A SysML-based Approach to Requirements Traceability using BPMN and DMN

Corina Abdelahad and Daniel Riesco
Departamento de informática
Universidad Nacional de San Luis
San Luis, Argentina
e-mail: cabdelah@unsl.edu.ar, driesco@unsl.edu.ar

Carlos Kavka
Research and Development Department
ESTECO SPA
Trieste, Italy
e-mail: kavka@esteco.com

***Abstract*—Model-based system engineering is a well-established methodology where the use of models has a central role. One of its main contributions is to support the documentation of the requirements and decisions that are taken during the design process. In particular, requirements traceability, or in other words, the capability to follow the life-cycle of the requirements, plays an important role in this methodology. Of course, system engineers' work in this area is supported by the use of standards. One of the most used standards in this domain is Systems Modeling Language (SysML), which being defined as an extension of the well-known and widely-used Unified Modeling Language (UML) standard, supports the definition and specification of requirements and their relations with the other components of the whole system. The main goal of this paper is to present an extension of SysML for requirements traceability particularly useful in the design of systems where decision-making activities are essential. The proposed extension to SysML, defined by following its standard extension mechanism, supports the association of processes and decision-making activities to system requirements, greatly improving their traceability. Processes and decision-making activities are defined in terms of the widely-used Business Process Model and Notation (BPMN) and Decision Model and Notation (DMN) standards, respectively. Our contribution is illustrated by means of a small case study.**

Keywords-SysML; BPMN; DMN; requirements traceability.

I. INTRODUCTION

Abstraction is one of the main tools used by engineers to deal with complexity, permitting them to focus only on the information that is considered significant or relevant. Based on it, Model-Based Systems Engineering (MBSE) is a successful methodology for the design of complex systems, which emphasizes the use of models when performing systems engineering activities [1]. These models, which can be executable or not, are used to describe the structure and the behavior of the systems.

With the evolution of systems engineering, the need for a consistent standard modeling language arose. International Council on Systems Engineering (INCOSE) together with the Object Management Group (OMG) [2] defined SysML, a general-purpose modeling language based on UML, which can be used for specifying, analyzing, designing, and verifying complex systems, including hardware, software, information, personnel, procedures, and facilities [4]. SysML is based on the so-called four pillars, which give the possibility to view a system from four different perspectives: Requirements, Structure, Behavior and Parametrics, each one of them defined in terms of diagrams [3]. Requirements modeling [20] is implemented in terms of the requirement diagram, which allows for capturing, analyzing and maintaining traceability of requirements in the modeled system. Structure modeling has a block definition diagram as the main diagram, representing structural elements (blocks) with their properties, relationships, and composition. Behavior modeling has different kinds of behavior diagrams like activities diagram, state machine, and sequence diagram. Parametric modeling has a parametric diagram that can be used to identify the system constraints [4]. In SysML, requirements can be related to other requirements, as well as to other model elements via one or more relationships, making possible the traceability of requirements. Furthermore, SysML can be integrated into other tools including spreadsheets and design and simulation software, such as Matlab or Modelica [19], enabling requirements verification.

The specification of business processes also followed the same path requiring standards for its definition. In particular, the Business Process Management Initiative (BPMI) together with the OMG developed the widely used BPMN notation for modeling business processes [5]. BPMN defines an abstract representation for the specification of business processes, which can include human intervention or not. BPMN couples an expressive graphical representation with a rigorous Extensible Markup Language (XML) encoding of processes and the interactions among them, supporting not only modeling activities but also process execution by using appropriate BPMN engines. Since many activities within a business process involve decision-making, the OMG defined recently the DMN standard for the elicitation and representation of decision models, effectively separating decision logic and control flow logic in business processes [6]. DMN was designed to be usable alongside the standard BPMN. At present, many companies have adopted BPMN not only because of its popularity, but because it is strongly related to DMN. This standard is already receiving adoption in the industry, with many tools being developed to assist users in modeling, checking, and applying DMN models.

