems. Thus, four categories were created:

**UML Diagrams to Represent Aspect-oriented Software Systems:** The keywords used in this category refer to the UML techniques (version 2.2): Use Case Diagram, State Machine Diagram, Timing Diagram, Interaction Overview Diagram, Communication Diagram, Sequence Diagram, Activity Diagram, Profile Diagram, Composite Structure Diagram, Package Diagram, Deployment Diagram, Component Diagram, Class Diagram, and Object Diagram. We are interested in knowing which UML techniques have been more investigated to model aspect-oriented software systems;

**Main Keywords:** Keywords of this category refer to the most relevant keywords obtained from the selected primary studies. To do this, we firstly identified all the keywords from studies included. Then, these keywords were classified according to their frequency of occurrence. Finally, we selected the keywords that seemed more relevant or referred to software engineering research areas;

**Main Events and Journals:** Keywords of this category refer to name of the events (such as conference and workshop) and journals in which the included primary studies were published; and

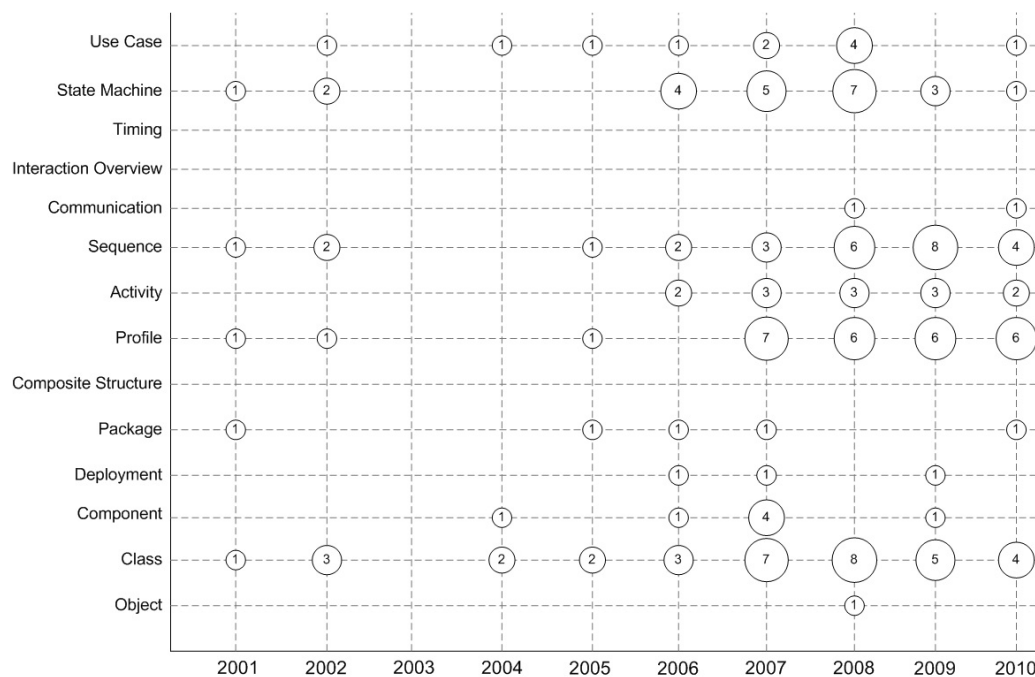
**Main Research Groups:** Keywords of this category refer to the research groups that have published work related to UML extension to model aspect-oriented software systems. For this, we firstly identified all research groups related to included primary studies identified in our mapping. Then, research groups that had two or more primary studies (that were included in our mapping) were chosen as keywords of this category.

Following, we classified the primary studies into keywords in the categories. It is observed that a study can be classified sometimes into more than one keyword per category. Moreover, for category “UML Diagrams to Represent Aspect-oriented Systems”, we read all relevant primary studies and separated them among the keywords. The same strategy was conducted to other three category: “Main Keywords”, “Main Events and Journal”, and “Main Research Groups”. As result, we build four maps, one for each category.

Figure 4 shows the map to the category “UML Diagrams to Represent Aspect-oriented Systems”. The reader can refer to Appendix C in order to find the complete list of primary studies classified in each category established for this map. It is observed that from first studies in 2001, there is an increasing interest in extending UML diagrams to model aspect-oriented software systems. It is important to notice that there is an overlap between Class Diagram and Profile Diagram<sup>5</sup>, since primary studies extend the Class Diagram using new stereotypes, such as <<aspect>> and <<crosscuts>>, that are specific to aspect-oriented systems. Among the behavioral diagrams, Sequence Diagram and State Machine Diagram have been more investigated to represent aspect-oriented systems. These diagrams make possible to describe new behaviors introduced by the aspects and represent more easily when and how these aspects will be executed. Otherwise, diagrams that are not so disseminated and used (for example, Timing Diagram, Interaction Overview Diagram, and Composite Structure Diagram) if compared with, for instance, Class Diagram, have not been considered to model aspect-oriented systems.

Figure 5 shows the map to the category “Main Keywords”. The keywords of this category are: Aspect Composition, Case Study, Component, Composition Mechanism, Framework, Join Point Model, Model-driven Architecture, Model-driven Development, Requirement, Reverse Engineering, Software Architecture, Tool Support, UML Based, UML Extension, UML Profile, and UML Standard. The reader can refer to Appendix C in order to find the complete list of primary studies

<sup>5</sup>Profile Diagram allows to define custom stereotypes, tag definitions, and constraints in a UML diagram.



**Figure 4:** Map related to category “UML Diagrams to represent aspect-oriented systems”

classified in each category established for this map. This map shows which are the keywords that are more directly related to work that extend UML to model aspect-oriented systems. Thus, for instance, the keywords “UML extension” and “UML Based” are two important keywords that could be used in the future searches for primary studies. Otherwise, only few primary studies on UML extension to aspect-orientation domain are related to “Requirements” and “Reverse Engineering”.

Figure 6<sup>6</sup> shows the map to the category “Main Events and Journals”. The keywords of this category are events and journals in which the 84 primary studies selected in our mapping were published. It is important to notice that the primary studies were published in a diversity of events and journal (51 different events and 7 journals). Furthermore, these studies are not only concentrate in well-known events or those specific of aspect-orientation, such as ICSE (ACM/IEEE International Conference on Software Engineering) and AOSD<sup>7</sup> (International Conference on Aspect-Oriented Software Development). Therefore, if the search space was limited to those more known events and journals, important and relevant primary studies could not be found. For instance, AOSD published three primary studies and ICSE published only one among our 84 studies. Thus, systematic mapping was a fundamental and important technique to conduct our search and achieve positive and more extensive result.

