A ReqIF/SysML profile example – Requirements exchange and roundtrip

Languages, process, methods and tools topic: Engineering tools integration and interoperability

Olivier CASSE olivier.casse@atego.com [1] Manuel Reis Monteiro manuel.monteiro@atego.com [2]

1: Atego SAS 143 bis Avenue de Verdun 92130 Issy-Les-Moulineaux -France

2: Atego Systems GmbH Major-Hirst-Str. 11 D-38442 Wolfsburg - Germany

Abstract: This paper presents a model-based approach for requirements interoperability. It relies on SysML for system requirements modelling, and is exploring which means could be used for exchanging requirements from the RERM toolset, either managed in the same company, or between the client and its subcontractors We will review relevant parts of SysML for such synchronization, that is mainly the Requirements view and its relationships to model elements, where links are the essential value of RERM tools and SysML models. We'll then study how the SysML profile could be extended for supporting a round trip process between stakeholders. Time will come to valuate which means are possible for interoperability: CSV based files, XMI, AP-233 or ReqIF. A short comparison between those notations will be summarized as none as emerged yet in the market. For instance XMI is not supported the same way by all tool vendors, leading to a poor compatibility between SysML tools. Based on going work started by ProSTEP iViP for the OMG we will browse possible SysML profile extensions for ReqIF compliancy.

Keywords: requirements, methodology, SysML, model-driven engineering, exchange, roundtrip.

1. Introduction

Today's complexity of systems leads to a large number of requirements to be managed; those requirements live in the sense that they are updated, refined more and more frequently. As a consequence stakeholders also need to be aware of new versions, for that purpose PDF or HTML documents can be used, but back-annotating for giving structured feedback and comments (or prove requirements are fulfils or correctly implemented, tested...) is simply not done; lack of means and tools is the main reason. Another barrier is consistency between RERM tools and modelling tools. No standard currently helps in doing so, each tool vendor provides a dedicated interface with its own process, either on the Requirements side OR the model side, but not for both. It is worth mentioning a new initiative has started within the OMG: the MIWG, Model Interchange Working Group. Alternatively, RIF (former ReqIF name) has started its deployment in the German car industry, and is slowly being taken up in other industries and countries, therefore we will present basic features and principles in a nutshell.

2. REQUIREMENTS

Requirements are customer needs, are mainly textual, are business oriented, and show the 'what' and not the 'how'. They should be independent from the solution. They often lack of validation, making roundtrips through modelling is the first step towards product full life cycle management from specification to test. This is useless if models are not validated against requirements

Which kind of requirements is needed: textual or graphical? The answer could be both actually!

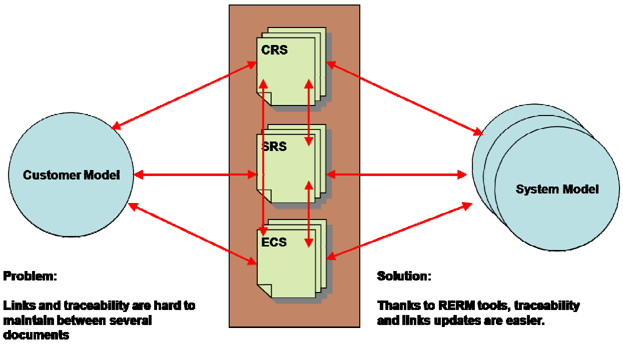

CRS
SRS
ECS
Customer Model
System Model
Problem:
Links and traceability are hard to maintain between several documents
Solution:
Thanks to RERM tools, traceability and links updates are easier.

Figure 1: Typical Requirements Exchange

3. SYSML PROFILE

We will focus on the requirements aspect of SysML, which is one of the major contributions to UML, truly needed for system engineering. SysML offers basic support on it 1.1 specification annex C2 the SysML standard [1] offers as an example to extend its profile to support. For instance we have two interesting requirement properties represented by enumerated types: *RiskKind* and *VerificationMethodKind*.

RiskKind indicates the level of risk; High, Medium or Low, whilst *VerificationMethodKind* could be valued as analysis, demonstration, Inspection or Test.