As the main contribution, this work presents an innovative approach to enhance requirements traceability in the context of MBSE, by combining SysML, BPMN and DMN. This approach can help systems engineers to improve the design of requirements, to understand and cover their different views, improving maintenance and verification activities while contributing to refine the level of detail of the models.

The rest of this paper is organized as follows: Section II introduces related work, while Section III summarizes the basic concepts used in this paper. Section IV addresses the proposed approach with a case study presented in Section V. Section VI shows the conclusion.

II. RELATED WORK

Several works in the field of software engineering are related to the concept of requirements traceability using SysML. For example, the authors in [7] show how requirements traceability for mechatronic design can be achieved using MBSE and SysML. SysML is used for linking system requirements to the system elements of the different domain models while guaranteeing the traceability. This paper presents a case study of a mechatronic system in order to show this traceability.

In [8], a solution for SysML model verification and validation in an industrial context is presented. The authors provide a method and a list of the existing challenges; besides that, they show experimental results. A case study is presented, verification rules are in Object Constraint Language (OCL), while the validation rules are in a formal text format evaluated by a script. The authors mention that the verification of these rules ensures a certain degree of traceability.

In [9], an approach to construct true model-based requirements in SysML is presented. This approach proposes that every requirement can be modeled as an input/output transformation. This proposal uses SysML behavioral and structural models and diagrams, with specific construction rules derived from Wymore’s mathematical framework for MBSE and taxonomies of requirements and interfaces. The authors consider that this proposal provides several benefits, including traceability, and improved precision over the use of natural language.

In [10], the authors propose a model-based approach to automotive requirements engineering for the general development of vehicles of passengers. The SysML requirement element is extended, through stereotype, to functional and non-functional requirements. The paper validates the advantages that include classified and modeled requirements graphically, as well as their relationships that are explicitly mapped. This article presents a case study that shows the proposed extension and the performed requirements traceability.

In [11], the authors propose a model-driven requirement engineering approach for the embedded software domain. This approach is based on UML, MARTE and SysML standard notations, which are integrated in order to improve requirements specification and traceability. MARTE is used to allow domain-specific non-functional requirements to improve the software specification and SysML is combined with UML/MARTE models to support requirements management, to follow their changes. The approach is illustrated by means of a case study.

In [12], the authors propose a metamodel, which establishes the traceability links among the requirement model, the solution model and the verification and validation model for embedded system design. This approach enables traceability of requirements by considering heterogeneous languages for modeling and verifying real-time embedded systems. A case study illustrates the approach with the use of languages such as SysML, MARTE, SIMULINK, among others.

However, to the best of our knowledge, no research work about requirements traceability has considered the decision requirement, a kind of requirement that involves decision making. This requirement appears in the decision requirement diagram, which represents human decision making or automated decision making within a process. The main motivation of this work is the need to provide support to decision requirements, by offering adequate tools to the system engineers that improve the design and handling of these types of requirements. Considering this, the approach presented in this paper is a step forward to support decision requirements, completing the different system engineering views by combining of SysML, BPMN and DMN.

III. BASIC CONCEPTS

This section presents the basic concepts on which the proposed approach is based. Section A introduces traceability related concepts in SysML, Section B describes the SysML requirements diagram and block definition diagram used in the approach. Section C shows some concepts about DMN and its relationship with BPMN.

A. Traceability in SysML

In [13], INCOSE indicates that “requirements traceability refers to the ability to describe and follow the life of a requirement in both a forward and backward direction along the design stages”. Traceability plays an important role as part of any MBSE methodology [1]. MBSE emphasizes the use of models to perform the systems engineering activities, as mentioned before. In fact, “MBSE is 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” [14].

Modeling with SysML allows good traceability because it defines relationships between requirements and among other modeling elements [15]. Figure 1 describes the approach in which SysML accomplishes traceability by means of the 4 pillars presented in Section I. This figure shows the system model as an interconnected set of model elements. The arrows that cross the pillars, as seen in Figure 1, illustrate how the different elements belonging to the different types of diagrams that participate in the pillars are related, supporting requirements traceability.

