5-15-2019

A REVIEW OF PROBLEMS AND CHALLENGES OF USING MULTIPLE CONCEPTUAL MODELS

Danilo Ong
Queensland University of Technology, danilojr.ong@connect.qut.edu.au

Mohammad Jabbari
Queensland University of Technology, m.jabbarisabegh@qut.edu.au

A REVIEW OF PROBLEMS AND CHALLENGES OF USING MULTIPLE CONCEPTUAL MODELS

Research paper

Danilo Jr. Ong, Queensland University of Technology, Brisbane, Australia, danilojr.ong@connect.qut.edu.au

Mohammad Jabbari, Queensland University of Technology, Brisbane, Australia, m.jabbarisabegh@qut.edu.au

Abstract

Conceptual models are used to visualise, envisage, and communicate the requirements, structure, and behaviour of a system. Particularly, during design and analysis phases, a model can serve as a tool to recognise different components, elements, actors, and relationships involved in a system. However, as a domain becomes complex, multiple models are needed to capture different aspects of a system. Further, each conceptual model develops using different grammars, methods, and tools. Therefore, using multiple models to represent a complex system may result in several problems, and challenges. This research aims to identify, analyse, and classify the different problems and issues encountered when using multiple models during information systems analysis and design through a structured literature review. Several problems are identified and are classified into seven main categories based on their common characteristics. The results of this study may serve as a baseline information for researchers in further understanding different modelling approaches and how multiple models can be used in harmony and reduce risks and issues. Also, the list of problems gathered will give insights to professionals on which issues they may possibly encounter when inter-relating various models.

Keywords: Conceptual Model, Multiple Models, Challenges, Problems, Systems Analysis and Design.

1 Introduction

System analysts and designers often develop graphical representations of information systems for analysing, understanding, and communicating different aspects of a system with different stakeholders. These graphical representations are often called conceptual models. Information systems (IS) professionals normally develop and use multiple and different types of conceptual models when they are dealing with a complex phenomena (Sabegh and Recker, 2017). They use different types of conceptual models because different models are developed for different purposes. For instance, data models are used to represent data structure of systems (Chen, 1976) and process models are developed to describe the behavioural aspects of systems, organisational activities, and business operations (Petri, 1962; Scheer and Nüttgens, 2000). Systems analysts also use multiple models because a single conceptual model is not sufficient to envisage all requirements of a system (Green et al., 2011).

Previous studies highlighted the use of multiple types of conceptual models in practice (Dobing and Parsons, 2008; Fettke and Loos, 2007). Studies indicate that IS professionals use multiple types of conceptual models for different purposes such as better and complete representation of a system under study (Kim, Hahn and Hahn, 2000; da Silva, 2015). Some studies investigated the use of multiple models, but most of these studies investigated how *users understand* multiple conceptual models or how using different models *improves domain understanding*. For example, Green et al. (2011) investigated the use of multiple modelling grammars[1] and how they can complement each other. Recker and Green (forthcoming) developed a theory to explain how model users read and understand multiple models. Kim, Hahn and Hahn (2000) and Siau and Lee (2004) analysed how using multiple types of models helps to improve understanding of a real-world domain and Sabegh and Recker (2017) analysed why practitioners use multiple models and how they select what types of models to use.

However, there is *a lack of studies on potential problems and challenges of using multiple models*. The plethora and availability of different types of models, methods to develop, and use of different models trigger its own challenges such as investment decisions on appropriate tools to support different types of models (Rosemann, 2006), possible inconsistency between developed models (Lucas, Molina and Toval, 2009), as well as challenges of integrating different models (Ahmad and Nadeem, 2010).

In this study, we therefore aimed to investigate "what are the possible challenges and problems of using different types of multiple conceptual models". To address this research question, a structured literature review of academic publications is conducted to assess the current research studies and potential problems and challenges they have identified, if any. Academic references such as journal articles and conference proceedings published since 2000 are considered in this study. Fifty-six academic papers from information systems and computer science databases were found to be relevant and are included in the literature review results and analysis. Findings of this study are significant to both information systems practitioners and academic researchers in numerous ways: (1) For information systems practitioners, particularly those that have intensive exposures to systems modelling and architecture for a longer period, this study serves as a guideline of collective understanding on the existing problems that plagued the simultaneous use of multiple models, hindering the effectivity of model representations. By knowing, and subsequently avoiding these issues, it will allow for modelling tasks to be more accurate and efficient and as well as reduce ambiguities and misunderstandings among different stakeholders and users on the interpretation of multiple models; (2) For academic researchers, the identified modelling problems serves as a baseline reference for future studies related to the improvement and seamless integration of systems modelling. Further, for researchers that are developing and designing tools and frameworks to support the integrated modelling methods, the problems identified in this paper are useful insights on the functional features the tool or framework should consider.

The remainder of this paper is structured as follows. Section 2 provides a brief background of the studies on conceptual modelling. Section 3 discusses the research methodology used in searching and identifying relevant literature publications through a structured review process. Section 4 presents the results gathered from the literature review and highlights the modeling problems. Section 5 provides an in-depth discussion of the findings, the significance of the findings to different audiences, and potential future studies to be done. Section 6 provides a summary of this literature review.

2 Background

Systems analysts and designers normally use more than one conceptual model in most system analysis and design tasks (Sabegh and Recker, 2017). There have been numerous research studies regarding the applicability of conceptual modelling grammars, methods, and tools to support systems analysis and design (Bera and Evermann, 2014; Wand and Weber, 2002; Burton-Jones et al., 2017). However, inconsistent with practice, most of these studies focused on evaluating and understanding of *a single model*. Practitioners' reports and previous studies (Fettke, 2009; Dobing and Parsons, 2008; da Silva, 2015) indicated that, apparently, using multiple models during systems analysis and design tasks is common practice. But, still, few studies only acknowledged the use of multiple models to represent a complex system (Ahmad and Nadeem, 2010; Schatz et al., 2003). Lillieskold (2003) identified how various conceptual models can be used to simplify the representation of a complex system. Similarly,

[1] Modelling grammars are set of constructs and rules how to use those constructs to represent real-world phenomena.

(Gustas, 2010) presented substantial insights how to use different models to depict the behaviour and structure of conceptual models, however, the study mainly focused on analysing and using models separately and no explicit multiple model usages were illustrated. Same is the case with the research of Park and Bae (2011) where multiple UML models were individually analysed and compared amongst each other. Previous studies also provide methods, such as formalised approaches, to improve conceptual modelling and information transformation (Labiak et al., 2011), developed tool-support to improve the design veracity, consistency, and information synchronicity (Pancham and Millham, 2015).

While, these studies provided beneficial avenues in improving the integrity and reliability of various modelling techniques, however, these studies mainly focused on understanding models or improving the quality of conceptual models. While, empirical evidence suggest that the use of multiple models introduces its own challenges. For example, Spanoudakis and Zisman (2001) and (Lucas, Molina and Toval, 2009) reviewed the reported inconsistency problems between Unified Modelling Language (UML) models (Lucas, Molina and Toval, 2009; Spanoudakis and Zisman, 2001). There is lack of studies that incorporate the potential issues and challenges that may arise in using multiple models (Sabegh and Recker, 2017) and there is no insight in the academic literature how those problems affect the outcome of conceptual modelling activities or influence on effectivity of a tool.

