Towards Effective SysML Model Reuse

Roy Mendieta, Jose Luis de la Vara, Juan Llorens and Jose María Álvarez-Rodríguez
Computer Science and Engineering Department, Carlos III University of Madrid, Madrid, Spain
roy.mendieta@kr.inf.uc3m.es, jvara@inf.uc3m.es, {llorens, jmalvarez}@kr.inf.uc3m.es

Keywords: Systems Modeling Language, SysML, Model Reuse, Knowledge Reuse, RSHP, CAKE.

Abstract: The Systems Modeling Language (SysML) is spreading very fast. Most modelling tool vendors support it and practitioners have adopted it for Systems Engineering. The number of SysML models is growing, increasing the need for and the potential benefit from platforms that allow a user to reuse the knowledge represented in the models. However, SysML model reuse remains challenging. Each tool has its own implementation of SysML, hindering reuse between tools. The search capabilities of most tools are also very limited and finding reusable models can be difficult. This paper presents our vision and initial work towards enabling an effective reuse of the knowledge contained in SysML models. The proposed solution is based on a universal information representation model called RSHP and on existing technology for indexing and retrieval. The solution has been used to index models of all SysML diagram types and preliminary validated with requirements diagrams. The results from the validation show that the solution has very high precision and recall. This makes us confident that the solution can be a suitable means for effective SysML model reuse.

1 INTRODUCTION

Since the OMG (Object Management Group) proposed SysML (Systems Modeling Language) (OMG, 2016) to unify different ways of modelling systems, it has been established as a recognized standard. SysML is intended to facilitate the application of Model-Based Systems Engineering, to create a cohesive and consistent process (Friedenthal, et al., 2014). The language includes nine types of diagrams: block definition, internal block, package, use case, requirements, activity, sequence, state machine, and parametric diagram.

SysML supports the specification, analysis and validation of a wide range of systems and systems-of-systems (OMG, 2015). SysML can be used to model different systems, both hardware and software ones, at a high level of abstraction. In addition, SysML is spreading very fast and most of the leading modelling tool vendors such as IBM, No Magic, PTC, and Sparx support it (Bombieri et al., 2013).

As a result, the amount of information represented in SysML models is increasing, thus the amount of knowledge that could be reused. This situation increases the need for platforms that allow a user to search for, select, and thus reuse models effectively and efficiently, which is expected to lead to higher productivity and quality in system modelling (Marincic et al., 2013).

However, SysML model reuse still needs several improvements in practice. Each tool vendor makes its own implementation of SysML, thus reuse between tools is hindered (IBM, 2013). The search capabilities of most tools are also limited. For example, based on our experience, MagicDraw does not support natural language-based search within a model. Papyrus and Rhapsody require the use of regular expressions so that search results are not too imprecise. This impacts usability. Finally, despite the widespread use of SysML and that it already is nine years old, very few publications have dealt with the reuse of the knowledge represented in its models.

We aim to answer the following question: How can the knowledge contained in SysML models be reused effectively? We are working on a modelling tool-independent solution that uses RSHP, a universal information representation model (Llorens, et al., 2004), and CAKE (Computer-Aided Knowledge Environment) (Silva, 2005), an ontology-based framework for information indexing and retrieval.

The solution is called SYSML2RSHP. We have used it to index models from the literature (Holt, Jon and Perry, 2008) for all the types of SysML diagrams. Next, we have conducted a preliminary validation with 44 SysML requirements diagrams and used 25

Mendieta, R., Vara, J., Llorens, J. and Álvarez-Rodríguez, J.
Towards Effective SysML Model Reuse.
DOI: 10.5220/0006267605360541
In *Proceedings of the 5th International Conference on Model-Driven Engineering and Software Development (MODELSWARD 2017)*, pages 536-541
ISBN: 978-989-758-210-3

queries to study the effectiveness of the solution in model finding, i.e., how well the solution is doing at finding relevant items for a query (Croft et al., 2010). All the models have been created with Papyrus.

