UML consistency rules: a systematic mapping study

Damiano Torre [§,¶] Yvan Labiche [§] Marcela Genero [¶]

[§] Carleton University, Department of Systems and Computer Engineering,
Software Quality Engineering Laboratory
1125 Colonel By Drive, Ottawa, ON K1S5B6, Canada

[¶] University of Castilla-La Mancha, Department of Technologies and Information Systems,
ALARCOS Research Group,
Paseo de la Universidad, 4 13071 Ciudad Real, Spain

dctorre@sce.carleton.ca labiche@sce.carleton.ca marcela.genero@uclm.es

ABSTRACT

Context: The Unified Modeling Language (UML), with its 14 different diagram types, is the de-facto standard modeling language for object-oriented modeling and documentation. Since the various UML diagrams describe different aspects of one, and only one, software under development, they are not independent but strongly depend on each other in many ways. In other words, the UML diagrams describing a software product must be consistent. Inconsistencies between these diagrams may be a source of faults in software systems. It is therefore paramount that these inconsistencies be detected, analyzed and hopefully fixed.

Objective: The aim of this article is to deliver a comprehensive summary of UML consistency rules as they are described in the literature to date to obtain an extensive and detailed overview of the current research in this area.

Method: We performed a Systematic Mapping Study by following well-known guidelines. We selected 95 primary studies from a search with seven search engines performed in December 2012.

Results: Different results are worth mentioning. First it appears that researchers tend to discuss very similar consistency rules, over and over again. Most rules are horizontal (98.10%) and syntactic (88.21%). The most used diagrams are the class diagram (71.58%), the sequence diagram (47.37%) and the state machine diagram (42.11%).

Conclusion: The fact that many rules are duplicated in primary studies confirms the need for a well-accepted list of consistency rules. This paper is a first step in this direction. Results indicate that much more work is needed to develop consistency rules for all 14 UML diagrams, in all dimensions of consistency (e.g., semantic and syntactic on the one hand, horizontal, vertical and evolution on the other hand).

Categories and Subject Descriptors

D.2.4 [**Software Engineering**]: Software/Program Verification - Model checking

I.6.5 [**Computing Methodologies**]: Model Development

General Terms

Documentation, Languages, Verification.

Keywords

Unified Modeling Language (UML), UML consistency rules, Systematic Mapping Study.

1. INTRODUCTION

The Model Driven Architecture (MDA) [1] promotes a set of transformations between successive models from requirements to analysis, to design, to implementation, and to deployment [2]. Recent years have seen a lot of attention into MDA in academia and industry [3-5], which resulted in models gaining even more importance in software development. The Unified Modeling Language (UML) [6] is the Object Management Group (OMG) most-used specification and the de-facto standard modeling language for object-oriented modeling and documentation [7-13]. It is the privileged modeling tool to implement the MDA. It is however not to be used in every single software development project [14]. The architecture of the UML is based on a four-layer meta-model structure, and it provides 14 diagram types [6] for describing a system from different perspectives (e.g., structure, behavior) or abstraction levels (e.g., analysis, design), which helps with addressing complex systems, distribute responsibilities among stakeholders, among other benefits. Since the various UML diagrams describe different aspects of one, and only one, software under development[1], they are not independent but strongly depend on each other in many ways. In other words, the UML diagrams describing a software product must be consistent. As UML is not a formal notation, inconsistencies may arise in the design specification of a complex system (i.e., between the UML diagrams of that specification) when such specification requires multiple diagrams to describe different perspectives of the software [15]. When UML diagrams portray contradicting or conflicting meaning, the diagrams are said to be inconsistent [16]. Inconsistencies between different diagrams/views of a model may be a source of the considerable increase of faults in software systems [17, 18]. It is therefore paramount that these inconsistencies be detected, analyzed and hopefully fixed [19].

Even though many researchers have proposed, explicitly or not, rules to prevent or detect different types of inconsistencies, no well-accepted, as complete as possible set of consistency rules has so far been described and published. Although the UML standard itself contains some consistency rules, often referred to as well-formedness rules, the standard does not offer a complete list since for instance some consistency rules may be specific to the way the UML notation is used. This lack of well-accepted list of rules forces researchers to systematically define the consistency rules they rely on for their research [15]. Although this is good practice, this results, as confirmed by some of our results, in researchers describing similar or even identical consistency rules, over and over again. Our overall objective is to identify a set, as complete as possible, of well-accepted consistency rules for UML diagrams. In other words, the main research question that is guiding our work is the following: What is the current state of the art in terms of UML consistency rules? To achieve this goal, we need a systematic, as objective as possible identification of the rules which have been applied, or have been described, to ensure consistency between UML diagrams. Hence, the aim of this article

[1] We see only one exception to this statement: the analysis and design of product lines. However, even then, diagrams ought to be consistent.

is to deliver a comprehensive summary of the existing UML consistency rules (to the best of our knowledge) to obtain an extensive and detailed overview of the current research in this area.

To achieve this goal, we performed a Systematic Mapping Study (SMS) [20] as this is a research method that provides an objective procedure for identifying the quantity of existing research related to a research question. Performing a SMS has several benefits [21]: it gives a starting point for PhD students and in the longer term, it provides a body of knowledge to the next generation of researchers. To carry out the SMS detailed in this paper we followed the guidelines of Kitchenham and Charters [22]. These guidelines are admittedly for systematic literature reviews. However they can be readily applied, and have been applied by others, when conducting systematic mapping studies.

This paper is structured as follows. In section 2 we provide a brief discussion on related work. This is followed by a description of the SMS protocol we followed [22]: the SMS planning (section 3), the SMS execution (section 4), and the results (section 5). A preliminary discussion with the main findings is provided in section 6. Threats to validity are in section 7. Finally, section 8 draws the conclusions and provides directions for future works.

2. RELATED WORK

As we have mentioned before, there are a lot of works on the consistency of UML diagrams. In the run-up to this SMS, we searched for surveys, literature reviews, mapping studies, or similar work on the topic of UML consistency. We only found six such publications, which we discuss in this section. It is important to note, as summarized later in this section that none of them answered our main research question (section 1).

To the best of our knowledge, the closest piece of work to our problem is a review on UML consistency management [3]. It is different from our SMS in several ways. The first important difference is the research protocol used during the review: they presented a Systematic Literature Review (SLR) while we present a SMS. The second main difference is the purpose: They focused only on the management of UML (in) consistencies, i.e., they focused on techniques to identify and fix inconsistencies, without discussing in details which inconsistencies had to be identified and fixed. In contrast our SMS focuses on those inconsistencies that need to be identified and fixed. A direct consequence of this difference is that we reviewed a broader number of papers about UML consistency (95 primary studies instead of 43), that approximately half of their primary studies are not primary studies in our SMS (only 24 of their primary studies are also primary studies in our SMS), and that our research overlaps (three of their six research questions are updated in our SMS, the other three being irrelevant in our context). Moreover the search periods are different: they cover the [2001-2007] period while we cover papers in the [2000-2012] period.