Figure 7 shows the map to the category “Main Research Groups”. For the sake of space, only research groups that published two or more studies were considered. Therefore, 19 research groups were considered in this map. For instance, research group of the Hebei University of Engineering (in China) published seven primary studies and each one of them deals with one UML diagram: Sequence Diagram, State Machine Diagram, Class Diagram, and in the design of connectors to represent aspect-oriented components. There is also an effort in applying these concepts in the framework of MDA (Model Driven Architecture). Meanwhile, other research groups in 17 universities published two or three studies. Furthermore, it is worth to highlighting that we identified other 59 different institutions that have published only one primary study. Therefore, we have observed that researches involving UML extension to aspect-orientation domain have been conducted by many different institutions.

<sup>6</sup>The meaning of the abbreviations can be found in the Appendix B.

<sup>7</sup><http://www.aosd.net/>



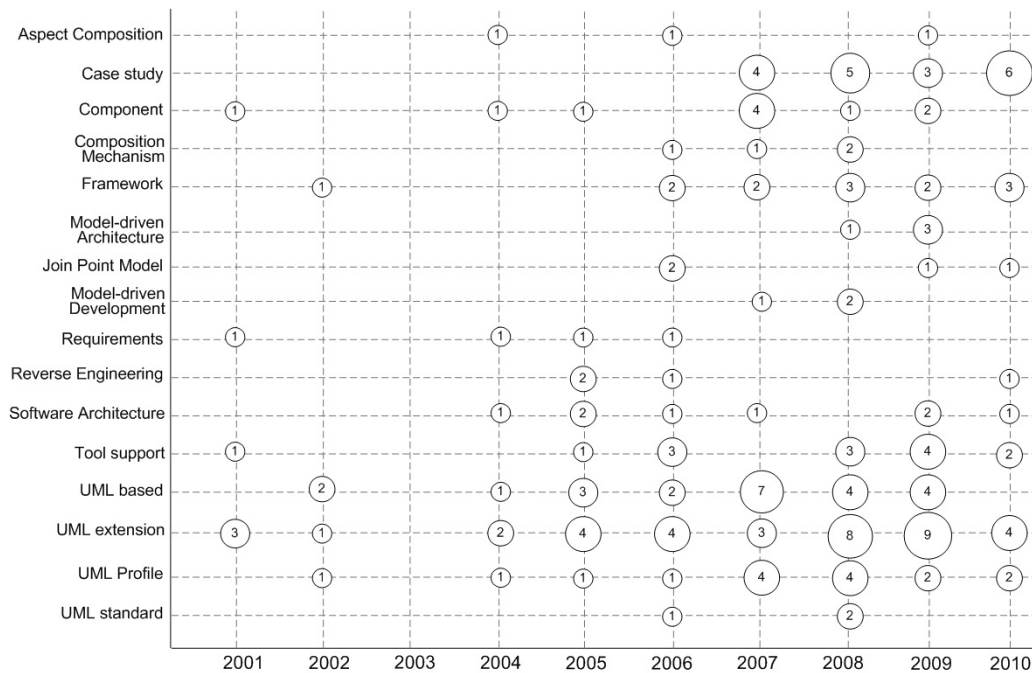


Figure 5: Map related to category "Main Keywords"

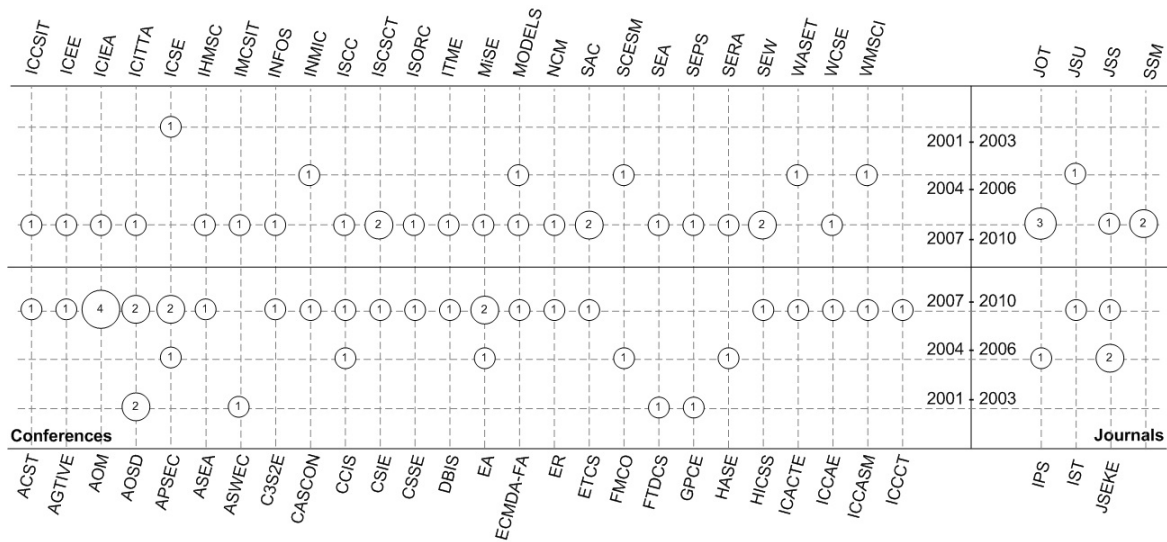
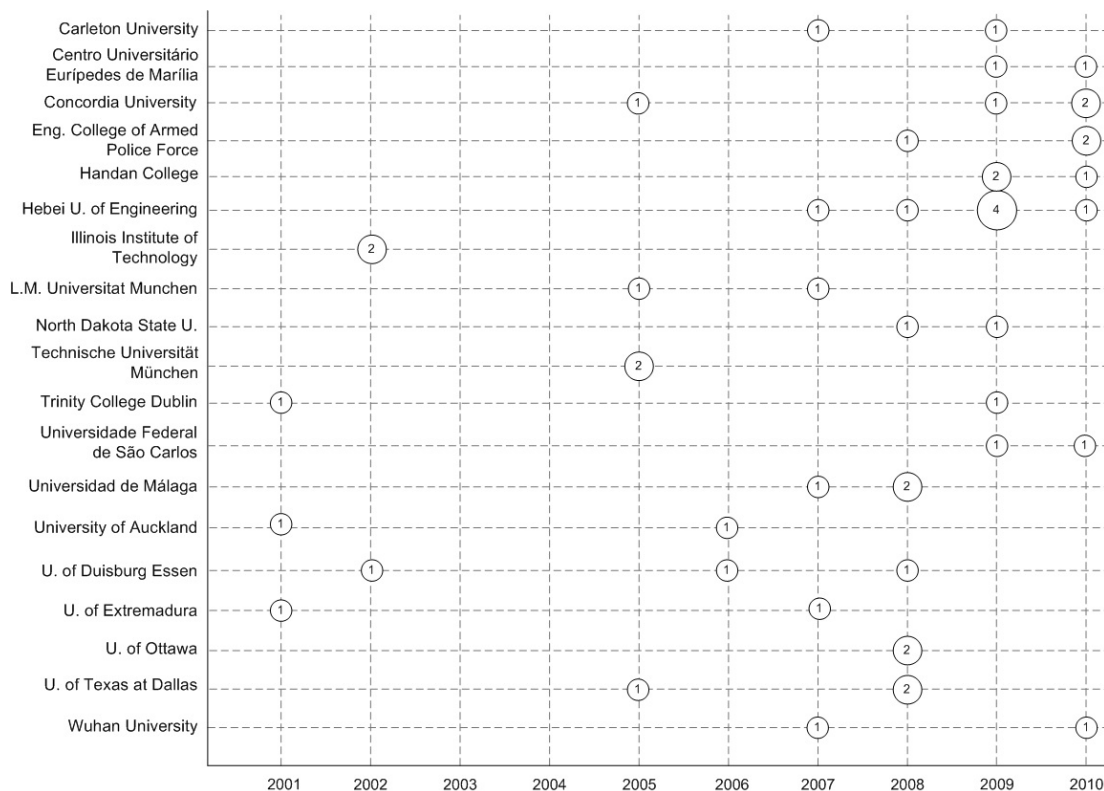