The main contributions of this paper are: (1) the definition of a conceptual architecture that allows semantic interpretation of the information in SysML models; (2) the presentation of an initial solution for SysML model reuse based on RSHP and CAKE, and; (3) the demonstration that the precision and recall of the solution can be good or excellent.

The paper is organised as follows. Section 2 presents the background. Section 3 describes the proposed solution and Section 4 its preliminary validation. Section 5 summarises our conclusions.

2 BACKGROUND

This section describes the main basis for our work, including a review of the related work.

2.1 RSHP

RSHP (Llorens, et al., 2004) is a universal information representation model. It allows a user to handle all kinds of artefacts (text, diagrams, etc.) using the same schema. It is thus possible to generalize the management of the different artefacts.

The main elements of the model (Figure 1) are:

1. *Artifact* is defined as a knowledge container and described by Relationships, e.g. a diagram node; an Artifact can be represented through only Knowledge Elements or through other Artifacts.
2. *Term* represents a semantic concept in a natural language, e.g. the terms “system” and “sensor”.
3. *Knowledge Element* is the smallest knowledge unit and is related to the occurrence of a Term, e.g. the Term “sensor” in an Artifact.
4. *Relationship (RSHP)* is the core component of a model. It is used as a descriptor of all types of Artifacts information and to connect Knowledge Elements, e.g. to indicate that a “sensor” is part of a “system”.
5. *Metaproperty* represents metadata about an Artifact through key and value pairs, e.g. “version” and “2.0” for a “system”.
6. *RSHP Semantics* depict the meaning of a Term or Relationship, e.g. “component” for “sensor”.

2.2 CAKE

CAKE (Silva, 2005) is a framework of tools, applications, and methodologies to identify, classify, organise, and reuse knowledge. The framework aims to allow a user to manage any kind of “knowledge assets”. It uses RSHP as base information representation model.

Among its main functionalities, CAKE support information indexing and retrieval. Indexing of an artefact’s information (i.e., the knowledge represented in it) is performed according to the information types and structure in RSHP. CAKE also uses ontologies as reference knowledge bases from which further knowledge can be derived. This is especially important for retrieval. The use ontologies allow CAKE to e.g. exploit synonyms for information search. The terms in a RSHP model are part of the ontologies.

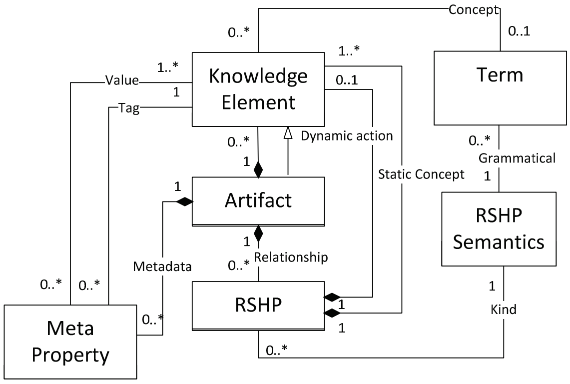

Figure 1: The RSHP representation model using UML.

2.3 Related Work

Reuse has always been an important area amongst system and software companies in order to increase their productivity and the quality of their products (Robles et al., 2012). Prior work includes approaches for model reuse, e.g. for UML models (Robles et al., 2012) (Adamu and Zainoon, 2016)

Regarding publications that has dealt with SysML model reuse, (Jobe and Johnson, 2008) propose the reuse of engineering analysis models through a specific framework. Unlike the solution proposed in this paper, this framework requires the design of SysML models under specific taxonomies, which constrains how the models can be created. Therefore, reuse becomes less flexible and less applicable, thus not effective enough.

(Favaro et al., 2012) developed an approach based on the use of semantic wiki technology to enable users to specify structured, semantically-rich requirements associated with canonical system designs specified with SysML. This proposal is not oriented to all kinds of SysML diagrams, thus its reuse support is limited.