 ISBN: 978-1-61208-752-8

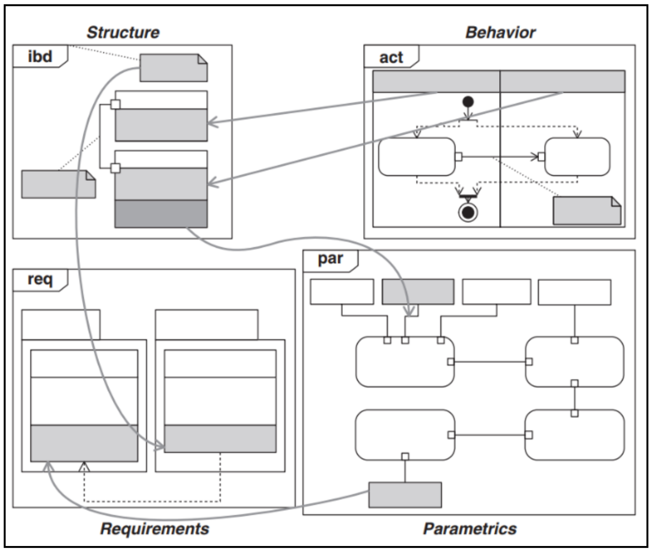

Figure 1. A system model example in SysML where requirements traceability is indicated with the connecting arrows (from [3]).

B. SysML Requirements diagram

In SysML, the requirements diagram shows the set of requirements and the relationship between them. A requirement specifies a function that must be satisfied or a condition that a system must achieve. Requirements modeling provides a bridge among different SysML diagrams because a requirement can appear on other diagrams to show its relationship to other modeling elements. The relationships that allow relating requirements with other requirements or with other modeling elements are [4]:

- Containment: a relationship which 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 which describes that a requirement is derived from other 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 which 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.

The requirements are related to the blocks through the relationship «satisfy», as mentioned before. The block definition diagram captures the relation between blocks, such as a block hierarchy. Since the activities can be seen as a block, they can have associations between each other, including composition associations. Activities in block definition diagrams appear as regular blocks, except for the «activity» keyword [4].

SysML enables characterization of any type of requirements for the system, including user, technical or others. A modeler can then define relationships between the specified requirements, providing the opportunity to create traceability among them. There is also an opportunity to create traceability from the logical and structural architecture design to their requirements, one of the most critical activities in systems engineering [16].

C. BPMN and DMN

The OMG provides the DMN notation for modeling decisions, which is not only understandable to stakeholders but it is also designed to be used in conjunction with the BPMN standard notation [6].

DMN provides constructs to both decision requirements and decision logic modeling. For decision requirements modeling, it defines the concept of Decision Requirements Graph (DRG) depicted with the Decision Requirements Diagram (DRD). This latter shows how a set of decisions depends on each other, on input data, and on business knowledge models. A decision element determines an output from the inputs, using decision logic, which may reference one or more business knowledge models. This denotes a function encapsulating business knowledge, e.g., as business rules, a decision table, or an analytic model. A decision table is a representation of decision logic, based on rules that determine the output depending on the inputs [6]. Decision-making modeled in DMN may be mapped to BPMN tasks or activities (Business Rules) within a process modeled with BPMN. The combined use of both thus provides a graphical language for describing decision-making, i.e., the BPMN tasks involving a decision can invoke a DMN decision model.

IV. MBSE AND REQUIREMENTS TRACEABILITY WITH SYSML

In this section, our contribution of traceability of requirements using SysML, BPMN, and DMN is detailed. Section A presents an extension to SysML for BPMN tasks while Section B describes details on the proposed approach for requirements traceability.