Another work about UML consistency presents a survey of consistency checking techniques for UML models [5]. The authors argue that formalizing UML models is preferable to verify consistency because this helps removing ambiguities and enforcing consistency. They briefly reviewed 17 articles, which represent less than a quarter of the number of primary studies in our SMS (95): only ten primary studies are common. During this survey the authors did not follow any SLR or SMS protocol, and they simply provided an initial summary of their findings about UML consistency checking techniques (not consistency rules).

Spanoudakis and Zisman [19] presented a survey on the problem of managing inconsistencies in software models, not specifically UML ones. The authors presented a conceptual framework that views inconsistency management as a process which incorporates activities for detecting overlaps and inconsistencies between software models, diagnosing and handling inconsistencies, tracking the information generated along the way, and specifying and monitoring the exact way of carrying out each of these activities. They then surveyed all the research works published prior to 2001 that address one or more of the aspects of their conceptual framework/process. It is worth noting that the process they carried out did not follow a SLR or SMS protocol.

Another piece of work [23] showed a rule based method for consistency checking in UML models supported by a software prototype for the MagicDraw UML CASE tool. The authors presented 50 UML consistency rules involving one or more UML diagrams. They obtained those rules by reviewing eight articles, seven of which are also considered in our SMS. Their brief review did not follow a systematic protocol (SLR or SMS): no clear process to obtain the papers, no clear process to include or exclude such documents. After conducting our SMS we compared the list of primary studies between their work and our work and identified the missing paper (the one of the eight). We confirmed our search could not find it. We nevertheless read the paper and identified it was describing only one consistency rule and that this rule was about the consistency between requirements and classes. Since we already had such a rule in our list (between use case descriptions and classes) we stopped investigating this paper.

Ahmad and Nadeem [7] presented a survey focusing only on Description Logic (DL) based consistency checking approaches. As one result of their research, they said that only class diagram, sequence diagram and state chart diagram inconsistencies were covered in the surveyed papers and a few common types of inconsistencies were discussed. They briefly described the background of the DL formalism and they reviewed three articles, which are also reviewed in our SMS. Their survey did not follow any SLR or SMS protocol.

Finally Genero et al conducted a SMS about the quality of UML diagrams [4]. Since they were interested in UML diagram quality in general they did not focus on UML consistency and even less so on UML consistency rules. They nevertheless discuss UML consistency and write that semantic consistency is by far the semantic quality subtype that has been researched most (42% of their primary studies). They mention that 70.27% of the papers that research semantic quality focused on consistency issues. Moreover they mention that the majority of methods attempt to improve semantic quality do improve the consistency of UML diagrams. In addition, most of the rules, modeling conventions, guidelines and checklists related to semantic quality that they discuss were especially related to consistency problems. This confirms that identifying inconsistencies between UML diagrams is a very important activity in improving UML model quality.

To summarize, our search for answers to our main research question failed, which confirmed the need for a SMS about UML consistency rules. It is also important to note that published works that relate to our SMS are in general more informal literature surveys or comparisons with no defined research questions, no search process, no defined data extraction or data analysis processes. Instead, our SMS follows a strict, well-known protocol.

3. PLANNING THE MAPPING STUDY

In this section we present the main components of the protocol required to carry out a SMS [22].

3.1 Research Questions

The underlying motivation for the research questions was to determine the current state of the art about UML consistency rules and this guided the design of the review process. In order to identify the current state of the art on UML consistency rules, we considered seven research questions (RQs): Table 1.

Table 1. Research questions

Research questions	Main motivation
RQ1: What are the UML versions used by researchers in the approaches found?	To discover what UML versions are used in the approaches that handle the UML consistency.
RQ2: Which types of UML diagrams have been tackled in each approach found?	To discover the UML diagrams that research has focused upon, to reveal the UML diagrams that are considered more important than others, as well as to identify opportunities for further research.
RQ3: What are the UML consistency rules to check?	To find the UML consistency rules to check and to assess the state of the field.
RQ4: Which types of consistency problems have been tackled in the rules found?	To find the types of consistency problems tackled in the rules. The data found are categorized into three consistency dimensions split into three sub-dimensions: 1) horizontal, vertical and evolution consistency; 2) syntactic and semantic consistency; 3) observation and invocation consistency.
RQ5: Which research type facets are used in research on UML model consistency?	To determine if the field is generally more applied or more basic research as well as to identify opportunities for future research. The papers found were categorized into six types: evaluation research, validation research proposal of solution, philosophical papers, opinion papers and personal experience papers.
RQ6: Is the approach presented automatic, manual or semi-automatic?	To discover how the approaches to check the UML consistency are implemented, in other word if their check system is presented with an automatic, manual or semi-automatic way.
RQ7: How the UML consistency rules are specified? How the UML consistency rules are checked?	To discover how the consistency rules to check the consistency of the UML diagrams are specified (e.g., Plain English, OCL, Promela) and to discover with which tools those consistency rules are checked (e.g., SPIN, OCL-Checker)

3.2 Search strategy

Conducting a search for primary studies requires the identification of search strings (SS), and the specification of the parts of primary studies (papers) in which the search strings are looked for (the search fields). To identify our search strings, we followed the procedure of Brereton et al [24]:

1. Define the major terms;
2. Identify alternative spellings, synonyms or related terms for major terms;
3. Check the keywords in any relevant papers were already available;
4. Use the Boolean OR to incorporate alternative spellings, synonyms or related terms;
5. Use the Boolean AND to link the major terms.

The major search terms were “UML” and “Consistency” and the alternative spellings, synonyms or terms related to the major terms are presented in Table 2.

Table 2. Search string

Major Terms	Alternative terms
UML	(uml OR unified modeling language OR unified modelling language)
Consistency	(consistency OR inconsistency)

In the selection of the SS, we considered various alternatives. For example the SS used in the SLR on consistency management [3] was discarded due to the fact that it might not strictly focus on UML consistency rules: we are much more interested in collecting rules than in identifying consistency management issues and solutions. (This is the main reason why we obtain a different set of primary studies.) Other SSs were experimented with, but due to space limits, we cannot discuss below all those alternative search strings. In the set of alternative SSs, we selected the following one as it allowed us to retrieve the largest number of useful papers, i.e., the largest number of papers focusing on UML consistency:

((uml OR unified modeling language OR unified modelling language) AND (consistency OR inconsistency))

We did not establish any restriction on publication years up to December 12, 2012. The SMS process started in September 2012 and was totally finished on October 2013. We used the above mentioned SS with the following seven search engines: IEEE Digital Library, Science Direct, ACM Digital Library, Scopus, Springer Link, Google Scholar, and WILEY. The searches were limited to the following search fields: title, keywords and abstract.

