A Model-Driven Method for Documentation and Analysis of Software-Intensive Systems Requirements

Breno Lisi Romano  
Brazilian Aeronautics Institute of Technology  
Brazil – São Paulo – São José Dos Campos  
Praça. Marechal Eduardo Gomes 50  
055 12 3947-5843  
bromano@gmail.com  
Denis Ávila Montini  
M.Sc. Production Engineering  
TCS Brazil – São Paulo – Barueri  
Av. Aruanã 70, Tamboré  
055 11 6496-5107  
denisavila.montini@tcs.com

Luiz Alberto Vieira Dias  
PhD. Space Physics  
Brazil – São Paulo – São José Dos Campos  
Praça. Marechal Eduardo Gomes 50  
055 12 3947-5843  
vdias@ita.br

Henrique Campos Fernandes  
Brazilian Aeronautics Institute of Technology  
Brazil – São Paulo – São José Dos Campos  
Praça. Marechal Eduardo Gomes 50  
055 12 3947-5843  
henriquefc@gmail.com  
Adilson Marques da Cunha  
Doctor of Science  
Brazil – São Paulo – São José Dos Campos  
Praça. Marechal Eduardo Gomes 50  
055 12 3947-5843  
cunha@ita.br

ABSTRACT

This paper tackles the development of a Model-Driven Method for Documentation and Analysis (MDM-DA) of software-intensive systems requirements. This method is supported by Model-Driven Development (MDD) features to improve quality and efficiency for the fulfillment of Requirements’ Engineering activities. Besides, a Conceptual Model for applying MDM-DA was designed, by considering the theoretical concepts of Systems and Requirements Engineering. At the end, this method was successfully applied within a real project component providing requirement traceability from the beginning up to the end of the development process.


Key words: Model-Driven Development (MDD), Systems and Requirements Engineering, System Modeling Language (SysML), Kano Method, and SysML Profile.

1. INTRODUCTION

There are considerable efforts in research and development aiming to improve the production process of Software-Intensive Systems (SIS). Considered as part of critical areas in systems development, they are comprised of hardware, software, firmware, and other elements especially related to requirements engineering activities. Environments have been necessary for continuous improvement of sophisticated systems by using new tools, technologies, modeling languages, and development [1].

Systems’ Engineering is a multidisciplinary approach to develop solutions for SIS, in response to the diversity of stakeholders’ needs. It includes technical and management processes to achieve a balanced development, by reducing risks that may impact its success.

The System Modeling Language (SysML) can be used to support System Engineering technical processes, allowing specification, design, and verification of the SIS target.

The SysML was developed in order to fill up semantic gaps among systems, software, and other engineering disciplines. By definition [2], it allows system engineers from different areas to analyze, specify, design, and verify complex systems, by enhancing SIS quality.

This language also allows visual SIS requirements representation with models, by showing the contract between stakeholders and systems engineers. However, more detailed specifications of SIS requirements and other elements related to them are not covered by SysML.

Within this context Model-Driven Development (MDD) considers models as key artifacts during SIS development. It is an emerging approach that addresses complexity in systems development by intensively using models [3].

Researchers have been working on MDD approaches to transform models into source-codes intending to increase the production level in SIS development, such as the use of Model-Driven Architecture (MDA). However, few of them have considered models as the main artifacts of requirements’ engineering which is the major focus of this paper [4].

The MDA, as an MDD approach, uses models at different abstraction levels, aiming to separate system conceptual architecture from their specific implementations, focusing on solutions’ development rather than technology details. The definition of transformation mechanisms allows those models translations into high level source-codes to generate executable application of them [5].

Figure 1 presents an overview of each one of the four different abstraction levels proposed by the MDA, illustrating the possible transformations between them. The four different models’ types are: Computation-Independent Model (CIM), Platform-Independent Model (PIM), Platform-Specific Model (PSM), and Source-Codes.

This paper tackles the development of a Model-Driven Method for Documentation and Analysis (MDM-DA) of SIS requirements, in order to improve quality and efficiency in performing requirements’ engineering activities, and to reduce wastage of resources on activities after the development process, regarding requirements priority and traceability, and it’s changing management.

The remaining of this paper is organized as follows. Second section shows some related works. The third describes the
conceptual model to apply the MDM-DA. A model-driven method for documentation and analysis of SIS requirements is shown in the fourth section. Some obtained results of the MDM-DA applied on a real project are presented on the fifth section. Finally, the sixth section presents some conclusions.