A. SysML extensions for BPMN tasks

In order to support the modeling of BPMN tasks in SysML, the element of SysML block diagram must be extended through stereotypes. The stereotypes are one of the extensibility mechanisms of UML, therefore also of SysML, that enable to extend its vocabulary allowing the creation of new kinds of building blocks that are derived from existing ones but specific to a problem [17]. Stereotypes are shown as text strings surrounded by the symbols “« »” [18]. The stereotypes change or add semantics to a base SysML element.

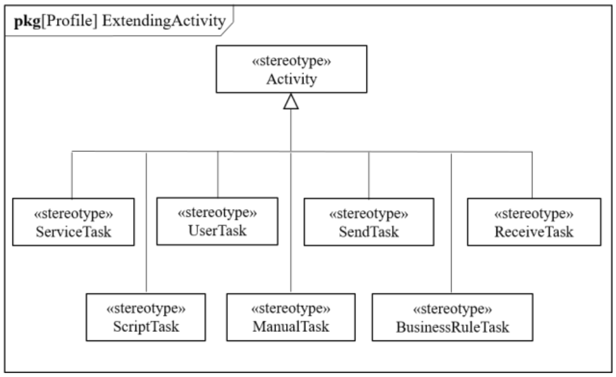

Figure 2. Extension of the SysML «Activity» stereotype.

Figure 2 shows the proposed extension of the SysML «Activity» stereotype in order to support all types of BPMN tasks [4].

This extension consists of the following stereotypes:

- «serviceTask»: represents a task that uses a web service or an automated application.
- «sendTask»: represents a simple task that is designed to send a message to an external participant.
- «receiveTask»: represents a simple task that is designed to wait for a message to arrive from an external participant.
- «userTask»: represents a task where a person performs the task with the assistance of a software application.
- «manualTask»: represents a task that is expected to be performed without the aid of any business process execution engine or any application.
- «scriptTask»: represents a task executed by a business process engine.
- «businessRuleTask»: represents a task that involves decision-making.

The business rule task was defined in BPMN as a placeholder for (business-rule-driven) decisions, being the natural placeholder for a decision task [6].

B. Requirements Traceability using SysML and BPMN-DMN

The interaction between the process and the decision models plays a crucial role because a decision can affect the process behavior or flow [5]. Therefore, it is important that decision-making must be considered as a requirement that should be performed and satisfied.

The approach will be illustrated with an example intended to carry out the traceability of the requirements through forward engineering, mainly focusing on those requirements involved in the decision-making activities, with the aim of integrating and covering their different views.

As it was mentioned before, the SysML requirement diagram has several relationships used to connect requirements. For example, Figure 3 presents a SysML requirements diagram labeled "Example Requirements Diagram", which shows the relationship between requirements. In particular, it can be observed that the requirement with *id="2"* has a relationship with the requirement with *id="1.2"* through the «deriveReqt» relation. This relation specifies that the requirement with *id="2"* is derived from the requirement with *id="1.2"*.

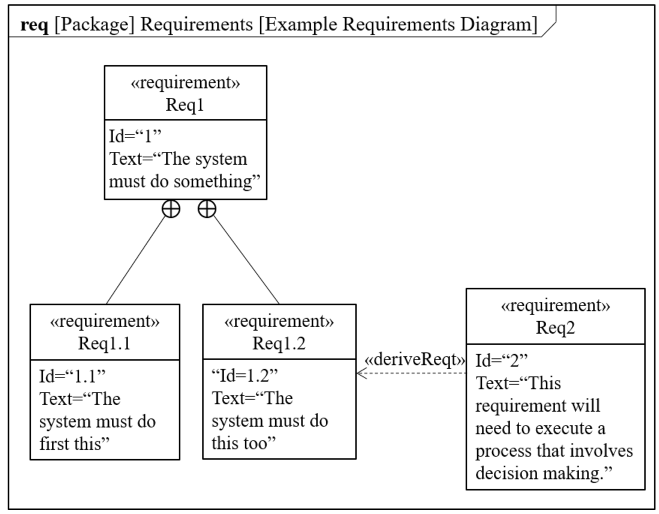

Figure 3. An example of a requirements diagram.