3.3 Selection procedure and inclusion and exclusion criteria

In this section we discuss the inclusion and exclusion criteria we used. We then discuss the process we followed to include a primary studies in this SMS. The inclusion criteria were:

- Electronic Papers (EPs) focusing on UML diagrams consistency which contained at least one UML consistency rule;
- EPs written in English language;
- EPs published in peer-reviewed journals, international conferences and workshops;
- EPs published up to December 12, 2012.
- EPs which proposed UML consistency rules with a restriction (or extension) of the UML models that don't strictly follow the OMG standard [6].

The exclusion criteria were:

- EPs not focusing on UML diagrams consistency;
- EPs which did not present a full-text paper (title, abstract, complete body of the article and references) but were reduced to an abstract for instance;
- EPs focusing on UML diagrams consistency which did not contain at least one UML consistency rule;
- Duplicated EPs (e.g., returned by different search engines);

- EPs which discussed consistency rules between UML diagrams and other, non-UML sources of data, such as requirements or source code.

3.4 Data extraction strategy

We extracted the data from the primary studies according to a number of criteria, which were directly derived from the research questions detailed in Table 1. Using each criterion to extract data required that we read the full-text of each of the 95 primary studies. Once recorded, we collected data in an Excel spreadsheet that represent our data form. From each primary study the following information was extracted and collected into the Excel data form:

- Search engines: where the paper was found (see section 3.2);
- Inclusion and Exclusion Criteria (see section 3.3);
- Data related to Research Questions (see Section 3.1):
    - What UML version was used;
    - What are the UML consistency rules discussed (see [25]);
    - What diagrams are involved in consistency rules: Class Diagram (CD), Collaboration Diagram (COD), Use Case Diagram (UCD), Communication Diagram (COMD), State Chart Diagram (SCD), Sequence Diagram (SD), Protocol State Machine Diagram (PSMD), Object Diagram (OD), Interaction Diagram (ID), Activity Diagram (AD), Composite Structure Diagram (CSD), Timing Diagram (TD), Interaction Overview Diagram (IOD), and Deployment Diagram (DD);
- Which are the types of UML consistency dimensions. Several possible dimensions of consistency appear in the literature, since there isn't yet any standard for reasoning about consistency. Three UML consistency dimensions have been proposed though [26]:
    - Horizontal, Vertical and Evolution Consistency: *Horizontal consistency*, also called intra-model consistency, refers to consistency within a model or between different diagrams of the model at the same level of abstraction, and within the same version [18]. *Vertical Inconsistency*, also called inter-model consistency, refers to consistency between models (and therefore their diagrams) at different levels of abstraction [27]. *Evolution consistency* refers to consistency between different versions of the same model (and therefore their diagrams), and has to be maintained when the model is in the process of evolution [18].
    - Syntactic versus Semantic consistency: *Syntactic consistency* ensures that a specification conforms to the abstract syntax specified by the meta-model, and requires that the overall model has to be well formed [27]. *Semantic consistency* requires that the behavior of diagrams be semantically compatible [27]. Note that this does not mean semantic consistency is necessarily restricted to behavioral diagrams. For instance, operation contracts available in the class diagram specify behavior. Semantic consistency applies at one level of abstraction (with horizontal consistency), at different levels of abstraction (vertical consistency), and during model evolution (evolution consistency) [7].
    - Observation versus Invocation consistency: *Observation consistency* requires that an instance of a subclass behave like an instance of its superclass, when viewed according to the superclass description [28]. In terms of UML state chart diagrams (corresponding to protocol state machines) this can be rephrased as "after hiding all new events, each sequence of the subclass state chart diagram should be contained in the set of sequences of the superclass state chart diagram." *Invocation consistency* requires that an instance of a subclass of a parent class can be used wherever an instance of the parent is required [28]. In terms of UML state chart diagrams (corresponding to protocol state machines), each sequence of transitions of the superclass state chart diagram should be contained in the set of sequences of transitions of the state chart diagram for the subclass.
    - Tool support (Automatic, Semi-Automatic, Manual);
        - Automatic means that a tool automatically checks the UML consistency rules with no human intervention;
        - Semi-automatic means that checking the UML consistency rules were partially automated (for instance when the check of a UML model needs the support of user to finish the process);
        - Manual means that the UML consistency rules were not supported by any implemented and automatic tool.
    - What mechanisms were used to specify the rules: e.g., plain language, Promela, etc.;
    - How the UML consistency rules are checked: e.g., SPIN, OCL-Checker, etc.;
    - Research type facet followed in the paper, for which we used the following classification [29]:
        - Evaluation research (ER): this is a paper that investigates techniques that are implemented in practice and an evaluation of the technique is conducted. That means, the paper shows how the technique is implemented in practice (solution implementation) and what are the consequences of the implementation in terms of benefits and drawbacks (implementation evaluation).
        - Proposal of solution (PS): this is a paper that proposes a solution to a problem and argues for its relevance, without a full-blown validation.
        - Validation Research (VR): this is a paper that investigates the properties of a solution that has not yet been implemented in practice.
        - Philosophical papers (PP): this is a paper that sketches a new way of looking at things, a new conceptual framework, etc.
        - Opinion papers (OP): this is a paper that contains the author's opinion about what is wrong or good about something, how something should be done, etc.
        - Personal experience papers (PEP): this is a paper that emphasizes more on what and not on why.

4. EXECUTION

The planning for this SMS with the seven search engines begun in September 2012 and was completed on December 12, 2012. In this section we present the execution of the SS into the seven search engines and the selection of primary studies according to the inclusion/exclusion criteria previously described. In order to document the review process with sufficient details [22], we describe the multi-phase process of four sub-phases we followed:

- First sub-phase (SP1): the search string was used to search with the seven search engines as mentioned earlier.
- Second sub-phase (SP2): we deleted duplicates automatically, by using the RefWorks tool [30]; we also removed duplicates manually.
- Third sub-phase (SP3): we obtained an initial set of studies by reading the title, abstract and keywords of all the papers obtained after SP2 while enforcing the inclusion and exclusion criteria. When reading just the title, abstract and keywords of a paper was not enough to decide to include or exclude it, we checked the full-text.
- Fourth sub-phase (SP4): all the papers identified in SP3 were read in their entirety and the exclusion criteria were applied again. This resulted in the final set of primary studies.

Table 3 breaks down the number of papers we have found by sub-phases. Row SP1 in Table 3 shows the first results which were obtained by running the SS into the seven search engines selected. The next two rows show the results obtained after applying SP2 and SP3 of the studies selection process. In the end, we collected 95 primary studies for further analysis. The complete list of references can be found elsewhere [25].

