Quest Journals
Journal of Software Engineering and Simulation
Volume 10 ~ Issue 12 (2024) pp: 26-37
ISSN(Online) :2321-3795 ISSN (Print):2321-3809
www.questjournals.org

Research Paper

Decision Requirements with SysML for Industrial Processes

[1]Corina Abdelahad
National University of San Luis, Argentina
[2] Daniel Riesco
National University of San Luis, Argentina
[3]Carlos Kavka
Luxembourg Institute of Science and Technology, Luxembourg

ABSTRACT: *An important activity in system development is to ensure that all requirements are met. SysML is based on four pillars with each offering a unique perspective of the system. The diagrams used in these pillars enable requirements traceability, a feature that allows for the tracking of the lifecycle of requirements in both forward and backward directions. This traceability has an important role in Model-Based Systems Engineering. On one hand, the OMG proposed the DMN standard notation to model decision requirements. Currently, DMN is on the rise and is widely used to model decision requirements. On the other hand, the OMG proposed SysML to model system requirements and their relationships to other modeling elements, excluding decision requirements. The aim of this paper is to use SysML as a single language to model requirements, including decision requirements. To carry out this objective, it is not admissible to completely discard the DMN models already built and reconstruct the SysML requirement model, that is why it is proposed to convert these DMN diagram into a SysML requirements diagram allowing not only to have a single requirements model but also to show the relationship that these requirements have with the other requirements of the system and with the other modeling elements. Our contribution is illustrated by means of a case study of an industrial process.*
KEYWORDS: *Requirement, DMN, SysML, traceability*

R*eceived 05 Dec., 2024;* ***R****evised 12 Dec., 2024;* ***A****ccepted 17 Dec., 2024 © The author(s) 2024.*
P*ublished with open access at* ***www.questjournas.org***

I. INTRODUCTION

For some years, system development has no longer been focused on coding but on building models. These models are abstractions of reality and help engineers manage the complexity of systems by focusing only on relevant information.

Model-Based Systems Engineering (MBSE) emphasizes the use of models [1]. Requirements traceability is a crucial process in MBSE, this is defined as “the formalized application of modeling to support system requirements, design, analysis, verification and validation activities beginning in the conceptual design phase and continuing throughout development and later life cycle phases” [2].

On the other hand, INCOSE [3], together with the Object Management Group (OMG) [4], defined SysML, a standard general-purpose modeling language based on UML, which can be used to specify, analyze, design, and verify systems, including hardware, software, information, personnel, procedures, and features [5]. SysML is based on four pillars that allow the visualization of a system from four different perspectives: Requirements, Structure, Behavior, and Parametrics, each defined in terms of diagrams [6]. Requirements modeling [7] is implemented through the requirements diagram, which allows for capturing, analyzing, and maintaining the traceability of system's requirements; structural modeling includes structural diagrams; behavioral modeling involves behavior diagrams; and parametric modeling has a parametric diagram to identify system's constraints [5]. In SysML, requirements can be related to other requirements and to other modeling elements through the relationships it provides. However, this diagram is designed to model functional and non-functional requirements, leaving aside the modeling of decision requirements that appear in several systems.

The specification of business processes also requires standards for their definition. BPMN is the OMG standard and today is widely used to model business processes [8]. Although UML activity diagrams can model business processes, BPMN was specifically designed for this purpose [9], and the OMG adopted it as a core standard [8]. Since many activities within a business process involve decision-making, the OMG defined a standard specifically for decision modeling, the DMN (Decision Model and Notation). It allows the decision models to be represented, separating decision logic and control flow logic in business processes [10]. DMN was designed to be used alongside BPMN; however, it can also be used independently. DMN is already being adopted in the industry, and many tools are being developed for its modeling such as: Cardanit [11], BPMN.iO [12], Camunda [13], among others.

As the main contribution, this work presents a metamodel level correspondence and subsequently a conversion of a DMN requirements diagram into a SysML requirements diagram, allowing not only to have a single requirements model but also to show the relationship that these requirements have with the other requirements of the system and with the other modeling elements. Once the SysML requirements diagram is constructed, it is possible to follow the traceability of these requirements.

This approach can help systems engineers enhance the design of requirements, including decision requirements, understanding and covering different perspectives while refining the level of detail in the models using SysML.

The rest of this paper is organized as follows: Section 2 summarizes the basic concepts used in this paper. Section 3 describes the motivations of current research. The DMN and SysML metamodels used in this research are described in section 4. Section 5 presents the proposed approach with a case study presented in Section 6. We conclude with the conclusions in Section 7.

II. BASIC CONCEPTS

This section introduces the fundamental concepts of the proposed approach. Firstly, traceability in SysML and the requirements diagram are presented, showing their relationships with other elements belonging to other diagrams, such as with the block definition diagram. Secondly, the basic concepts of DMN notation are explained, along with its decision requirements diagram. Finally, some types of requirements and stereotypes are exposed to extend the SysML requirements, necessary to carry out this work.

2.1 SysML Requirements Diagram and its Relationships with other Elements

As previously mentioned, requirements traceability, defined as the ability to follow the life of a requirement throughout the design stages, is crucial in the MBSE methodology [14]. Modeling with SysML allows traceability because it defines relationships between requirements and other modeling elements [15] [16], since, SysML achieves traceability through its four pillars.

SysML defines a requirement as "a capability or condition that must (or should) be satisfied. A requirement can specify a function that a system must perform or a condition and/or constraint that the system must satisfy" [5]. In SysML, the requirements diagram shows all the requirements and the relationships between them. This facilitates connections with different SysML diagrams, as a requirement can appear in other types of diagrams, showing its relationship with other elements of the model.

The relationships that allow relating requirements with other requirements or with other modeling elements are [5]:

- Containment: a relationship that is used to represent how a compound requirement can be partitioned into a set of simpler requirements (denoted graphically with a circle containing a + symbol).
- «deriveReqt»: a relationship that describes that a requirement is derived from another requirement.
- «satisfy»: a relationship that describes that a design element satisfies a requirement. Usually, a requirement is satisfied by a block.
- «verify»: a relationship that connects a test case with the requirement that is verified by that test case.
- «refine»: a relationship that specifies that a model element describes the properties of a requirement in more detail.
- «trace»: a general-purpose relationship between a requirement and any other model element.