Figure 6: Map related to category "Main Events and Journals"

#### 4. DISCUSSION

In this section, we discuss some issues related to the conducted systematic mapping, what also includes lessons learned and limitations we have noticed.

Regarding research questions established for our systematic mapping, it is observed that all of them were successfully answered. This suggests that, the general knowledge about representing aspect-oriented systems using UML techniques has been mapped. Besides that, we believe that results presented in this work are representative of the whole universe of



**Figure 7:** Map related to category “Main Research Groups”

existing approaches to model aspect-oriented systems, since systematic mapping has provide mechanism to achieve it.

Considering knowledge arisen from this work, it is possible to identify interesting and important research lines that can be investigated in future work. For instance, the establishment of a unified UML notation to aspect-oriented modeling and extension of UML diagram that have not been explored in aspect-orientation domain, such as Timing Diagram.

Regarding limitation of this work, other categories could have been established and, as a consequence, related maps could have been designed. In another perspective, other research questions could be established, considering specific topics of research. Moreover, it is worth highlighting that systematic mapping conduction is not a trivial task because the amount of papers that need to be manipulated. Besides that, both paper reading and category analysis are manually conducted. Many of the studies provided none substantial information in the abstract or representative keywords to facilitate the analysis phase of the mapping. In those cases, it was also necessary to read the introduction and conclusion sections and, sometimes, the entire text of the primary study to determine the context and contributions of the study. Sometimes it was necessary to infer a conclusion to make some decisions. Besides that, during the conduction of our systematic mapping, more specifically after the primary studies had already been selected for our mapping, IEEE Xplore changed its search engine. Thus, the results presented here might be different from those obtained if a new search is conducted in that database.

## 5. CONCLUSIONS

The main contribution of this work was to present an overview about UML extensions to represent aspect-oriented software systems. To establish this overview, it was essential to systematically apply the set of steps provided by systematic mapping. As results, we found that there are several initiatives being proposed aiming at extending UML for the aspect-orientation approach. It is also important to notice that most studies were published in the last four years. In spite of that,

there is a lack of consensus about which extensions should be disseminated and more widely used. Thus, we believe that this overview can contribute to the aspect-oriented software development area, getting new lines of research that are open yet. Besides that, in the same perspective, other mappings must be conducted, what could contribute to the establishment of a unified solution to the aspect-oriented modeling.

**Acknowledgments** This work is supported by Brazilian funding agencies: FAPESP, CAPES, and CNPq.

## References

- [1] R. Afzal, W.; Torkar and R. Feldt, “A systematic mapping study on non-functional search-based software testing,” in *Proceedings of the 20<sup>th</sup> International Conference on Software Engineering and Knowledge Engineering (SEKE 2008)*, San Francisco, USA, 2008, pp. 1–6.
- [2] E. Baniassad, P. C. Clements, J. Araujo, A. Moreira, A. Rashid, and B. Tekinerdogan, “Discovering early aspects,” *IEEE Software*, vol. 23, no. 1, pp. 61–70, 2006.
- [3] P. Brereton, B. A. Kitchenham, D. Budgen, M. Turner, and M. Khalil, “Lessons from applying the systematic literature review process within the software engineering domain,” *Journal of Systems and Software*, vol. 80, no. 4, pp. 571–583, 2007.
- [4] M. B. P. Budgen, D.; Turner and B. Kitchenham, “Using mapping studies in software engineering,” in *Proceedings of the 20<sup>th</sup> Annual Meeting of the Psychology of Programming Interest Group (PPIG 2008)*, Lancaster, United Kingdom, 2008, pp. 195–204.
- [5] S. Clarke and E. Baniassad, *Aspect-Oriented Analysis and Design: The Theme Approach*, ser. Object Technology, 2005.
- [6] N. Condori-Fernandez, M. Daneva, K. Sikkil, R. Wieringa, O. Dieste, and O. Pastor, “A systematic mapping study on empirical evaluation of software requirements specifications techniques,” in *Proceedings of the 3<sup>rd</sup> International Symposium on Empirical Software Engineering and Measurement (ESEM 2009)*, Lake Buena Vista, USA, 2009, pp. 502–505.
- [7] W. L. L. X. Cui, Z. and D. Xu, “Modeling and integrating aspects with UML activity diagrams,” in *Proceedings of the 2009 ACM Symposium on Applied Computing (SAC 2009)*, New York, USA, 2009, pp. 403–437.
- [8] O. Dieste, A. Grimán, and N. Juristo, “Developing search strategies for detecting relevant experiments,” *Empirical Software Engineering*, vol. 14, no. 5, pp. 513–539, 2009.
- [9] T. Dybå, B. Kitchenham, and M. Jorgensen, “Evidence-based software engineering for practitioners,” *IEEE Software*, vol. 22, no. 1, pp. 58–65, 2005.
- [10] T. Dybå, T. Dingsoyr, and G. K. Hanssen, “Applying systematic reviews to diverse study types: An experience report,” in *Proceedings of the 1<sup>st</sup> International Symposium on Empirical Software Engineering and Measurement (ESEM 2007)*, Los Alamitos, USA, 2007, pp. 225–234.
- [11] M. Guessi, L. B. R. Oliveira, and E. Y. Nakagawa, “Modeling Aspect-oriented Software Systems Using UML: A Systematic Mapping,” in *Proceedings of the 36<sup>th</sup> Latin American Conference of Informatics (CLEI 2010)*, Asuncion, Paraguay, 2010, pp. 1–14.
- [12] J. Ivers, P. Clements, D. Garlan, R. Nord, B. Schmerl, and J. R. O. Silva, “Documenting component and connector views with UML 2.0,” Tech. Rep., 2004, cMU/SEI-2004-TR-008.
- [13] S. A. Khan and A. Nadeem, “UML Extensions for Modeling Aspect Oriented Software: A Survey,” in *Proceedings of the National Software Engineering Conference (NSEC 2010)*, Rawalpindi, Pakistan, 2010, pp. 1–6.