(Bombieri et al., 2013) propose an integrated methodology to abstract already existing heterogeneous information about chips into equivalent SysML behavioural models. This work strongly focused on chip information and cannot be directly applied to domains.

Prior work has also presented approaches for reuse based on RSHP. (Gallego et al., 2015) addressed the storage and reuse of physical system models represented with Modelica, whereas (Llorens et al. , 2004) dealt with UML models represented with XMI. In both cases, the authors propose a mapping between the language under consideration and RSHP. Although these publications are a basis for the solution presented in this paper and made us confident that a RSHP-based reuse solution for SysML was feasible, their insights are not directly applicable to SysML models because of SysML's different nature and specific reuse challenges.

In summary, none of the above publications provides a general and thus fully effective solution to SysML model reuse.

3 SYSML2RSHP: A TOOL FOR INDEXING AND RETRIEVAL OF SYSML MODELS

SYSML2RSHP can transform SysML models into RSHP models. It has been developed in Visual Studio 2010 and uses the CAKE framework as supporting technology. The tool performs the processes of indexing and retrieval of information, which can be used to analyse its reuse capabilities. Currently, the solution can interpret SysML models created with Papyrus. Nonetheless, it aims to be tool-independent and thus support the interpretation of SysML models created with any modelling tool.

Figure 2 shows the conceptual architecture based on which the technological solution for indexing and retrieval of SysML models was developed. The architecture has the following main elements:

- SysML Tool (Papyrus): tool with which a SysML model is created.
- XML Converter: a component that interprets the various implementations of the XMI standard made by the developers of SysML tools and convert them into a XML generic structure.
- Indexer: Component that performs the conversion of SysML model information in the generic structure into RSHP.
- CAKE: the component that supports the storage and retrieval of model information.
- SysML Models (Knowledge) Database: repository where information about the indexed SysML models is stored.

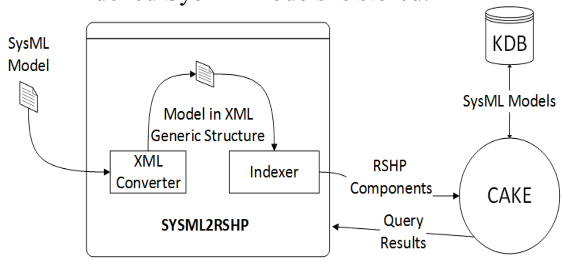

Figure 2: Conceptual architecture.

The solution was developed with the following technologies: (1) XSLT platform for transformation of the information in Papyrus SysML models into the XML generic structure (see Figure 3); (2) a class library developed in Visual Studio .Net 2010 to parse the generic structure and map it to RSHP elements, and; (3) the CAKE framework for RSHP model storage and for retrieval.

SysML models are converted to the structure shown in Figure 3. For Papyrus, the information is interpreted directly from .uml files. These files contain more detailed information than .xmi ones. The same conversion process can be applied to models created with other tools. A key aspect that make SYSML2RSHP tool-independent is the use of intermediate XML generic files.

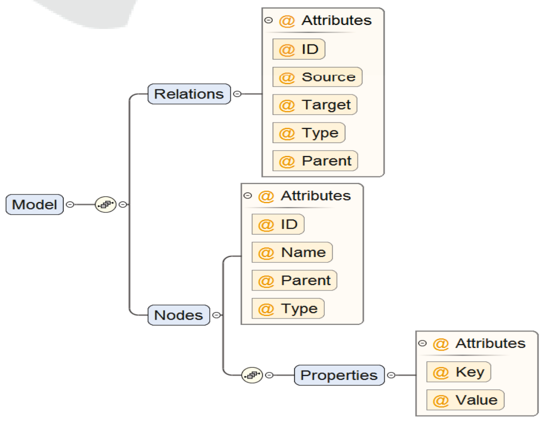

Figure 3: XML generic schema structure based on model, nodes and relationships.