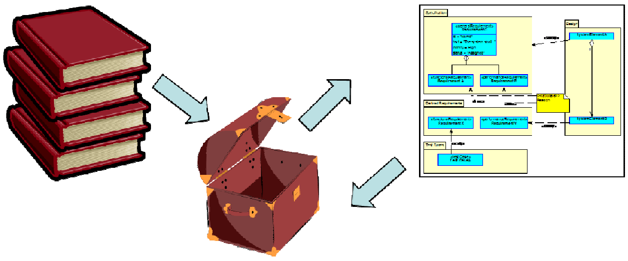


Figure 2: Model Based Requirements Management

4. POSSIBLE EXTENSIONS

The C2 Annex from the SysML 1.1 specification is giving us a possible start, with ReqIF we could go further by importing root and user attributes into the SysML Requirements model:

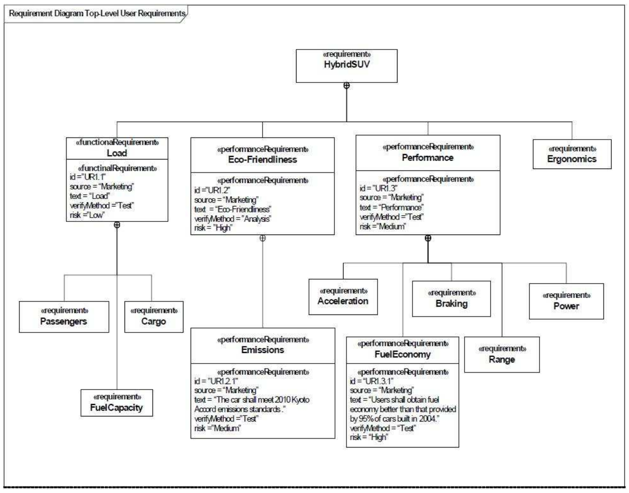
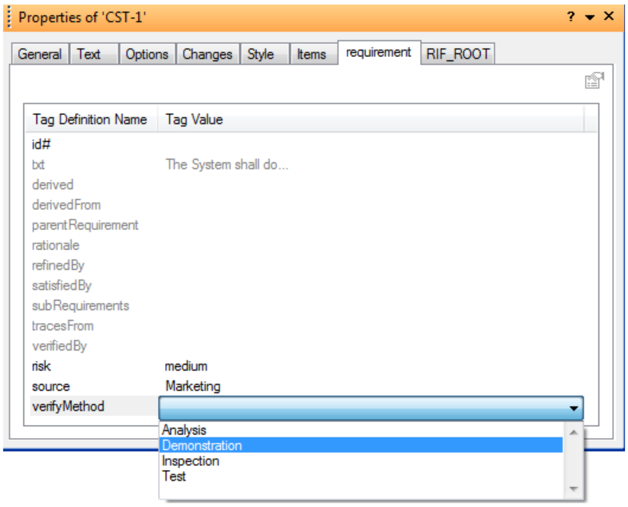


Figure 3: Example extensions to Requirement

Here is a very simple example on how to implement this in a standard SysML tool. We added on top of the basic attributes Id, Test and name, the recommended attributes, here using enumerated types.
Figure 4 gives a practical example on extending Requirements attributes in a modelling tool.



Properties of 'CST-1'

Tag Definition Name	Tag Value
id#	
txt	The System shall do...
derived	
derivedFrom	
parentRequirement	
rationale	
refinedBy	
satisfiedBy	
subRequirements	
tracesFrom	
verifiedBy	
risk	medium
source	Marketing
verifyMethod	

Analysis
Demonstration
Inspection
Test

Figure 4: Example Modelled in a SysML tool

5. INTEROPERABILITY

File based exchanges: A common practice is for exchanges between customers and suppliers to be based on IT tools such as textual documents or spreadsheets. For instance .csv or .xls either using Excel Office from Microsoft or the Calc Opensource from OpenOffice format is a straightforward, and is a convenient way to specify requirements, put comments and then offer a communication mechanism, which is tool independent. The drawback of this approach is that the exchange process cannot be fully controlled; each partner can modify the structure of the requirements on-the-fly. Furthermore, with the increase in maturity in the requirement engineering field, industrial specifications tend to become large and complex, with increasing importance being given to traceability; office-type documents cannot handle traces conveniently for the user, and managing the size of specifications is left to users. Finally, for companies having invested in RERM solutions, ad-hoc solutions need to be developed to handle synchronization with the source repositories.