Figure 1. The overview of models and transformations proposed by MDA

2. RELATED WORKS

Initially, it was investigated a recent research from Soares [6] with the same scope of this paper. It proposed the requirements specification and modeling of a real-time distributed system to manage the Netherlands highways system, by using requirements, diagrams, and tables as proposed in the SysML.

The Soares research was continued in 2008 [3], by formalizing the previous approach to specify and model SIS requirements, proposing an extension of the SysML Requirements’ Diagram. This extension was defined to include the requirements’ classification of Functional, Non-Functional, and External Interfaces. In addition to this extent, it was recommended a strategy of grouping SIS requirements through packages in order to struggle with the difficulty of large and complex systems’ development.

Other research work produced by Boulanger [7] presented an application of a methodology based on Requirements’ Engineering to facilitate the mapping of requirements validation and traceability into embedded software for the French automotive sector.

To better assist the implementation of Requirements’ Engineering phases on this automotive project, Boulanger defined an extension of SysML requirements stereotype, by adding the new fields of priority, author, justification, and requirements’ classification [7].

3. THE MDM-DA CONCEPTUAL MODEL

With the concepts defined by Systems and Requirements Engineering, it was possible to conceive, as seen in Figure 2, a conceptual model to apply the MDM-DA. In this model, the target problem and the stakeholders’ needs have been identified and discussed with system engineers, in order to look for rationales and assumptions from the identified needs representing the basis for an appropriate requirements specification.

In Figure 2, green boxes represent the three technical processes of System Engineering: System Specification and Design; Components Design, Implementation, and Testing; and System Integration and Testing.

The first process of “System Specification and Design” was improved by appending Requirements’ Engineering Activities shown within the blue box, and identifying and classifying functional, non-functional, and organizational requirements shown within the red boxes.

On the second process of “Components Design, Implementation, and Testing”, the components’ requirements were categorized right after the system requirements’ identification process, representing the inputs for this process in order to verify them.

Once finished the components verification, all activities of the third process of “System Integration and Testing” were conducted taking into account all systems’ components. At this point, it is recommended to send feedbacks to the previous processes, aiming to improve system requirements’ specification throughout this iterative and incremental process.

Notice that by using an appropriate documentation of the Conceptual Model processes, it became possible to define traceability between needs, requirements, and functionalities of any SIS. The traceability feature was an important element considered within the MDM-DA, as shown on the next section.

4. THE MDM-DA

Within this context, it was possible to design the Model-Driven Method for Documentation and Analysis (MDM-DA) of SIS requirements. This was performed by applying the MDD concepts, taking into account models as key artifacts produced during its development. For this reason, it was adopted the SysML to support SIS modeling and specification, aiming to fulfill the semantic gap between systems, software, and other engineering disciplines.

One of the major contributions of this research was the adoption of SysML within the MDM-DA, providing diagrams and associations focusing on SIS requirements specification through models. When it was combined with some Unified Modeling Language (UML) diagrams for designing software, SysML constructions can fulfill the gap between the requirements specification, usually written in natural language, and use case diagrams, as an initial system requirements specification.

By considering Systems Engineering features, as the SysML represents an UML extension, it was necessary that the MDM-DA took into account other features not yet provided by SysML, which in its turn represented also some extensions of this language.
Thus, in order to document and analyze the SIS requirements using the MDD concepts, it was defined the MDM-DA on Figure 3, designed with a total of 12 steps. This method was applied iteratively and incrementally in systems development by systems engineers and team members managed by them.

The MDM-DA was conceived using: a) the MDD to graphically represent the SIS requirements, rather than only a textual representation; b) the SysML to model, specify, and document SIS requirements; and c) the Kano Method for requirements prioritizing and negotiation. Moreover, the MDM-DA was applied on the scenario presented on Figure 2, using the proposed SysML extension.

During this research, it became clear that MDM-DA was not just a set of heuristics and/or good practices, but a method mainly defined and characterized by the application of scientific and/or technical concepts previously defined within the MDM-DA steps, as shown in Figure 3.

![Figure 3. The MDM-DA of SIS Requirements](image)

In order to support this approach, this research proposed a SysML profile extension. It focused on those aspects covered by the proposed method and not covered so far by the SysML.

The proposed SysML extension is shown in Figure 4, highlighting: the main aspects included in the new extension; the relationships among them; and the stereotypes created for using in the produced models.

Following, it will be presented a brief description of each MDM-DA step and also some visual constructions of the SysML extension used in each of them.