In SysML, requirements are related to blocks through the «satisfy» relationship. The block definition diagram captures the relationships between blocks as a hierarchy. Activities can be viewed as blocks, and they appear as regular blocks, except for the keyword «activity» [5]. Depending on the nature of the block, it may have an associated behavior.

Figure 1 illustrates a generic SysML requirements diagram, showing the relationships existing in it and elements belonging to other diagrams.

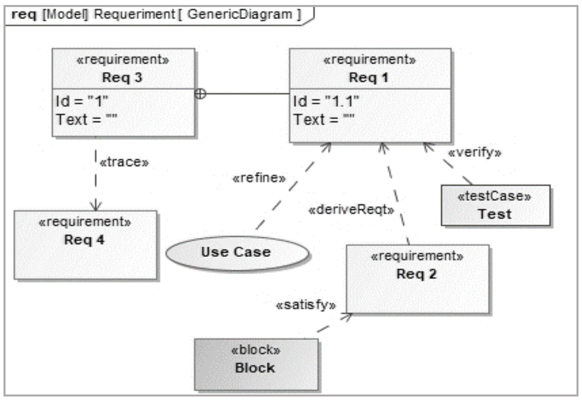


Figure 1: Generic SysML Requirements Diagram.

2.2 DMN

DMN is the OMG standard for decision modeling. This notation provides constructs for modeling decision requirements and decision logic. BPMN is an OMG standard and is used for business process modeling. It is currently considered the most understandable and used notation for this purpose. DMN was designed to complement BPMN and, as previously mentioned.

DMN defines decision requirements as: "the act of determining one or more output values from a set of input values, using decision logic that defines how the output(s) are determined from the inputs" [10]. Decision requirements are connected to a network called the Decision Requirements Graph (DRG), which shows the most important elements involved and the dependencies between them. DRGs can be represented by decision requirements diagrams. This diagram illustrates how decision requirements relate to each other, how they connect with the input data involved in these requirements, their relationship with Business Knowledge Models, and with Knowledge Sources [10].

Each element of the DMN diagram [10], [17] is detailed below:

- A decision requirement determines an output from inputs by applying some decision logic that can refer to one or more Business Knowledge Models. Decisions can be decomposed into sub-decisions.
- A Business Knowledge Model (BK) represents a function that encapsulates business knowledge.
- An input data represents information used as input for one or more decisions and Knowledge Source.
- A Knowledge Source (KS) denotes an authority for a decision or for a Business Knowledge Model. Knowledge Source represents the source of technical knowledge for making a decision, such as regulations or policies on how a decision should be made.

The relationships in the decision requirements diagram are described below:

- An information requirement denotes the input to a decision (⟶).
- A knowledge requirement denotes the invocation of a Business Knowledge Model (- - >).
- An authority requirement denotes the dependency of one element on another that will act as a Knowledge Source. It allows linking a Knowledge Source with a decision (- - -●).

Figure 2 shows a generic decision requirements diagram, highlighting the essential and most commonly used relationships between elements of DMN.

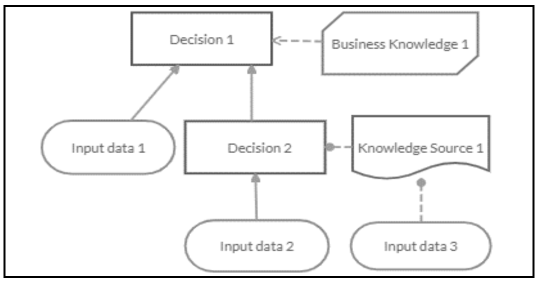


Figure 2: Generic Decision Requirements Diagram.

2.3 Requirements and stereotypes

Requirements are the basis of the development of any software system. They determine the functionalities that the system offers and specify characteristics and/or properties. That is, the requirements establish what the system should do and define the constraints that it has [18].

Requirements, depending on their type, can be classified as Functional and Non-Functional categories [19], [20]. These types of requirements are the most commonly used; however, there are other categorization models. Robertson [21] describes categories such as policies, operation, and security; the ISO/IEC 9126 standard [22] defines quality characteristics, and Robert Grady [23] presents the FURPS model, which has five groups: Functionality, Usability, Reliability, Performance, and Support.

SysML encompasses all categories with the stereotype «requirement»; however, it is frequently necessary to explicitly specify a type of requirement to enhance the model's semantics.

One extension mechanism that SysML provides is stereotypes [5]. These allow for the creation of new building blocks with additional properties and constraints. Thus, using stereotypes, it is possible to explicitly reference non-functional requirements with the stereotype «nonFunctionalRequirement» and decision requirements with the stereotype «decisionRequirement», as presented in [24].

III. MOTIVATIONS

As mentioned before, to carry out Model-Based Systems Engineering, a modeling language is necessary, and knowing it allows for sketching of design ideas, in addition, enables quick and effective communication among stakeholders.

On one hand, SysML includes a diagram for requirements modeling, while DMN provides a requirements diagram specifically designed to model decision requirements; both are OMG standards. However, when modeling requirements using two different notations complexity increases, and it becomes difficult to follow the traceability of all requirements.

Currently, DMN is gaining popularity in different areas, as evidenced by research works found such as in Medical [25] and [29], Learning technics [26], Public service [27], Internet of Things [28], among others. On the other hand, SysML is used to model system requirements and their relationships with other modeling elements, excluding decision requirements. The goal of this work is to use SysML as a single language to model all requirements, including decision requirements.

To achieve this aim, it is not admissible to completely discard DMN models already built and rebuild the SysML requirements model. Therefore, it is proposed to convert these DMN models into a SysML requirements diagram, allowing for not only a unified requirements model but also showing the relationships between these requirements and other system requirements and modeling elements.