Given the importance of conceptual modelling in systems analysis and design (Schmidt, 2006), understanding the challenges and problems of using multiple models will provide empirical contributions for practitioners to avoid potential drawbacks and may serve as a baseline information for researchers in further analysing and understanding different modelling approaches and how multiple models can be used in harmony to reduce risks and problems.

Unlike the previous studies by Spanoudakis and Zisman (2001) and Lucas, Molina and Toval (2009), this study doesn't entirely focus on UML models however aims to answer a broader question of "what are the possible challenges and problems of using different types of multiple conceptual models?" and present the underlying issues encountered in using multiple conceptual models and grammars[2] to represent different aspects, different operations, and components of a complex information system. In the next section, we explain the research method we have used to address our research question.

3 Research Method

We have selected a structured literature review method to address the research question of this study. In conducting the structured literature review, we mainly followed the approach proposed by Paré et al. (2015) and Webster and Watson (2002). The descriptive review approach as presented by Paré et al. (2015) was followed, and to ensure that all academic publications identified had been carefully evaluated, the steps proposed by Webster and Watson (2002) were pursued. To achieve the objectives of this study, first, we formalised search keywords and a search strategy based on our research question. Second, we conducted searches in relevant databases to retrieve relevant papers. Third, we analysed the relevancy of the papers based on our inclusion and exclusion criteria. Fourth, we selected relevant papers for further evaluation. Fifth, we evaluated relevant papers and identified potential problems and challenges. Finally, we clustered the identified challenges and problems based on the similarities and characteristics of challenges and problems mentioned in the papers. In order to be consistent with the previous reviews in the field, we also followed the procedure suggested by Lucas, Molina and Toval (2009) where ever it was applicable. Figure 1 represents an overview of the literature review process.

[2] In practice, modelling grammars and methods can also be attributed as languages and approaches.

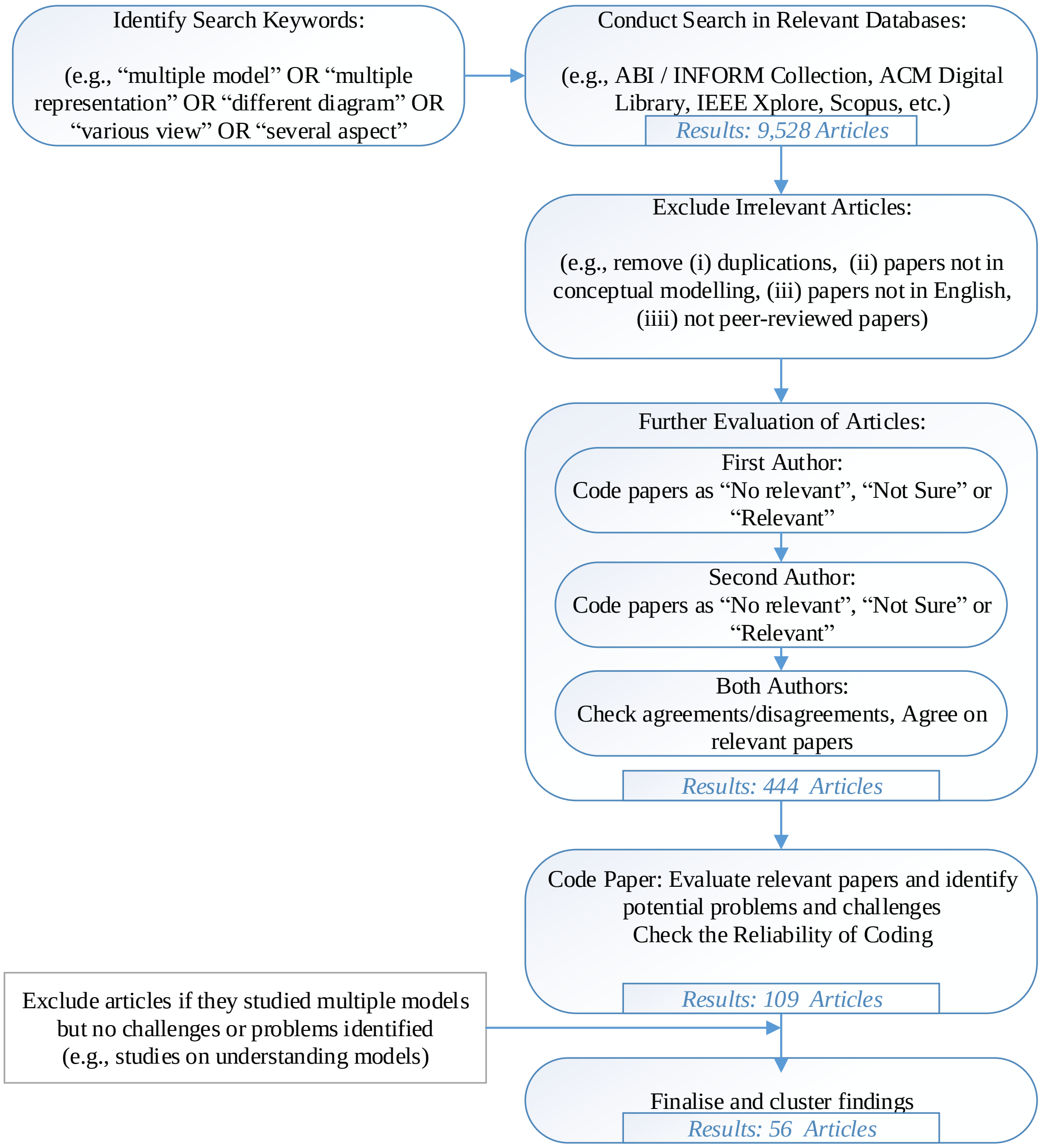
Identify Search Keywords:
(e.g., "multiple model" OR "multiple representation" OR "different diagram" OR "various view" OR "several aspect"
Conduct Search in Relevant Databases:
(e.g., ABI / INFORM Collection, ACM Digital Library, IEEE Xplore, Scopus, etc.)
Results: 9,528 Articles
Exclude Irrelevant Articles:
(e.g., remove (i) duplications, (ii) papers not in conceptual modelling, (iii) papers not in English, (iiii) not peer-reviewed papers)
Further Evaluation of Articles:
First Author:
Code papers as "No relevant", "Not Sure" or "Relevant"
Second Author:
Code papers as "No relevant", "Not Sure" or "Relevant"
Both Authors:
Check agreements/disagreements, Agree on relevant papers
Results: 444 Articles
Code Paper: Evaluate relevant papers and identify potential problems and challenges
Check the Reliability of Coding
Results: 109 Articles
Exclude articles if they studied multiple models but no challenges or problems identified
(e.g., studies on understanding models)
Finalise and cluster findings
Results: 56 Articles

Figure 1. Overview of the Literature Review Process

3.1 Search Queries

To identify the search keywords and formulate search criteria and search queries, we conducted an environmental scan of the closely-related papers, keywords, and terminologies. In instances where the search query did not fit the allowed format in the search database, corresponding changes were made. Table 1 displays the search query that was implemented and utilised.