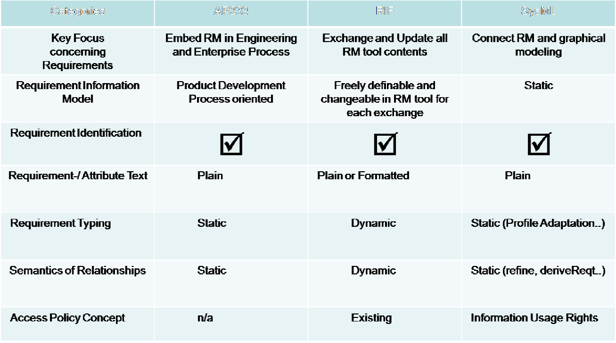
MIWG: OMG™ members have formed the Model Interchange Working Group (MIWG) to demonstrate and facilitate interoperability between UML®-based modelling tools. The group's initial focus is on model interchange between UML®, OMG SysML™, and Unified Profile for DoDAF and MODAF (UPDM) -capable tools. MIWG comprises end users, tool vendors and experts in the UML and XMI® standards. This group is focusing on models and not directly targeting requirements interchange between tools. Issues to be resolved involve items like formatted text, image/document embedding, etc. This won't solve the issue to import/export links with the RERM tools.

ProSTEP iViP IntRIF Project Group[3]:
This group is focusing on standardizing ReqIF, as the RIF community has now the desire to standardize RIF (renamed ReqIF for the OMG submission) on international level through the OMG. Intention is to make ReqIF an added value to SysML. ReqIF is specified by a MOF model and corresponding XML Schema. Timing is good and might be aligned with the upcoming SysML 2.0 effort.

Meanwhile here are described some possible paths for achieving a successful roundtrip.

AP 233:
AP233 is designed as a neutral information model for the exchange of data between Systems Engineering, Systems Architecture Description and related tools. In most cases, the specifications within this site are the definition of mappings between the schema or meta-model of an application, database or standard and the AP233 XML Schema.

For Requirements the data model that captures requirements as text strings with traceability, allocation, weighting and risk identified with each requirement [Text-based Requirements (TBR)] and that describes requirements as structured and quantified formalisms that may be decomposed from text-based requirements; can include tables, spreadsheets, graphs, charts, pictures and equations [Property-based Requirements (PBR)].



Categories	AP233	RIF	SysML
Key Focus concerning Requirements	Embed RM in Engineering and Enterprise Process	Exchange and Update all RM tool contents	Connect RM and graphical modeling
Requirement Information Model	Product Development Process oriented	Freely definable and changeable in RM tool for each exchange	Static
Requirement Identification	☑	☑	☑
Requirement-/ Attribute Text	Plain	Plain or Formatted	Plain
Requirement Typing	Static	Dynamic	Static (Profile Adaptation..)
Semantics of Relationships	Static	Dynamic	Static (refine, deriveReqt..)
Access Policy Concept	n/a	Existing	Information Usage Rights

Figure 5: Possible Exchange Means Matrix

6. ReqIF background

The Requirement Interchange Format (ReqIF, formerly RIF) is the product of an initiative of the automotive industry (the initiator was the HIS, a group of car manufacturers including Audi, BMW, Daimler-Chrysler, Porsche, and Volkswagen from the OEM side, and the participation of Continental AG and Robert Bosch). It was designed to exchange requirements between car manufacturers and suppliers. ReqIF goes beyond the automotive industry and has been successfully applied by other industries (e.g. rail transportation or medical devices).

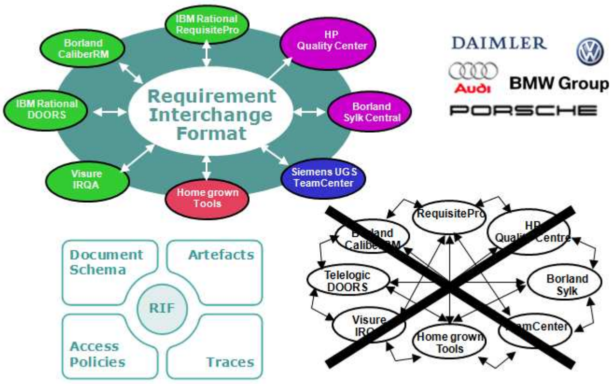


Figure 6: HIS RIFproject

Figure 6 illustrates main goals of the former project where HIS decided to define a standard to supporting Requirements Exchanges, including round-tripping.

7. ReqIF in a nutshell

The Requirements Interchange Format has been recently renamed from RIF to ReqIF and is currently under a request for comment at the OMG ([4]1.0 Alpha 25th March 2010). Submitters are Atego and ProSTEP iViP Association; supporters are 88solutions, Audi AG, BMW AG, HOOD GmbH, International Business Machines, MKS GmbH, ModelAlchemy Consulting, PROSTEP AG, Robert Bosch GmbH, and Volkswagen AG.

The principal/contractor constellation is a typical scenario in which requirements have to be exchanged. Between the two roles there is normally the corporation boundary, and access to a common requirements database is hardly ever possible. The friction losses, and thus errors, costs, time delays, and discords can easily exceed the limits to pain. And when something hurts, it's time to change it.

