How Inconsistencies Between Multiple Conceptual Modeling Scripts Affect Readers’ Understanding

Mohammad Jabbari
Université Laval, mohammad.jabbari@fsa.ulaval.ca

Karl Werder
University of Cologne

How Inconsistencies Between Multiple Conceptual Modeling Scripts Affect Readers' Understanding

Emergent Research Forum (ERF)

Mohammad Jabbari
Université Laval
mohammad.jabbari@fsa.ulaval.ca

Karl Werder
University of Cologne
werder@wiso.uni-koeln.de

Abstract

IS professionals often use multiple conceptual modeling scripts to develop an understanding of a domain. However, using multiple scripts introduces potential inconsistencies between scrips which can reduce script readers' cognitive ability to develop an understanding. While there are computational methods to avoid or detect inconsistencies, there is a lack of studies on how individuals deal with inconsistencies when they are performing different tasks. We developed a 2x2 between-subject experimental design to investigate the effects of syntactic vs semantic inconsistency on two different systems analysis and design tasks. We expect to contribute to conceptual modeling research, by investigating the effect of inconsistencies, comparing the effects of two tasks, and by elaborating the on role of a pragmatic factor, domain familiarity.

Keywords

Conceptual modeling scripts, multiple scripts, inconsistencies, different tasks, performance.

Introduction

Developing and using conceptual modeling scripts (*scripts* from now on) are fundamental activities of IS professionals to validate requirements and understand the system's interactions with its environment (Wand & Weber, 2002). IS professionals use multiple scripts developed using different grammars because IS have become ever more complex and any one grammar does not provide sufficient constructs to develop a script that represents all aspects of a system (Rodrigues da Silva, 2015). Therefore, IS professionals' benefit from different viewpoints that result from using multiple grammars (Delen & Benjamin, 2003). For example, scripts that represent the system's behavior and scripts that represent the system's structure. When multiple scripts are available, readers need to understand how different scripts are related to each other and then search, integrate, and combine information to form a single, composite mental model of the represented domain in their working memory (Delen & Benjamin, 2003; Kim et al., 2000).

Prior studies indicate that overlapping constructs enable readers to detect how different scripts are related to each other (Jabbari & Recker, 2017; Jabbari et al., 2022). Overlapping constructs between scripts refer to constructs in each script that represent the same entities of a real-world domain across scripts. However, overlapping constructs between scripts may lead to inconsistencies (Ong & Jabbari, 2019). Inconsistency between scripts refers to a state in which two or more elements in different scripts of the same domain—that should be overlapping constructs—are dissimilar or incompatible (Spanoudakis & Zisman, 2001). For example, when the meaning of constructs in one script is not compatible with their respective constructs in another script, or when overlapping constructs do not comply with abstract syntax of the modeling grammar (Lucas et al., 2009). When there are inconsistencies between scripts, readers might struggle to integrate information across scripts and need additional cognitive effort to form a composite mental model. Inconsistencies between scripts are common in practice (Feldmann et al., 2019). For instance, combining and integrating multiple enterprise modeling scripts is a challenge because inconsistencies increase readers' cognitive effort of understanding how different scripts are related (Sandkuhl et al., 2018).

Existing research mainly focused on identifying, classifying, or compensating for inconsistencies. For example, prior work developed methods to detect inconsistencies (Iren et al., 2019), derived a taxonomy (Lucas et al., 2009b), and identified rules to avoid inconsistencies (Torre et al., 2020). However, there is a lack of knowledge of how inconsistencies affect cognitive information processing behaviors and readers' understanding. It is important to evaluate the cognitive process of understanding because a complete understanding of a domain requires integrating information from multiple scripts (Jabbari & Recker, 2017), the existence of inconsistencies may decrease readers' cognitive capabilities to integrate relevant information and performing different systems analysis and design tasks requires different cognitive effort (Malinova & Mendling, 2021). Therefore, this study investigates the following research question: *What is the effect of inconsistencies between overlapping constructs on readers' performance in verifying requirements and understanding interactions during systems analysis and design?*