Table 3. Summary of primary studies selection

Sub phase	IEEE	Scopus	Springer Link	Google Scholar	WILEY	ACM	Science Direct	Total
SP1: Raw results	363	601	163	341	9	87	39	1603
SP2: No duplicates	279	325	159	247	9	80	36	1135
SP3: First selection	62	64	62	28	4	33	14	267
SP4: Primary studies	16	21	21	12	1	16	8	**95**

5. RESULTS

To reach the goal of this SMS, i.e., addressing the research questions listed in section 3.1, the 95 primary studies selected were classified according to the criteria detailed in section 3.4, then the results of the SMS reported in this section show the answers to the seven research questions previously presented.

A quantitative summary of the results for research questions RQ1, RQ2, RQ4, RQ5 and RQ6 is presented in **Error! Reference source not found.**. More details are provided in the following sub-sections.

5.1 UML version (RQ1)

Figure 1 plots the number of papers presenting rules for specific versions of the UML.

The presence of 29.47% (28 of 95 papers) of the primary studies with an old version (1.x) of the UML shows that the issue of the UML consistency rules started to be relevant from the initial launch of the UML (which has been evolving since the second half of the 1990s [6]). UML 2.0 is the UML version mostly used in the primary studies: 32.63% (31 of 95 papers).

Table 4. Results of SMS

Research question	Possible Answer		Result # Papers	Result Percent
RQ1: UML versions	UML 1.1		1	1.05%
	UML 1.3		13	13.68%
	UML 1.4		6	6.32%
	UML 1.5		8	8.42%
	UML 2.0		31	32.63%
	UML 2.1.X		10	10.53%
	UML 2.2		2	2.11%
	UML 2.3		1	1.05%
	UML 2.4.1		1	1.05%
	NF		19	20.00%
	Ext.		1	1.05%
	Sim.		2	2.11%
RQ2: UML diagrams	Class Diagram		68	71.58%
	State Chart Diagram		40	42.11%
	Protocol State Machine Diagram		5	5.26%
	Sequence Diagram		45	47.37%
	Collaboration Diagram		8	8.42%
	Activity Diagram		12	12.63%
	Use Case Diagram		14	14.74%
	Object Diagram		4	4.21%
	Communication Diagram		2	2.11%
	Composite Structure Diagram		1	1.05%
	Interaction Diagram		4	4.21%
RQ4: Types of consistency problems	1st Dimension	Horizontal	258	98.10%
		Vertical	5	1.90%
		Evolution	0	0.00%
	2nd Dimension	Syntactic	232	88.21%
		Semantic	31	11.79%
	3rd Dimension	Invocation	3	1.14%
		Observation	3	1.14%
RQ5: Research type facets	ER		16	16.84%
	VR		29	30.53%
	PS		47	49.47%
	PP		0	0.00%
	OP		0	0.00%
	PEP		3	3.16%
RQ6: Type of support	Automatic		24	25.26%
	Semi-Automatic		29	30.53%
	Manual		42	44.21%

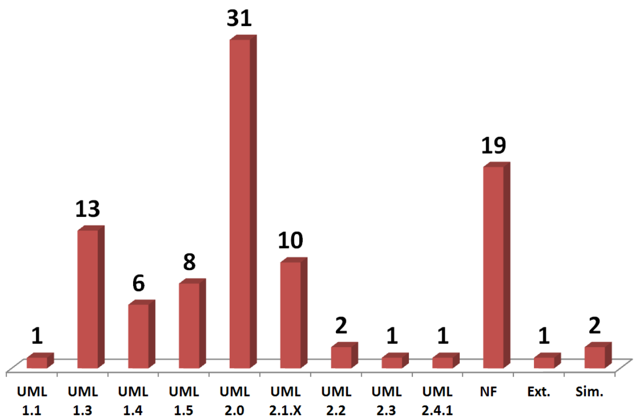

Figure 1. UML version

The subsequent UML versions (2.1, 2.1.1 and 2.1.2) were merged into 2.1.X to obtain a more readable graph. NF means “not found” and it represents all those primary studies which did not report on the UML version used and for which we were not able to guess the UML version by reading the text. “Ext.” and “Sim.” represent primary studies which use an extension or simplification of the UML notation that do not strictly follow the UML standard [6].

5.2 Types of UML diagrams (RQ2)

In this section we discuss the different types of UML diagrams involved in primary studies. Figure 2 indicates that collected rules describe consistency on only eleven of the 14 UML diagrams. (We did not collect any rule involving the timing, interaction overview and deployment diagrams.)

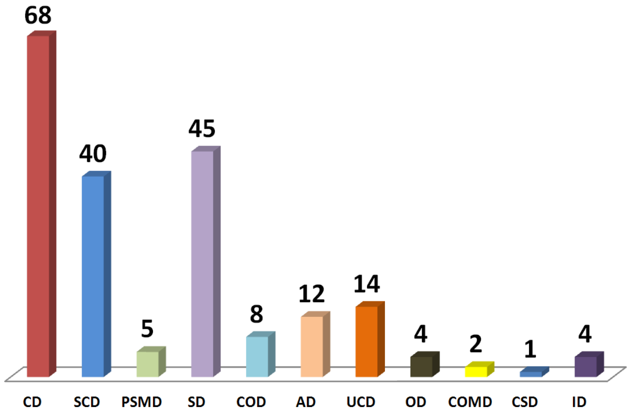

Figure 2. UML diagrams

Not surprisingly, since these are the most used diagrams [31], the Class Diagram (71.58%), the Sequence Diagram (47.37%), and the State Machine Diagram (42.11%) are the diagrams mostly involved in consistency rules. Research on UML consistency rules has placed much less attention on the Use Case Diagram (14.74%) and the Activity Diagram (12.63%). The Collaboration Diagram was found in 8.42% of the primary studies. The least used diagrams are the Protocol State Machine Diagram, the Object Diagram, the Interaction Diagram, the Communication Diagram and the Composite Structure Diagram.

5.3 UML consistency rules (RQ3)

The principal aspect shown in this RQ is that researchers involved into UML consistency rules typically define a number of similar consistency rules over and over again. Specifically, we collected a list of 603 UML consistency rules from the primary studies. After removing duplicates, or rules that are implied by another rule, we obtained a list of 263 UML consistency rules: The complete list of 263 UML consistency rules is presented elsewhere due to space limits [25]. In other words, only 43.33% (263 of 607) of the UML consistency rules initially collected were unique. The rest of the UML consistency rules were mostly due to duplications or implications (33.11%, 201 of 603). Other rules (23.56%, 143 of 603) were eliminated for a couple of reasons: they were not consistency rules (e.g., rules describing good modeling practices); they were explained in an ambiguous language; they were out of the scope of our research (e.g., focused on aspect-oriented multi-view modeling); yet others were simply inexact (i.e., either contradicting the UML metamodel, or contradicting UML-based modeling principles).

5.4 UML consistency dimensions (RQ4)

This sub-section presents the results about the number of UML consistency rules divided into the UML consistency dimension presented in section 3.4.