Requirements can be related to other requirements and to other modeling elements through a specific set of relationships as mentioned before. Relationships between requirements and other modeling elements can appear on various types of diagrams. Figure 4 presents an example of a «satisfy» relationship between a *SimAct* activity and the *Req2* requirement which appears in the requirement diagram presented in Figure 3. The interpretation of this «satisfy» relationship is that the design of the activity depends on the requirement, meaning that if the requirement changes, the design of the activity must be changed.

Once the main requirements have been captured, the elements responsible to satisfy those requirements are modeled through a block definition diagram. As previously mentioned, activities can be seen as a block, except for the «activity» keyword [4]. This later provides a means for representing activity decomposition. Figure 5 shows an example of decomposition of the *SimAct* activity which was presented in Figure 4 by using the stereotypes proposed in Section IV-A. The block definition diagram shown in this figure indicates that the *SimAct* is an activity composed of other activities, including *Task 1*, *DecisionMaking Task* and *Task 2*, all these activities being of BPMN activity types.

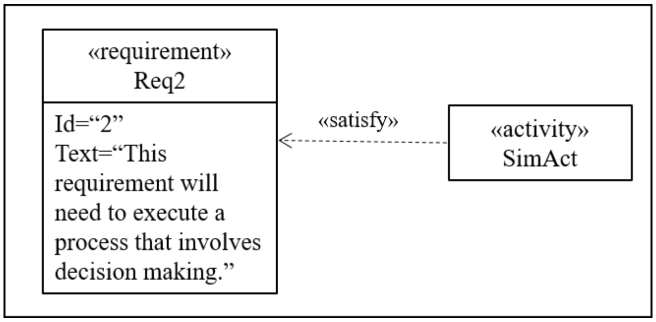

Figure 4. Example of a «satisfy» relationship between an Activity and a Requirement.

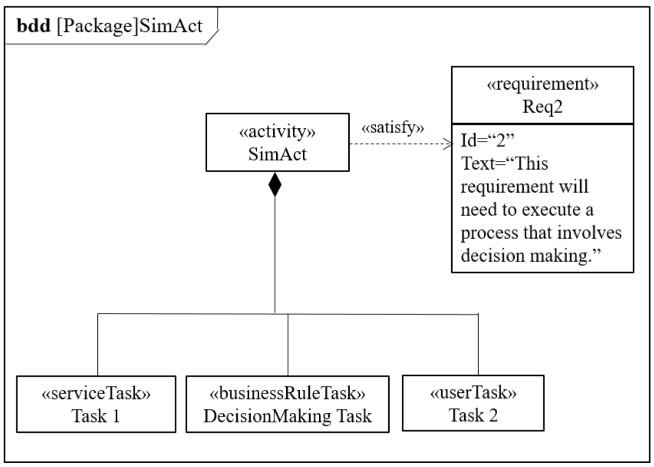

Figure 5. Block definition diagram with activities as blocks.

Finally, in order to cover the different views of the requirements, a BPMN model is constructed and associated to *SimAct* activity in order to show its behavior, as can be seen in Figure 6. In this figure, the activities that compose *SimAct*, which were modeled in Figure 5, are explicitated in BPMN format.

To conclude, the decision model related to the business rule task is built, since when BPMN and DMN are used, the BPMN tasks (business rule) have a link associated to the decision model, as mentioned in Section III-C. The *DecisionMaking Task* can then be implemented in terms of the associated decision requirements diagrams and decision tables.

V. CASE STUDY

To demonstrate our approach, we conducted a case study that includes the partial modeling of a Biodiesel Distiller and its requirements management. This case study illustrates how to carry out the traceability of the requirements through forward engineering.

Biodiesel is a type of biofuel that is similar to petroleum-based diesel, which can replace fossil fuel diesel. It is a sustainable fuel that is produced from fatty acids derived from animals such as beef fat, pork fat, chicken fat; and vegetable oils such as corn oil and cooking oil like those from restaurants that have already been used and disposed of. These oils are converted to diesel fuel through a chemical process.