In order to convert a decision requirements diagram into a SysML requirements diagram, it is necessary to identify not only semantic correspondences [24] but also metamodel level correspondences between these diagrams to ensure a legitimate conversion.

IV. DMN AND SYSML METAMODELS

Figure 3 shows the simplified DMN metamodel [10] with the classes involved in this work. It illustrates how decisions are related to other modeling elements. *Decisions* are connected to each other and to *InputData* through *InformationRequirement*. Through *AuthorityRequirement*, decisions are related to *Knowledge Source*, and the latter to *InputData*. And through *KnowledgeRequirement*, decisions are related to the *Business Knowledge Model*.

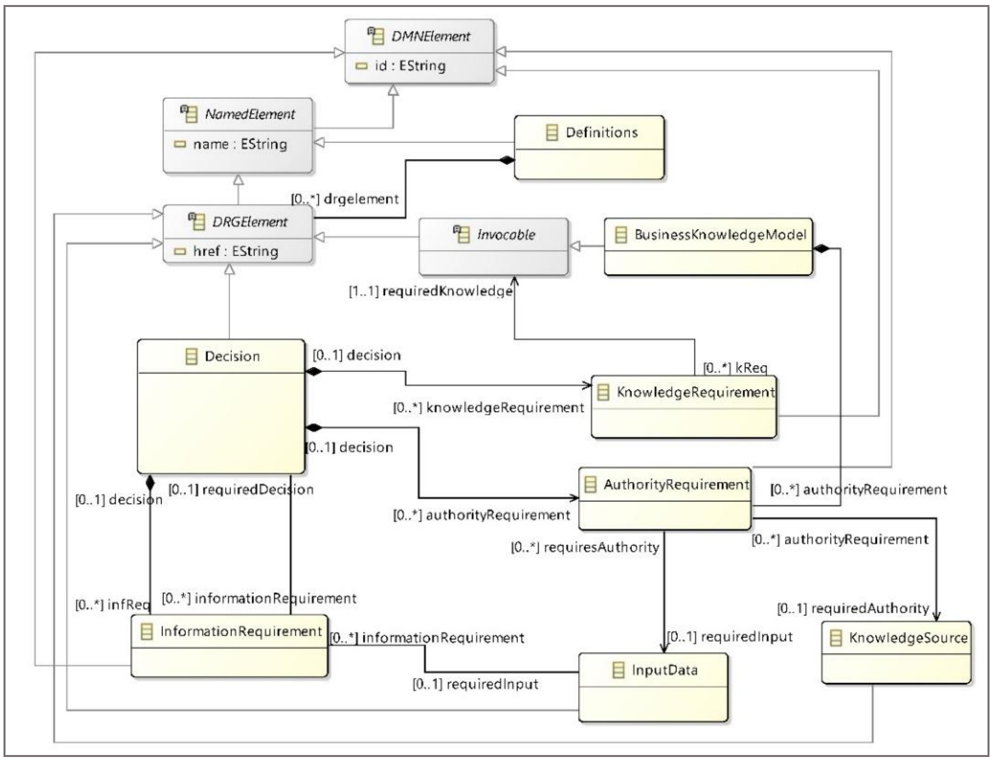


Figure 3: Simplified DMN Metamodel

On the other hand, SysML has been defined as an extension of UML, and therefore some SysML classes are instances or generalizations of UML metaclasses. In figure 4, the simplified SysML metamodel is shown with the classes involved in this article. This contains the classes along with their relationships, extracted from the SysML specification [5] and the UML specification [30]. As can be seen in the figure, the notation for extending a class is an arrow with the tip filled in from the stereotype to the extended class.

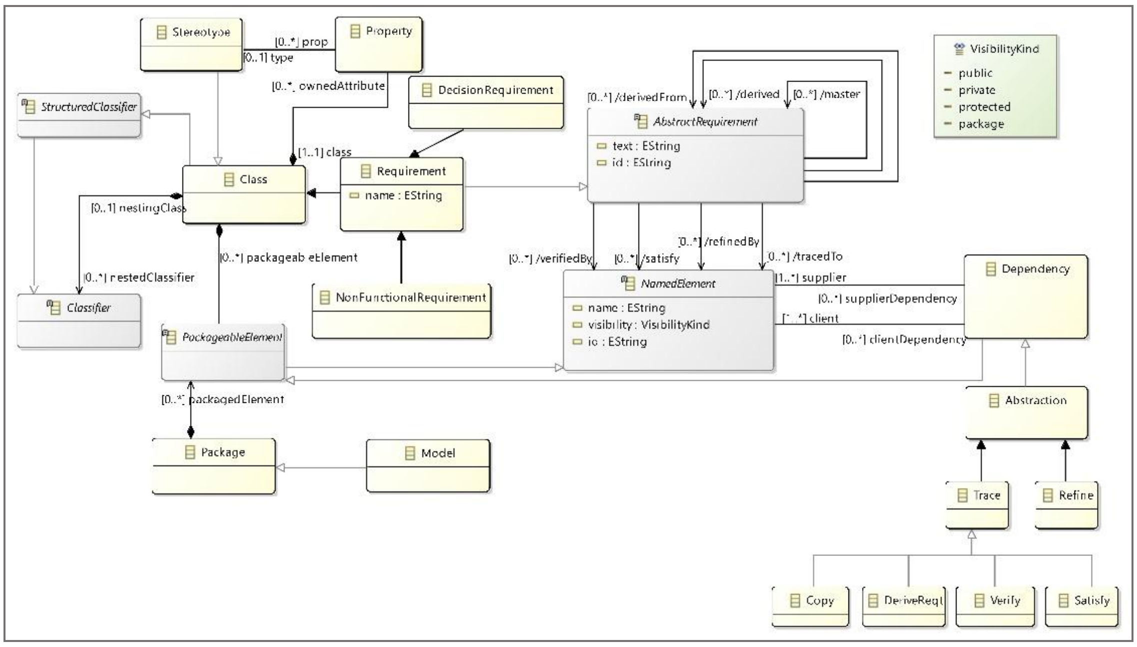
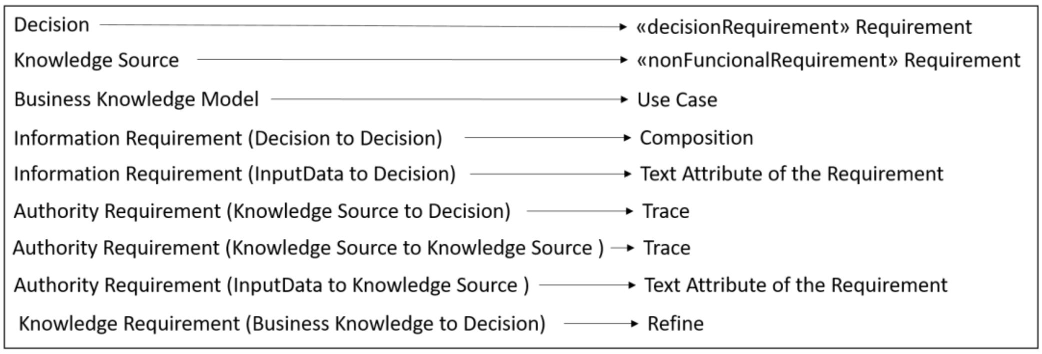