("multiple model" OR "different model" OR "various model" OR "several model" OR "multiple representation" OR "different representation" OR "various representation" OR "several representation" OR "multiple diagram" OR "different diagram" OR "various diagram" OR "several diagram" OR "multiple view" OR "different view" OR "various view" OR "several view" OR "multiple aspect" OR "different aspect" OR "various aspect" OR "several aspect") AND ("systems analysis and design" OR "system development" OR "systems development")

Table 1. Search Query

3.2 Source Selection

To ensure that highly-relevant academic publications were included in this study, a search was conducted for peer-reviewed journal and conference papers published in the Information Systems, Information Technology, and Computer Science databases. Eleven relevant databases were selected: ABI / INFORM Collection, ACM Digital Library, AISeL, INFORMS PubsOnline, IEEE Xplore, JSTOR, ScienceDirect, Scopus, SpringerLink, Web of Science, and Wiley Online. Before any keyword searches were conducted, the search structure of each database was also assessed with respect to the limited number of search keywords, Boolean operators, and section filters allowed.

3.3 Synthesizing and Analysing Results

3.3.1 Inclusion and Exclusion Criteria

To exhibit uniformity and ensure quality control of search results across various databases, an inclusion and exclusion criteria, in accordance with the scope and limitations, were applied as follows:

Peer-reviewed journal and conference papers published in the English Language since 2000 were selected. All non-academic papers, commentaries, editorial boards were excluded. In addition, duplicated publications across different databases were excluded to avoid repetition.

3.3.2 Review Process

With the removal of irrelevant and duplicated papers the remaining quantity of papers remained at 9,528. Table 2 presents the distribution of these papers across the selected databases.

Database Name	**No. of Papers**
ABI / INFORM	298
ACM	30
AISeL	98
INFORMS PubsOnline	34
IEEE Xplore	1979
JSTOR	105
Web of Science	207
ScienceDirect	5663
Scopus	50
SpringerLink	988
Wiley Online	76
Total	**9528**

Table 2. Search Database

To ensure that all remaining papers were relevant, the first author reviewed all papers based on their title and abstract. Through this process, the first author coded papers as *"Not Relevant"* if the papers were not focused on information systems or computer science and did not discuss any notions of modelling. The papers are coded as *"Not Sure"* if the papers were in scope, but the proximity of the findings and discussions included in the abstract can't be formally verified. The papers coded as *"Relevant"* were in the scope and consisted of an abstract that clearly discussed the use of multiple models.

444 papers were coded as either *"Not Sure"* or *"Relevant"* papers. To ensure the reliability of this round of the review, the second author reviewed all the 444 papers following the same procedure. Then, the two authors discussed their coding and disagreements. Finally, 109 papers were coded as relevant papers which both authors agreed on.

The 109 relevant papers went through a full text detailed review. The first author used a predefined coding scheme and coded the papers as 1) if they discussed the challenges of using multiple models, 2) types of issues and challenges identified in the paper, 3) any possible solutions discussed by each publication to address the identified challenges 4) types of paper (e.g. empirical or theoretical) 5) type of supports provided, if any.

Out of 109 papers, 53 papers were identified of which these papers either analysed or compared various models or have discussed the use of multiple models but did not point out possible problems and challenges. Therefore, these papers were excluded from further discussions.

The second author then individually coded a random sample of 12 papers out of 56 remaining papers. Both authors then met and discussed the disagreements and raised any concerns. After review, the first author recoded the 56 papers to implement the discussed solutions.

4 Findings

Once we identified the problems and challenges mentioned in the reviewed papers, we clustered them into groups based on their similarities and characteristics. The identified challenges and problems fall into seven main groups. We will explain these groups in the next section. Most of the papers reviewed in this study have a main focal point on UML models. This is not surprising, because UML provides a wide range of different diagrams that can be used for different purposes (Booch, Rumbaugh and Jacobson, 2005). The main theme emerging from these studies indicate that using multiple diagrams have helped deliver various information, improved understanding of the functions and features of domains, as well as enhanced the collaboration among different types of users from different perspectives (Sabegh and Recker, 2017).

4.1 Redundancy and Overlap

Information redundancy or overlap occurs when one particulate construct of a domain is presented in multiple models. In other words, repeated information exists between multiple models in which this overlapping information represent a same construct in the real-word phenomena. One particular example could be using an activity diagram together with a BPMN diagram in which both models represent a same set of activities or transformations of a real-world phenomenon. It may happen because of organisational conventions or agreement on requirements (Sabegh and Recker, 2017, 69). Similarly, redundancy occurs when several modelling constructs or model elements refer to the same classes, things or properties in the problem domain (Opdahl and Henderson-Sellers, 2005; Wand and Weber, 1995). For example, consider if an element of a problem domain is presented as a use case in a use case diagram, as a node in deployment diagram and as an object in an object diagram. Opdahl and Henderson-Sellers (2005) refers to this situation as referential redundancy and argues that referential redundancy compromises the quality of conceptual modelling as it may influence the alignment of information presented in different models and the reuse of model content between different models.

In sum, when a domain is represented through different models, some aspects of the envisaged system can be featured in multiple models (Jabbari Sabegh, Recker and Green, 2016; Recker and Green, forthcoming). As we will explain in the next section, such overlapping information may cause inconsistency between models, but it does not necessarily equate to inconsistency between models. There could be overlapping but consistent information represented through multiple models.

Papers	Problems and challenges
(Schatz et al., 2003); (El Asri, Nassar and Kriouile, 2009); (Andrade et al., 2004); (Ahmad and Nadeem, 2010); (Yan, Cheng and Chai, 2015); (Bruckner, Repa and Chlapek, 2014); (Gustas and Gustiené, 2017); (González et al., 2011); (Fenning, Dogan and Phalp, 2014); (Opdahl and Henderson-Sellers, 2005); (Dobing and Parsons, 2005); (Gustas and Gustiene, 2011); (Bjerkander and Kobryn, 2003); (Lange, Chaudron and Muskens, 2006); (Attiogbe, Poizat and	Redundancy & Overlap

Salaun, 2007); (Sewchurran and Petkov, 2007); (Große-Rhode, 2004); ; (Dobing and Parsons, 2008); (Lee and Ling, 2003)	
(Zur Muehlen, Indulska and Kittel, 2008); (Liu et al., 2004); (El Asri, Nassar and Kriouile, 2009); (Andrade et al., 2004); (Ahmad and Nadeem, 2010); (Venkatesh, Bhaduri and Joseph, 2001); (Kramer, 2015); (Yan, Cheng and Chai, 2015); (Bowman et al., 2002); (Lucas, Molina and Toval, 2009); (Sewchurran and Petkov, 2007); (Bashir et al., 2016)	Inconsistency
(Ahmad and Nadeem, 2010); (Basile, Chiacchio and Del Grosso, 2008); (Bolloju, 2006); (Moody, 2003); (Gustas, 2010); (Keet and Fillottrani, 2015); (Opdahl and Henderson-Sellers, 2005); (Anda and Sjøberg, 2005); (Gustas and Gustiene, 2011); (Lange, Chaudron and Muskens, 2006); (Boufares and Bennaceur, 2004); (Kim, Hahn and Hahn, 2000); (Bashir et al., 2016); (Torres, Galante and Pimenta, 2011)	Dependencies
(Schatz et al., 2003); (Malik, Truscan and Lilius, 2010); (Xu et al., 2017); (Attiogbe, Poizat and Salaun, 2007); (Kim, Hahn and Hahn, 2000); (Dobing and Parsons, 2008); (Heuer et al., 2013)	Complexity
(Nassar, 2003); (Cohen and Soffer, 2007); (Torngren, Chen and Crnkovic, 2005); (Moody, 2003); (Kramer, 2015); (Bruckner, Repa and Chlapek, 2014); (Gustas and Gustiené, 2017); (Bolloju, Schneider and Sugumaran, 2012); (George et al., 2012); (Bjerkander and Kobryn, 2003); (Lange, Chaudron and Muskens, 2006); (Palpanas et al., 2007); (Torres, Galante and Pimenta, 2011)	Maintainability & Traceability
(Malik, Truscan and Lilius, 2010); (Cohen and Soffer, 2007); (Hamou-Lhadj, Gherbi and Nandigam, 2009); (Torngren, Chen and Crnkovic, 2005); (Koivulahti-Ojala and Kakola, 2010); (Kernschmidt et al., 2013); (Skersys and Gudas, 2006); (Opdahl, 2010); (Zikra, Stirna and Zdravkovic, 2011); (Keet and Fillottrani, 2015); (Bolloju, Schneider and Sugumaran, 2012); (Gustas and Gustiene, 2011); (Lange, Chaudron and Muskens, 2006); (Steele, Son and Wysk, 2001); (Green et al., 2011); (Fuentes-Fernández, Pavón and Garijo, 2012); (Heuer et al., 2013); (Torres, Galante and Pimenta, 2011)	Language Variations & Tool Support
(Hamou-Lhadj, Gherbi and Nandigam, 2009); (Torngren, Chen and Crnkovic, 2005); (Adamu and Zainon, 2017); (Siau and Tian, 2005); (Tan, Alter and Siau, 2008); (Moody, 2003); (Keet and Fillottrani, 2015); (Bolloju, Schneider and Sugumaran, 2012); (Dobing and Parsons, 2008)	User Comprehension