ReqIF closes the gap, allowing you to smoothly exchange requirements beyond tool limits and company boundaries. It describes a generic format for filing requirements. In addition to the requirements themselves, you can also describe groups, hierarchies, relationships, access privileges, etc.

Requirements are assets that must be shared between the different stakeholders to ensure a successful completion of a project. Solid requirements management is one of the most important factors of successful development. As a result, all stakeholders must be in a position to seamlessly exchange or synchronize requirements and specification documents in a form that can be readily used by all stakeholders involved in a project. Requirements management tools provide only restricted solutions to the problem of transfer and exchange of requirements, often leaving the users to deal with complex programming to ensure a smooth interoperability between tools.

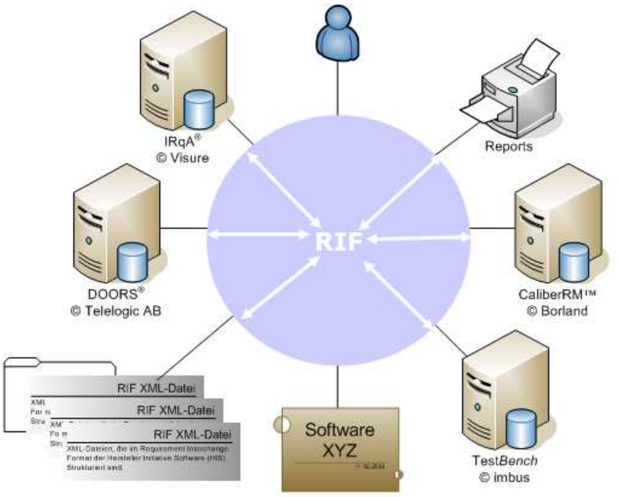


Figure 7: Implementation of ReqIF

8. Objectives of the ReqIF

[5] Requirements management has been an integral part of the development process in various industries (especially in the military, aeronautical or the medical device industry) for years. Other industries have been adopting requirements management recently.
The automotive industry for example introduced requirements management around 1999. As requirements management spread in the automotive industry over the years, more and more car manufacturers and suppliers have been applying requirements management and making use of dedicated requirements authoring tools. Large improvements have been made in these organizations and requirements management has been established as a key discipline in this collaborative engineering environment. Now with this established discipline in place, manufacturers and suppliers strive for collaborative requirements management where requirements management does not stop at company borders.
For technical and organizational reasons, two companies in the manufacturing industry are rarely able to work on the same requirements repository and sometimes do not work with the same requirements authoring tools or even with different incompatible versions. A generic, non-proprietary format for requirements information is required to cross the chasm, and to satisfy the urgent industry need for exchanging requirement information between different companies without losing the advantage of requirements management at the organizations' borders.
With the help of a dedicated interchange format for requirements specifications, it is possible to bridge the gap:

- The collaboration between partner companies is improved by the benefits of applying requirements management methods across company borders.
- The partner companies do not have to use the same requirements authoring tool and suppliers do not need to have multiple requirements authoring tools to fulfil the need of their customers with regards to compatibility
- Within a company, requirement information can be exchanged even if various tools are used to author requirements.

The Requirements Interchange Format (ReqIF) described in this specification defines such an open, non-proprietary exchange format. Requirement information is exchanged by transferring XML documents that comply to the ReqIF format.
See the following figure for an example scenario between two partners who are exchanging a Customer
Requirements Specification and the corresponding System Requirements Specification:

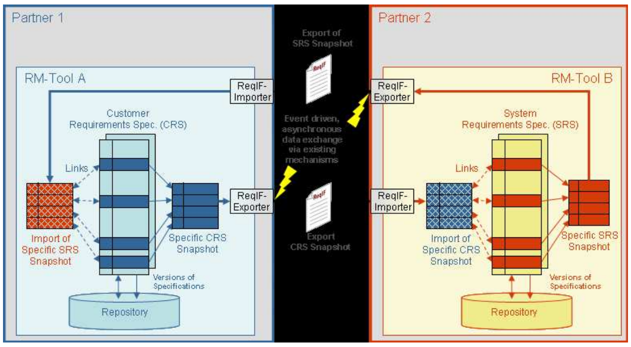

Partner 1
RM-Tool A
ReqIF-Importer
Customer Requirements Spec. (CRS)
Links
Import of Specific SRS Snapshot
Specific CRS Snapshot
ReqIF-Exporter
Versions of Specifications
Repository
Export of SRS Snapshot
Event driven, asynchronous data exchange via existing mechanisms
Export CRS Snapshot
Partner 2
RM-Tool B
ReqIF-Exporter
System Requirements Spec. (SRS)
Links
ReqIF-Importer
Import of Specific CRS Snapshot
Specific SRS Snapshot
Versions of Specifications
Repository