Distilled biodiesel is a clean fuel that has been purified through the process of distillation. Biofuel distillation is a method that consists of taking a biofuel and removing particles and impurities within the liquid through an evaporation and condensation process.

The requirements diagram in Figure 7 illustrates the breakdown of the Biodiesel Distiller's requirements into a hierarchy of more refined requirements. This diagram named "Biodiesel Distiller Requirements Diagram" shows the relationship between its elements. In particular, it can be observed that the requirement *Initial Statement* is partitioned into a set of simpler requirements: *Generate Biodiesel*, *Heat Exchanger*, *Boiler*, *Biodiesel Properties* and *Distill Water*.

Once the main requirements of *Biodiesel Distiller* have been captured, the elements responsible to satisfy them are modeled through a block definition diagram. As previously mentioned, activities can be seen as a block and as a set of requirements that can be related to other requirements and to other modeling elements through a specific set of relationships. Figure 8 illustrates how the *Generator* activity satisfies the *Generate Biodiesel* requirement which appears in the requirement diagram presented in Figure 7, also showing how the *Machine* activity is composed.

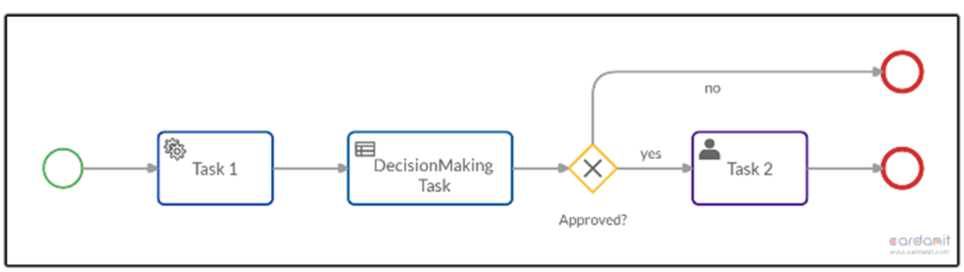

Figure 6. BPMN diagram of SimAct.

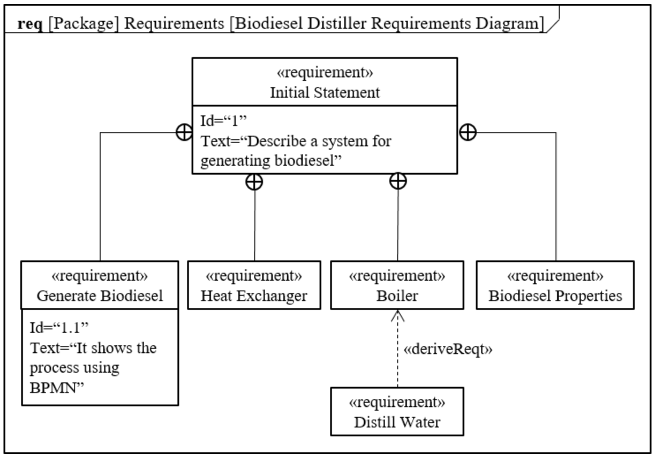

Figure 7. Requirements diagram: *Biodiesel Distiller Requirements Diagram.*

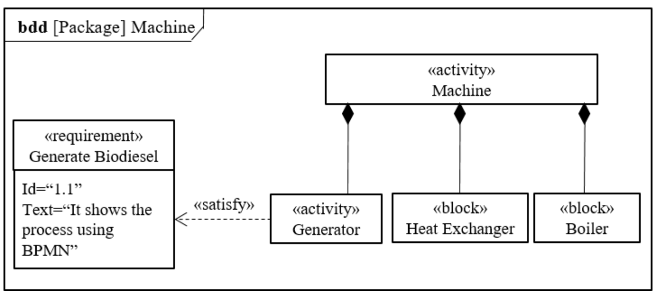

Figure 8. *Generator* activity satisfies the *Generate Biodiesel* requirement.

