

Reuse of requirements in the context of product lines of a complex system

Gauthier Fanmuy
ADN
<http://www.adneurope.com>
39-41 rue Louis Blanc
92038 Paris La Defense
France
gauthier.fanmuy@adn.fr

Olfa Djebbi
Paris 1 University
Phantheon Sorbonne
<http://www.univ-paris1.fr/>
90 rue de Tolbiac
75013 Paris - France
Olfa.Djebbi@malix.univ-paris1.fr

Abstract: this paper is about Product lines. A product line is the development of products that have commonalities. It is based on reusing engineering artifacts (requirements, architecture, tests...) and on managing variants of products. The development of a set of products rather than developing single products independently, which minimizes costs and completion time. This paper is about reusing a set of common requirements. It describes the different techniques for requirements reuse and a panorama of the different tools. It presents an implementation in DOORS.

Introduction

A product line / family of products (goods or services) is the set of products offered by an organization to satisfy the same needs in a given market sector. Product Line Engineering is a discipline that covers the new issues introduced by the design and development of a set of products which have some commonalities rather than developing single products independently, which minimizes costs and completion time.

Engineering a product line is focused on two key concepts: maximize the reuse and manage changes. The principle is to define a platform defining the commonalities and the variabilities. The products are then derived by reusing the commonalities and making choices about the alternatives.

In this paper, we focus mainly on the reuse of requirements in the context of product lines. Several constraints are related:

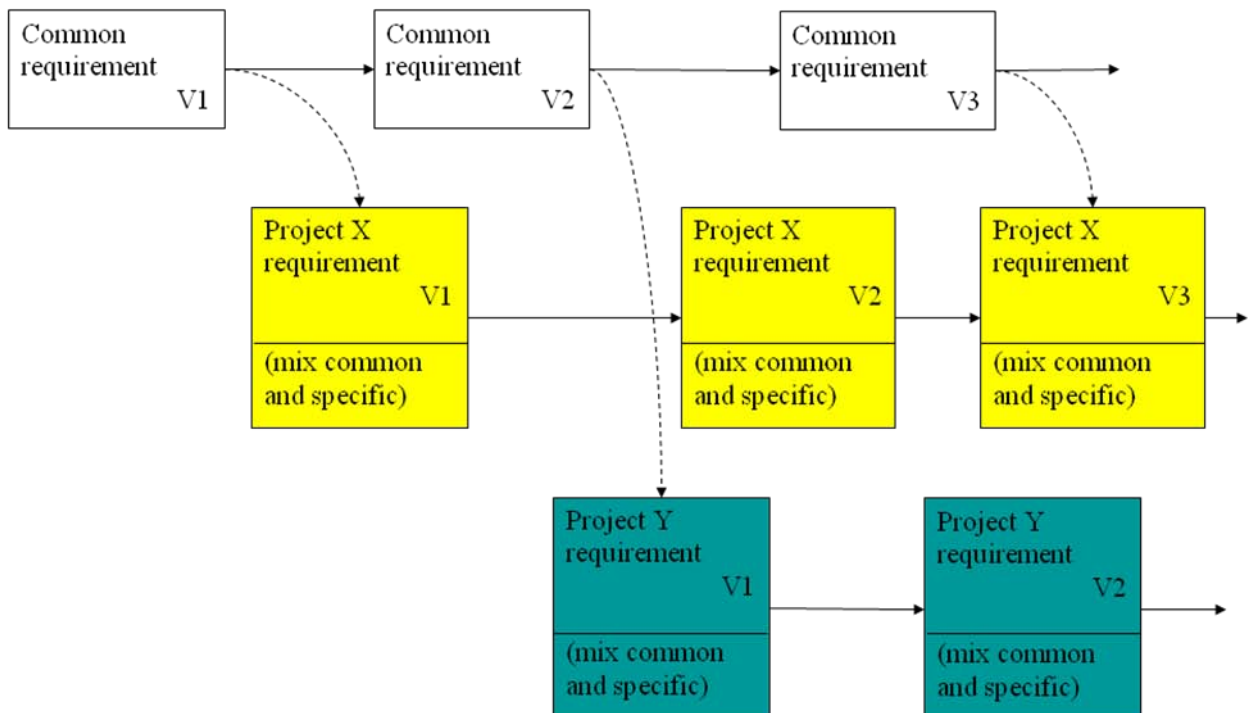
- To know at any time, for each project, the common requirements reused and the gap with the common (common requirements tailored to the project or additional specific requirements of the project).

