Managing (Requirements) Evolutions of High Assurance Systems

Michel Lemoine
ONERA CERT Centre de Toulouse
DPRS/SAE
Michel.Lemoine@cert.fr

Jack Foisseau
ONERA CERT Centre de Toulouse
DTIM/MIB
Jack.Foisseau@cert.fr

Abstract

Long lifetime HAS (High Assurance Systems) present, among others, a peculiar property: evolutions are numerous.

Because current standards [1] for producing such HAS are not accurate enough regarding evolutions, we have considered that all the artefacts, which are produced during their development, should be recorded. Recording artefacts means developing an IS (Information System), and using it in the same way it is done with classical IS.

Applying well-known IS principles supported by RDB (Relational Data Base), we have first of all considered their models, and then their exploitation.

For the modelling part we have taken into account all the artefacts, and their relationships, according to accurate representative UML abstract diagrams. Indeed UML [2] allows representing both static and dynamic aspect of any system.

Managing evolutions being the most difficult part of the HAS lifetime, we have put a special emphasis on modelling requirements, and evolutions.

We then have shown how these abstract UML meta-models, and their instantiations, can be used in two different ways, according to we are developing a system or we are managing its evolution.

In the former we have built up a Web DB (Data Base), which takes advantage of existing browsers.

Consequently one is then able to navigate through HAS artefacts. Depending on his role in the development, one is able either to follow the HAS development, and consequently to study whether HAS requirements are met, or to apply various analyses such as the impact analysis of any requirement change.

In the latter, because recording all the artefacts is a bit heavy, we have translated the abstract meta-models into a set of verification rules that allow checking manually some HAS properties such as release compatibility.

The used standard is the EAI-632. It has been improved, and, to a certain extent, automated in such a way that both recording development artefacts is feasible, and analysing impact of any requested evolution is no more done manually.

The industrial experiments that have been conducted along a very large project have shown the availability of the chosen approach.

A by-product that has appeared during the first experiment was the introduction of new relationships between end-users and the development team [3]. Indeed it has been confirmed that new or improved technologies must be thought carefully before any real implementation. In our case, a mutual trust between teams has appeared.

[1] Processes for Engineering a System, Dec 1998, www.geia.org/eoc/G47/main.html

[2] J. Rumbaugh et al., *The Unified Modelling Language Reference Manual*, Addison-Wesley, 1998

[3] J. Foisseau and M. Lemoine, *Gestion des exigences pour la maîtrise de la pérennité*, in Actes de la 2e Conférence Annuelle d'Ingénierie Système organisée par l'AFIS, Juin 2001, Toulouse, F

Proceedings of the IEEE Joint International Conference on Requirements Engineering (RE'02)
1090-705X/02 $17.00 © 2002 **IEEE**