Table 3. List of papers and identified challenges and problems[3]

4.2 Inconsistency

Inconsistency refers to a state in which two or more elements, which overlap in different models of the same system, are dissimilar or incompatible (Spanoudakis and Zisman, 2001). Inconsistency may also arise between different models with refined description of the system (Lucas, Molina and Toval, 2009; Huzar et al., 2004) due to different models being developed and used during different phases or iteration of IS development lifecycle. Similar to Lucas, Molina and Toval (2009), semantic, syntactic, inter-model and intra-model inconsistency problems were identified.

Semantic inconsistency happens when the modelling context is not homogenous with its intended requirement stipulations (Yan, Cheng and Chai, 2015). For example, when a State Machine Diagram (SM) describes the behaviour of a class and at the same time relates to a Sequence Diagram (SD) to depict system interactions, the sequence of exchanged messages should coordinate to the order of triggers and effects presented in its state machine diagram to properly describe the behaviour associated in the class diagram (Bashir et al., 2016).

[3] Note that some papers identified more than one problem or challenge. These papers were coded in each group, respectively.

Syntactic inconsistency arises when there is a defect in the modelling construct and convention such as naming, operation messages, and interface labels (Kenzi et al., 2009). Examples include messages referencing null operations, incomplete connector specifications, and a model referencing to a null class from another diagram (Ahmad and Nadeem, 2010). Problems with syntactic inconsistency can result in irregular model-to-model and model-to-code configurations (Bashir et al., 2016).

Intra-model inconsistency, also referred to as horizontal inconsistency, happens among different types of models created on similar abstraction levels (Lucas, Molina and Toval, 2009). And finally, inter-model, referred as vertical consistency, occurs among similar type of models created on various abstraction levels that incorporates refinement points (Lucas, Molina and Toval, 2009).

4.3 Dependencies

When different models are used to represent the same domain, relevant elements of the domain scatter across different models. Dependencies between models becomes critical when a task requires to integrate information presented across different models and understand the associations and relationships between models (Lange, Chaudron and Muskens, 2006), or when changes in one model requires to make additional changes in another model to keep the consistency; see the previous section. It becomes more problematic when there is no information about what consequence a change in a model may cause in another model, which (Torres, Galante and Pimenta, 2011) refers to as a problem of hidden dependencies between models.

An additional problem that may arise due to the dependencies between models is the issue of information translation when extracting and transitioning requirements from functional models to operational models. For instance, where a use case diagram is translated into a class diagram and subsequently then causes for logic inaccuracies to occur because of the context differences of each model (Anda and Sjøberg, 2005; Boufares and Bennaceur, 2004).

Relationship mismatch happens when a model is incomplete or if there is disproportion in the abstraction level of two dependent models (Gustas, 2010). A single conceptual model cannot provide a complete representation of a domain. Models are always partial and incomplete. For example, statistic aspects of a domain can be presented by a class diagram. But, not all of its behaviour can be constrained by the class diagram. However, it may happen that a single model may not provide enough details of system aspects and appropriate associations and therefore relationships cannot be modelled correctly to map two dependent models (Moody, 2003).

4.4 Complexity

Often, conceptual models become too complex to manage. For example, in Sequence Diagram and Entity Relationship Diagram (ERD) where their modelling elements easily get too big during complex representations and result in illegible textual information and inadvertent connector intersections (Moody, 2003; Cohen and Soffer, 2007).

Complexity can be viewed in different ways; symbols and elements the models used are hard to digest, understanding the process through multiple integrated diagrams, or the lack of clarity that hampers the reflection of accurate information. The studies mainly discussed the complexity in relation to the effects on ambiguity and integration.

Ambiguity refers to the level of distinct information included in the models and the lacking substantial clarity on how each model interact with each other and how this result in misunderstandings. However, ambiguity may also contradict with the effect of readability (Heuer et al., 2013) wherein presenting more formal, intricate, and detailed information result to in a higher level of understanding but the readability of these diagrams, due to excessive element representations, is decreased.

Integration refers to assimilating different models representing different aspects of a domain. While, having multiple models is preferred as they assist information clarity, they also create integration complexities (Attiogbe, Poizat and Salaun, 2007). Findings indicate that, as the number of different models increases, a model user can only see a small portion of the total system at a time. This is a result of limited human working memory. Therefore, availability of multiple and different types of models complicates perceptual and conceptual processes in diagrammatic reasoning (Jinwoo et al., 2000).

4.5 Maintainability and Traceability

Multiple models and diagrams are needed to capture different segments of a complex system and incorporate several models with various structural elements, context, and notations make the modelling view larger. The ability of the models to be flexible with integrations and reprocessed efficiently for manageability is important.

Traceability is a critical aspect especially during systems analysis, where accountability and visibility of how information is passed on from one model to another need to be highlighted. For example, during troubleshooting issues where multiple models are used to trace back system events and logs, it is a challenge to rollback some implemented changes as there are other affected models that weren’t explicitly identified either by the user who created it or the functional limitations of the modelling language (Kramer, 2015). Furthermore, when an activity is linked with a BPMN diagram where an actor represents the doer for the tasks and actions, BPMN swimlanes are unable to automatically comprehend with activity diagram changes and processes are always needed to be recreated (Bruckner, Repa and Chlapek, 2014). Another example is when informal labels such as comments and annotations that are diagram-specific are sometimes not carried across the other associated diagrams (Bjerkander and Kobryn, 2003).

4.6 Language Variations and Tool Supports

Each modelling approach adapts their own grammar and syntax and follows their own construction guidelines and relationship constraints. Having these aspects during multiple model integrations cause problems as there is no concrete and homogeneous approach to unify these different grammars into a common platform of understanding. The bulk of the academic publications identified reflected ambiguity concerns since each model based their context and interpretation to their own set of rules. Other problems also include the lack of enough constructs that cut across multiple models and as well as conflicts with naming conventions and explicit definitions of roles and actors.