Figure 4: Simplified SysML Metamodel.

V. CORRESPONDENCES

As mentioned in Section 2, a DMN requirements diagram includes decisions, input data, knowledge sources (KS), and business knowledge models (BK). Decisions can be decomposed into sub-decisions, and according to the DMN specification, these decisions are considered requirements [10]. KS can represent company policies, regulations, etc., and based on the classification of non-functional requirements provided by Somerville in [18], we can infer that KS can be represented as non-functional requirements. BK denotes a function, meaning it corresponds to functional requirements and, therefore, can be represented as use cases in SysML. Lastly, input data refers to the information needed for decisions and for knowledge sources, and it can be reflected through the text property that each requirement has. When a KS is related to a decision, it signifies that the decision needs knowledge of its content, i.e., both have a relationship with a purpose. When a BK is related to a decision, this BK refines the decision. In the SysML requirements diagram, the relationships of purpose and refinement are represented by *trace* and *refine*, respectively, as mentioned in Section 2. The semantic correspondences proposed can be observed in figure 5.



Decision	→ «decisionRequirement» Requirement
Knowledge Source	→ «nonFuncionalRequirement» Requirement
Business Knowledge Model	→ Use Case
Information Requirement (Decision to Decision)	→ Composition
Information Requirement (InputData to Decision)	→ Text Attribute of the Requirement
Authority Requirement (Knowledge Source to Decision)	→ Trace
Authority Requirement (Knowledge Source to Knowledge Source)	→ Trace
Authority Requirement (InputData to Knowledge Source)	→ Text Attribute of the Requirement
Knowledge Requirement (Business Knowledge to Decision)	→ Refine

Figure 5: Semantics Correspondences.

Using object diagrams, the correspondences at the metamodel level will be illustrated. That is, the object diagrams shown below are instances of the metamodels. Figure 6 shows the relation between *decisions*, and as shown in the metamodel of figure 3, the decisions are related between them through *Information Requirement*. As previously mentioned, this relation among decisions in SysML is a composition relation.

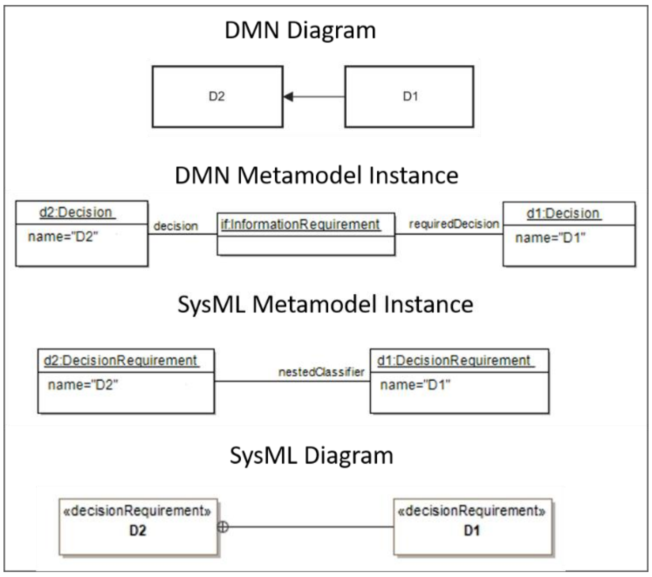


Figure 6: Information Requirement, Decision to Decision, is a Composition.

Figure 7 shows the relation between a *decision* and a *knowledge source*, and as presented in the metamodel of figure 3, this relation is through *Authority Requirement*. As previously mentioned, see figure 4, this relation in SysML is a *trace* dependency relation.

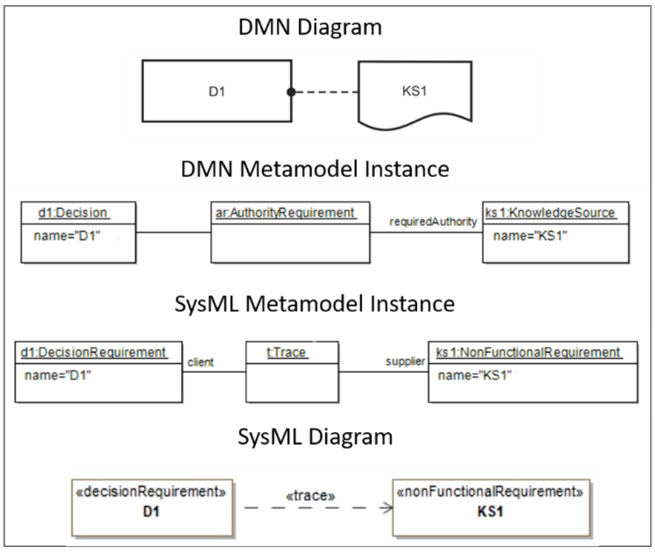


Figure 7: Authority Requirement (Knowledge Source to Decisión).