Figure 8: Example ReqIF exchange scenario

Figure 8 represents a common scenario how requirements specifications are exchanged between partners. Both partners in the scenario use different requirements management (RM) tools to create, manage and evolve their requirements specifications. The process is usually initiated by Partner 1. Customer requirements that are relevant for Partner 2 are consolidated in a snapshot document. The Partner 2 specific CRS snapshot is exported out of the RM-Tool A by means of the ReqIF-Exporter and transferred asynchronously to Partner 2 via existing data transfer mechanisms. The result of the export is a ReqIF compliant XML document representing the specific CRS snapshot.

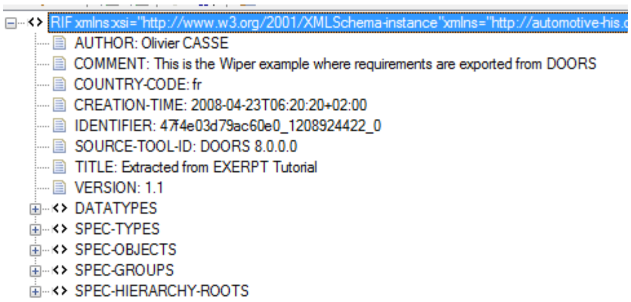
The data transfer mechanism is out of scope of ReqIF. Having received the exported CRS snapshot Partners 2 imports the information into RM-Tool B in order to analyze the customer requirements imposed by Partner 1. For traceability reasons Partner 2 links the received customer requirements with the corresponding system requirements.
As an answer to the customer requirements Partner 2 creates a consolidated SRS snapshot that contains the system requirements realizing the imposed customer requirements of Partner 1. The SRS snapshot is fed back to Partner 1 as an exported ReqIF compliant XML document. Having imported the SRS snapshot Partner 1 can analyze within a RERM Tool how the customer requirements are fulfilled by the system requirements specified by Partner 2. As specifications evolve over time the exchange via ReqIF is an event driven, asynchronous data exchange.

Car manufacturers and their tier suppliers face this problem every day. With the ever increasing complexity of the onboard electronics, the industry faces major difficulties in completing projects on-cost and on-time. Under the hospice of Hersteller Initiative Software (HIS)—a joint standard initiative of the German automotive industry including members such as Audi, BMW, DaimlerChrysler, Porsche and Volkswagen—a working group was tasked with the job of specifying a standard to resolve the problem of exchanging requirements and specification documents; the resulting Requirement Interchange Format (ReqIF) is based the Extendible Markup Language (XML), which is both machine readable text and is easily formatted for human reading. The ReqIF model is described in UML and implemented in XML.

Here is a small extract of a ReqIF File:

```
<SPEC-OBJECT>
<IDENTIFIER>47f4e03d79ac60e0_1208924423_57</IDENTIFIER>
<LAST-CHANGE>2008-04-23T06:20:20+02:00</LAST-CHANGE>
<LONG-NAME>R-9</LONG-NAME>
<TYPE>
<SPEC-TYPE-REF>47f4e03d79ac60e0_1208924423_14</SPEC-TYPE-REF>
</TYPE>
<VALUES>
<ATTRIBUTE-VALUE-EMBEDDED-DOCUMENT>
<IDENTIFIER>47f4e03d79ac60e0_1208924423_54</IDENTIFIER>
<LAST-CHANGE>2008-04-23T06:20:20+02:00</LAST-CHANGE>
<LONG-NAME>VALUE-EXERPT_ID_OEM</LONG-NAME>
<DEFINITION>
<ATTRIBUTE-DEFINITION-COMPLEX-REF>47f4e03d79ac60e0_1208924423_15</ATTRIBUTE-DEFINITION-COMPLEX-REF>
</DEFINITION>
<XHTML-CONTENT><rif-xhtml:div>45f11c317ddc4710_00000043_9<rif-xhtml:br/></rif-xhtml:div>
</XHTML-CONTENT>
</ATTRIBUTE-VALUE-EMBEDDED-DOCUMENT>
<ATTRIBUTE-VALUE-EMBEDDED-DOCUMENT>
<IDENTIFIER>47f4e03d79ac60e0_1208924423_55</IDENTIFIER>
<LAST-CHANGE>2008-04-23T06:20:20+02:00</LAST-CHANGE>
<LONG-NAME>VALUE-Object Text</LONG-NAME>
<DEFINITION>
<ATTRIBUTE-DEFINITION-COMPLEX-REF>47f4e03d79ac60e0_1208924423_17</ATTRIBUTE-DEFINITION-COMPLEX-REF>
</DEFINITION>
<XHTML-CONTENT><rif-xhtml:div>There are two states in operation:<rif-xhtml:br/></rif-xhtml:div>
</XHTML-CONTENT>
</ATTRIBUTE-VALUE-EMBEDDED-DOCUMENT>
<ATTRIBUTE-VALUE-ENUMERATION>
<IDENTIFIER>47f4e03d79ac60e0_1208924423_56</IDENTIFIER>
<LAST-CHANGE>2008-04-23T06:20:20+02:00</LAST-CHANGE>
<LONG-NAME>VALUE-Supplier Status</LONG-NAME>
<DEFINITION>
<ATTRIBUTE-DEFINITION-ENUMERATION-REF>47f4e03d79ac60e0_1208924423_18</ATTRIBUTE-DEFINITION-ENUMERATION-REF>
</DEFINITION>
<VALUES>
<ENUM-VALUE-REF>47f4e03d79ac60e0_1208924422_12</ENUM-VALUE-REF>
</VALUES>
```