Language ambiguity identified as one of the main issues in conceptual modelling practices in terms of the depth and specificity needed, as a concern during multiple modelling integrations. Comparing the component-based (CBD) with model-based development (MBD), for example, the representation and understanding of information are done differently where CBD makes use of explicit execution platforms while MBD presents the system through visualisations of the system context (Torngren, Chen and Crnkovic, 2005). Further, the linguistics and breadth of vocabulary used by different stakeholders affect the interpretation of requirements and is usually observed during the requirements gathering where narratives are modelled into use case diagrams (Zikra, Stirna and Zdravkovic, 2011). As such, defining different criteria on how ambiguity is understood by various users are crucial.

Another essential component to support the effective delivery, management, and maintainability of multiple models is having a reliable tool. Computer-aided tools can store libraries of elements, symbols, and grammars of different modelling languages which help in creating diagrams and aid in assigning relationship cardinalities and associations. Some sophisticated tools even go further by providing automation and execution rules to instantly reflect and coordinate updates across different models.

Although there have been numerous studies regarding tool compatibility and suitability depending on the type of modelling language and functional requirements, there is still no refined tool that unifies all languages and addresses the challenges of using multiple models. For example, there is no tool with the capability of integrating different modelling grammars into a unified and aligned notation structure. For example, activity diagrams’ allowable variability technique doesn’t coincide well with the formalised approach of Petri Nets and no particular tool has been developed yet to fix this disjointed reflection of information (Heuer et al., 2013)

Koivulahti-Ojala and Kakola (2010) highlighted the lack of managing different update versions when multiple and parallel updates are made across different models that represent a same system. Utilising a central information storage has been implemented as an alternative approach, however, it did not capture the changes made by different users well, and at times, it hindered simultaneous collaborations and updates. Torngren, Chen and Crnkovic (2005) indicate that there is a lack of available supporting tools on the management of multiple models that allows structure comprehensibility and easy navigation.

There have already been numerous studies on modelling support tools but most of them only address specific problem types. In the research of Boufares and Bennaceur (2004) that examined ER schema consistencies, they utilised the Computer Aided Software Engineering (CASE) Tool during database design and as well as the Fourier-Motzkin Elimination method for schema analysis and validation. In Lange, Chaudron and Muskens (2006) also examined model defects and consistencies using a tool called Metric View to determine class dynamicity and the functional relationships of classes with use cases. The CASE Tool was also used by Lucas, Molina and Toval (2009) coupled with Eclipse Modeling Framework (EMF) development platform that enabled integration of various models and streamlining of model logic for easy verification and management. Further, in 2012, a study by Bolloju, Schneider and Sugumaran (2012) integrated the Object Model Analyzer (OMA) extension to the CASE tool to assist in uncovering patterns through Natural Language Processing (NLP) and reduce invalid elements and semantic inconsistencies. With all these tools developed and studied, however, a significant number of academic publications from 2012 onwards still tackled issues with multiple modelling and the lack of a sophisticated and unified tool that can cover most of these problems.

4.7 User Comprehension

Any conceptual modelling work requires model users' comprehension in one way or another. Different stakeholders with different background and skills may be involved in conceptual modelling practices. Findings of reviewed papers indicate that different shareholders have different preferences on what types of models they would like to use. Further, as different users have various level of knowledge and understanding with respect to information system modelling, oftentimes there is a disconnect on how model elements and symbols are denoted by different users (Siau and Tian, 2005).

Moreover, different users have different cognitive capability how to comprehend the models presented to them. Past user experiences and exposures to various modelling languages play a role in how quickly a user can grasp the modelling concept. Although, regardless of prior knowledge of users with system modelling, it has been commonly noted that as diagrams get too large, problems with ambiguity, cognitive capacity, and context validation will persist (Bolloju, Schneider and Sugumaran, 2012; Moody, 2003; Siau and Tian, 2005).

5 Discussion

While systems analysts and designers often use multiple and different types of conceptual models during their design and analysis tasks, except few studies on the inconsistency between models, there were lack of knowledge on possible challenges and problems of using multiple types of models. This study, therefore, reviewed previous studies on conceptual modelling, to identify what are the common problems and challenges reported in the scientific studies. To achieve the objective of this study, relevant papers are analysed and selected 56 studies were evaluated and explored. Through the literature review, seven main problems and challenges of using multiple models were identified from previous studies.

Granular root causes of the problems were identified under each category to further substantiate and support when, where, and how the problems came up. Although there were common terminologies across different problems, such as overlap and inconsistency between models, they were analysed according to the context of what caused the problem and why it occurred.

The analysis indicated that, existence of redundancy and overlap between models may cause other problems such as inconsistency, dependency, or traceability. It was identified that model redundancies could be a result of inaccurate dependency representations. The idea about cardinality constraints, multiplicities, or relationship associations were some of the examples shown. If relationship associations were linked and interpreted incorrectly, duplicate process executions would have occurred and in return resulted in different types of redundancy concerns such as repetition of information or declaration of similar objects in various modelling languages. Problems concerning Traceability can also be associated with both Redundancy and Dependency where the lack of established verification and trace methods for accountability results into uncoordinated information and duplication of data. It was also found that Complexity and User Comprehension problem categories were also dependent on each other. In User Comprehension, the user's preference and adaptability were affected by how they perceived diagram notations and symbols and the level of user cognition and professional exposure they've had with multiple modelling techniques. These, in turn, raised issues with complexity on how to deal with information specificity and ambiguity while keeping the models easily comprehensible, and as to how different models were preferred and chosen by different user types where they can grasp and appreciate the context without difficulties. These are some of the reasons why the problem types were broken down into definite sections to capture more issues pertaining to the specific problem categories.

Most of the academic papers examined also highlighted the lack of an efficient supporting tool to reduce and regulate these problems. Many tools have already been developed such as CASE, Object Model Analyzer, and Natural Language Processor, to help users design and analyse models, however, issues with multiple model integration still exist. Most of the tools developed only targeted specific problems for improvement but none have captured the breadth of multiple modelling concerns that enabled seamless, efficient, and reliable alignment of information. Although, in the case of Bolloju, Schneider and Sugumaran (2012), they utilised the existing ArgoUML tool, based from CASE, and adopted an extension to supplement the inherent features and provided further functionalities.

Since each modelling technique is used according to its purpose, combining them to describe a complex system requires multiple integration points and refinements. UML Use Case Diagrams and Scenarios are actor-centric and define the sets of activities an actor performs in relation to the tasks and resources associated with it. Class diagrams and activity diagrams, on the other hand, model both the actor and other relevant business objects to define their relationships and associations. Business Process Modeling (BPM) also models the tasks an actor does, however, it allows representation of multiple actors using swimlanes to perform process analysis and improvements. Meanwhile, Petri Nets further enables process execution functions for capacity planning and resource simulations. As such, transitioning use case models into a class diagram needs information accuracy and explicit definition of their association constraints. Same is the case with integrating use case models with BPM where swimlanes constantly needs refinement to align the relationships among different actors. Furthermore, integrating activity diagram to Petri Nets need formalisation to semantically and structurally comply with PN's verification requirements before the execution and analysis functions can be maximised.

The strengths of this literature review include the establishment of a rigid search query that used keyword combinations highly-relevant to the research problem, and all search databases were finalised before any search procedures were made. By having the same search query applied across different databases, most of the retrieved academic publications included modelling techniques that were used and implemented in different IS. The four-stage review methodology of evaluating the academic publications ensured that all the papers assessed in the study have used several models and diagrams to design and analyse a complex IS. Publication types were also highly regarded where only academic references such as journal articles and conference proceedings were considered, and further, the publication date range used also warranted that the problems found are current and relevant. By presenting the results in a concept matrix, it easily reflected the modelling problems identified in each academic paper that was analysed.