Figure 8 shows the relation among *knowledge sources*, and as exposed in the metamodel, figure 3, this relation is through *Authority Requirement*. As before mentioned, see figure 4, this relation, in SysML is a *trace* dependency relation.

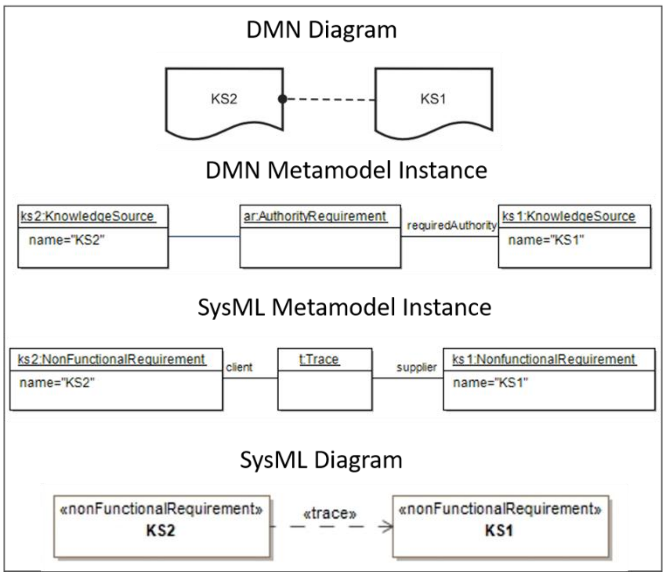


Figure 8: Authority Requirement (Knowledge Source to Knowledge Source).

Both *decisions* and *knowledge sources* can have *input data*, and as mentioned above, this *input data* refers to necessary information, and it can be reflected through the *text* property that each requirement has. These correspondences can be seen in figures 9 and 10.

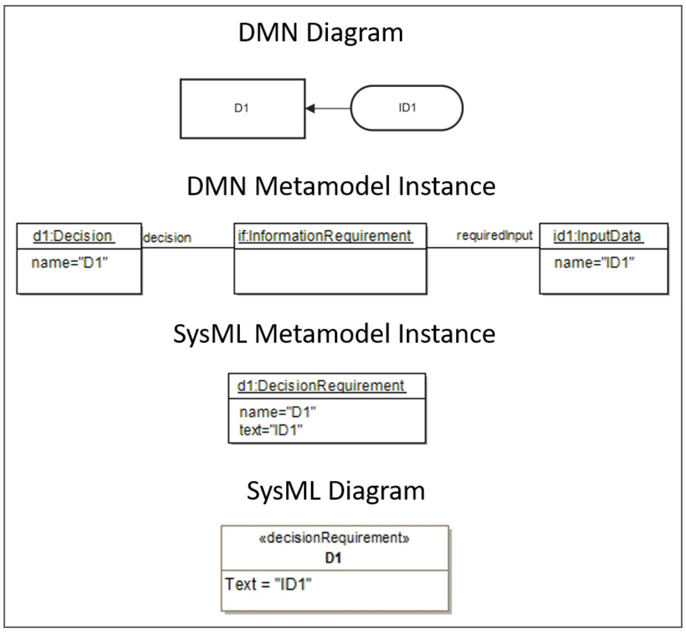


Figure 9: Information Requirement (InputData to Decision).

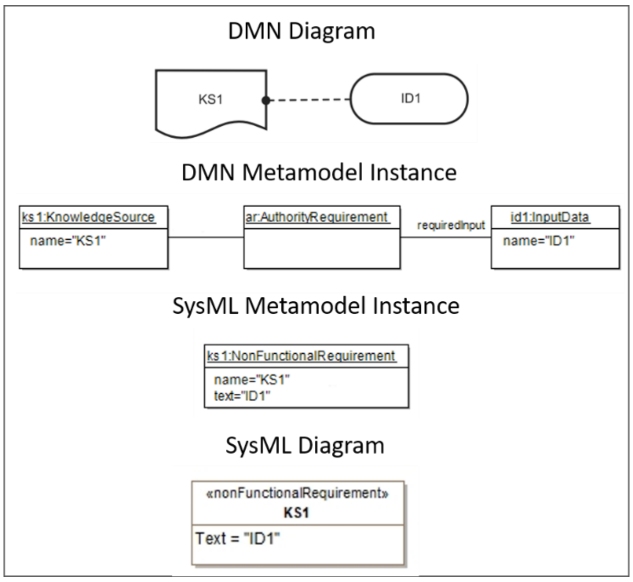


Figure 10: Authority Requirement (InputData to Knowledge Source).

Through a case study, the correspondences at the metamodel level will be shown. That is, how a DMN requirements diagram can be converted into a SysML requirements diagram and subsequently the traceability of these requirements.

VI. CASE STUDY

As previously mentioned, requirement traceability is understood as the ability to follow the lifecycle of a requirement. The most important step when building a system is to consider all types of requirements.

Traceability will be illustrated through forward engineering, focusing mainly on requirements that involve decisions, showing the blocks with which they are related and that have a particular behavior, which will be modeled with BPMN. The aim is to integrate and cover its different views.

The example selected to illustrate traceability is a case study of Alfalfa Production for their own use and for export. Figure 11 shows the DMN decision requirements diagrams, which represent the decisions to carry out the Alfalfa Production requirement.

As can be observed, for sowing, a) the extended weather forecast and the previous crop must be considered. It is important to note that sowing is not done at any time of the year, considering the Southern

Hemisphere, sowing is only done in March/April and/or September. The extended forecast is crucial because rainfall is required in the months following sowing.

Once the sowing has been done, wait a few months and evaluate the percentage of effectiveness b). Depending on the values, different decisions are made (replanting, if possible, due to the weather, or continuing to wait for the plant to grow). This percentage is determined by agronomists.