Once a SysML model is represented with the XML generic format, the process of indexing the SysML information as a RSHP model starts. It is a transformation of the generic objects obtained from the previous conversion process into a set of RSHP artifacts, relationships and properties. The mapping for the transformation is presented in Table 1. SYSML2RSHP can index the nine types of SysML diagrams. We confirmed it by indexing a model of each SysML diagram type, selected from (Holt, Jon and Perry, 2008).

Table 1: Equivalence between generic objects and elements of the RSHP Model.

Generic Object	RSHP Object
Model	RSHP Artifact
Relation	RSHP
Relation Type	RSHP Semantics
Node	RSHP Artifact
Node Property Attribute	RSHP Metaproperty

Regarding the information retrieval process, it can be defined as the process of returning SysML models (or references to them) that present a similarity with a query. The CAKE framework contains information retrieval routines that use the information stored in a knowledge repository in Microsoft Access. This retrieval process is based on graph pattern matching. To make use of these routines, the queries must be represented in the form of RSHP elements.

The retrieval process can be performed in two different ways:

- Natural language query: a query in natural language (e.g. a sentence) is indexed and then related models are searched.
- Partial diagram query: instead of a query in natural language, a model file is selected; this model is then indexed with the SYSML2RHP indexer and finally compared to others in the knowledge database.

4 PRELIMINARY VALIDATION

The validation is presented using the schema proposed by (Juristo and Moreno, 2001), where experimentation is divided into four main activities: definition of the objectives of the experimentation, design of the experiment, execution of the experiment, and analysis of the results.

4.1 Objectives

The following objective was defined for the validation: To study the effectiveness of the proposed solution in finding reusable SysML models (i.e. suitable for reuse) from a model repository.

4.2 Design

The metrics used to validate the solution are:

- Precision: fraction of retrieved information that is relevant.
- Recall: fraction of relevant information that is retrieved.
- F1: a combination of precision and recall.

We also use the levels of "goodness" for precision and recall defined in (Hayes et al., 2005) as a reference to analyse these metrics. The levels are:

- Precision: above 20% it is acceptable, good above 30%, and excellent above 50%.
- Recall: above 60% it is acceptable, good above 70%, and excellent above 80%.

We used the following data and formulas to calculate the metrics (Croft et al., 2010):

- True Positives (TP), i.e. retrieved and relevant items.
- False Positives (FP), i.e. retrieved but irrelevant items.
- True Negatives (TN), i.e. not retrieved and irrelevant items.
- False Negatives (FN), not retrieved but relevant items.

$$Precision=\frac{TP}{TP+FP}$$

$$Recall=\frac{TP}{TP+FN}$$

$$F1=\frac{2*Precision*Recall}{Precision+Recall}$$

We created 44 SysML models for validation. They are based on existing requirements diagrams from several sources (Friedenthal et al., 2014) (OMG, 2016). We selected 11 diagrams and made three variations of each by removing elements, adding elements, and changing terms for synonyms.

We also designed a set of 25 queries to evaluate the retrieval capabilities of the solution. The queries are listed in Table 2. These queries were designed considering functional aspects common to several of the models, the components of the models, and the terminology of the models.

After creating the queries, the relevant items to be retrieved were defined. This allowed us to calculate the metrics.

Table 2: Queries for validation.

No	Query Search String
Q1	System availability
Q2	Maximum rate of failure
Q3	Manage Traffic flow
Q4	System to purify water
Q5	System using remote control component
Q6	System using cameras
Q7	System with a statistical data component
Q8	System Performance Requirements
Q9	Requirements of System Usability
Q10	System with Simulation Component
Q11	Group Creation
Q12	System Restrictions Requirements
Q13	System that uses Sensors
Q14	Gather and Interpret Information Module
Q15	Adaptive Control
Q16	Consistency in transaction
Q17	Manual Control
Q18	Intruders detection
Q19	Time Validation
Q20	Computer response time
Q21	System validation cards
Q22	Tasks and scenarios
Q23	Traffic management based on the region
Q24	Semaphores automatic operation
Q25	Control standard