One of the limitations of this study is the heavy reliance on pre-existing research studies where some of these issues may or may not have already been solved in real practice. Also, due to time constraints, 109 academic publications rated as relevant were read and evaluated in full text. For instance, with the problems related to tool support, there may be other papers apart from the 109 identified may have also presented more useful and reliable new tools and frameworks. In addition, this study did not assess in detail how every problem relates and deeply affects each of the challenges identified. As a future research, we will study the correlation between identified problems and challenges, and we will analyse potential positive or negative relationships between them.

The results of this study provide contributions to practitioners and academics. Professional practitioners that are heavily exposed to systems modelling and analysis can use the problems identified in this study to anticipate the challenges they may encounter, and as a result, will make their design and analysis activities more efficient and accurate. Additionally, it gives them further insight on which modelling techniques are compatible to use in terms of the complexity of integration, language, dependencies, and comprehension. By knowing which problems affect the different phases of using multiple models, it allows them to evaluate what tools or plug-ins are further needed to address the modelling deficiencies and constraints in terms of grammar, syntax, semantics, and viewpoints analysis. For the academic researchers, the findings will serve as their initial guideline in understanding the possible implications and challenges with multiple models. For those further studies that require the analysis of existing modelling techniques and development of new modelling grammars, the problems identified in this study serve as their baseline considerations in coming up with new and improved construct and rules and methods.

Conducting the structured literature review to assess the current state of research done with the impacts of using multiple models in complex information systems is just the first step of understanding the research problem. It is recommended to further conduct studies in applying how each problem relate to each modelling language and what possible solutions and workarounds are available. To further validate the theoretical examinations of the problems, it will be best to conduct surveys, interviews, or action research with real-world industry practitioners that are exposed to systems modelling and use the problems presented in this research paper as baseline criteria. It is important to know if these problems are frequently encountered in real life and note the different solutions each practitioner did to solve the modelling problems they came across.

6 Conclusion

This research study conducted a structured literature review to determine the different problems, challenges, and issues when using multiple models and diagrams during the design and analysis of information systems. Models have been used to represent the state and process behaviours of a system and the more complex the system becomes, the more it demands for multiple models to be simultaneously utilised and integrated among each other. A single model cannot fully suffice system representations as they often only represent either the structure, behaviour, or executory aspects of a system. To achieve consistency and rigor in identifying the academic publications included in this study, search queries were first finalised, and all information systems related search databases were exhausted before any review and analysis are made.

There were different modelling problems identified, and they were categorised according to the common characteristics they reflected. As a result, seven main challenges and problems are identified, including overlap and redundancy, inconsistencies, dependency, complexity, maintainability and traceability, language variations and tools support and users' comprehension. These results may serve as a baseline information for researchers in further under-standing different modelling approaches and how multiple models can be used in harmony and reduce risks and issues. Also, the list of problems gathered may give insights to professionals on which issues they may possibly encounter when interrelating various models.

References

Adamu, Alhassan and Wan Mohd Nazmee Wan Zainon. 2017. "Multiview Similarity Assessment Technique of UML Diagrams." *Procedia Computer Science* 124: 311-318.

Ahmad, Mahreen Aziz and Aamer Nadeem. 2010. "Consistency checking of UML models using Description Logics: A critical review." In *2010 6th International Conference on Emerging Technologies (ICET)*, edited, 310-315: IEEE.

Anda, Bente and Dag I. K. Sjøberg. 2005. "Investigating the Role of Use Cases in the Construction of Class Diagrams." *Empirical Software Engineering* 10 (3): 285-309. doi: 10.1007/s10664-005-1289-3.

Andrade, Javier, Juan Ares, Rafael García, Juan Pazos, Santiago Rodríguez and Andrés Silva. 2004. "A methodological framework for viewpoint-oriented conceptual modeling." *IEEE Transactions on Software Engineering* 30 (5): 282-294.

Attiogbe, C. , P. Poizat and G. Salaun. 2007. "A Formal and Tool-Equipped Approach for the Integration of State Diagrams and Formal Datatypes." *IEEE Transactions on Software Engineering* 33 (3): 157-170. doi: 10.1109/TSE.2007.21.

Bashir, Raja Sehrab, Sai Peck Lee, Saif Ur Rehman Khan, Victor Chang and Shahid Farid. 2016. "UML models consistency management: Guidelines for software quality manager." *International Journal of Information Management* 36 (6, Part A): 883-899. doi: https://doi.org/10.1016/j.ijinfomgt.2016.05.024.

Basile, Francesco, Pasquale Chiacchio and Domenico Del Grosso. 2008. "Modelling automation systems by UML and Petri Nets." In *2008 9th International Workshop on Discrete Event Systems*, edited, 308-313: IEEE.

Bera, Palash and Joerg Evermann. 2014. "Guidelines for using UML association classes and their effect on domain understanding in requirements engineering." *Requirements Engineering* 19 (1): 63-80.

Bjerkander, M. and C. Kobryn. 2003. "Architecting systems with UML 2.0." *IEEE Software* 20 (4): 57-61.

Bolloju, Narasimha. 2006. "Exploring Quality Dependencies among UML Artifacts Developed by Novice Systems Analysts." *AMCIS 2006 Proceedings*: 472.

Bolloju, Narasimha, Christoph Schneider and Vijayan Sugumaran. 2012. "A knowledge-based system for improving the consistency between object models and use case narratives." *Expert Systems with Applications* 39 (10): 9398-9410. doi: https://doi.org/10.1016/j.eswa.2012.02.126.

Booch, Grady, James Rumbaugh and Ivar Jacobson. 2005. *The Unified Modeling Language User Guide*. 2nd ed. Upper Saddle River, NJ: Addison Wesley Longman Inc.

Boufares, Faouzi and Hachemi Bennaceur. 2004. "Consistency problems in ER-schemas for database systems." *Information Sciences* 163 (4): 263-274. doi: https://doi.org/10.1016/j.ins.2003.06.015.

Bowman, H., M.W.A. Steen, E.A. Boiten and J. Derrick. 2002. "A Formal Framework for Viewpoint Consistency." *Formal Methods in System Design* 21 (2): 111-166. doi: 10.1023/a:1016000201864.

Bruckner, Tomas, Vaclav Repa and Dusan Chlapek. 2014. "Consistency Issues in Large Business Process Model Environment, a Case Study." In *International Conference on Business Informatics Research*, edited, 172-186: Springer.

Burton-Jones, Andrew, Jan Recker, Marta Indulska, Peter Green and Ron Weber. 2017. "Assessing Representation Theory with a Framework for Pursuing Success and Failure." *MIS Quarterly* 41 (4): 1307-1333.

Chen, Peter Pin-Shan. 1976. "The entity-relationship model—toward a unified view of data." *ACM Transactions on Database Systems (TODS)* 1 (1): 9-36.

Cohen, Shalom and Avi Soffer. 2007. "Scrutinizing UML and OPM modeling capabilities with respect to systems engineering." In *2007 International Conference on Systems Engineering and Modeling*, edited, 93-101: IEEE.

da Silva, Alberto Rodrigues. 2015. "Model-driven engineering: A survey supported by the unified conceptual model." *Computer Languages, Systems & Structures* 43: 139-155.