The results show that the great majority of UML consistency rules are Horizontal and Syntactic rules, respectively with 98.10% (258 of 263 rules) and 88.21% (232 of 263 rules) of the total of collected UML consistency rules. Moreover, 31 (11.79%) Semantic rules involved in UML consistency were found. Researchers described strikingly many more syntactic than semantic consistency rules. Also, although we have not yet compared the 263 rules with well-formedness rules of the UML standard, we suspect that a large majority of the syntactic consistency rules we collected are already in the UML standard: for instance several authors present the rule whereby a class cannot be a descendant (or ancestor) of itself in a class diagram, which is already a constraint of the UML metamodel. Proposals of UML consistency rules have placed much less attention on Vertical (1.90%), Invocation (1.14%) and Observation (1.14%) consistency. We were surprised to discover that no one Evolution consistency rule (0%) was proposed by researchers.

5.5 Research type facets (RQ5)

The results of the research type facet classification show that 49.47% (47 of 95 papers) of primary studies proposed solutions to the inconsistency problem (PS), 30.53% (29 of 95 papers) presented validation research (VR), 16.84% (16 of 95 papers) presented evaluation research (ER), and only 3.16% (3 of 95 papers) presented personal experience (PEP). We did not find any philosophical paper (PP) nor opinion paper (OP) (0% for both). This suggests the field is about problem solving and requires more evaluations of the consistency rules that have been proposed.

5.6 Tool support (RQ6)

The UML consistency rules presented by researchers are supported by automatic tools (25.26%, 24 of 95 papers), semi-automatic tools (30.53%, 29 of 95 papers), and finally the larger number of publications presented manual verification (44.21%, 42 of 95 papers).

5.7 UML consistency rules: specification and support (RQ7)

Figure 3 shows that plain english (29.47%) is the most used language to specify UML consistency rules, followed by the Object Constraint Language (OCL) [6] (22.11%), Communicating Sequential Processes (CSP) and Promela (5.26% each). Using OCL makes sense since this is a constraint language that is part of the UML; it is mostly used in syntactic rules. Languages such as CSP and Promela have been used to specify semantic rules between the sequence diagram and the state machine diagram. The category "other" in Figure 3 summarizes all those proposals (23.16%, 22 of 95 papers) that present a specification mechanism that appears in only one primary study (for instance XML Equivalent Transformation, Prolog, Constraint Logic Programming).

The majority of the papers (55.79%) presented tool support to check UML consistency rules: Figure 4. IBM Rational Rose is the most used tool (10.53%), followed by Spin, UML/Analyzer and OCL interpreters, the last two contributing respectively 6.32% and 5.26% of the total. The category "other" in Figure 4 summarizes all those primary studies that present a UML consistency checking tool that is used in only one primary study (e.g., Poseidon, ArgoUML). 44.21% of the primary studies did not present any tool support (in Figure 4 "NI" means not implemented) for their UML consistency rules.

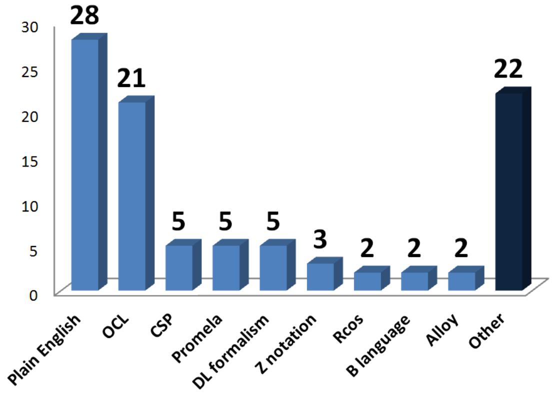

Figure 3. Language of UML consistency rules

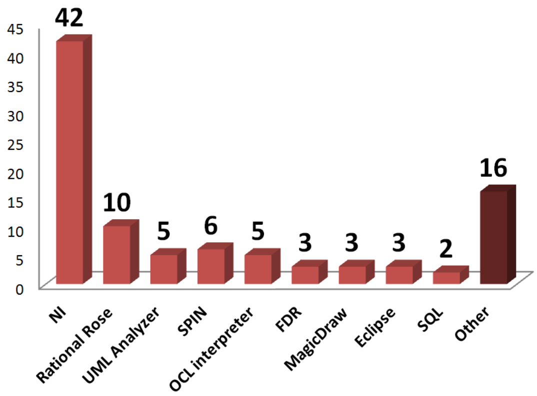

Figure 4. Tool used to check UML consistency rules

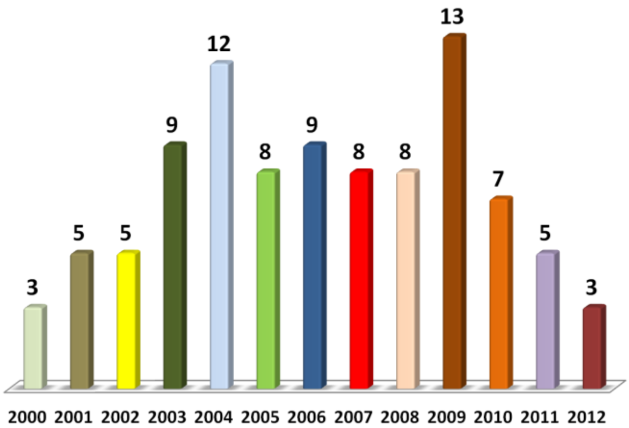

Figure 5. Number of papers per year

5.8 Additional results

Table 5 shows the publication venues with the largest number of papers on UML consistency rules. The first three have the same number of five papers each, together representing 15.79% (15 papers) of the total. The next three, all of them with four papers together represent 12.63% (12 papers) of the total.

The distribution per year of the 95 primary studies is shown in Figure 5. Between 2003 and 2010, the number of publications remained relatively stable from 7 to 13 articles a year, except in 2004 and 2009. We also notice that the release of UML 2.0 in 2005 did not impact numbers much. All this suggests that the topic of UML diagrams consistency remains important to the research community. The number of publications decreased in 2012, but it is likely due to the fact that many papers published in that year were not yet available at the time we performed searches.

Table 5. Number of papers per type of publication

Publication	# Paper	Percent
International conference on Model driven engineering languages and systems (MODELS)	5	5.26%
IEEE/ACM International Conference on Automated Software Engineering (ASE)	5	5.26%
International Conference on Conceptual Modeling (ER)	5	5.26%
Australian Conference on Software Engineering (ASWEC)	4	4.21%
Electronic Notes in Theoretical Computer Science	4	4.21%
International Conference on Software Engineering (ICSE)	4	4.21%
IEEE Transactions on Software Engineering	2	2.11%
Proceedings of the IEEE Region 10 (TENCON)	2	2.11%
IEEE International Conference on Software Engineering and Formal Methods (SEFM)	2	2.11%
ACM symposium on Applied computing (SAC)	2	2.11%
International Conference on Computer Systems and Technologies (CompSysTech)	2	2.11%
ACS/IEEE International Conference on Computer Systems and Applications (AICCSA)	2	2.11%

6. DISCUSSION

The following sub-sections describe the analysis of the results for RQ1 to RQ6, defining bubble plots in order to report the frequencies of combining the results from different research questions. A bubble plot is basically two x–y scatter plots with bubbles in category intersections. This synthesis method is useful to provide a map and it gives a quick overview of a research field [32].