- [14] G. Kiczales, J. Irwin, J. Lamping, J. Loingtier, C. Lopes, C. Maeda, and A. Menhdhekar, "Aspect-oriented programming," in *Proceedings of the 11<sup>th</sup> European Conference on Object-Oriented Programming (ECOOP 1997)*, Jyväskylä, Finland, 1997, pp. 220–242.
- [15] B. Kitchenham, "Procedures for performing systematic reviews," Keele University and National ICT Australia Ltd, Tech. Rep. TR/SE-0401 and NICTA Technical Report 0400011T.1, 2004.
- [16] B. Kitchenham and S. Charters, "Guidelines for performing systematic literature reviews in software engineering," Keele University and Durham University Joint Report, Tech. Rep. EBSE 2007-001, 2007. [Online]. Available: <http://www.dur.ac.uk/ebse/resources/Systematic-reviews-5-8.pdf>
- [17] R. Laddad, "Aspect-oriented programming will improve quality," *IEEE Software*, vol. 20, no. 6, pp. 90–91, 2003.
- [18] Y. Lau, W. Zhao, X. Peng, Y. Chen, and Z. Jiang, "A unified formal model for supporting aspect-oriented dynamic software architecture," in *Proceedings of the International Conference on Convergence Information Technology (IC-CIT 2007)*, Los Alamitos, USA, 2007, pp. 450–455.
- [19] E. Navarro, P. Letelier, and I. Ramos, "Requirements and scenarios: Running aspect-oriented software architectures," in *Proceedings of the 6<sup>th</sup> Working IEEE/IFIP Conference on Software Architecture (WICSA 2007)*, Mumbai, India, 2007, p. 23.
- [20] OMG, "Unified modeling language version 2.2," [Online], 2009, Available:<http://www.omg.org/spec/UML/2.2/>.
- [21] V. Paula and T. Batista, "Revisiting a formal framework for modeling aspects in the design phase," in *Proceedings of the Early Aspects at ICSE: Workshops in Aspect-Oriented Requirements Engineering and Architecture Design*, Washington, DC, USA, 2007, p. 6.
- [22] K. Petersen, R. Feldt, M. Shahid, and M. Mattsson, "Systematic mapping studies in software engineering," in *Proceedings of the 5<sup>th</sup> IEEE Workshop on Engineering of Autonomic and Autonomous Systems (EASE 2008)*, Belfast, Northern Ireland, 2008, pp. 1–10.
- [23] M. Pinto, N. Gamez, and L. Fuentes, "Towards the architectural definition of the health watcher system with AO-ADL," in *Proceedings of the Early Aspects at ICSE: Workshops in Aspect-Oriented Requirements Engineering and Architecture Design*, Washington, DC, USA, 2007, p. 5.

## Appendix A

Table 6 presents the complete reference of each primary study included in the systematic mapping.

**Table 6:** Complete reference of primary studies included

Id	Reference
S1	Abeywickrama, D. & Ramakrishnan, S., "Towards engineering models of aspectual pervasive software services", In <i>Proceedings of the 3rd ACM workshop on Software engineering for pervasive services (SEPS 2008)</i> , 2008, pp. 3–8, Sorrento, Italy.
S2	Alam, F. E., Evermann, J. & Fiech, A., "Modeling for dynamic aspect-oriented development", In <i>Proceedings of the 2nd Canadian Conference on Computer Science and Software Engineering (C3S2E 2009)</i> , 2009, pp. 143–147, Montreal, Canada.
S3	Albunni, N. & Petridis, M., "Using UML for Modelling Cross-Cutting Concerns in Aspect Oriented Software Engineering". In <i>Proceedings of the 3rd International Conference on Information and Communication Technologies: From Theory to Applications (ICTTA 2008)</i> , 2008, pp. 1–6, Damascus, Syria.
S4	Aldawud, O. M., "Aspect-oriented modeling with UML", PhD thesis, Illinois Institute of Technology, 2002, Chicago.
S5	Anthonyamy, P. & Somé, S., "Aspect-oriented use case modeling for software product lines", In <i>Proceedings of the 2008 AOSD workshop on Early aspects (EA-AOSD 2008)</i> , 2008, pp. 1–8, Brussels, Belgium.
S6	Ballal, R. & Hoffman, M. A., "Extending UML for Aspect Oriented Software Modeling", In <i>Proceedings of the WRI World Congress on Computer Science and Information Engineering (CSIE 2009)</i> , 2009, volume 7, pp. 488–492, Cayro, Egypt.
S7	Bustos, A. & Eterovic, Y., "Modeling aspects with UML's class, sequence and state diagrams in an industrial setting", In <i>Proceedings of the 11th IASTED International Conference on Software Engineering and Applications (SEA 2007)</i> , 2007, pp. 403–410, Anaheim, USA.
S8	Carton, A., Driver, C., Jackson, A. & Clarke, S., "Model-Driven Theme/UML", In <i>Transactions on Aspect-Oriented Software Development VI</i> , 2009, pp. 238–266 (LNCS v.5560), Berlin, Heidelberg.
S9	Clarke, S. & Baniassad, E., "Aspect-Oriented Analysis and Design", Addison-Wesley Professional, 2005.
S10	Clarke, S. & Walker, R., "Composition patterns: An approach to designing reusable aspects", In <i>Proceedings of the 23rd International Conference on Software Engineering (ICSE 2001)</i> , 2001, pp. 5–14, Toronto, Canada.
S11	Clemente, P. J., Hernandez, J. & Sanchez, F., "Driving Component Composition from Early Stages Using Aspect-Oriented Techniques", In <i>Proceedings of the 40th Annual Hawaii International Conference on System Sciences (HICSS 2007)</i> , 2007, pp. 257a, Big Island, USA.
S12	Cui, Z., Wang, L., Li, X. & Xu, D., "Modeling and integrating aspects with UML activity diagrams", In <i>Proceedings of the 24th ACM symposium on Applied Computing (SAC 2009)</i> , 2009, pp. 430–437, Honolulu, USA.
S13	Dai, L., Cooper, K. & Eric Wong, W., "Modeling and analysis of performance aspects for software architecture: A UML-based approach", <i>International Journal of Software Engineering and Knowledge Engineering (IJSEKE)</i> , 2006, 16(3):347–378.
S14	Debnath, N., Baigoria, L., Riesco, D. & Montejano, G., "Metrics applied to Aspect Oriented Design using UML profiles", In <i>IEEE Symposium on Computers and Communications (ISCC 2008)</i> , 2008, 654–657, Marrakech, Morocco.
S15	Deng, Z. & Xu, X., "Aspect-oriented Technology-Based Production System Simulation Modeling Method", In <i>Proceedings of the 2010 International Conference on E-Business and E-Government (ICEE 2010)</i> , 2010.
S16	Deubler, M., Meisinger, M., Rittmann, S. & Kruger, I., "Modeling crosscutting services with UML sequence diagrams", In <i>Proceedings of the 8th International Conference on Model Driven Engineering Languages and Systems (MODELS 2005)</i> , 2005, pp. 522–536 (LNCS v.3713), Montego Bay, Jamaica.
S17	Domokos, P. & Majzik, I., "Design and Analysis of Fault Tolerant Architectures by Model Weaving", In <i>Proceedings of the 9th IEEE International Symposium on High-Assurance Systems Engineering (HASE 2005)</i> , 2005, pp. 15–24, Heidelberg, Germany.
S18	El Maghawry, N. & Dawood, A., "Aspect oriented GoF design patterns", In <i>Proceedings of the 7th International Conference on Informatics and Systems (INFOS 2010)</i> , 2010, pp. 1–7, Cairo, Egypt.
S19	Elrad, T., Aldawud, O. & Bader, A., "Aspect-oriented modeling: Bridging the gap between implementation and design", In <i>Proceedings of the 1st International Conference on Generative Programming and Component Engineering (GPCE 2002)</i> , 2002, pp. 189–201 (LNCS v.2487).
S20	Evermann, J., "A meta-level specification and profile for AspectJ in UML", <i>Journal of Object Technology (JOT)</i> , 2007, 6(7):27–49.
S21	Filman, R., Elrad, T., Clarke, S. & Akşit, M. <i>Aspect-oriented software development</i> , Addison-Wesley Professional, 2004.
S22	Foumani, A. A. & Constantinides, C., "Aspect-oriented reverse engineering", In <i>Proceedings of the 9th World Multi-Conference on Systemics, Cybernetics and Informatics (WMSCI 2005)</i> , 2005, volume 1, pp. 253–260, Int. Inst. of Informatics & Systemics.
S23	France, R., Ray, I., Georg, G. & Ghosh, S., "Aspect-oriented approach to early design modelling", <i>IEE Proceedings Software (IPE)</i> , 151(4):173–185.
S24	Fuentes, L., Manrique, J. & Sánchez, P., "Execution and simulation of (profiled) UML models using Pópulo", In <i>Workshop on Modeling in Software Engineering (MiSE 2008)</i> , <i>30th International Conference on Software Engineering (ICSE 2008)</i> , 2008, pp. 75–81, Leipzig, Germany.
S25	Fuentes, L. & Sánchez, P., "Designing and weaving Aspect-Oriented executable UML models", <i>Journal of Object Technology (JOT)</i> , 6(7):109–136.