- To anticipate verification / validation of common requirements to avoid carrying multiple corrections on project requirements that reuse these common requirements.
- To ensure consistency between the common and project specific requirements.
- To manage configuration of common requirements.
- To manage configuration of the requirements of the project reusing the common requirements.
- To synchronize changes between the common and the project specific requirements under change management.
- To follow the assembly rules and conditions for reuse of common requirements.

Reuse of requirements

Requirements reuse is a central part of the product lines engineering. The major challenge of this activity is to maintain traceability and managing the changing requirements in both the product line platform and the instantiated project. Different techniques for reuse of requirements are identified (source AFIS, Requirements WG):

Cloning:



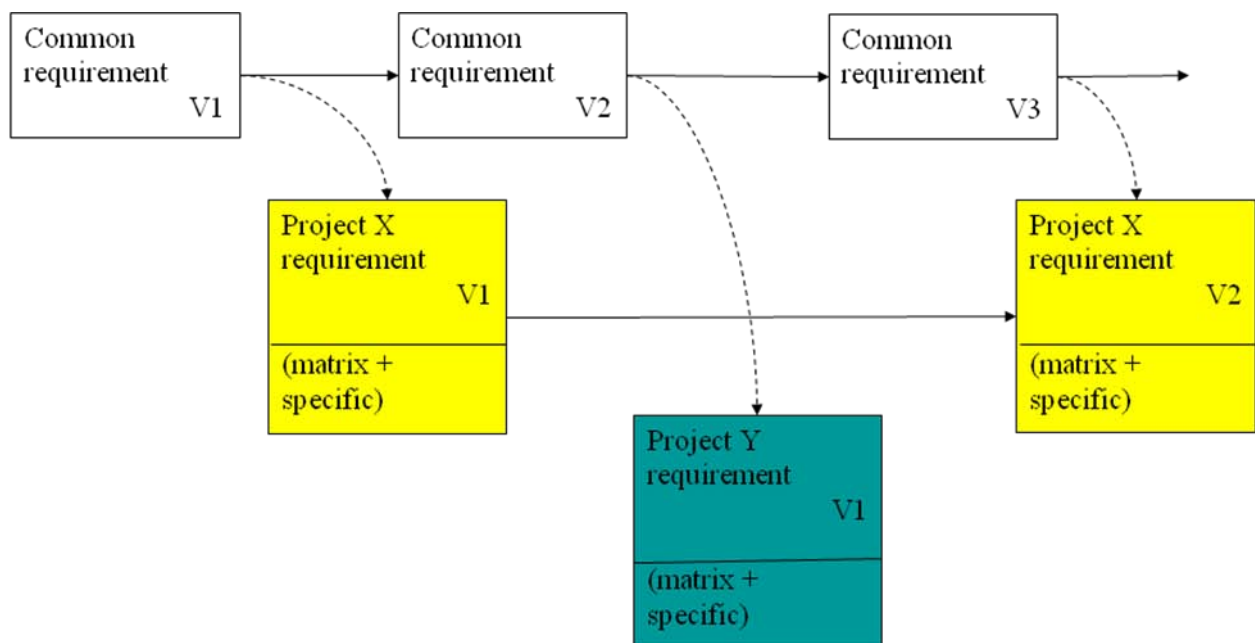
This technique can also be called “copy/paste”. It is characterized by the absence of physical relationship between the common requirements and the project requirements: at a time T the common is copied into the project. The common is a 'reference' to this time T. If the common

changes, the project can be updated by cloning the new version of the common requirements involved.

In a project specification, the cloned common requirements and specific requirements are not separated (no links between the common and specific).

