

RE in Flatness Measurement and Control Systems Development at ABB

Nur Yilmaztürk

ABB Automation Technology Products AB, 721 59 Västerås, SWEDEN

nur.yilmazturk@se.abb.com

1. Introduction

The Stressometer product line (for more information, refer to www.abb.com) involves software intensive systems, which have been providing rolling mills with accurate online control of the flatness of cold rolled strips for more than 30 years. Over these years, ABB has been continuously improving these products parallel to the technological progress in software engineering in an attempt to keep its number one position in the market.

New generation Stressometer systems are implemented by using SUP (Stressometer Unified Process) that is RUP[®] (Rational Unified Process[®]) tailored to fit the Stressometer department's development projects' needs. It facilitates use-case driven, iterative development during which modeling is done via UML. This paper presents the requirements engineering activities that are involved in the SUP, and discusses the problems faced during practice thus, putting forward issues for the attention of research.

2. Practice and Issues

The first step in the requirements engineering process via SUP is the elicitation of requirements. Representatives of the external customers, who buy the system, and the internal customers, who do the development, installation, and maintenance of the system as well as the research department, take part in the requirements elicitation process. The internal customers bear the required domain knowledge. Such a rich blend of different interests in the final product is expected to increase the completeness of the requirements; nevertheless, it is not a homogeneous group hence, increase in the severity and number of conflicts faced. Firstly, some of these stakeholders do not have a software engineering background. Secondly, those who take part in requirements elicitation have too busy schedules to fit number of requirements workshops or interviews into it. Thirdly, the domain experts have a large amount of tacit knowledge which is highly difficult to retrieve.

Accordingly, the research should focus on developing methods that will require very little or no active participation of the stakeholders or domain experts. Alternatively, one can also work on the available methods to make the elicitation process more efficient. For example, we see a great potential in preparing use-case model and use-case specification patterns for different industrial domains. Another example could be to provide the stakeholders, domain experts with no software engineering background with automated tools that can

direct them entering their knowledge and translates it both to and from UML notation thus, facilitating also the specification and validation of requirements.

In industrial systems like Stressometer where advanced control theories are applied, mathematical functions impose requirements on the way the users perform their tasks. Moreover, they are a part of the domain knowledge. However, use-cases, traditionally, should describe only the outwardly visible requirements of systems.

The way we handle this issue with SUP is to document the mathematical descriptions in a separate document, which is a Supplementary Specification to the Use-Case Model, and refer to the sections of the document from the relevant part of a use-case specification. Could we extend the UML notations to include such knowledge explicitly in the use-case model or in the use-case specifications?

Finally, by the end of each iteration, and before delivering a system to a customer, we need to verify the requirements at system level. For function testing, one can identify the scenarios out of the use-cases by describing the paths through each use case that traverse the basic flow and alternate flows start to finish through the use case; subsequently, test cases are developed for each use-case scenario by varying the conditions.

Issues raise during this activity are two-folded: (i) No method exists describing how to prepare a complete set of test cases out of use-case scenarios. Can we determine full set of conditions to generate functional test cases by only looking at the use-cases? (ii) Using UML, which is a semi-formal language, for requirements specification brings about the risk of incompleteness, inconsistency, and ambiguity in the requirements definition which will be inherited to the verification process. Could we formalize the UML notations so that we could avoid ambiguities, and detect inconsistencies and incompleteness in requirements at the early stages of development? Could UML notations be complemented by the existing formal notations in such a manner to increase the quality of the requirements specification? Could we prepare tools that would enable us analyzing formal UML models? Could these tools also aid in early and fast prototyping of the systems hence, verification of the requirements much earlier than the system test, even before implementation?

3. Conclusions

The paper presented requirements engineering activities performed throughout the SUP (Stressometer Unified Process). It put the emphasis on the problems faced during practice thus, suggested issues to be dealt with by research.