Continued...

Table 6 – Continued

Id	Reference
S26	wei Ge, J., Xiao, J., qiu Fang, Y. & dan Wang, G., “Incorporating aspects into UML state machine”, In <i>Proceedings of the 3rd International Conference on Advanced Computer Theory and Engineering (ICACTE 2010)</i> , volume 2, pp. 505–508, Chengdu, China.
S27	Gowthaman, K., Mustafa, K. & Khan, R. A., “Reengineering Legacy Source Code to Model Driven Architecture”, In <i>Proceedings of the 4th Annual ACIS International Conference on Computer and Information Science (CCIS 2005)</i> , 2005, pp. 262–267, Jeju Isl, South Korea.
S28	Grundy, J. & Patel, R., “Developing Software Components with the UML, Enterprise Java Beans and Aspects”, In <i>Proceedings of the 13th Australian Conference on Software Engineering (ASWEC 2001)</i> , 2001, pp. 127–136, Canberra, Australia.
S29	Grønmo, R. & Møller-Pedersen, B., “Aspect Diagrams for UML Activity Models”, In <i>Proceedings of the 3rd International Symposium on Applications of Graph Transformations with Industrial Relevance (AGTIVE 2007)</i> , 2007, pp. 329–344 (LNCS v.5088), Kassel, Germany.
S30	Herrero, J., Sanchez, F. & Toro, M., “Fault tolerance as an aspect using JReplika”, In <i>Proceedings of the 8th Workshop on Future Trends of Distributed Computing Systems (FTDCS 2001)</i> , 2001, pp. 201–207, Bologna, Italy.
S31	Ho, W.-M., Jézéquel, J.-M., Pennaneac’h, F. & Plouzeau, N., “A toolkit for weaving aspect oriented UML designs”, In <i>Proceedings of the 1st international conference on Aspect-oriented software development (AOSD 2002)</i> , 2002, pp. 99–105, Enschede, The Netherlands.
S32	Hovsepyan, A., Baelen, S., Berbers, Y. & Joosen, W., “Generic Reusable Concern Compositions”, In <i>Proceedings of the 4th European conference on Model Driven Architecture Foundations and Applications (ECMDA-FA 2008)</i> , 2008, pp. 231–245, Berlin, Germany.
S33	Jacobson, I. & Ng, P.-W., <i>Aspect-Oriented Software Development with Use Cases</i> , Addison-Wesley Object Technology Series, Addison-Wesley Professional, 2004.
S34	Juerjens, J., “Model-based security engineering with UML: Introducing security aspects”, In <i>Proceedings of the 4th International Symposium on Formal Methods for Components and Objects (FMCO 2005)</i> , 2005, pp. 64–87 (LNCS v.4111), Amsterdam, Netherlands.
S35	Júnior, J. U., Camargo, V. V. & Chavez, C. V. F., “UML-AOF: a profile for modeling aspect-oriented frameworks”, In <i>13th Aspect-oriented modeling workshop (AOM 2009)</i> , 8th International Conference on Aspect-Oriented Software Development (AOSD 2009), 2009, pp.1–6, Charlottesville, USA.
S36	Júnior, J. U., Penteado, R. D. & Camargo, V. V., “An overview and an empirical evaluation of UML-AOF: an UML profile for aspect-oriented frameworks”, In <i>Proceedings of the 2010 ACM Symposium on Applied Computing (SAC 2010)</i> , 2010, pp. 2289–2296, Sierre, Switzerland.
S37	Katara, M. & Katz, S., “A concern architecture view for aspect-oriented software design”, <i>Software and Systems Modeling (SSM)</i> , 2007, 6(3):247–265.
S38	Kienzle, J., Al Abed, W. & Klein, J., “Aspect-oriented multi-view modeling”, In <i>Proceedings of the 8th international conference on Aspect-oriented software development (AOSD 2009)</i> , 2009, pp. 87–98, Charlottesville, USA.
S39	Li, H., Zhang, J. & Chen, Y., “Aspect-oriented modeling in software architecture pattern based on UML Computer and Automation Engineering (ICCAE)”, In <i>Proceedings of the 2nd International Conference on Computer and Automation Engineering (ICCAE 2010)</i> , volume 3, pp. 575–578, Singapore, Malaysia.
S40	Mosconi, M., Charfi, A., Svacina, J. & Wloka, J., “Applying and evaluating AOM for platform independent behavioral UML models”, In <i>Proceedings of the 12th international workshop on Aspect-oriented modeling (AOM 2008)</i> , 2008, pp. 19–24, Brussels, Belgium.
S41	Mostefaoui, F. & Vachon, J., “Design-level detection of interactions in Aspect-UML models using Alloy”, <i>Journal of Object Technology (JOT)</i> , 6(7):137–165.
S42	Mouheb, D., Talhi, C., Lima, V., Debbabi, M., Wang, L. & Pourzandi, M., “Weaving security aspects into UML 2.0 design models”, In <i>Proceedings of the 13th workshop on Aspect-oriented modeling (AOM 2009)</i> , 2009, pp. 7–12, Charlottesville, USA.
S43	Muley, K., Suman, U. & Ingle, M., “Representing join point in UML using pointcuts”, In <i>Proceedings of the International Conference on Computer and Communication Technology (ICCCCT 2010)</i> , 2010, pp. 557–561, Allahabad, Uttar Pradesh.
S44	Nakajima, S. & Tamai, T., “Aspect-oriented software design with a variant of UML/STD”, In <i>Proceedings of the 2006 international workshop on Scenarios and state machines: models, algorithms, and tools (SCESM 2006)</i> , 2006, pp. 44–50, Shanghai, China.
S45	Naufal, R. A., Silveira, F. F. & Guerra, E. M., “A Collaborative Support Approach on UML Sequence Diagrams for Aspect-Oriented Software”, In <i>Workshop Proceedings of the 28th of the International Conference on Conceptual Modeling (ER 2009)</i> , 2009, pp. 116–125 (LNCS v.5833), Gramado, Brazil.
S46	Naveed, M., Abdullah, M. K., Rashid, K. & Ahmad, H. F., “Representing Shared Join Points with State Charts: A High Level Design Approach”, In <i>World Academy of Science Engineering and Technology (WASET 2006)</i> , 2006, volume 15, pp. 80–84, Melbourne, Australia.
S47	Nemuraite, L. & Balandyte, M., “Aspect-Oriented Use Cases and Crosscutting Interfaces for Reconfigurable Behavior Modeling”, In <i>Proceeding of the 7th International Baltic Conference on Databases and Information Systems (DBIS 2006)</i> , 2006, pp. 189–202, Vilnius, Lithuania.
S48	Netinant, P. & Pukdesree, S., “Separation of concerns in automation of code generation”, In <i>Proceedings of the 5th IASTED International Conference on Computer Science and Technology (ACST 2008)</i> , 2008, pp. 91–96, Langkawi, Malaysia.
S49	Nouh, M., Ziarati, R., Mouheb, D., Alhadidi, D., Debbabi, M., Wang, L. & Pourzandi, M., “Aspect weaver: a model transformation approach for UML models”, In <i>Proceedings of the 2010 Conference of the Center for Advanced Studies on Collaborative Research (CASCON 2010)</i> , 2010, pp. 139–153, Toronto, Canada.