Background

Inconsistencies in Conceptual Modeling Scripts: The use of multiple scripts is a common practice and their use is a focal topic during systems analysis and design (Jabbari et al., 2022) and enterprise modeling (Delen & Benjamin, 2003; Sandkuhl et al., 2018). Multiple scripts developed using different grammars, methods, and tools are necessary to represent different aspects of complex systems or enterprises. Therefore, multiple different types of separate, but interdependent scripts are developed during enterprise modeling and systems analysis and design tasks. Interdependent scripts incorporate elements which refer to common aspects of the system under development (Feldmann et al., 2019). For instance, an activity diagram may present the activity and behavior of an entity represented in a class diagram. Therefore, inconsistency may occur between these heterogeneous and interdependent scripts. Inconsistencies can be classified according to the different characteristics and causes (Lucas et al., 2009): semantic, syntactic, inter-script, and intra-script. Semantic inconsistency refers to the incompatibility between respective constructs in relevant scripts. When the meaning of constructs in a class diagram—for example, a class representing "students"—is not compatible with their respective constructs in an activity diagram—for example, a swimlane that represent the same activities related to the operations in the "students" class—but conveys a different meaning (rather than students in our example). Syntactic inconsistency refers to overlapping constructs that do not comply with abstract syntax of the modeling grammar. For example, when a class in the class diagram is represented as an activity in the activity diagram. Intra- and Inter-script inconsistency refer to discrepancies in the level of abstraction within the same or among different scripts. Prior studies on inconsistencies focused on the development of computational approaches such as description logics or reasoning tools to detect inconsistencies (van der Straeten et al., 2003), or proposed over 119 rules to avoid potential inconsistencies between scripts (Torre et al., 2018). However, most of these rules are used in computational methods and there is a lack of studies on individuals' cognitive efforts in dealing with inconsistencies. Therefore, we focus on syntactic and semantic between different types of scripts at the same level of abstraction. We investigate how syntactic and semantic inconsistencies affect participants' performance in two different task settings and how this relationship is affected by individual differences, that is, domain familiarity (Khatri & Vessey, 2016).

***Cognitive Processing of Conceptual Modeling Scripts*:** The CogniDia theory (Malinova & Mendling, 2021) summarizes the cognitive processing of script readers. The theory suggests that when reading scripts, different cognitive processes take place that draw on different knowledge sources from the long-term memory. The central executive orchestrates the overall cognitive process and integrates knowledge from the long-term memory (Anderson et al., 2004). The cognitive process consists of four steps (Malinova & Mendling, 2021). First, information enters the visual processing, that is, the information is captured through the eyes. Gestalt knowledge is needed to process ideograms, lines, and shapes. Second, the information is processed further in two ways: i) the words presented in the script enter the verbal processing where words are linked, symbols are combined using associative connections, and words and symbols are linked using referential connections (Mayer, 2002), requiring natural language knowledge; or ii) the images are organized in a visual mental model. However, script readers often require both, visual and verbal processing, to generate a visual and a verbal mental model from the script. Third, both models are processed further semantically (Anderson et al., 2004). Images are organized, integrated with the words, and interpreted using prior knowledge (e.g., domain knowledge). Prior knowledge reduces the cognitive efforts needed to read scripts through automated processing of the information (Khatri & Vessey, 2016). Fourth, the interpreted model will be subject to a goal hierarchy when processing a particular task.

Cognitive-fit theory demands a fit between the task and problem representation to achieve higher performance (Vessey, 1991). Systems analysis and design is performed using five central tasks (Malinova & Mendling, 2021): Elicit requirements, refine requirements, specify design, decompose design, and implement the system. CogniDia theory suggests cognitive process for different tasks varies based on task characteristics. However, there is a lack of empirical evaluation of CogniDia`s propositions.

Hypotheses

To evaluate how semantic factors, that is, inconsistency (Bera et al., 2014) and pragmatic factors, that is, domain knowledge (Khatri & Vessey, 2016) may affect individual's performance for different tasks we draw on the CogniDia theory (Malinova & Mendling, 2021). Overall, we suggest that the effect of inconsistency on performance varies with the type of inconsistency (i.e., syntactic vs semantic) and the task at hand (i.e., verifying requirements vs understanding interactions). We suggest that these relationships are affected by domain familiarity. We present our research model in Figure 1.

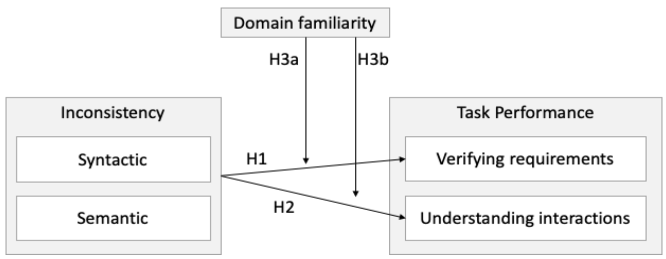

Figure 1. Research model.

The effect of inconsistencies during requirements verification: Inconsistencies lead to increased efforts in the cognitive search process by script readers. Detecting inconsistencies always requires the integration of what is captured within the scripts with the readers' prior knowledge. While syntactic inconsistencies relate to the compliance of pre-defined rules that require assessment against prior modelling knowledge, semantic inconsistencies relate to the language's consistency and require assessment against prior domain knowledge. We suggest that script readers identify more errors with semantic inconsistencies in contrast to syntactic inconsistencies. It is easier to spot inconsistencies in two presented information than identifying missing information. The processing of semantic inconsistencies relies on the same cognitive functions, whereas the processing of syntactic inconsistencies relies on multiple cognitive functions (i.e., processing of images and processing of analytical rules). For example, prior studies have shown that the semantic processing of two scripts is influenced by script readers' ability to integrate those scripts and therefore influences performance (Jabbari et al., 2022). We formulate the following hypothesis:

H1: Script readers facing semantic inconsistencies will have better task performance during requirements verification than script readers facing syntactic inconsistencies.