The implementation of this technique is very easy and doesn't not require any specialized requirements engineering tool. However, it has two major drawbacks: (i) it is difficult to maintain the project requirements in relation to changing common requirements (if an impact is anticipated, it must be done manually), and (ii) capitalization is difficult because it is not clear how to identify what are the common requirements that have been instantiated in the different projects.

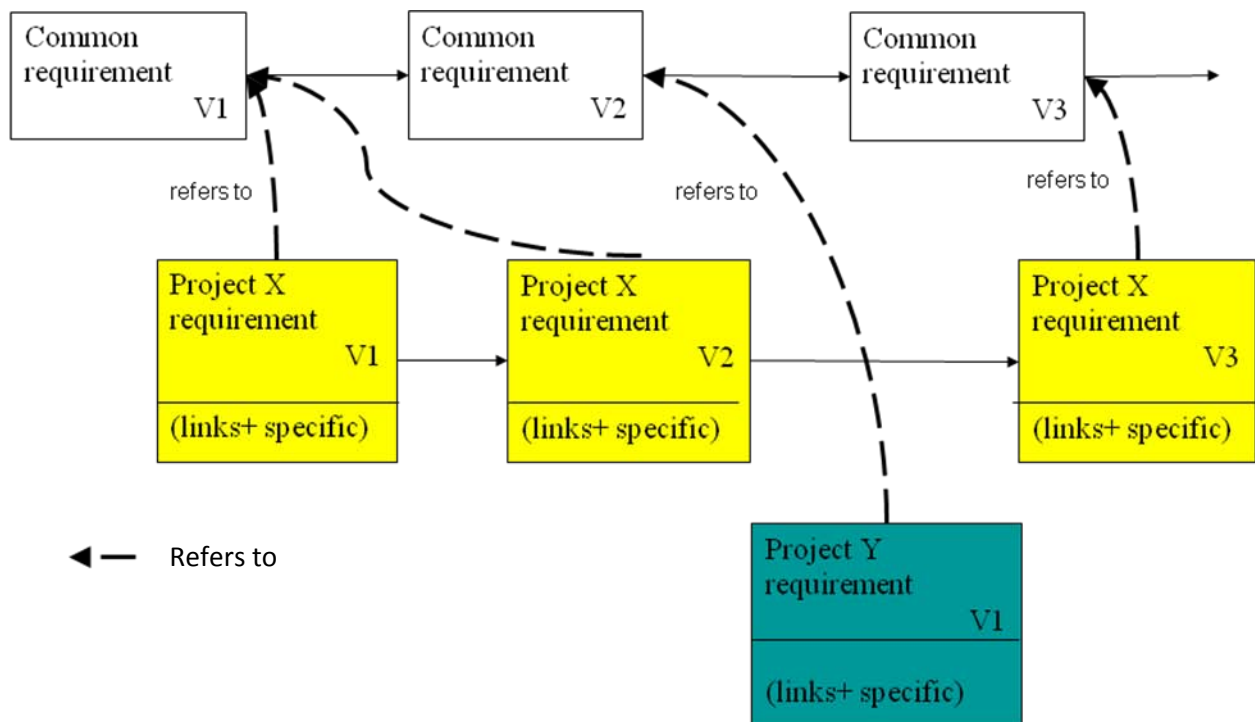
Applicability matrices:



In this technique, a common requirement is instantiated in a project with an applicability matrix. An applicability matrix defines for each common requirement if it is applicable or not on a given project and what is the adaptation made in the project. A project specification is composed of the applicability matrix and the specific requirements. If the common changes, the project can be maintained by updating the applicability matrix.

This technique does not require special tools. It allows a quick review of what is applicable or not and what are the adaptations of the common made in the project, which allow to work in gap with the common. But the maintenance of matrices is difficult since hundreds of requirements can be managed. If an impact analysis is expected, it is better to use a tool.

Reference to common requirements by links:



In this technique, a project requirement is instantiated with a link to a common requirement. A project requirement is composed of the common requirement description (referred with the link) and its possible adaptation in the project. Projects should be alerted for changes in the common requirements, to decide whether their inclusion in the project.