Continued...

Table 6 – Continued

Id	Reference
S50	Petriu, D. C., Shen, H. & Sabetta, A., “Performance analysis of aspect-oriented UML models”. <i>Software and Systems Modeling (SSM)</i> , 6(4):453–471
S51	Ping, Z. & Yang, S., “Understanding the aspects from various perspectives in aspects-oriented software reverse engineering”, In <i>Proceedings of the International Conference on Computer Application and System Modeling (ICCASM 2010)</i> , 2010, volume 11, pp.311–314, Taiyuan, China.
S52	Przybyłek, A., “Separation of crosscutting concerns at the design level: An extension to the UML metamodel”, In <i>Proceedings of the International Multiconference on Computer Science and Information Technology (IMCSIT 2008)</i> , 2008, volume 3, pp. 551–557, Wisla, Poland.
S53	Reddy, Y. R., Ghosh, S., France, R. B., Straw, G., Bieman, J. M., McEachen, N., Song, E. & Georg, G., “Directives for composing aspect-oriented design class models”, In <i>Transactions on Aspect-Oriented Software Development I</i> , pp. 75–105 (LNCS v.3880)
S54	Rui, W., Xiao-Guang, M., Zi-Ying, D. & Yan-Ni, W., “Extending UML for Aspect-Oriented Architecture Modeling”, In <i>Proceedings of the 2nd International Workshop on Computer Science and Engineering (WCSE 2009)</i> 2009, pp. 362–366, Qingdao, China.
S55	Sánchez, P., Stein, D. & Hanenberg, S., “Statecharts as an intermediate representation of JPDDs which help developers to map them to executable artifacts”, In <i>Proceedings of the 12th international workshop on Aspect-oriented modeling (AOM 2008)</i> , 2008, pp. 31–36, Brussels, Belgium.
S56	Sharafi, Z., Mirshams, P., Hamou-Lhadj, A. & Constantinides, C., “Extending the UML Metamodel to Provide Support for Crosscutting Concerns”, In <i>Proceedings of the Eighth ACIS International Conference on Software Engineering Research, Management and Applications (SERA 2010)</i> , 2010, pp.149–157, Montreal, Canada.
S57	Sharif, H., Rehan, S. & Farrukh, G. A., “Aspect-oriented requirements engineering: A use case based approach”, In <i>Proceedings of the 9th International Multitopic Conference (INMIC 2005)</i> , 2005, pp. 566–570, Karachi, Pakistan.
S58	Shen, W. & Wong, W. E., “Flexible Aspect-Oriented Design Model Checking”, In <i>Proceedings of the 15th Asia-Pacific Software Engineering Conference (APSEC 2008)</i> , 2008, pp. 537–544, Beijing, China.
S59	Somé, S. & Anthonysamy, P., “An approach for aspect-oriented use case modeling”, In <i>Proceedings of the 13th international Workshop on Early Aspects (WEA 2008)</i> , 2008, pp. 27–33, Leipzig, Germany
S60	Stein, D., Hanenberg, S. & Unland, R., “A UML-based aspect-oriented design notation for AspectJ”, In <i>Proceedings of the 1st international conference on Aspect-oriented software development (AOSD 2002)</i> , 2002, pp. 106–112, Enschede, The Netherlands.
S61	Stein, D., Hanenberg, S. & Unland, R., “Join point designation diagrams: A graphical representation of join point selections”, <i>International Journal of Software Engineering and Knowledge Engineering (IJSEKE)</i> , 2006, 16(3):317–346.
S62	Su, Y., Li, F., Hu, S.-M. & Chen, P., “Aspect-oriented software reverse engineering”, <i>Journal of Shanghai University (JSU)</i> , 2006, 10(5):402–408.
S63	Su, Y., Zhou, X.-W. & Zhang, M.-Q., “Research on Modeling Traversing Features in Concurrent Software System”, In <i>Proceedings of the 2008 International Conference on Computer Science and Software Engineering (CSSE 2008)</i> , 2008, pp. 81–84, Wuhan, China.
S64	Wang, B., Zhu, C. & Sheng, J., “A Formal Description Method for Aspect-Oriented Statechart Based on CSP”, In <i>Proceedings of the 2008 International Symposium on Computer Science and Computational Technology (ISCST 2008)</i> , 2008, volume 2, pp. 750–753, Wuhan, China.
S65	Wang, C.-M., Huang, C.-C., Chen, H.-M. & Wang, S.-T., “Conformance Checking of Running Programs in Dynamic Aspect-Oriented Systems”, In <i>Proceedings of the 14th Asia-Pacific Software Engineering Conference (APSEC 2007)</i> , 2007, pp. 183–190, Nagoya, Japan.
S66	Wang, Y., Singh, S., Hosking, J. & Grundy, J., “An aspect-oriented UML tool for software development with early aspects”, In <i>Proceedings of the 2006 international workshop on Early aspects at ICSE (EA 2006)</i> , 2006, pp. 51–58, Shanghai, China.