Regarding the cutting requirement d), other decisions and a knowledge source (KS) are involved. To cut, it is necessary to analyze: How tall is the plant? How many leaves does it have? What is the flowering percentage? What percentage of humidity does it have? Is the protein content high, medium or low? What is the color of the plant?

As for quality standards (KS), the plant needs to have 14% humidity and 18% protein, which is acceptable for export, and the color must be bright green.

For baling c), it is necessary to decide what type of machinery will be used according to what is needed, as the machines can generate into cubes or rolls.

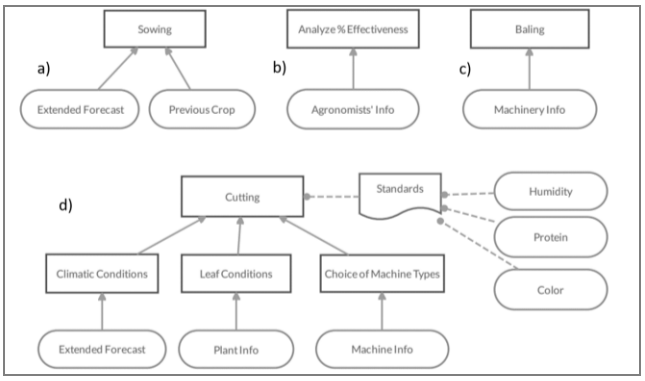


Figure 11: DMN DRD for Alfalfa Production.

To convert a DMN diagram into a SysML diagram, the process will first be shown with object diagrams, instances of the metamodels, and the correspondences found when analyzing metamodels; see figure 12 and figure 13. For reasons of length and as an example, only point d) of figure 11 will be shown.

In figure 12 it can be seen that the *InformationRequirements* if1, if3, and if5 are used to relate *decisions*, as mentioned above, this represents the composition of decisions and therefore the composition of requirements. For example, *InformationRequirement* if1 specifies that the Cutting decision needs the Climatic Conditions decision.

The *DataInputs* related by *InformationRequirement* if2, if4, and if6 to the decisions are modeled through the text attribute that the requirements have. For example, in figure 13, the requirement related to Climatic Conditions shows in the *text* field the input data: Extended Forecast, necessary to make the decision.

In figure 12, object a1 links a *decision* with *KnowledgeSource* ks1, as mentioned above; this is modeled with the SysML trace relation. The *DataInputs* related to the *KnowledgeSource* ks1, through the *AuthorityRequirements*, are also modeled with the *text* attribute of the requirement; see figure 13.

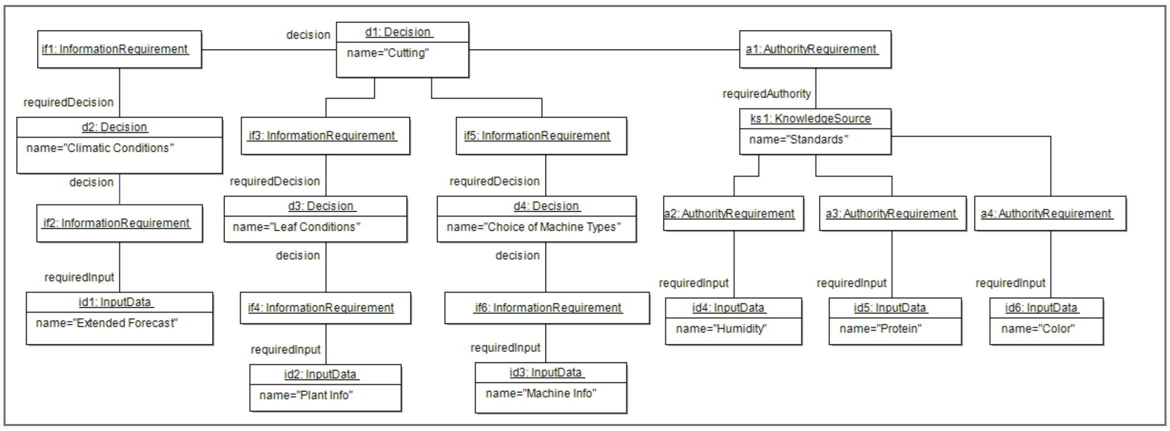


Figure 12: Object diagram DMN Cutting.

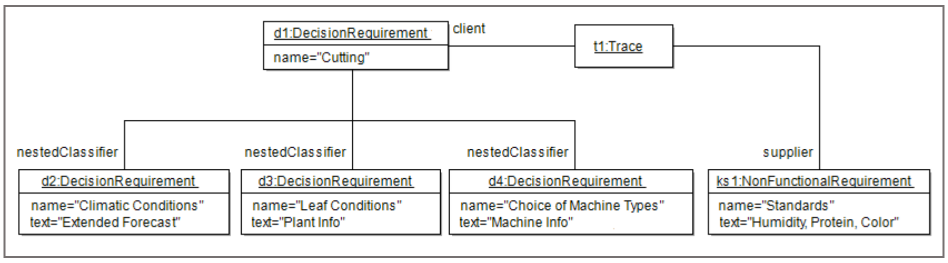


Figure 13: Object diagram SysML Cutting.

Once the metamodel level correspondences by object diagrams have been shown, the DMN requirements diagram for the case study is converted into a SysML requirements diagram; see figure 14.