This technique is the most relevant. If the common changes, the project can be updated by updating the links. Also in the project, you can choose to automatically redirect the links to the new common requirements. However, managing the links between the project and the common can be a difficult task if not tooled properly.

Panorama of practices and tools

Panorama practices

The cloning technique is common practice. We might vulgarly call the technique "copy / paste". It does not require the implementation of specific tools. Any project starts from a repository and a specification from a previous project to initialize its specification.

The technique of the matrix is already somewhat less widespread. Although it does not require special tools, it requires a minimum level of maturity e.g. have (i) unique identified requirements, (ii) an enabling document management, versioning and archiving specifications and (iii) management changes synchronized between the common and specific project.

The technique involves links the implementation of tools for engineering requirements for versioning and traceability requirements, with the same prerequisites as the technique of matrix.

The techniques of matrix and links have been applications in satellite, automotive and aeronautics. Studies ROI level satellites have shown gains of about 3% of the cost of labor overall (source: Thales Alenia Space, Space Bus Product Family).

Panorama Tools

The tool offers on the product lines is very confidential. Many tools are based on research but deal with the management of the variability of marketing functions ("features") more than requirements engineering and traceability. These tools are more or less operational and are independent from requirements engineering tools. We can cite DecisionKing, AHEAD, XFeature, FMP and FAMA.

The commercial tools are more mature. However, the problem of product lines in systems engineering is relatively new, these tools are mostly in the software area (eg Big Lever, Pure:: Variant) and does not cover the problematic system (hierarchy of systems, concept of products, product variants / variant functions ...). Tools emerge at the system level (eg Requirements Central Enovia v6) but cover only some of the issues raised. Other commercial tools for requirements engineering manager of product lines are being developed (eg DOORS-RMF PFM - Product Family Management).

Reuse of requirements with DOORS RMF RCM

Background

DOORS tool is the market leader in the field of engineering requirements. It allows to define and track requirements. RMF means Requirements Management Framework. DOORS-RMF is an Add-On DOORS incorporating a flexible data model and utilities to accelerate the use of DOORS. RMF was originally developed by Thales and then taken over by IBM (Telelogic). RCM (Requirements Management System) in RMF allows to equip the versioning and change management requirements.

The choice to implement the capability to reuse the generic requirements in DOORS-RMF has quickly established itself as DOORS-RMF RCM allows control requirements through a process ad 'hoc prerequisite for reuse of requirements by links traceability. This monitoring requirement is assured by the RCM following functions:

- The versioning requirements,
- Change management in DOORS modules: a requirement is changed only if a 'Change Request' is associated

- The availability of spaces of reference / work
- The archiving of baselines and associated links,
- Managing the delivery of modules DOORS: archiving of exports a set of modules.

Placing a module controlled RCM means that we want to have traceability of all actions will be performed (creation, modification, ...). But above all, be aware configure RCM to limit access. Version 5.3 p1 of RMF requires the following configuration:

- One user must be Administrator RCM and CR (Change Request) Manager at the same time.
- The RCM administrator's role is to put the modules involved in RCM.
- The QC Manager is responsible for the creation and approval of Change Request.
- Under RCM RMF any modification requirements triggers a Change Request.

Developments made

There are no tools on the market able to instantiate application projects from joint projects. It should therefore develop language functions DXL (programming language DOORS: Doors eXtended Language).

The following functions have been developed:

1. A function to initialize a project specification (DOORS module) from a generic specification in a generic project. This function creates a draft specification to the same by selecting the generic requirements of latest approved under the management changes. The project requirements are identified and created automatically via links pointing to the generic requirements approved. The actor then has to draft a specification "ready to use" whose demands can be reused as is or adapted. This feature allows therefore an important time saving in the establishment of a project specification, safe operating requirements obsolete or unapproved.

2. A function to control the instantiation of generic requirements for the application projects. The purpose of this function is to indicate the generic customization of the generic requirements for projects subscribers (list of projects and list of requirements and customization by project). Thus a business manager may have a global vision of applying its repository in the different projects.