S67	Wehrmeister, M., Freitas, E., Pereira, C. & Wagner, F., “An aspect-oriented approach for dealing with non-functional requirements in a model-driven development of distributed embedded real-time systems”, In <i>Proceedings of the 10th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing (ISORC 2007)</i> , 2007, pp. 428–432, Santorini Island, Greece.
S68	Whittle, J., Jayaraman, P., Elkhodary, A., Moreira, A. & Araújo, J., “MATA: A unified approach for composing UML aspect models based on graph transformation”, In <i>Transactions on Aspect-Oriented Software Development VI</i> , 2009, pp. 191–237 (LNCS v.5560).
S69	Woodside, M., Petriu, D. C., Petriu, D. B., Xu, J., Israr, T., Georg, G., France, R., Bieman, J. M., Houmb, S. H. & Jürjens, J., “Performance analysis of security aspects by weaving scenarios extracted from UML models”, <i>Journal of Systems and Software (JSS)</i> , 82(1):56–74.
S70	Xu, D., Xu, W. & Wong, W. E., “Testing aspect-oriented programs with UML design models”, <i>International Journal of Software Engineering and Knowledge Engineering (IJSEKE)</i> , 18(3):413–437.
S71	Xu, Y., Tang, S., Xu, Y. & Tang, Z., “Towards aspect oriented web service composition with UML”, In <i>Proceedings of the 6th IEEE/ACIS International Conference on Computer and Information Science (CCIS 2007)</i> , 2007, pp. 279–284, Melbourne, Australia.
S72	Yang, S. & Jun, Q., “Approach on modeling crosscutting features in concurrent system”, In <i>Proceedings of the Sixth International Conference on Networked Computing and Advanced Information Management (NCM 2010)</i> , 2010, pp. 133–138, Seoul, Korea.
S73	Zhang, G., “Towards Aspect-Oriented Class Diagrams”, In <i>Proceedings of the 12th Asia-Pacific Software Engineering Conference (APSEC 2005)</i> , 2005, pp. 763–768, Taipei, Taiwan

Continued...

**Table 6 – Continued**

<b>Id</b>	<b>Reference</b>
S74	Zhang, G., Hoelzl, M. & Knapp, A., “Enhancing UML state machines with aspects”, In <i>Proceedings of the 10th International Conference on Model Driven Engineering Languages and Systems (MODELS 2007)</i> , 2007, pp. 529–543 (LNCS v.4735), Nashville, USA.
S75	Zhang, J., Chen, Y. & Liu, G., “The Realization of Aspectual State Model with UML Extension Mechanism”, In <i>Proceedings of the 4th IEEE Conference on Industrial Electronics and Applications (ICIEA 2009)</i> , 2009, pp. 3059–3063, Xian, China.
S76	Zhang, J., Chen, Y. & Liu, G., “Modeling Aspect-Oriented Programming with UML Profile”, In <i>Proceedings of the 1st International Workshop on Education Technology and Computer Science (ETCS 2009)</i> , 2009, volume 2, pp. 242–245, Wuhan, Hubei.
S77	Zhang, J., Chen, Y., Liu, G. & Li, H., “Using Sequence Diagram to Support Aspect-Oriented Programming in MDA”, In <i>Proceedings of the 1st International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC 2009)</i> , 2009, pp. 359–362, Hangzhou, China.
S78	Zhang, J., Chen, Y., Zhang, Y. & Li, H., “Aspect-oriented modeling and mapping driven by Model Driven Architecture”, In <i>Proceedings of the 2nd IEEE International Conference on Computer Science and Information Technology (ICCSIT 2009)</i> , 2009, pp. 180–184, Beijing, China.
S79	Zhang, J., Li, F. & Zhang, Y., “Aspect-Oriented Requirements Modeling”, In <i>Proceedings of the 31st IEEE Software Engineering Workshop (SEW 2008)</i> , 2008, pp. 35–40, Columbia, USA.
S80	Zhang, J., Li, H. & Li, F., “Research on Aspect Connectors for Software Architecture Adaptation”, In <i>Proceedings of the 31st IEEE Software Engineering Workshop (SEW 2007)</i> , 2007, pp. 63–66, Columbia, USA.
S81	Zhang, L., “Aspect-Oriented Analysis for Embedded Real-Time Systems”, In <i>Proceedings of the Advanced Software Engineering and Its Applications (ASEA 2008)</i> , 2008, pp. 53–56, Hainan Island, China.
S82	Zhou, X.-c., Liu, C., Niu, Y.-t. & Lai, T.-z., “Towards a Framework of Aspect-Oriented Modeling with UML”, In <i>Proceedings of the International Symposium on Computer Science and Computational Technology (ISCST 2008)</i> , 2008, volume 2, pp. 738–741, Shanghai, China.
S83	Zhou, Y., Lei, G., Li, P. & Kong, L., “Realizing extension use cases with AOP”, In <i>Proceedings of the International Symposium on IT in Medicine and Education (ITME 2008)</i> , 2008, pp. 1040–1044, Xiamen, China.
S84	Zhu, Z. J. & Zulkernine, M., “A model-based aspect-oriented framework for building intrusion-aware software systems”, <i>Information and Software Technology (IST)</i> , 51(5):865–875.