This means that it can be imported to and exported from a SysML model. The hard part that remains in all variants is the traceability of requirements beyond model boundaries. Though SysML integrates several arrangements to improve the situation, the successful realization depends on the modelling tools.



```
<> RIF xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"xmlns="http://automotive-his.c
    AUTHOR: Olivier CASSE
    COMMENT: This is the Wiper example where requirements are exported from DOORS
    COUNTRY-CODE: fr
    CREATION-TIME: 2008-04-23T06:20:20+02:00
    IDENTIFIER: 47f4e03d79ac60e0_1208924422_0
    SOURCE-TOOL-ID: DOORS 8.0.0.0
    TITLE: Extracted from EXERPT Tutorial
    VERSION: 1.1
  <> DATATYPES
  <> SPEC-TYPES
  <> SPEC-OBJECTS
  <> SPEC-GROUPS
  <> SPEC-HIERARCHY-ROOTS
```

Figure 9: RIF file structure example

9. Relationships between ReqIF, XMI & SysML

[6] Both the Requirements Interchange Format (ReqIF) and XMI represent technologies for the serialization of information models. XMI provides a generic means to serialize MOF-based models, while ReqIF is specialized for the exchange of requirements specifications between requirements authoring tools. This requires the exchange of rich formatted information, a capability provided by ReqIF, but not directly available in the current form of XMI. The German automotive and transport industry uses the Requirements Interchange Format already heavily in daily operations. Users and vendors are equally interested to further stabilize the Requirements Interchange Format through standardization by the OMG. However, this requires the preservation of the rich formatting capabilities of the Requirements Interchange Formats. The relationship between the Requirements Interchange Format and SysML has been the subject of several discussions between the Systems Engineering DSIG and the ManTIS taskforce (during the OMG Technical Meeting in St. Antonio 2009 and the OMG Technical Meeting in Washington D.C. 2009). Both SysML and the Requirements Interchange Format support the concept of a requirement, the concept of hierarchically structuring requirements and the possibility to define relationships between requirements. However, state-of-the-art requirements authoring utilizes a broader spectrum of features, in particular rich text formatting, than currently provided by SysML. At the St. Antonio Meeting in 2009, a mapping from SysML to the Requirements Interchange Format has been proposed and discussed in the ManTIS task force. The integration of ReqIF with SysML enables requirements engineers to visualize requirements defined in a requirements authoring tool in SysML. System architects can trace design artefacts to requirements in standardized form to establish seamless traceability. Requirements engineers can use modelling and text based approaches to specify requirements and integrate them in a standardized way.

Figure 10 shows an overview of how RERM tools and Modelling Tools may interact.