In the first MDM-DA step, it was defined the SIS target problem solved within its development. To achieve this goal, it was suggested regular meetings between client team and system engineer team, which was documented in meeting minutes.

![Figure 4. Proposed SysML Profile Extension for MDM-DA](image)

**As the MDM-DA was supported by MDD, the main models resulting from the first step have used the SysML Block Definition Diagram to define the SIS context, which represented its context within the client’s internal processes and also their interactions with other systems. So, the SysML Profile Extension proposed the following visual constructions, in order to represent these interactions: Internal Process, Related System, and External Element.**

After the SIS context already defined, it was identified and documented all stakeholders involved with the system. For each identified stakeholder, the second step of the MDM-DA proposed to fulfill the following information: a) an identifier; b) the category in which each stakeholder belongs, such as: system user, corporate manager, customer, or domain expert; c) the name and contact person that represents the stakeholder in the corporation; and d) the priority of each stakeholder which in its turn impacts on the requirements’ priority related to it.

The SysML Profile Extension has defined a visual construction for stakeholders and two enumerations: one representing its category and the other its priority.

In the early SIS development stages, each stakeholder has different needs that must be considered. Often, these needs are already documented within the company artifacts. The third step of MDM-DA proposed the identification and documentation of stakeholder’s needs and their documentations sources. Each defined need had an identifier and a description, and every documentation source had a filename and its storage site. It was noteworthy that there was a direct link between the stakeholders
and the needs, and also between the documentation sources and the needs. The SysML Profile Extension defined two visual constructions in this step: one for the needs and other for the documentation sources.

From the stakeholders' needs, system requirements were identified in the fourth step of the MDM-DA, as well as their assumptions and rationales. The system requirements were documented by an identifier, a description, and additional information. Besides, the needs were linked to every system requirements derived from them. In this step, it was also used two visual constructions from the original SysML specification: requirements and rationales. Additionally, one visual construction was defined to represent assumptions on the Proposed SysML Profile Extension.

In the eighth step of the MDM-DA named “defining development production lines”, classification, prioritization, and componentization of each requirement were finalized, in order to configure the SIS requirements implementation.

Notice that, the fifth step of the MDM-DA has proposed the classification of each SIS requirements as: functional; non-functional (usability, performance, quality, maintainability, and others); and organizational. On this step, one enumeration was defined on the Proposed SysML Profile Extension, aiming to define requirements' classification.

In order to configure the system requirements priority, in the sixth step of the MDM-DA, it was proposed the Kano Method application considering two important dimensions within the prioritization and negotiation context: the requirement compliance and the stakeholder satisfaction feeling [8].

According to the Kano Method, requirements were classified into three types which influenced the stakeholder satisfaction such as: necessary, normal, and attractive. The Proposed SysML Profile Extension has defined one visual construction representing the partial result of the Kano Method application and two other enumerations that support its application, as show on Figure 4. Besides, the requirement visual construction stored the final result from the Kano Method application.

Based in the MIL-STD-498, the seventh step of the MDM-DA aimed to cluster system requirements in components, with the purpose of reducing SIS development complexity, integrating the similar or related SIS requirements, and facilitating the production lines definition. This standard defines four integration levels for systems development [9], as shown on Figure 5. Depending on the SIS complexity, it was suggested its decomposition into a more appropriate integration level.

The SysML Profile Extension defined seven visual constructions, in order to represent integrations levels for hardware or software components on systems development such as: Computer Software/Hardware System (SYSTEM); Computer Software Configuration Item (CSCI); Computer Hardware Configuration Item (CHCI); Computer Software Component (CSC); Computer Hardware Component (CHC); Computer Software Unit (CSU); and Computer Hardware Unit (CHU).

The SIS development in an iteratively and incrementally way began after the production lines definition development (the MDM-DA eighth step).

During the ninth step of the MDM-DA, for each production line, it was defined and documented their use cases, business rules, test cases, and also links among all these model elements. In this step, within the Proposed SysML Profile Extension, it was also defined one visual construction to represent business rules and also a new association to link business rules from use cases, named “implement” association.

Moreover, in the tenth step of the MDM-DA, it was proposed the traceability definition among system requirements, use cases, and test cases. Consequently, it allowed requirements to be analyzed in the twelfth step, checking if they had already been implemented and tested.