4.3 Execution

We used models created with Papyrus for execution. Table 3 presents the results. Out of the 25 queries:

- The precision and the recall are at least good for all the queries, and excellent on average.
- The precision is good for 8 queries (32%), and excellent for 17 (68%); it is perfect (1,00) for 13 queries (52%).
- The recall is good for 3 queries (12%) and excellent for 22 (88%); it is perfect (1,00) for 20 queries (80%).
- F1 is perfect (1,00) for 10 queries (40%).

4.4 Analysis

It is clear to us that SYSML2RSHP can be a great solution to make SysML reuse effective. The level of precision and recall has been excellent for the vast majority of the queries. This is especially important for recall. The results suggest that a user would be able to find almost all the models that could be reused for a given purpose (query in the validation). We consider that it is more important to find all the reusable models that retrieving models that are not reusable actually. A user could discard the non-reusable models. The very promising percentage of perfect F1 and its high average also suggest that SYSML2RSHP can enable effective reuse with reduced additional effort to discard models that are not reusable actually.

We think that the good results of the validation are, to a large extent, a consequence of CAKE capabilities, e.g. to process and analyse natural language. The framework performs an information harmonisation process and can interpret synonyms using an ontology. Nonetheless, a suitable indexing is also necessary, and a specific indexing approach has been defined for SYSML2RSHP. We conjecture that the results might vary with different indexing strategies, e.g. only transforming models into RSHP Artifacts.

The current validation is preliminary and has limitations. For example, only requirements diagrams have been used. Nonetheless, we regard these limitations as minor for the current, initial status of our work. We will address them in future validations, once the solution is also further developed.

Table 3: Precision, Recall and F1 for each query.

	Precision	**Recall**	**F1**
Q1	0,45	0,94	0,61
Q2	1,00	0,83	0,91
Q3	0,67	1,00	0,80
Q4	1,00	1,00	1,00
Q5	1,00	0,75	0,86
Q6	0,36	1,00	0,53
Q7	1,00	1,00	1,00
Q8	0,33	1,00	0,50
Q9	0,38	1,00	0,55
Q10	1,00	1,00	1,00
Q11	0,50	1,00	0,67
Q12	0,50	1,00	0,67
Q13	0,36	1,00	0,53
Q14	1,00	0,73	0,84
Q15	0,75	1,00	0,86
Q16	1,00	1,00	1,00
Q17	1,00	1,00	1,00
Q18	1,00	1,00	1,00
Q19	0,70	1,00	0,82
Q20	1,00	1,00	1,00
Q21	1,00	1,00	1,00
Q22	1,00	1,00	1,00
Q23	0,78	1,00	0,88
Q24	1,00	1,00	1,00
Q25	0,43	0,75	0,55
Average	0,77	0,96	0,82

5 CONCLUSIONS

The state of the art and the state of the practice related to knowledge reuse from SysML models reflect that several issues have not been solved yet. Although there are solutions that can support SysML model reuse, the solutions can be ineffective, mostly as a result of usability and applicability limitations.

We have introduced SYSML2RSHP, a new solution for indexing and retrieval of SysML models. The implementation of the solution is based on the RSHP information representation model and the CAKE framework. SysML models created with Papyrus have been converted into a XML generic structure, next indexed according to RSHP, and finally stored through CAKE, which also supports model retrieval.

We argue that the proposed solution is a very promising alternative towards effective SysML model reuse. In the preliminary validation, the level of precision and recall of all the queries is good or excellent. F1 is perfect for 40% of the queries and above 0.8 on average. SYSML2RSHP is the first step towards a mature platform for modelling tool-independent SysML model reuse.