6.1 Combining RQ1, RQ2 and RQ5

Combining the results of RQ1, RQ2 and RQ5, we obtained (Figure 6) the mapping of the research type facets used depending on the year of publications and the type of UML diagrams. In the same way the different UML versions are shown according to the UML diagrams and year of papers published. The results about the UML versions show that with 23 proposals, the Class Diagram is the most used UML diagram with UML version 2.0. It is closely followed by the Sequence Diagram with 20 papers in the same UML version. This is an observation that we can consistently make across UML versions. Proposals which used State Chart Diagrams were constant (in numbers) between UML versions 1.3 and 2.1.X; in fact the number of publications remained relatively stable from 4 to 9 articles for version reaching its peak with 9 articles for the UML version 2.0. Little has been proposed for UML versions 2.2 and 2.3, perhaps because of the small changes to the metamodel from UML 2.1.

As shown in Figure 6, most of the primary studies which present a PS paper, present rules that involve the Class Diagram, Sequence Diagram and State Chart Diagram, respectively with 72.34% (34 papers), 53.19% (25 papers) and 48.94% (23 papers) of the 47 PS primary studies. It is important to note that the vast majority of the primary studies (24 of 29, 82.76%) that perform some validation (classified as VR) focus on the class diagram. We also observe an imbalance of the ratio ER + VR primary studies (evaluation and validation) over PS primary studies for the three main diagrams: i.e., we see a higher ER+VR / PS for the class diagram (97%) than

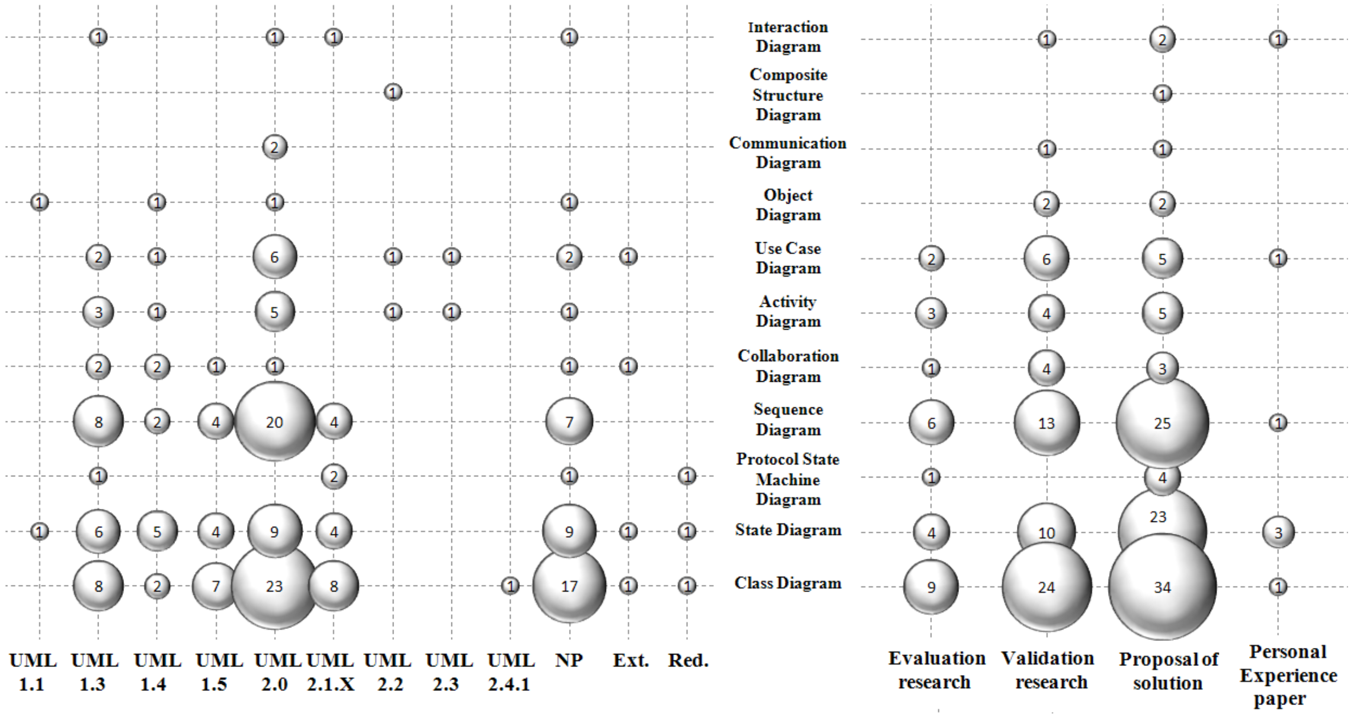

Figure 6. Combining RQ1, RQ2 and RQ5

for the sequence diagram and state machine diagram (76% and 60.8%, respectively).

6.2 Combining RQ2 and RQ3

As a consequence of RQ2 and RQ3, Table 6 shows that the pairs of diagrams mostly involved in rules are CD-SD, CD-COD, and SD-SCD (52.10%).

Table 6. Consistency between two diagrams

Consistency between 2 diagrams	# Rules	Percent
Class Diagram and Sequence Diagram	26	21.85%
Class Diagram and State Chart Diagram	25	21.01%
Class Diagram and Collaboration Diagram	11	9.24%
State Chart Diagram and Sequence Diagram	9	7.56%
Sequence Diagram and Activity Diagram	9	7.56%
Sequence Diagram and Use Case Diagram	5	4.20%
Activity Diagram and Use Case Diagram	5	4.20%
Class Diagram and Use Case Diagram	5	4.20%

Table 7 shows that CD, SCD and SD are the diagrams mostly used in rules involving only one diagram (84.72%).

Table 7, Consistency in one diagram

Consistency in one diagram	# Rules	Percent
Class Diagram	60	41.67%
State Chart Diagram	52	36.11%
Sequence Diagram	10	6.94%
Composite Structure Diagram	8	5.56%
Activity Diagram	6	4.17%
Use Case Diagram	3	2.08%
Collaboration Diagram	3	2.08%

6.3 Combining RQ2, RQ3 and RQ4

First of all regarding the several dimensions that can be used to classify consistency rules (section 3.4), we note that 69.47% (66 of 95 papers) of the primary studies did not mention any such dimension, that 17.89% (17 of 95 papers) presented horizontal and vertical consistency rules, and only 4.21% mentioned also evolution consistency with those two dimensions.