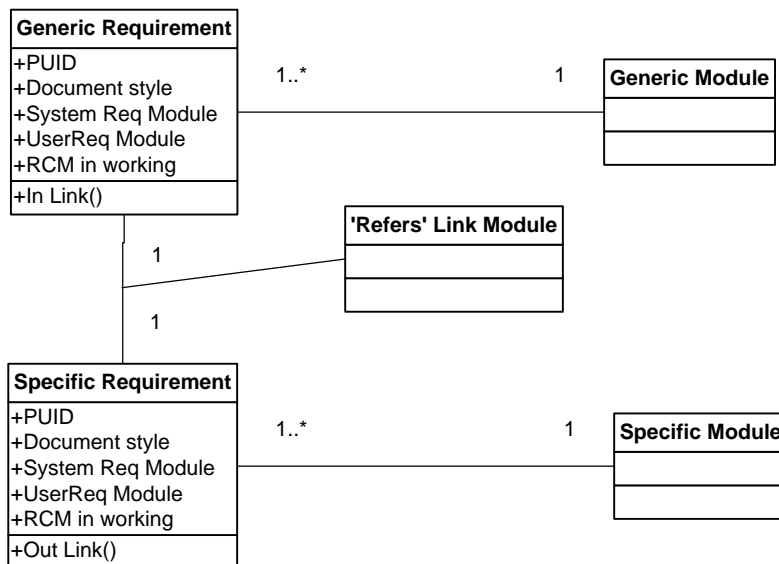


Figure 1. Data Model Structure Module generic / specific modules

3. An alert function that allows when changing generic requirements following the approval of a Change Request, to notify the application projects of an evolution of the generic. This alert is displayed as an icon at the relevant project requirements.

In the case of the issuance of alerts, it is possible to accept all changes and each change individually. This acceptance has the effect of automatically redirect the link to trace the requirement to draft the final version of the approved generic. Where the alert is refused, the link between the requirement of the project and the existing generic requirement is retained.

This feature therefore allows transactions to eliminate non-value added as manual redirection of links in case of changing the generic. The project stakeholders are masters of the decision whether to apply the generic developments.

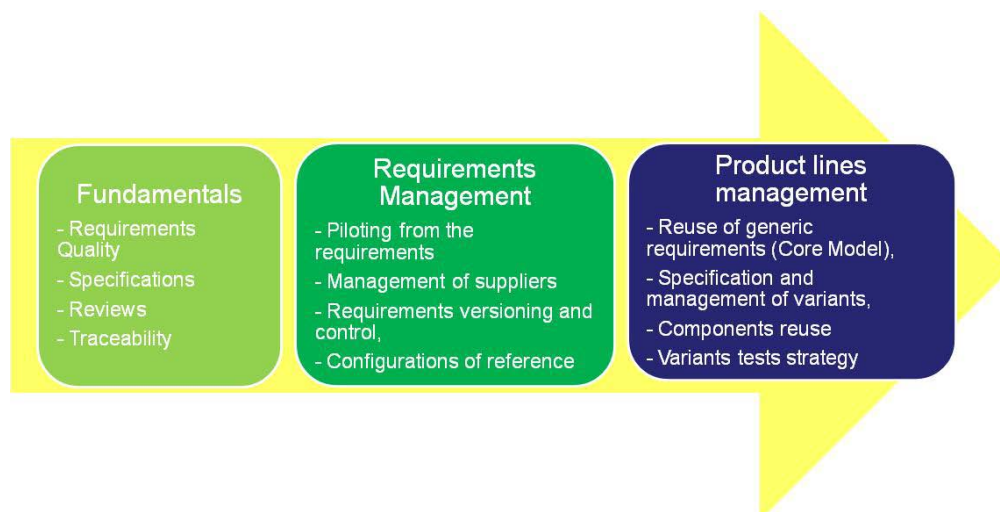
The following figure shows the computer implementation in DOORS. The first column contains the identifiers of requirements of the project. The third column contains the image of generic requirements in the project. The second column contains the customization requirements. The project requirement is the union of the 2nd and 3rd column. A project requirement is either generic as it is implemented (ie no text in the 2nd column), is adapted from the generic (ie text in the 2nd column) or specific (ie no links, not text in the 3rd column).