Dobing, Brian and Jeffrey Parsons. 2005. "Current Practices in the Use of UML, *Berlin, Heidelberg*, edited, 2-11: Springer Berlin Heidelberg.

Dobing, Brian and Jeffrey Parsons. 2008. "Dimensions of UML diagram use: a survey of practitioners." *Journal of Database Management* 19 (1): 1-18.

El Asri, Bouchra, Mahmoud Nassar and Abdelaziz Kriouile. 2009. "A model driven framework for multiview service oriented system development." In *2009 IEEE/ACS International Conference on Computer Systems and Applications*, edited, 404-411: IEEE.

Fenning, Ross, Huseyin Dogan and Keith Phalp. 2014. "Applicability of SSM and UML for Designing a Search Application for the British Broadcasting Corporation (BBC), *Berlin, Heidelberg*, edited, 472-486: Springer Berlin Heidelberg.

Fettke, Peter. 2009. "How conceptual modeling is used." *Communications of the Association for Information Systems* 25 (1): 571-592.

Fettke, Peter and Peter Loos. 2007. "Ontological Evaluation of Scheer's Reference Model for Production Planning and Control Systems." *International Journal of Information Systems Interoperability in Business* 2 (1): 9-28.

Fuentes-Fernández, Rubén, Juan Pavón and Francisco Garijo. 2012. "A model-driven process for the modernization of component-based systems." *Science of Computer Programming* 77 (3): 247-269. doi: https://doi.org/10.1016/j.scico.2011.04.003.

George, M.;, K.; Fischer-Hellmann, M.; Knahl, U.; Bleimann and S. Atkinson. 2012. "Traceability in model-based testing. ." *Future Internet* 4 (4): 1026-1036.

González, Arturo, Sergio España, Marcela Ruiz and Óscar Pastor. 2011. "Systematic Derivation of Class Diagrams from Communication-Oriented Business Process Models, *Berlin, Heidelberg*, edited, 246-260: Springer Berlin Heidelberg.

Green, Peter, Michael Rosemann, Marta Indulska and Jan Recker. 2011. "Complementary use of modeling grammars." *Scandinavian Journal of Information Systems* 23 (1): 59-86.

Große-Rhode, Martin. 2004. "On Model Integration and Integration Modelling." In *Integration of Software Specification Techniques for Applications in Engineering: Priority Program SoftSpez of the German Research Foundation (DFG), Final Report*, edited by Hartmut Ehrig, Werner Damm, Jörg Desel, Martin Große-Rhode, Wolfgang Reif, Eckehard Schnieder and Engelbert Westkämper, 567-581. Berlin, Heidelberg: Springer Berlin Heidelberg. doi: 10.1007/978-3-540-27863-4_30.

Gustas, Remigijus. 2010. "Conceptual modeling and integration of static and dynamic aspects of service architectures." In *Workshop on Ontology, Conceptualization and Epistemology for Information Systems, Software Engineering and service Science*, edited, 17-32: Springer.

Gustas, Remigijus and Prima Gustiene. 2011. "A Method for Conceptual Modeling of Semantically Integrated Use-Case Scenarios, *Berlin, Heidelberg*, edited, 138-152: Springer Berlin Heidelberg.

Gustas, Remigijus and Prima Gustiené. 2017. "Principles of Semantically Integrated Conceptual Modelling Method, *Cham*, edited, 1-26: Springer International Publishing.

Hamou-Lhadj, Abdelwahab, Abdelouahed Gherbi and Jagadeesh Nandigam. 2009. "The impact of the model-driven approach to software engineering on software engineering education." In *2009 Sixth International Conference on Information Technology: New Generations*, edited, 719-724: IEEE.

Heuer, André, Vanessa Stricker, Christof J. Budnik, Sascha Konrad, Kim Lauenroth and Klaus Pohl. 2013. "Defining variability in activity diagrams and Petri nets." *Science of Computer Programming* 78 (12): 2414-2432. doi: https://doi.org/10.1016/j.scico.2012.06.003.

Huzar, Zbigniew, Ludwik Kuzniarz, Gianna Reggio and Jean Louis Sourrouille. 2004. "Consistency problems in UML-based software development." In *International Conference on the Unified Modeling Language*, edited, 1-12: Springer.

Jabbari Sabegh, Mohammad Ali, Jan Recker and Peter Green. 2016. "Designing experiments to test the theory of combined ontological coverage.

Jinwoo, Kim, Kim Jimwoo, Hahn Jungpil and Hahn Hyoungmee. 2000. "How Do We Understand a System with (So) Many Diagrams&quest; Cognitive Integration Processes in Diagrammatic Reasoning." *Information Systems Research* 11 (3): 284-303. doi: 10.1287/isre.11.3.284.12206.

Keet, C. Maria and Pablo Rubén Fillottrani. 2015. "An ontology-driven unifying metamodel of UML Class Diagrams, EER, and ORM2." *Data & Knowledge Engineering* 98: 30-53. doi: https://doi.org/10.1016/j.datak.2015.07.004.

Kenzi, Adil, Bouchra El Asri, Mahmoud Nassar and Abdelaziz Kriouile. 2009. "A model driven framework for multiview service oriented system development, edited, 404-411: IEEE Publishing. doi: 10.1109/AICCSA.2009.5069357.

Kernschmidt, Konstantin, Giacomo Barbieri, Cesare Fantuzzi and Birgit Vogel-Heuser. 2013. "Possibilities and challenges of an integrated development using a combined SysML-model and corresponding domain specific models." *IFAC Proceedings Volumes* 46 (9): 1465-1470.

Kim, Jinwoo, Jungpil Hahn and Hyoungmee Hahn. 2000. "How do we understand a system with (so) many diagrams? Cognitive integration processes in diagrammatic reasoning." *Information Systems Research* 11 (3): 284-303.

Koivulahti-Ojala, Mervi and Timo Kakola. 2010. "Framework for evaluating the version management capabilities of a class of UML modeling tools from the viewpoint of multi-site, multi-partner product line organizations." In *2010 43rd Hawaii International Conference on System Sciences*, edited, 1-10: IEEE.

Kramer, Max E. 2015. "A Generative approach to change-driven consistency in multi-view modeling." In *Proceedings of the 11th International ACM SIGSOFT Conference on Quality of Software Architectures*, edited, 129-134: ACM.

Labiak, Grzegorz, Marian Adamski, Jacek Tkacz, Michal Doligalski and Arkadiusz Bukowiec. 2011. "Role of UML Modelling in Discrete Controller Design, edited, 480-481: IEEE Publishing. doi: 10.1109/ICSEng.2011.97.

Lange, C. F. J., M. R. V. Chaudron and J. Muskens. 2006. "In practice: UML software architecture and design description." *IEEE Software* 23 (2): 40-46. doi: 10.1109/MS.2006.50.

Lee, Mong Li and Tok Wang Ling. 2003. "A Methodology for Structural Conflict Resolutionin the Integration of Entity-Relationship Schemas." *Knowledge and Information Systems* 5 (2): 225-247. doi: 10.1007/s10115-003-0077-x.

Lillieskold, J. 2003. "Coordinating dependencies in complex system development projects, *USA*, edited, 400-404: IEEE. doi: 10.1109/IEMC.2003.1252302.