The most immediate pieces of future of work are the improvement, further development, and further validation of SYSML2RSHP. It must be shown that it can index models created in modelling tools other than Papyrus, e.g. MagicDraw and Rhapsody, and the search capabilities of the different tools should be compared. With the ability to index information from SysML models, another possible line of future research is the analysis of SysML model quality, e.g. completeness and consistency. This would also allow a user to select and reuse SysML models taking their quality into account.

ACKNOWLEDGEMENTS

The research leading to this paper has received funding from the AMASS project (H2020-ECSEL grant agreement no 692474; Spain's MINECO ref. PCIN-2015-262).

REFERENCES

Adamu, A., & Zainoon, W. M. N. W. (2016). A Framework for Enhancing the Retrieval of UML Diagrams. In *International Conference on Software Reuse* (pp. 384–390).

Bombieri, N., Ebeid, E., & Fummi, F. (2013). On the Reuse of Heterogeneous IPs into SysML Models for Integration Validation, 647–667.

Croft, W. B., Metzler, D., & Strohman, T. (2010). *Search Engines: Information Retrieval in Practice*. Pearson Education. Retrieved from http://comjnl.oxfordjournals.org/cgi/reprint/54/5/831\n http://comjnl.oxfordjournals.org/cgi/doi/10.1093/comjnl/bxq039.

Favaro, J., Schreiner, R., & Olive, X. (2012). Next Generation Requirements Engineering. In *INCOSE International Symposium* (pp. 461–474).

Friedenthal, S., Moore, A., & Steiner, R. (2014). *A practical guide to SysML: the systems modeling language*. Morgan Kaufmann.

Gallego, E., Álvarez-rodríguez, J. M., & Llorens, J. (2015). Reuse of Physical System Models by means of Semantic Knowledge Representation : A Case Study applied to Modelica, 747–757.

Hayes, J. H., Dekhtyar, A., & Sundaram, S. K. (2005). Improving after-the-fact tracing and mapping: Supporting software quality predictions. *IEEE Software*, *22*, 30–37.

Holt, Jon and Perry, S. (2008). *SysML for systems engineering*. IET.

IBM. (2013). Rhapsody and MagicDraw Integration. Retrieved from http://www.ibm.com/support/knowledgecenter/SSB2MU_8.1.0/com.ibm.rhp.oem.pdf.doc/pdf/sodius/Rhapsody_MagicDraw_Integration.pdf.

Jobe, J. M., & Johnson, T. A. (2008). Multi-Aspect Component Models: A Framework for Model Reuse in SysML, 1–13.

Juristo, N., & Moreno, A. M. (2001). *Basics of Software Engineering Experimentation*. *Analysis* (Vol. 5/6). Springer Science & Business Media.

Llorens, J., Carlos, U., Madrid, I. I. I. De, Llorens, J., Morato, J., & Genova, G. (2004). RHSP : an Information Representation Model Based on Relationship on relationships, (January 2004).

Llorens, J., Fuentes, J. M., & Morato, J. (2004). Uml retrieval and reuse using xmi. *Proceedings of the IASTED International Conference on Software Engineering*, 740–746.

Marincic, J., Mader, A., Wieringa, R., & Lucas, Y. (2013). Reusing knowledge in embedded systems modelling. *Expert Systems*, *30*(3), 185–199. http://doi.org/10.1111/j.1468-0394.2012.00631.x.

OMG. (2015). OMG Systems Modeling Language (OMG SysML ™) v.1.4, 320. Retrieved from http://www.omg.org/spec/SysML/1.4/

OMG. (2016). The Official OMG SysML site. Retrieved from http://www.omgsysml.org/

Robles, K., Fraga, A., Morato, J., & Llorens, J. (2012). Towards an ontology-based retrieval of UML Class Diagrams. *Information and Software Technology*, *54*(1), 72–86.

Silva, A. R. (2005). Tools Exhibits. In *UML Modeling Languages and Applications SE - 34* (Jardim Nun, Vol. 3297, pp. 281–291). Berlin, Heidelberg: Springer Berlin Heidelberg.