Continuing with the approach, Figure 9 illustrates the decomposition of the *Generator* activity by using the stereotypes proposed in Section IV-A. The block definition diagram shown in this figure indicates that the Generator is an activity composed of other activities such as: *Choose method* business rule, which involves decision-making, *Prepare reactors* activity, *Notify* user task, *Heat Water* service task, *Separate materials* service task, *Decant* activity and *Washing*

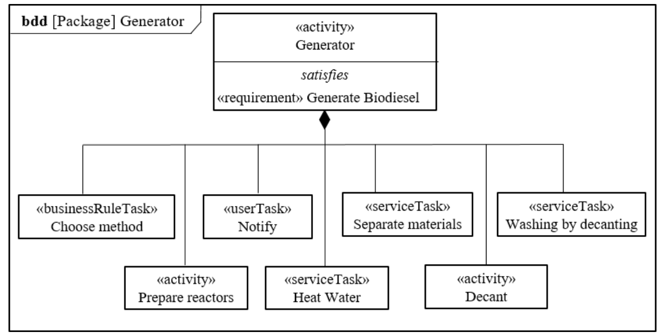

Figure 9. Decomposition of the *Generator* activity.

by decanting service task, all these activities being of BPMN activity types.

Following the approach presented in this work, a BPMN model is constructed in order to cover the different views of the requirements. This model shows the process which is carried out to generate biodiesel associated with the *Generator* activity. Its behavior can be observed in Figure 11.

To conclude, the decision model related to the business rule task is built. To prepare the reactors, the type of method to be used must be known and this depends on the type of material that will be used for the generation of biodiesel. The materials can be beef fat, pork fat, chicken fat, and vegetable fat. In the case of study, this decision making is shown in terms of the decision table as shown in Figure 10.

U	material	Method
	Text [beef, pork, chicken, vegetable]	Text [M1,M2,M3,M4]
1	beef	M1
2	pork	M2
3	chicken	M3
4	vegetable	M4

Figure 10. Decision table to *Choose method*.

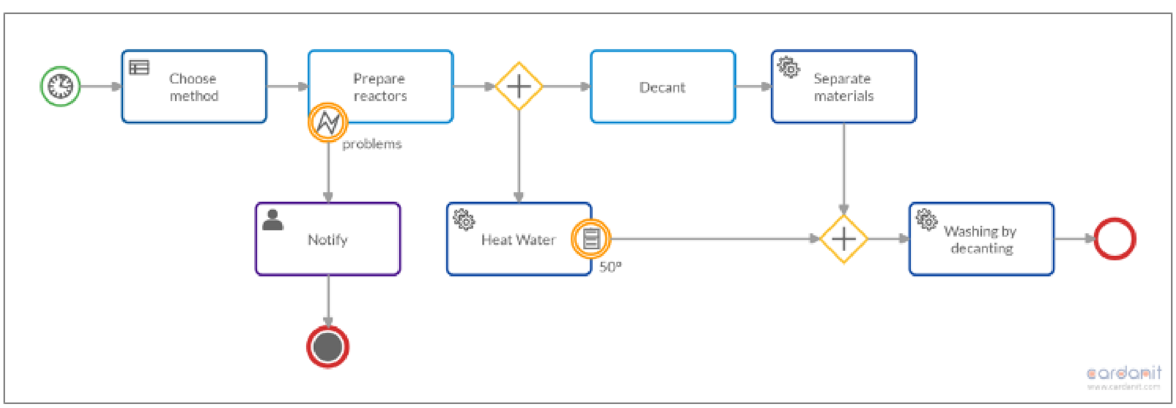

Figure 11. BPMN model of *Generator*.