## Appendix B

This section presents the meaning of the abbreviations found in the map to the category “Main Events and Journals” (in Figure 6). Table 7 lists the journals and Table 8 lists the events where the primary studies were published.

**Table 7: Abbreviations of Journals**

<b>Abbreviation</b>	<b>Journal</b>
IJSEKE	International Journal of Software Engineering and Knowledge Engineering
IPS	IEEE Proceedings-Software
IST	Information and Software Technology
JOT	Journal of Object Technology
JSS	Journal of Systems and Software
JSU	Journal of Shanghai University
SSM	Software and Systems Modeling

**Table 8: Abbreviations of Events**

<b>Abbreviation</b>	<b>Event</b>
ACST	Advances in Computer Science and Technology
AGTIVE	Applications of Graph Transformations with Industrial Relevance
AOM	Aspect-oriented modeling
AOSD	Aspect-oriented software development
APSEC	Asia-Pacific Software Engineering Conference
ASEA	Advanced Software Engineering and Its Applications
ASWEC	Australian Conference on Software Engineering

Continued...



**Table 8 – Continued**

<b>Abbreviation</b>	<b>Event</b>
C3S2E	Canadian Conference on Computer Science and Software Engineering
CASCON	Center for Advanced Studies on Collaborative Research
CCIS	Conference on Computer and Information Science
CSIE	Computer Science and Information Engineering
CSSE	Computer Science and Software Engineering
DBIS	International Baltic Conference on Databases and Information Systems
EA	Workshop on Early Aspects
ECMDA-FA	European conference on Model Driven Architecture
ER	International Conference on Conceptual Modeling
ETCS	Education Technology and Computer Science
FMCO	Formal Methods for Components and Objects
FTDCS	Future Trends of Distributed Computing Systems
GPCE	International Conference on Generative Programming and Component Engineering
HASE	High-Assurance Systems Engineering
HICSS	Hawaii International Conference on System Sciences
ICACTE	International Conference on Advanced Computer Theory and Engineering
ICCAE	International Conference on Computer and Automation Engineering
ICCASM	International Conference on Computer Application and System Modeling
ICCSCT	International Conference on Computer and Communication Technology
ICCSIT	International Conference on Computer Science and Information Technology
ICEE	International Conference on E-Business and E-Government
ICIEA	IEEE Conference on Industrial Electronics and Applications
ICSE	International Conference on Software Engineering
ICITTA	International Conference on Information and Communication Technologies: From Theory to Applications
IHMSC	Intelligent Human-Machine Systems and Cybernetics
IMCSIT	International Multiconference on Computer Science and Information Technology
INFOS	International Conference on Informatics and Systems
INMIC	International Multitopic Conference
ISCC	IEEE Symposium on Computers and Communications
ISCST	International Symposium on Computer Science and Computational Technology
ISORC	International Symposium on Object and Component-Oriented Real-Time Distributed Computing
ITME	IT in Medicine and Education
MiSE	Models in Software Engineering
MODELS	International Conference on Model Driven Engineering Languages and Systems
NCM	Networked Computing and Advanced Information Management
SAC	Symposium on Applied Computing
SCES	Scenarios and state machines: models, algorithms, and tools
SEA	Software Engineering and Applications
SEPS	Software Engineering for Pervasive Services
SERA	Software Engineering Research, Management and Applications
SEW	Software Engineering Workshop
WASET	World Academy of Science Engineering and Technology
WCSE	Workshop on Computer Science and Engineering
WMSCI	World Multi-Conference on Systemics, Cybernetics and Informatics

## Appendix C

Table 9 presents the complete list of primary studies classified in each category established for the map “UML Diagrams to Represent Aspect-oriented Systems”. Table 10 presents the complete list of primary studies classified in each category established for the map “Main Keywords”.

**Table 9:** List of primary studies related to the map “UML Diagrams to Represent Aspect-oriented Systems”

UML Diagram	Primary Studies
Use Case	S5, S33, S47, S48, S51, S57, S59, S60, S62, S67, S83
State Machine	S1, S4, S7, S13, S19, S25, S26, S30, S34, S37, S38, S44, S46, S48, S55, S63, S64, S67, S68, S74, S75, S79, S81
Communication	S39, S81
Sequence	S4, S7, S8, S10, S11, S16, S32, S37, S38, S39, S40, S42, S43, S45, S48, S51, S55, S56, S60, S61, S62, S68, S69, S70, S77, S82, S84
Activity	S3, S12, S29, S24, S34, S40, S49, S50, S62, S69, S71, S72, S76
Profile	S1, S2, S4, S8, S11, S12, S14, S15, S17, S18, S20, S30, S32, S35, S36, S37, S40, S42, S49, S50, S51, S56, S58, S65, S67, S71, S78, S82
Package	S10, S34, S37, S39, S73
Deployment	S34, S50, S69
Component	S11, S21, S25, S50, S54, S66, S80
Class	S2, S4, S7, S15, S17, S19, S18, S20, S21, S23, S24, S25, S28, S31, S32, S34, S38, S40, S41, S49, S52, S53, S56, S61, S65, S67, S68, S70, S71, S73, S75, S79, S81, S82, S84
Object	S48

**Table 10:** List of primary studies related to the map “Main Keywords”

Main Keywords	Primary Studies
Aspect Composition	S21, S46, S69
Case Study	S1, S8, S12, S18, S25, S26, S32, S36, S40, S45, S50, S51, S56, S63, S67, S72, S80, S82
Component	S11, S16, S21, S25, S28, S45, S50, S54, S64, S80
Composition Mechanism	S5, S32, S50, S53
Framework	S13, S31, S32, S35, S36, S37, S48, S49, S51, S62, S67, S82, S84
Join Point Model	S2, S43, S44, S46
Model-driven Architecture	S1, S8, S77, S78
Model-driven Development	S11, S24, S32
Requirements	S9, S10, S13, S21
Reverse Engineering	S22, S27, S51, S56, S62
Software Architecture	S13, S17, S23, S27, S35, S39, S54, S80
Tool Support	S5, S8, S12, S13, S17, S24, S28, S38, S51, S53, S56, S58, S66, S68
UML Based	S3, S7, S9, S11, S16, S19, S22, S23, S38, S47, S50, S55, S60, S61, S62, S63, S67, S68, S70, S71, S75, S77, S80
UML Extension	S5, S6, S8, S10, S12, S13, S15, S17, S21, S26, S27, S28, S29, S30, S31, S32, S33, S44, S45, S46, S51, S52, S54, S56, S57, S58, S59, S64, S65, S66, S69, S73, S74, S76, S78, S81, S83, S84
UML Profile	S1, S4, S14, S20, S21, S25, S34, S35, S36, S37, S40, S41, S42, S49, S82
UML Standard	S48, S53, S79