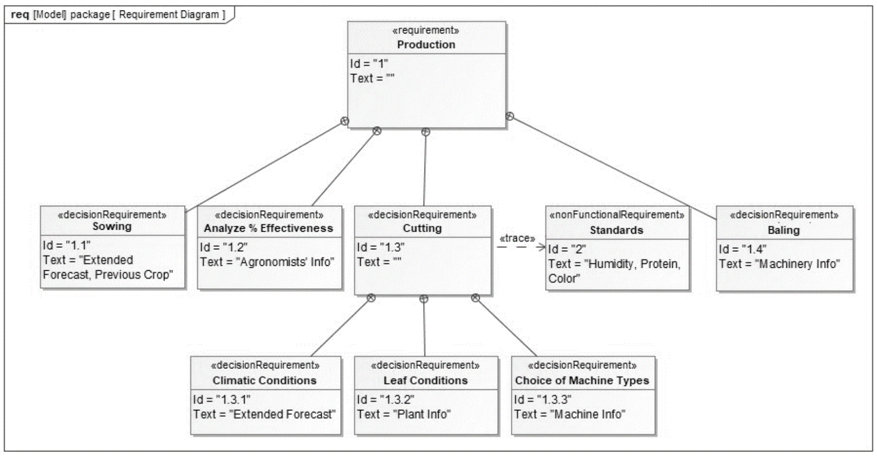


Figure 14: SysML Requirements Diagram for Alfalfa Production.

Once the model is built, the software engineer can add all the necessary relationships. As previously mentioned, requirements can be related to other modeling elements through a specific set of relationships.

Once the main requirements have been captured, the elements responsible for satisfying these requirements are modeled through a block definition diagram, structural view. As mentioned in Section 2, activities can be viewed as a block, except for the keyword «activity» [5]. In addition, depending on its nature, a block may have an associated behavior (see figure 15). It is important to mention that we emphasize requirements that involve decisions. The «satisfy» relationship links a block to a requirement. The interpretation of this relationship is that the block's design depends on the requirement, which means that if the requirement changes, the design of the block must be changed [5].

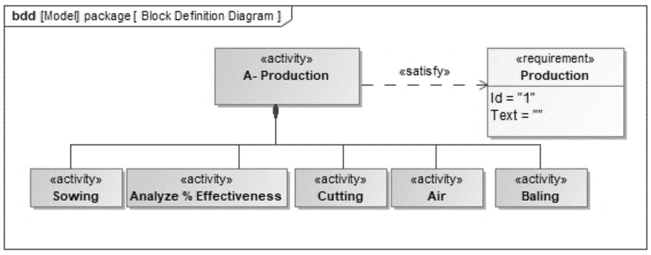


Figure 15: Block Definition Diagram – Production Activity.

Figure 15 illustrates how the Production activity is composed of the following activities: Sowing, Analyze % Effectiveness, Cutting, Air, and Baling. It is important to mention that these activities fulfill the requirements presented in the SysML requirements diagram in figure 14.

Following the approach presented in this work, a BPMN model is built to cover the different views of the requirements, in this case, the behavioral view. This process model is built and associated with the Production activity, showing the steps to follow to carry out alfalfa production. Its behavior can be observed in figure 16.

The process begins with sowing, and after several months, the effectiveness of the sowing is analyzed. If the effectiveness does not reach 70%, resowing is required; otherwise, the plant is allowed to grow. Once it reaches the expected height, it is cut, then aired, and once it is fully dry, it begins to be baled.

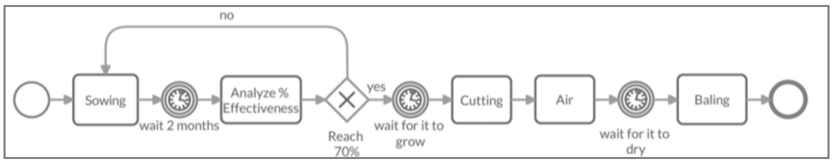


Figure 16: BPMN Alfalfa Production.

To conclude our approach, figure 17 illustrates the traceability process carried out for the case study. In the figure, the DMN requirements diagram and the SysML requirements diagram can be seen. Once the requirements diagram is built, the elements responsible for satisfying these requirements are modeled through a block definition diagram, structural view. Finally, it is shown how, based on this diagram, a BPMN is built with the events and activities required to carry out the process.

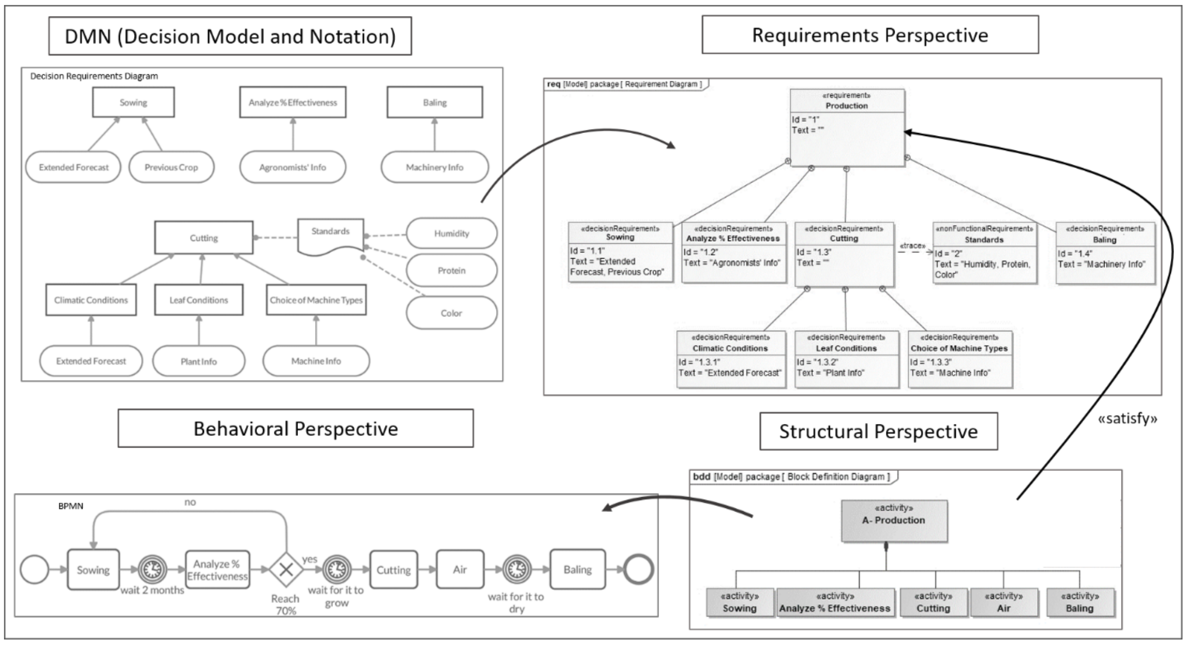


Figure 17: Case Study Traceability.