VI. CONCLUSION

Requirements traceability is the capability to follow the life-cycle of the requirement playing an important role in model-based system engineering where the use of models has a central role. SysML is a general-purpose modeling language, based on UML, which enables traceability because it defines relationships between requirements and other modeling elements. The approach presented in this paper has as the central proposition to consider the decision-making activities that participate in a process as a decision requirement and through traceability, reaching its decision logic. The combination of SysML and BPMN-DMN is interesting because the use of SysML in the industry grows day by day. Not less important is the fact that SysML is a well-defined standard. The combination of all these standards is a step forward that enhances the modeling of the different views of the system to be built including also decision requirements. The approach forces us to define new stereotypes in SysML to support all types of BPMN tasks. An example and a case study have been provided to show how traceability of a decision requirement can be performed by combining SysML and BPMN-DMN. This approach seeks to integrate and cover the different views of the decision requirements, helping systems engineers to improve the design of them.

In future work, we will consider analyzing the link between the relationships in the requirements diagram and the DMN decision requirements diagram.

REFERENCES

[1] 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), ser. Lecture Notes in Computer Science. Springer, vol. 8144, pp. 266-281, 2013.

[2] OMG https://www.omg.org/ [retrieved: October, 2019]

[3] S. Friedenthal, A. Moore, and R. Steiner, “A Practical Guide to SysML The Systems Modeling Language”, Burlington: Morgan Kaufmann/OMG, Elsevier, 2008.

[4] SysML https://www.omg.org/spec/SysML/1.5 [retrieved: October, 2019]

[5] OMG document number: “Business process model and notation”. formal/13-12-09 [retrieved: October, 2019]

[6] Decision Model and Notation https://www.omg.org/spec/DMN/1.2/ [retrieved: October, 2019]

[7] E. J. Vidal and E. R. Villota, “SysML as a Tool for Requirements Traceability in Mechatronic Design”. In Proceedings of the 2018 4th International Conference on Mechatronics and Robotics Engineering. ACM, pp. 146-152, 2018.

[8] R. Baduel, M. Chami, J. M. Bruel, and I. Ober, “SysML Models Verification and Validation in an Industrial Context: Challenges and Experimentation”. In European Conference on Modelling Foundations and Applications. Springer, Cham, pp. 132-146, 2018.

[9] A. Salado and P. Wach, “Constructing True Model-Based Requirements in SysML”. Systems, vol. 7, no 2, pp. 19, 2019.

[10] K. Gruber, J. Huemer, A. Zimmermann, and R. Maschotta, "Integrated description of functional and non-functional requirements for automotive systems design using SysML", 2017 7th IEEE Int. Conf. on System Engineering and Technology (ICSET), pp. 27-31, Oct 2017.

[11] M. R. S. Marques, E. Siegert, and L. Brisolara, “Integrating UML, MARTE and SysML to improve requirements specification and traceability in the embedded domain”. In 2014 12th IEEE International Conference on Industrial Informatics (INDIN). IEEE, pp 176-181, 2014

[12] H. Dubois, M. A. Peraldi-Frati, and F. Lakhal, “A model for requirements traceability in a heterogeneous model-based design process: Application to automotive embedded systems”. In 2010 15th IEEE International Conference on Engineering of Complex Computer Systems. IEEE, pp. 233-242. 2010

[13] INCOSE https://www.incose.org/ [retrieved: October, 2019]

[14] T. Weilkiens, “Systems engineering with SysML/UML: modeling, analysis, design” Elsevier, 2011.

[15] O. C.Z. Gotel and A. C.W. Finkelstein, "An Analysis of the Requirements Traceability Problem", Proc. IEEE Int. Conf. on Requirements Engineering, pp. 94-101, April 1994.

[16] http://www.omgsysml.org/INCOSE-OMGSysML-Tutorial-Final-090901.pdf [retrieved: October, 2019]

[17] UML 2.4 “Infrastructure Specification” https://www.omg.org/spec/UML/2.4.1/ [retrieved: October, 2019]

[18] G. Booch, J. Rumbaugh, and I. Jacobson, “The Unified Modeling Language User Guide” Addison-Wesley. 1999.

[19] Modelica: https://www.modelica.org/ [retrieved: October, 2019]

[20] 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). IEEE. p. 476-477, 2017.