| PUID | Project Requirements | Generic Requirements | Alert |
|--------------|--|--|-------------------------------------|
| | 3.1.1 [F1] Warn the driver of an abnormal speed | | |
| | 3.1.1.1 Common requirements | | |
| | 3.1.1.1.1 Detection of speed | | |
| [SR-REQ-002] | Applicable as is → | ▶ The system shall detect speed signs. | |
| [SR-REQ-004] | | ▶ The system shall identify speed limits on speed signs | |
| [SR-REQ-001] | | ▶ The system shall display to the driver the speed limit | It exists a more recent requirement |
| [SR-REQ-003] | The system shall warn the driver when the vehicle speed is more than 5km/h | ▶ The system shall warn the driver when the vehicle speed is more than 2km/h | |
| | 3.1.1.1.2 Markets to consider | | |
| [SR-REQ-005] | The system shall be capable of operating for Europe ← | | Specific to the project |
| | 3.1.1.1.3 Warning conditions | | |
| [SR-REQ-006] | 75km/h ← | ▶ The speed system shall operate on following conditions, at least. • Vehicle velocity : This system shall operate over __km/h (T.B.D during development phasis). | |
| [SR-REQ-007] | | ▶ The speed system shall not operate on following conditions: at least, is failure. | |
| | 3.1.1.2 Warn the d by a sound | | |
| | 3.1.1.2.1 MMI | | |
| [SR-REQ-008] | | ▶ The sound warning shall be perceivable haptic and/or audible. | |
| [SR-REQ-009] | | ▶ the buzzer sound pattern is defined as Fig.3. | |
| | | figure 3 | |
| [SR-REQ-010] | | ▶ The system shall equip two or more than buzzer | |

Figure 2. Implementation in DOORS-RMF

Conclusion

The reuse of requirements requires a minimum maturity Engineering System. First of all we control the fundamental requirements of quality, traceability practice but also repositories Trades capitalizing a robust feedback to reuse.



It is also necessary to coordinate change and configuration of both generic and

application projects, with an impact on the organization.

The reuse of requirements is the starting point of a wider re-use, allowing more savings for a line of products. This includes expanded reuse architectures but also testing. Thus, it is possible to control the reuse of architectural components and capitalize on generic tests already performed.

To go further in terms of requirements, further study would be to integrate the level of specific size variability.

Biography

Gauthier Fanmuy serves as Technical Director and head of the Skill Center "Systems Engineering" at ADN (<http://www.adneurope.com>), a consulting company. He has a particular responsibility for the expansion strategy of ADN in the industry outside the life sciences field, the technical coordination of Skill Centers, the deployment of methods and tools related to transverse Systems Engineering and Project Management.

Within AFIS (<http://www.afis.fr>), he is the leader of the Technical Committee "Global Processes".

Within INCOSE he is the Leader of the "Systems Engineering for Very Small and Medium Entities WG". He is involved in several WG such as bio-Medical, Lean and Requirements.

He previously worked at PSA Peugeot-Citroen, where he was responsible for the implementation of engineering requirements on a project vehicle. He was also responsible for the deployment of System Engineering in a department in charge of vehicles engineering and testing.

Prior to this experience, he worked at Dassault Aviation on weapons systems Rafale, Mirage 2000-9, F1CR Mirage and Atlantic 2, where he served various responsibilities in the quality, development of tactical functions, integration of opto-electronic equipment, covering all activities of the lifecycle of a system.

He is an Engineer from the Ecole Centrale de Marseille.

Olfa Djebbi is a PhD student in Computer Science in the ICC (Center for Research in Computer Science), University of Paris1-Panthéon-Sorbonne and member associations AFIS and INCOSE. She has also worked at ADN as a Systems Engineering consultant in the field of product lines and requirements engineering.