VII. CONCLUSION

Models allow us to manage complexity, visualize, and abstract multiple aspects of the system to be developed, which is why their use has increased in recent years. Based on this premise, Model-Based Systems Engineering (MBSE) is a methodology that emphasizes the use of models to carry out systems engineering activities, and traceability plays an important role in this type of methodology.

SysML is a general-purpose modeling language, based on UML, that allows the requirements traceability by defining relationships between them and with other modeling elements. SysML supports general requirements modeling through the «requirement» stereotype. BPMN was specifically designed for modeling processes, making the combination of SysML and BPMN attractive and improving the modeling of the different views of the system being developed.

On the other hand, DMN is the notation used for modeling decision requirements through the decision requirements diagram. However, using different notations to support all types of requirements increases the complexity of system development.

The objective of this work was to convert a DMN requirements diagram into a SysML requirements diagram by previously analyzing the correspondences at the metamodel level of both languages and to illustrate the traceability of these requirements. This work aims to improve the design by using a single language for requirements modeling and to understand all views of these requirements, including decision requirements. The approach was illustrated through a case study.

REFERENCES

[1] T. Stahl, M. Völter, and K. Czarnecki, Model-driven software development: technology, engineering, management, John Wiley & Sons, Inc, 2006.
[2] INCOSE Technical Operations, Systems Engineering Vision 2020, No. INCOSE-TP-2004-004-02. Seattle, WA: International Council on Systems Engineering, version 2.03 Edition, 2007.
[3] INCOSE https://www.incose.org/, 2024.
[4] OMG https://www.omg.org/, 2024.
[5] SysML https://www.omg.org/spec/SysML/1.6/PDF, 2024.
[6] S. Friedenthal, A. Moore, and R. Steiner, A Practical Guide to SysML: The Systems Modeling Language, Elsevier, 2014.
[7] P. Spoletini and A. Ferrari, Requirements elicitation: a look at the future through the lenses of the past, In 2017 IEEE 25th International Requirements Engineering Conference (RE), 2017, pp. 476-477.
[8] Business process model and notation https://www.omg.org/spec/BPMN/2.0.2/PDF, 2024.
[9] D. Q. Birkmeier, S. Klöckner, and S. Overhage, An Empirical Comparison of the Usability of BPMN and UML Activity Diagrams for Business Users, ECIS 2010 Proceedings 51, 2010. [Online]. Available: https://aisel.aisnet.org/ecis2010/51/
[10]Decision Model and Notation https://www.omg.org/spec/DMN/1.2/, 2024.
[11]Cardanit https://www.cardanit.com/
[12]BPMN.iO https://bpmn.io/toolkit/dmn-js/
[13]Camunda https://camunda.com/dmn/
[14]J. Jacobs and A. C. Simpson, Towards a Process Algebra Framework for Supporting Behavioural Consistency and Requirements Traceability in SysML, In Proceedings of the 15th International Conference on Formal Engineering Methods (ICFEM 2013), Lecture Notes in Computer Science, Springer, vol. 8144, 2013, pp. 265-280
[15]O. C. Z. Gotel and A. C. W. Finkelstein, An Analysis of the Requirements Traceability Problem, Proceedings of IEEE international conference on requirements engineering. IEEE, 1994, pp. 94-101,
[16]K. Hampson, Technical evaluation of the systems modeling language (SysML), Procedia Computer Science, vol. 44, 2015, pp. 403-412,
[17]B. Silver, DMN Method and Style. 2nd Edition: A Business Practioner's Guide to Decision Modeling ISBN-13:978-0982368176. Cody-Cassidy Press, 2018.
[18]I. Sommerville, Software Engineering, Seventh Edition, Pearson Education, 2005.
[19]G. Booch, I. Jacobson, and J. Rumbaugh, The Unified Process Software Development, 1999.
[20]J. Holt and S. Perry, SysML for systems engineering, vol.7, IET, 2008.
[21]S. Robertson, and J. Robertson, Mastering the requirements process: Getting requirements right, Addison-Wesley, 2012.
[22]ISO 9126-Software Engineering: Product Quality, 1991.
[23]R. B. Grady, Practical Software Metrics for Project Management and Process Improvement, Prentice Hall, Englewood Cliffs, NJ, USA, 1992.
[24]C. Abdelahad, D. Riesco, and C. Kavka, Transformando Diagramas de Requerimientos de Decisión DMN en Diagrama de Requerimientos SysML, CONAIISI, 2021.
[25]M. Berkhout, K. Smit, and J. Versendaal. Decision discovery using clinical decision support system decision log data for supporting the nurse decision-making process. BMC Medical Informatics and Decision Making, 2024, vol. 24, no 1, p. 100.
[26]A. Goossens, J. De Smedt and J. Vanthienen. Extracting Decision Model and Notation models from text using deep learning techniques. Expert Systems with Applications, 2023, vol. 211, p. 118667.
[27]T. Tavantzis, R. Promikyridis and E. Tambouris. Towards exploiting BPMN and DMN in public service modeling. In Proceedings of the 27th Pan-Hellenic Conference on Progress in Computing and Informatics. ACM. 2023. p. 211-216.
[28]Y. Kirikkayis, F. Gallik and M. Reichert. Modeling, executing and monitoring IoT-driven business rules with BPMN and DMN: current support and challenges. In International Conference on Enterprise Design, Operations, and Computing. Cham: Springer International Publishing, 2022. p. 111-127.
[29]A. Bianchi, et al. Putting BPMN and DMN to Work: a Pediatric Surgery Case Study. In 2021 IEEE International Conference on Digital Health (ICDH). IEEE, 2021. p. 154-159.
[30]UML https://www.omg.org/spec/UML/2.5.1/PDF, 2024.