The effect of inconsistencies during understanding interactions: We suggest that script readers develop a better understanding with syntactic inconsistencies in contrast to semantic inconsistencies. Developing an understanding of scripts requires a thorough understanding and synthesis of both presented scripts. While the semantic inconsistencies are easier to identify, they are more difficult to fully comprehend, as they lead to irresolvable ontological differences between two terms. For example, when modeling an e-commerce platform, we find the terms user and customer are used to describe the same thing. Syntactic inconsistencies on the other hand can often be resolved through logic and reasoning, following the ontological rules of the modeling grammar. For example, a prior study suggests that syntactic inconsistencies can be automatically identified (Torre et al., 2020), while others suggest more challenges to accomplish a fully automated approach of detecting semantic inconsistencies (Daun et al., 2015). We formulate the following hypothesis:

H2: Script readers facing syntactic inconsistencies will have a better task performance when understanding the system's interactions than script readers facing semantic inconsistencies.

The moderating effect of domain familiarity: We suggest that the effects of semantic vs syntactic inconsistencies on task performance are moderated by domain familiarity. When script readers are unfamiliar with a domain, they do not have an established mental representation of the domain. Given the lacking mental representation, the script readers strongly rely on the scripts presented with the scripts' limitations. In contrast, if script readers are familiar with a domain, they have a prior mental model against which the current scripts can be compared and evaluated. Missing information are identified more easily, and conflicts can be quickly resolved. For example, a prior study has shown that prior domain knowledge can override semantic effects (Bera et al., 2014). We suggest that domain familiarity positively moderates the effects of inconsistencies toward task performance. We hypothesize:

H3a: The effect of semantic inconsistencies toward task performance when verifying requirements is moderated by domain familiarity, so that the effect is stronger for script readers that are familiar with the domain and weaker for script readers that are less familiar with the domain.

H3b: The effect of syntactic inconsistencies toward task performance when understanding the system's interactions is moderated by domain familiarity, so that the effect is stronger for script readers that are familiar with the domain and weaker for script readers that are less familiar with the domain.

Research Method

We will conduct a 2x2 between subject experimental study with two factors: domain familiarity - high and low; and inconsistency - semantic and syntactic. We designed four sets of two scripts with either semantic inconsistencies or syntactic inconsistencies. We manipulate level of domain familiarity by developing UML scripts for two different domains: library domain (Khatri & Vessey, 2016) and pharmaceutical domain (Bera et al., 2014). We selected the library domain for that we can assume that our participants have some but varying prior knowledge and are more familiar with. We selected the pharmaceutical domain for that we can assume that our participants have low or no prior knowledge and are less familiar with.

We designed four sets of scripts constructed with two different UML grammars, that is, activity and class diagram. For each domain, we designed a set of two scripts with semantic inconsistencies, for example, an activity diagram with swimlanes to represent roles and their related activities, and a class diagram representing classes with related operations, where the swimlanes are not compatible with the related classes, and a set of two scripts with syntactic inconsistencies, for example, an activity diagram representing data flows or activities and a class diagram that does not have associations between related instances to enable exchanges or there are missing related operations (see Torre et al., 2018, p. 133). We will measure performance in two different task settings: requirements verifications and understanding interactions (Malinova & Mendling, 2021) as our dependent variables. To evaluate participants performance in requirements verification we measure the total number of errors identified in verifying the given requirements (Kim et al., 2000). To evaluate participants performance in understanding interactions, we measure the total number of correct answers to the problem-solving questions (Jabbari et al., 2022). All measures and scripts are available on an open science server (doi: DOI 10.17605/OSF.IO/MEQDN). We will conduct our study using a self-developed tool for online experiments. We will sample our data from participants recruited using Amazon Mechanical Turk (MTurk). We will follow existing guidelines (Hauser et al., 2019) to ensure attentiveness and language comprehension, mitigate learning effects and deceptions by participants, and avoid attrition and self-selection. We will apply job function "information technology" and employment industry "software and IT services" filters to identify individuals who are likely to have been involved in systems analysis and design tasks. Participants will be assigned randomly into four groups and each participant will be asked to complete two tasks. The order of tasks will be randomized to control the potential learning effect. An incentive of $15 will be paid for participation.