Table 8. Horizontal and Syntactic rules

Horizontal and Synaptic dimensions	# Rules	Percent
Class Diagram	127	38.60%
State Chart Diagram	75	22.80%
Sequence Diagram	48	14.59%
Activity Diagram	23	6.99%
Use Case Diagram	20	6.08%
Collaboration Diagram	18	5.47%
Composite Structure Diagram	8	2.43%

As a consequence of RQ2, RQ3 and RQ4, Table 8 ranks horizontal and syntactic rules by diagram, horizontal consistency and syntactic consistency being the two dimensions with the largest number of UML consistency rules gathered. The class diagram, with 38.60% of rules, is the most used UML diagram involved in the definitions of UML (horizontal and syntactic) consistency rules. It is followed, as expected by State Chart Diagram and Sequence Diagram respectively with 22.80% and 14.59% of the total of UML consistency rules presented in this work.

6.4 Combining RQ6 and RQ7

As shown earlier, most of the studies about UML consistency rules did not present any UML CASE tool to support those rules. In fact this aspect is confirmed by the fact that, the plain English is the language mostly used to specify UML consistency rules. There is still not a UML tool used by researchers that can be considered standard to execute UML consistency rules.

7. THREATS TO VALIDITY

The main threats to the validity of an SMS like ours are related to publication bias, selection bias, inaccuracy in data extraction, and misclassification [33].

As it is impossible to completely cover every publication written on our topic, we acknowledge that some relevant papers might not have been included. We used seven search engines to collect journals, conferences and workshops proceedings that are relevant to UML consistency rules; we did not consider grey literature [22] (e.g., PhD theses, books) or unpublished results (e.g., technical reports) because these might affect the validity of our results because they were not peer-reviewed. Selection bias refers to the distortion of a statistical analysis owing to the criteria used to select publications.

We attempted to solve this problem by defining our inclusion and exclusion criteria in order to gather the most relevant papers regarding UML consistency rules. To help ensure an unbiased selection process, we defined seven research questions in advance, organized the selection of articles and finally created and followed a multi-phase process to execute the SMS. We would also like to mention that during the data extraction process, there was the possibility of subjectivity when we decided what was (and what was not) related to our topic. This interpretation might have affected the results.

8. CONCLUSION

In recent years, a great number of UML consistency rules have been presented by researchers to fix inconsistencies between UML diagrams. However, no mapping study exists that summarizes these UML consistency rules since the majority of studies are informal literature surveys.

This work presented the results obtained after carrying out a Systematic Mapping Study (SMS) of literature with the aim to identify and evaluate the current state of the art about UML consistency rules. The SMS was carried out following well-known guidelines [22]. From an initial set of 1134 papers, a total of 95 primary studies were found by following a precise selection protocol driven by seven research questions. Primary studies were then classified according to several criteria, also derived from those research questions.

One important observation we made is that researchers typically define a number of similar UML consistency rules over and over again, which suggests there is a need for a documented list of accepted consistency rules. This is one of our next steps.

Based on our interpretation of the SMS carried out in this paper, we observe that (in no particular order of importance):

- There is little tool support for checking consistency rules, and there is no tool package that could check rules in various UML CASE tools, such as for instance Eclipse-based tools;
- The class diagram is the UML diagram mostly involved in UML consistency checking; it is followed in importance by the State Machine Diagram and the Sequence Diagram. This is not entirely surprising since these are likely the most used UML diagrams.
- A very few number of rules address the issue of vertical and evolution consistency. Even though the UML consistency topic is mature, it still needs to evolve to include definitions of UML consistency rules in all dimensions. Our SMS therefore shows areas where future work is needed.
- The UML version 2.0 is the most used standard to present UML consistency rules.
- There is no UML consistency rule suggested for Timing, Interaction Overview and Deployment Diagrams. Besides the class, sequence, and state machine diagrams, there is a need for much additional research on consistency rules involving all 14 UML diagrams.
- There is still a small number of evaluations of consistency rules reported in the literature.

These observations definitely call for future work.

We also consider additional work to consolidate further the list of consistency rules we have collected. First, as already mentioned earlier in this paper, we intend to compare the rules we collected with the well-formedness rules the UML standard already contains. Second, we believe we can collect additional consistency rules from other sources. For instance, textbooks on UML-based object-oriented software development (e.g., [34]) suggest, implicitly or explicitly, consistency rules. Also, we are aware of research activities where some UML diagrams are synthesized from other diagrams (e.g., [35]): in doing so the authors rely or want to enforce some consistency rules between diagrams.

9. ACKNOWLEDGMENTS

This research has been partly funded by a Discovery grant of the Natural Sciences and Engineering Research Council of Canada and the GEODAS-BC project (Ministerio de Economía y Competitividad and Fondo Europeo de Desarrollo Regional FEDER, TIN2012-37493-C03-01).

10. REFERENCES

[1] Mukerji, J., and Miller, J. 2003. *Overview and guide to OMG's architecture.* MDA Guide V1.0.1. Object Management Group. http://www.omg.org/mda/.

[2] Thomas, D. 2004. MDA: Revenge of the modelers or UML utopia? *IEEE Software*. 21, 3, 15–17.

[3] Lucas, F.J., Molina, F., and Toval, A. 2009. A systematic review of UML model consistency management. *Information and Software Technology*. 51, 12, 1631-1645.

[4] Genero, M., Fernández-Saez, A.M., Nelson, H.J., Poels, G., and Piattini, M. 2011. A Systematic Literature Review on the Quality of UML Models. *Journal of Database Management*. 22, 3 (July-September 2011), 46-70.

[5] Usman, M., Nadeem, A., Tai-hoon, K., and Eun-suk, C. 2008. A Survey of Consistency Checking Techniques for UML Models. In *Proceedings of the Advanced Software Engineering and Its Applications* (Hainan Island, China, December 13-15, 2008). ASEA 2008. IEEE Computer Society, 57-62.

[6] OMG. 2011. *OMG Unified Modeling LanguageTM.* Superstructure Version 2.4.1. Object Management Group.

[7] Ahmad, M.A., and Nadeem, A. 2010. Consistency checking of UML models using Description Logics: A critical review. In *Proceedings of the 6th International Conference on Emerging Technologies* (Islamabad, Pakistan, October 18-19, 2010). ICET '10. IEEE Computer Society, 310-315.

[8] Alanazi, M.N., and Gustafson, D.A. 2009. Super state analysis for UML state diagrams. In *Proceedings of the 2009 WRI World Congress on Computer Science and Information Engineering* (Los Angeles, California, USA, 31 March - 2 April, 2009). CSIE '09. EEE Computer Society, 560-565.