In the eleventh step of the MDM-DA, as requirements' changes occurred during a SIS development, it was proposed a control of these changes, in order to maintain the changes' history and also analyze their impact on the development. After that, a return from the eleventh step to the fourth step was defined to tolerate reviews in activities previously performed, mainly due to the requirements' changes resulted in modifications of assumptions, rationales, priorities, production lines, among other modifications.

5. THE ICA-MMH PROJECT CASE STUDY

In this section, some results are shown, obtained from the MDM-DA application on the Project of Amazon Integration and Cooperation for Modernization of Hydrological Monitoring (Projeto de Integração e Cooperação Amazônica para Modernização do Monitoramento Hidrológico - ICA-MMH).

This project has been developed in a collaborative effort involving the Brazilian Aeronautics Institute of Technology (Instituto Tecnológico de Aeronáutica - ITA) and the Brazilian National Water Agency (Agência Nacional de Águas – ANA), supported by the Brazilian Research and Projects Financing (Financiadora de Estudos e Projetos - FINEP) [10].

In the ICA-MMH Project, hydrometeorological information are collected by remote measurement stations comprised of data acquisition equipments which can be linked to Data Collection Platforms (DCP) located nearby river beds. These measurements are taken from different geographic points on regular or irregular intervals, or determined by hydrometeorological events. The collected data become available through the Internet to be accessed by users of hydro resources such as institutions,
corporations, involving neighboring countries, public and environmental organizations, farmers, research institutes, and other interested groups. Some of the main applications for these data are hydro resources availability estimations, hydrological variability analyses, climate changes, and meteorological and critical forecasting’s events.

Within this context, the ICA-MMH System of Systems (SoS) is comprised of the following systems, as shown in Figure 6: Data Acquisition System; Data Treatment System; Monitoring, Control, and Decision Support System; Data Diffusion System; and the proposed Application Database System. Figure 6 was produced as result of applying first step of the MDM-DA, defining the ICA-MMH Project context.

According to the information flow shown in Figure 6, besides hydrometeorological information collected from the DCP (Related System), there are needs for interoperability among the Data Acquisition System, External Databases (External Elements), and Brazilian States Situational Rooms (External Elements). On the other hand, the ICA-MMH Project must also interact with others ANA systems through Web Services (WS). Some examples are WS for providing security access, geographic data supply, among others.

Additionally, as a result of applying the first step of the MDM-DA, it was also defined another context diagram representing a general schematic of communications among components within the scope of the ICA-MMH Project, as shown in Figure 7.

This section demonstrates partly the MDM-DA application on Monitoring, Control, and Decision Support System. Table 1 shows just one example of a need, a system requirement, and a business rule. This example illustrates the component “Monitoring of Amazon Watershed Reference Network” (ICSC-SMCAD-MMRR).

The Figure 8 shows the results of applying steps 2, 3, and 4 of the MDM-DA on the ICSC-SMCAD-MMRR component of the Monitoring, Control, and Decision Support System.

Furthermore, Table 2 shows the results of applying steps 4 and 5 of the MDM-DA on the ICSC-SMCAD-MMRR's requirements.

Notice, from Table 2, that the ICSC-SMCAD-MMRR component is composed only by functional requirements justifying their implementation with use cases. Notice also that the majority of the ICSC-SMCAD-MMRR's requirements was considered attractive to stakeholders. According to the Kano Method application, the compliance of these requirements has provided higher levels of stakeholders’ satisfactions, though not necessarily representing their grievances in case of failure.

Table 1. ICSC-SMCAD-MMRR: Examples of Need, Requirement and Business Rule

<table>
<thead>
<tr>
<th>Identifier</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMCD-NEC005</td>
<td>Visualize hydrometeorological and operational temporal series.</td>
</tr>
<tr>
<td>System Requirement</td>
<td></td>
</tr>
<tr>
<td>Identifier</td>
<td>Description</td>
</tr>
<tr>
<td>SMCD-RF067</td>
<td>The ICSC-SMCAD-MMRR must be able to provide the statistical information visualization of used transmission medias.</td>
</tr>
<tr>
<td>Business Rule</td>
<td></td>
</tr>
<tr>
<td>Identifier</td>
<td>Description</td>
</tr>
<tr>
<td>SMCD-RN015</td>
<td>It should be accounted statistically collected results, according to input parameters defined by user.</td>
</tr>
</tbody>
</table>

Figure 10 shows an example of a model that can be designed from use cases to demonstrate a complete application of the MDM-DA and the Proposed SysML Profile Extension in a real project.