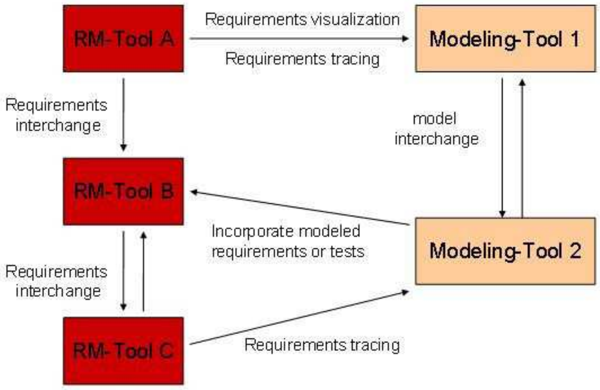


Figure 10: Overview of OMG current vision

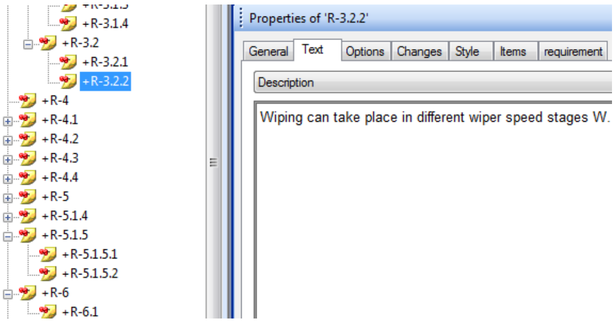
10. Mapping Proposal for ReqIF and SysML

Part of this study is based on several papers made by HOOD[5], some ideas are similar or different but the main difference is the pragmatic approach used related to the tooling experience of the authors both for ReqIF exchange platform deployment and modelling in SysML.

If we study the possible mapping between ReqIF and SysML, some correspondences are quite obvious, but a few others are trickier. We have classified elements in four categories:

1) Headers
2) Data-types
3) Relationships
4) Access rights

As for SysML Requirements, we may consider 2 different usages. The first one is importing, so reusing, requirements previously defined in a RERM tool, again this could be a simple Excel spreadsheet or a Word document. Then the added value of mapping ReqIF to SysML is to use SysML native relationships <<satisfy>>, <<refine>> etc. The second usage is a roundtrip between the RERM values for instance defined by the customer and the SysML model to formalize the specification that will be written by the supplier. Then a SysML to ReqIF mapping is also needed.



Wiping can take place in different wiper speed stages W.

Figure 11: ReqIF supports Textual Requirements

Also SysML allows to 'draw' requirements, in other words we can capture diagrams representing requirements and their relationships. We won't focus too much on this aspect as actually this is only a way to represent requirements and their relationships; the SysML modelling tool has the core feature we are interested in: requirements and relationships storage in a repository or file format.

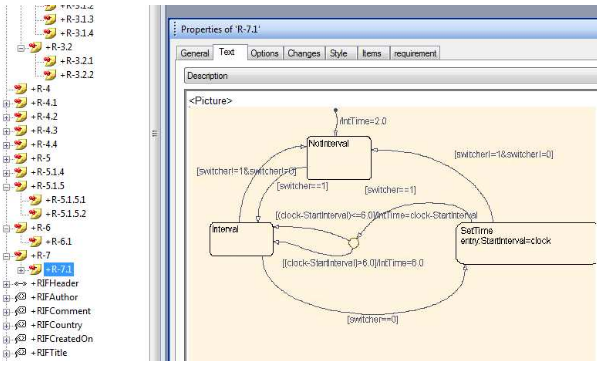


Figure 12: ReqIF can include graphical Requirements

As Requirement interchange is our main focus, both usages are important; the first one could be considered as a sub scenario of the second one. This approach could names the ninth Use Case or UC9 supported by ReqIF

Again we will target in this paper how data from a ReqIF file will be imported into a SysML model: those data are:

-Requirements

-Tracing (links)

-Test

-Access policies (highly tool dependent of course)

In ReqIF one considers the following Model element classes sets:

Root: RIF, Header, content

Identifiable: unique ID for each element

Spec: SpecObject (Requirement basically), SpecType (its type), SpecRelation (links/relationships), SpecGroup(group)

DataTypes: in ReqIF, there are three kinds of data types:

1) **Simple:** integer, real, Boolean, string
2) Enumerated
3) Complex

Other elements are defined but either not relevant for this paper or out of scope for such a draft example, for instance embedded files which support highly depends on the SysML tool used.

Access policies: even if again this is related to the tool used, this is of high interest for contractual exchanges; they basically represent access rights.

Hierarchy will also be supported; in SysML we use packages for this purpose.

<u>Implementation proposal:</u>

We saw in a previous chapter that the SysML profile is mainly composed with:

- a name
- a Text Description
- an id#

They are quite straightforward to map over ReqIF. More interesting is the links/relationships we'll use for traceability and hierarchy, the mapping is a bit more complex, due to the SysML richness, as we have <<satisfy>>, <<verify>>, <<refine>>, <<trace>>, <<copy>>...

Most RERM tools have loose links where it is difficult to further qualify them; probably by putting in line SysML with ReqIF will require ReqIF to take this into account, as those values are key for System engineering.