Expected Contributions

In this paper, we explained an experimental design and steps to study a largely ignored problem in practice: *how different inconsistencies between scripts affect script readers' performance in different tasks*. The resulting insights from our study can be used to derive practical guidelines that improve the detection of errors in the early stages of system development lifecycle. We also expect three important theoretical contributions. First, we contribute to conceptual modeling research by explaining and empirically evaluating how individuals deal with inconsistencies when reading multiple scripts. Second, we explore the varying effects of inconsistencies while accounting for known influential factors during systems analysis and design tasks, that is, pragmatic and semantic factors. Third, we extend current conceptual modeling theory by explaining and evaluating how different tasks settings require different cognitive effort.

REFERENCES

Anderson, J. R., Bothell, D., Byrne, M. D., Douglass, S., Lebiere, C., & Qin, Y. (2004). An integrated theory of the mind. *Psychological Review*, *111*(4), 1036.

Bera, P., Burton-Jones, A., & Wand, Y. (2014). Research note—how semantics and pragmatics interact in understanding conceptual models. *Information Systems Research*, *25*(2), 401–419.

Daun, M., Weyer, T., & Pohl, K. (2015). Detecting and correcting outdated requirements in function-centered engineering of embedded systems. *International Working Conference on Requirements Engineering: Foundation for Software Quality*, 65–80.

Delen, D., & Benjamin, P. C. (2003). Towards a truly integrated enterprise modeling and analysis environment. *Computers in Industry*, *51*(3), 257–268.

Feldmann, S., Kernschmidt, K., Wimmer, M., & Vogel-Heuser, B. (2019). Managing inter-model inconsistencies in model-based systems engineering: Application in automated production systems engineering. *Journal of Systems and Software*, *153*, 105–134.

Hauser, D., Paolacci, G., & Chandler, J. (2019). *Common concerns with MTurk as a participant pool: Evidence and solutions*.

Iren, Y. D., Aysolmaz, B., & Reijers, H. (2019). Detecting Role Inconsistencies in Process Models. *27th European Conference on Information Systems*.

Jabbari, M., & Recker, J. (2017). Combined Use of Conceptual Models in Practice. *Journal of Database Management*, *28*(2), 56–88.

Jabbari, M., Recker, J., Green, P., & Werder, K. (2022). How do Individuals Understand Multiple Conceptual Modeling Scripts? *Journal of the Association for Information Systems*.

Khatri, V., & Vessey, I. (2016). Understanding the role of is and application domain knowledge on conceptual schema problem solving: A verbal protocol study. *Journal of the Association for Information Systems*, *17*(12), 759 – 803.

Kim, J., Hahn, J., & Hahn, H. (2000). How do we understand a system with (so) many diagrams? Cognitive integration processes in diagrammatic reasoning. *Information Systems Research*, *11*(3), 284–303.

Lucas, F. J., Molina, F., & Toval, A. (2009). A systematic review of UML model consistency management. *Information and Software Technology*, *51*(12), 1631–1645.

Malinova, M. M., & Mendling, J. (2021). Cognitive Diagram Understanding and Task Performance in Systems Analysis and Design. *MIS Quarterly*, *45*(4), 2101–2157.

Mayer, R. E. (2002). Multimedia learning. In *Psychology of learning and motivation* (Vol. 41, pp. 85–139). Elsevier.

Ong, J. D., & Jabbari, M. (2019). A review of problems and challenges of using multiple conceptual models. *Proceedings of the 27th European Conference on Information Systems*, 1–18.

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

Sandkuhl, K., Fill, H.-G., Hoppenbrouwers, S., Krogstie, J., Matthes, F., Opdahl, A., Schwabe, G., Uludag, Ö., & Winter, R. (2018). From Expert Discipline to Common Practice: A Vision and Research Agenda for Extending the Reach of Enterprise Modeling. *Business & Information Systems Engineering*, *60*(1), 69–80.

Spanoudakis, G., & Zisman, A. (2001). Inconsistency management in software engineering: Survey and open research issues. In *Handbook of Software Engineering and Knowledge Engineering: Volume I: Fundamentals* (pp. 329–380). World Scientific.

Torre, D., Labiche, Y., Genero, M., & Elaasar, M. (2018). A systematic identification of consistency rules for UML diagrams. *Journal of Systems and Software*, *144*, 121–142.

Torre, D., Labiche, Y., Genero, M., Elaasar, M., & Menghi, C. (2020). UML consistency rules: a case study with open-source UML models. *Proceedings of the 8th International Conference on Formal Methods in Software Engineering*, 130–140.

van der Straeten, R., Simmonds, J., & Mens, T. (2003). Detecting Inconsistencies between UML Models Using Description Logic. *Description Logics*, *81*.

Vessey, I. (1991). Cognitive fit: A theory-based analysis of the graphs versus tables literature. *Decision Sciences*, *22*(2), 219–240.

Wand, Y., & Weber, R. (2002). Research Commentary: Information Systems and Conceptual Modeling—A Research Agenda. *Information Systems Research*, *13*(4), 363–376.