As the ICA-MMH Project has been designed using automatic code generation from models, Figure 10 presents: a) a state machine to the use case "Show Data Acquisition Evidence"; b) a Human-Machine Interface (HMI) produced with automatic code generation considering this state machine, using the framework AndroMDA [10].
It is also important to notice that MDM-DA was successfully applied to a real project component by providing requirements traceability since the beginning up to the end of the development process.

The main benefits of using the MDM-DA application were: a) an effective increase in communication among the members involved in system development and also with stakeholders; b) a centralization of all artifacts to be used to develop a SIS in the same environment; and c) a detailed analysis of the entire development process through full documentation of this process with traceability from documentation sources of the stakeholders’ needs up to the produced source-codes.

However, the major difficulty was that not all members involved in the development had a complete understanding of the semantic concepts related to the SIS produced models, but this lack of understanding was minimized during its development.

Table 2. The ICSC-SMCAD-MRR: Requirements Classification and Stakeholder Satisfaction

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Classification</th>
<th>Satisfaction</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMCAD-RF051</td>
<td>Functional</td>
<td>Must be</td>
</tr>
<tr>
<td>SMCAD-RF052</td>
<td>Functional</td>
<td>Attractive</td>
</tr>
<tr>
<td>SMCAD-RF066</td>
<td>Functional</td>
<td>Attractive</td>
</tr>
<tr>
<td>SMCAD-RF067</td>
<td>Functional</td>
<td>Attractive</td>
</tr>
<tr>
<td>SMCAD-RF068</td>
<td>Functional</td>
<td>Attractive</td>
</tr>
<tr>
<td>SMCAD-RF069</td>
<td>Functional</td>
<td>Attractive</td>
</tr>
</tbody>
</table>

Figure 8. The ICSC-SMCAD-MRR: Results of applying steps 2, 3 and 4 of the MDM-DA

Figure 9. The ICSC-SMCAD-MRR: Results of applying steps 9 and 10 of the MDM-DA
Figure 3. The traceability between some of the most important elements using the MDM-DA

6. CONCLUSION

In this paper, it was consolidated Systems and Requirements Engineering concepts. This was performed in order to support the documentation and the SIS requirements analysis. The major contributions of this paper were the definition and description of: the MDM-DA Conceptual Model; the MDM-DA method; and all the steps necessary for its application in a real project named ICA-MMH.

The MDM-DA was proposed by using: a) the MDD to graphically represent requirements rather than textual representations; b) the SysML for modeling, specification, and requirements documentation; and c) the Kano Method to define requirements priorities.

To make possible the SysML extension providing aspects considered by MDM-DA, it was necessary to redesign the SysML original specification. This was performed mainly because it was not found in the scientific and technological community, a SysML profile available as open source software.

The authors of this paper consider as an important contribution also the design of a new profile that includes the SysML original specification with all aspects of the MDM-DA approach. This new profile will be available as open source software, when the copyright and the patent processes related to the ICA-MMH Project have been completed.

Some possible adjustments in the MDM-DA and the Proposed SysML extension can still occur due to the fact that the ICA-MMH Project is underway.

As future works, it is recommended to improve some steps proposed by the MDM-DA, in order to support their applications on general systems, instead of only in Software-Intensive Systems (SIS).

Some possible examples of the MDM-DA steps’ adaptation to allow the application on these general systems are: a) Classifying Systems’ Requirements (Step 5); b) Clustering Systems’ Requirements in Components (Step 7); and c) Defining Development Production Lines (Step 8).

New aspects of Systems Engineering and Requirements Engineering will be also investigated as future works aiming to complement the MDM-DA to evaluate the impact of each requirement in costs and delivery times of a system as for example: risks analysis and effort estimation.

Finally, by using the MDM-DA and the SysML extension on the ICA-MMH Project, it was possible to improve quality and efficiency of system requirements traceability, mainly because they provided verification, implementation, and testing of these requirements.

7. ACKNOWLEDGMENTS

Authors of this paper would like to thank: the Brazilian Aeronautics Institute of Technology (ITA), for its technologic and scientific development incentives; the Brazilian National Water Agency (ANA), for the opportunity of participating in the ICA-MMH Project; the Brazilian Research and Projects Financing (FINEP), for this contract opportunity; the Casimiro Montenegro Filho Foundation (FCMF), for its infrastructure availability and FINEP scholarships; and IBM for supporting its modeling tool during the project thought an academic agreement with ITA.

8. REFERENCES