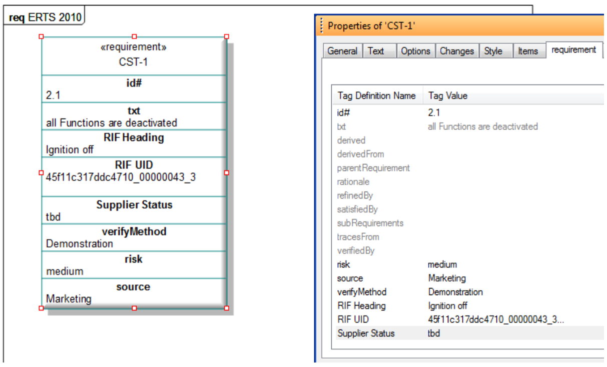
As far as hierarchy is concerned, SpecGroup or SpecHierachy elements will be used for structuring dependencies and locate groups of requirements into packages.

ReqIF Header details

Author, comment, creation Time, identifier, source ToolsID and title will be connected to the root package in the SysML model where imported requirements will be stored.

Requirements (aka SpecObjects):

The tags will be mapped using dedicated <<requirement>> extensions as illustrated in figure



«requirement»
CST-1

id#
2.1

txt
all Functions are deactivated

RIF Heading
Ignition off

RIF UID
45f11c317ddc4710_00000043_3

Supplier Status
tbd

verifyMethod
Demonstration

risk
medium

source
Marketing

Tag Definition Name	Tag Value
id#	2.1
txt	all Functions are deactivated
derived	
derivedFrom	
parentRequirement	
rationale	
refinedBy	
satisfiedBy	
subRequirements	
tracesFrom	
verifiedBy	
risk	medium
source	Marketing
verifyMethod	Demonstration
RIF Heading	Ignition off
RIF UID	45f11c317ddc4710_00000043_3...
Supplier Status	tbd

Figure 13: Requirement added attributes for ReqIF

As soon as ReqIF will become more popular and deployed on large projects where SysML modelling is used, we'll see the need for engineers to have an interface from ReqIF to SysML in order to import/export requirements, and what is the real value the links between requirements and the model elements. Putting in place the suggested profile will help to extend the ReqIF roundtrip feature beyond requirements engineering toward modelling activities.

11. Conclusion

Exchanging requirement and supporting round-tripping between stakeholders is possible using a combination of existing and emerging standards. The tooling implementation will be possible based on those standards, avoiding a proprietary solution even if dedicated solution might be proposed by the various tool vendors. A recommendation could be made by the SysML and/or the OMG working group to ease deployment. As a prototype for proof of concept, Atego has developed a profile for his tool ArtisanStudio which is already usable for importing requirements into a SysML model; this implementation works in conjunction with AtegoEXERPT to enable round-tripping with COTS RERM tools. Several key industrial companies are using both notations and could in a near future validate this interface. Round-tripping should be easily manageable as well.

We believe this could reinforce usage of MBSE techniques for supporting Requirements Engineering.

12. References

[1]. OMG Specification. *OMG System Modeling Language (OMG SysML^{TM}) Specification*. V1.1 formal/2008-11-01. November 2008.

[2]. MIWG Model Interchange Working Group : July 8, 2009 - OMG™

[3]. Internationalization of the Requirements Interchange Format (IntRIF) *http://www.prostep.org/en/project-groups/internationalization-of-the-requirements-interchange-format-intrif.htm*

[4]. Request For Comment OMG mantis/2010-03-07 http://www.omg.org/cgi-bin/doc?mantis/2010-3-7

[5]. Request For Comment OMG mantis/2010-03-07 http://www.omg.org/cgi-bin/doc?mantis/2010-3-7

Chapter 1.2

[6]. Side effects on existing OMG specifications

[7]. APP -233 RIF –SYSML : ManTIS Task Force Technical Session OMG Technical Meeting, 2009-03-24, Washington DC

[8]. HOOD: http://www.omg.org/docs/syseng/08-06-07.pdf

Glossary

AP-233	Application Protocol 233 is the STEP Systems Engineering Project
CSV	comma separated values used for the digital storage of data structured in a table of lists form
HIS	Hersteller Initiative Software
MBSE	Model Based System Engineering
MIWG	Model Interchange Working Group
MOF	Meta-Object Facility
OMG	Object Management Group
ProSTEP iViP	Solutions for the problem of integration in the manufacturing industry
RERM	Requirements Engineering & Requirements Management
RIF	Requirements Interchange Format
SysML	System Modeling Language
TBR	Text Based Requirements
UML	Unified Modeling Language
UPDM	UML Profile for DoDAF/MODAF
XMI	XML Metadata Interchange
XML	Extensible Markup language