Liu, Jing, Zhiming Liu, Jifeng He and Xiaoshan Li. 2004. "Linking UML models of design and requirement." In *2004 Australian Software Engineering Conference. Proceedings.*, edited, 329-338: IEEE.

Lucas, Francisco J, Fernando Molina and Ambrosio Toval. 2009. "A systematic review of UML model consistency management." *Information and Software Technology* 51 (12): 1631-1645.

Malik, Qaisar A, Dragos Truscan and Johan Lilius. 2010. "Using UML models and formal verification in model-based testing." In *2010 17th IEEE International Conference and Workshops on Engineering of Computer Based Systems*, edited, 50-56: IEEE.

Moody, Daniel. 2003. "Dealing with complexity in information systems modeling: Development and empirical validation of a method for representing large data models." *ICIS 2003 Proceedings*: 18.

Nassar, Mahmoud. 2003. "VUML: a Viewpoint oriented UML Extension." In *18th IEEE International Conference on Automated Software Engineering, 2003. Proceedings.*, edited, 373-376: IEEE.

Opdahl, A. L. and B. Henderson-Sellers. 2005. "A unified modelling language without referential redundancy." *Data & Knowledge Engineering* 55 (3): 277-300.

Opdahl, Andreas L. 2010. "Incorporating uml class and activity constructs into ueml." In *International Conference on Conceptual Modeling,* edited, 244-254: Springer.

Palpanas, Themis, Pawan Chowdhary, George Mihaila and Florian Pinel. 2007. "Integrated model-driven dashboard development." *Information Systems Frontiers* 9 (2): 195-208. doi: 10.1007/s10796-007-9032-9.

Pancham, J. and R. Millham. 2015. "Design phase consistency: A tool for reverse engineering of UML activity diagrams to their original scenarios in the specification phase, edited, 655-670: Springer Verlag. doi: 10.1007/978-3-319-21410-8_50.

Paré, Guy, Marie-Claude Trudel, Mirou Jaana and Spyros Kitsiou. 2015. "Synthesizing information systems knowledge: A typology of literature reviews." *Information & Management* 52 (2): 183-199.

Park, Wei-Jin and Doo-Hwan Bae. 2011. "A two-stage framework for UML specification matching." *Information and Software Technology* 53 (3): 230-244. doi: 10.1016/j.infsof.2010.11.001.

Petri, Carl A. 1962. "Fundamentals of a Theory of Asynchronous Information Flow." In *IFIP Congress 62: Information Processing,* edited by C. M. Popplewell, 386-390. Munich, Germany: North-Holland.

Recker, Jan and Peter Green. forthcoming. "How do Individuals Interpret Multiple Conceptual Models? A thoery of Combined Ontological Ccmpleteness and Overlap." *Journal of the Association for Information Systems*: In press.

Rosemann, Michael. 2006. "Potential pitfalls of process modeling: part A." *Business Process Management Journal* 12 (2): 249-254.

Sabegh, Mohammad and Jan Recker. 2017. "Combined Use of Conceptual Models in Practice: An Exploratory Study." *Journal of Database Management (JDM)* 28 (2): 56-88. doi: 10.4018/JDM.2017040103.

Schatz, Bernhard, Peter Braun, Franz Huber and Alexander Wisspeintner. 2003. "Consistency in model-based development." In *10th IEEE International Conference and Workshop on the Engineering of Computer-Based Systems, 2003. Proceedings.*, edited, 287-296: IEEE.

Scheer, August-Wilhelm and Markus Nüttgens. 2000. "ARIS Architecture and Reference Models for Business Process Management." In *Business Process Management: Models, Techniques, and Empirical Studies,* edited by W.M.P. van der Aalst, Jörg Desel and Andreas Oberweis, 376-389. Berlin, Heidelberg: Springer doi: 10.1007/3-540-45594-9_24.

Schmidt, Douglas C. 2006. "Model-driven engineering." *COMPUTER-IEEE COMPUTER SOCIETY-* 39 (2): 25.

Sewchurran, K. and D. Petkov. 2007. "A systemic framework for business process modeling combining soft systems methodology and UML." *Information Resources Management Journal* 20 (3): 46-62.

Siau, Keng Leng and Lih Yunn Lee. 2004. "Are use case and class diagrams complementary in requirements analysis? An experimental study on use case and class diagrams in UML." *Requirements engineering* 9 (4): 229-237.

Siau, Keng and Yuhong Tian. 2005. "A Semiotics View of Modeling Method Complexity-The Case of UML." *AMCIS 2005 Proceedings*: 318.

Skersys, Tomas and Saulius Gudas. 2006. "Class model development using business rules." In *Advances in Information Systems Development*, 203-215: Springer.

Spanoudakis, George and Andrea Zisman. 2001. "Inconsistency management in software engineering: Survey and open research issues." In *Handbook of Software Engineering and Knowledge Engineering: Volume I: Fundamentals*, 329-380: World Scientific.

Steele, Jay, Y. J. Son and R. A. Wysk. 2001. "Resource modeling for the integration of the manufacturing enterprise." *Journal of Manufacturing Systems* 19 (6): 407-427.

Tan, Xin, Steven Alter and Keng Siau. 2008. "Integrating Lightweight Systems Analysis into the United Process by Using Service Responsibility Tables." *AMCIS 2008 Proceedings*: 64.

Torngren, Martin, DeJiu Chen and Ivica Crnkovic. 2005. "Component-based vs. model-based development: a comparison in the context of vehicular embedded systems." In *31st*

EUROMICRO Conference on Software Engineering and Advanced Applications, edited, 432-440: IEEE.

Torres, Alexandre, Renata Galante and Marcelo S. Pimenta. 2011. "A synergistic model-driven approach for persistence modeling with UML." *Journal of Systems and Software* 84 (6): 942-957. doi: https://doi.org/10.1016/j.jss.2011.01.027.

Venkatesh, R, Purandar Bhaduri and Mathai Joseph. 2001. "Formalizing models and meta-models for system development extended abstract." In *Proceedings Eighth Asia-Pacific Software Engineering Conference*, edited, 155-158: IEEE.

Wand, Yair and Ron Weber. 1995. "On the deep structure of information systems." *Information Systems Journal* 5 (3): 203-223.

Wand, Yair and Ron Weber. 2002. "Research commentary: information systems and conceptual modeling—a research agenda." *Information Systems Research* 13 (4): 363-376.

Webster, J. and Rt Watson. 2002. "Analyzing the past to prepare for the future: Writing a literature review." *Mis Quarterly* 26 (2): XIII-XXIII.

Xu, Yifan, Jianwei Lv, Zhenyu Lian and Guangqiang Wang. 2017. "Multi-resolution modeling and application for complex system analysis based on design structure matrix." In *2017 Second International Conference on Reliability Systems Engineering (ICRSE)*, edited, 1-8: IEEE.

Yan, Rongjie, Chih-Hong Cheng and Yesheng Chai. 2015. "Formal consistency checking over specifications in natural languages." In *Proceedings of the 2015 Design, Automation & Test in Europe Conference & Exhibition*, edited, 1677-1682: EDA Consortium.

Zikra, Iyad, Janis Stirna and Jelena Zdravkovic. 2011. "Analyzing the Integration between Requirements and Models in Model Driven Development, *Berlin, Heidelberg*, edited, 342-356: Springer Berlin Heidelberg.

Zur Muehlen, Michael, Marta Indulska and K. Kittel. 2008. "owards integrated modeling of business processes and business rules." Paper presented at the Australian Conference on Information Systems, Christchurch.