[9] Balaban, M., and Maraee, A. 2006. Consistency of UML class diagrams with hierarchy constraints. In *Proceedings of the 6th international Conference on Next Generation Information Technologies and Systems*, Etzion, O., Kuflik, T., and Motro, A., Ed. (Kibbutz Shefayim, Israel, July 4-6, 2006). NGITS'06. Springer-Verlag, 71-82.

[10] Chen, Z., and Motet, G. 2009. A language-theoretic view on guidelines and consistency rules of UML. In *Proceedings of the 5th European Conference*, Paige, R.F., Hartman, A., and Rensink, A., Ed. (Enschede, The Netherlands, June 23-26, 2009). ECMDA-FA '09. Springer-Verlag, 66-81.

[11] Labiche, Y. 2008. The UML is more than boxes and lines. In *Proceedings of the Workshops and Symposia at MODELS 2008*, Chaudron, M.R.V., Ed. (Toulouse, France, September 28 - October 3, 2008). MODELS '08. Springer-Verlag, 375-386.

[12] Lano, K. 2007. Formal specification using interaction diagrams. In *Proceedings of the 5th IEEE International Conference on Software Engineering and Formal Methods* (London; United Kingdom, September 10-14, 2007). SEFM '07. IEEE Computer Society, 293-301.

[13] Sapna, P.G., and Mohanty, H. 2007. Ensuring consistency in relational repository of UML models. In *Proceedings of the 10th International Conference on Information Technology* (Orissa, India, December 17-20, 2007). ICIT '07. IEEE Computer Society, 217-222.

[14] Petre, M. 2013. UML in practice. In *Proceedings of the 35th International Conference on Software Engineering* (San Francisco, CA, USA, May 18th-26th, 2013). ICSE '13. IEEE Press, 722-731.

[15] Ibrahim, N., Ibrahim, R., Saringat, M.Z., Mansor, D., and Herawan, T. 2011. Consistency rules between UML use case and activity diagrams using logical approach. *International Journal of Software Engineering and its Applications*. 5, 3, 119-134.

[16] Simmonds, J., Straeten, R.V., Jonkers, V., and Mens, T. 2004. Maintaining Consistency between UML Models using Description LogicZ. *RSTI – L'Object LMO'04*. 10, 2-3, 231-244.

[17] Muskens, J., Bril, R.J., and Chaudron, M.R.V. 2005. Generalizing Consistency Checking between Software Views. In *Proceedings of the 5th Working IEEE/IFIP Conference on Software Architecture* (Pittsburgh, Pennsylvania, USA, November 6-10, 2005). WICSA '05. IEEE Computer Society, 169-180.

[18] Huzar, Z., Kuzniarz, L., Reggio, G., and Sourrouille, J.L. 2005. Consistency problems in UML-based software development. In *Proceedings of the International Conference on UML Modeling Languages and Applications*, Nunes, N.J., Selic, B., da Silva, A.R., and Alvarez, A.T., Ed. (Lisbon, Portugal, 2005). UML'04. Springer-Verlag, 1-12.

[19] Spanoudakis, G., and Zisman, A. 2001. *Inconsistency management in software engineering: Survey and open research issues.* Chang, S.K. World Scientific Publishing Co. 329-380.

[20] Arksey, H., and O'Malley, L. 2005. Scoping studies: towards a methodological framework. *International Journal of Social Research Methodology*. 8, 1.

[21] Budgen, D., Turner, M., Brereton, P., and Kitchenham, B. 2008. Using mapping studies in software engineering. In *Proceedings of the Psychology of Programming Interest Group Workshop* (Lancaster University, 2008). PPIG '08.195–204.

[22] Kitchenham, B., and Charters, S. 2007. *Guidelines for performing systematic literature reviews in software engineering.* EBSE-2007-01. Keele University.

[23] Kalibatiene, D., Vasilecas, O., and Dubauskaite, R. 2013. Rule Based Approach for Ensuring Consistency in Different UML Models. In *Proceedings of the 6th SIGSAND/PLAIS EuroSymposium 2013* (Gdańsk, Poland, September 26, 2013). SIGSAND/PLAIS '03. Springer-Verlag, 1-16.

[24] Brereton, P., Kitchenham, B., Budgen, D., Turner, M., and Khalil, M. 2007. Lessons from applying the systematic literature review process within the software engineering domain. *Journal of Systems and Software*. 80, 4, 571–583.

[25] Torre, D., Labiche, Y., and Genero, M. 2014. *UML consistency rules: a systematic mapping study.* Tecnical Report. Carleton University. http://squall.sce.carleton.ca/mediawiki/index.php/Technical_reports.

[26] Mens, T., Van der Straeten, R., and Simmonds, J. 2005. A framework for managing consistency of evolving UML models. In *Software Evolution with UML and XML*, Yang, H., Ed. (Hershey 2005) IGI Publishing, 1-30.

[27] Engels, G., Hausmann, J.H., and Heckel, R. 2002. Testing the consistency of dynamic UML diagrams. In *Integrated Design and Process Technology* (Pasadena, California, June 24, 2002). IDPT '02.

[28] Engels, G., Küster, J.M., Heckel, R., and Groenewegen, L. 2001. A methodology for specifying and analyzing consistency of object-oriented behavioral models. *Sigsoft Software Engineering Notes*. 26, 5 (September 2001), 186-195.

[29] Wieringa, R., Maiden, N.A.M., Mead, N.R., and Rolland, C. 2006. Requirements engineering paper classification and evaluation criteria: a proposal and a discussion. *Requirements Eng.* 11, 1, 102–107.

[30] ProQuest. 2014. *RefWorks - A web-based bibliography and database manager.* http://www.refworks.com/.

[31] Dobing, B., and Parsons, J. 2006. How UML is used. *ACM* 49, 5 (May 2006), 109-113.

[32] Petersen, K., Feldt, R., Mujtaba, S., and Mattsson, M. 2008. Systematic mapping studies in software engineering. In *Proceedings of the 12th International Conference on Evaluation and Assessment in Software Engineering*, Visaggio, G., Baldassarre, M.T., Linkman, S., and Turner, M., Ed. (Bari, Italy, 2008). EASE '08. British Computer Society, 71–80.

[33] Sjoberg, D.I.K., Hannay, J.E., Hansen, O., By Kampenes, V., Karahasanovic, A., Liborg, N.-K., and Rekdal, A.C. 2005. A Survey of Controlled Experiments in Software Engineering. *IEEE Trans. Softw. Eng.* . 31, 9 (September 2005), 733-753.

[34] Bruegge, B., and Allen H., D. 2009 *Object-Oriented Software Engineering Using Uml, Patterns, and Java (3rd ed.).*

[35] Kang, S., Kim, H., Baik, J., Choi, H., and Keum, C. 2010. Transformation Rules for Synthesis of UML Activity Diagram from Scenario-Based Specification. In *Proceedings of the 34th Annual Computer Software and Applications Conference* (Seoul, South Korea, 19-23 July, 2010). COMPSAC '10. IEEE, 431-436.