Towards an Efficient and Effective Process for Integration of Component-Based Software Systems

Stig Larsson
ABB Corporate Research
stig.bm.larsson@se.abb.com

Abstract

The integration of software components into working products is one of the major challenges in product development. Component-Based Software Engineering research shows initial results in increased efficiency and effectiveness. However, additional research is needed to secure that the specifications of software components include sufficient information to respond to requirements from the integration process.

The research outlined in this paper will be based on current industry experiences as well as research in software engineering. It will result in proposed additions to current software component models and integration processes.

1. Introduction

The research in Component-Based Software Engineering (CBSE) has resulted in descriptions of requirements on individual components as reusable units, and the system aspects related to combination of components [1]. The solutions proposed in the CBSE research offers gains in development effort, maintainability and time-to-market [2].

However, that additional information is needed in the proposed component models as described by Yacoub *et al.* [3]. Specifically, through recent experiences of current practices in product integration, we can foresee that additional requirements on the component models need to be specified. This will most likely also call for additions and changes to the integration process. We expect these additions in both component models and processes to improve the efficiency and effectiveness of the integration, test and verification processes.

This paper describes the challenges in product integration process and the research we have the intention to conduct to enhance the current components models and the integration process. The first section describes the problem, lays out a roadmap for the research needed and lists related work. The second section describes in more detail the plans for the research activities and the challenges that we anticipate. The third section describes the results expected from the research.

1.1. Software integration challenges

The product integration process for software products addresses the assembly of software components into complete products. Figure 1 shows some of the related process areas. The technical solution processes develop the components and the necessary component data that is used by the product integration process area to integrate the complete product. Both the technical solution and the product integration processes rely on the verification and validation processes to ensure that product components meet specified requirements and fulfill the expectations of the customer. Typically, the software product integration is done according to a defined plan, either in one stage or incrementally.

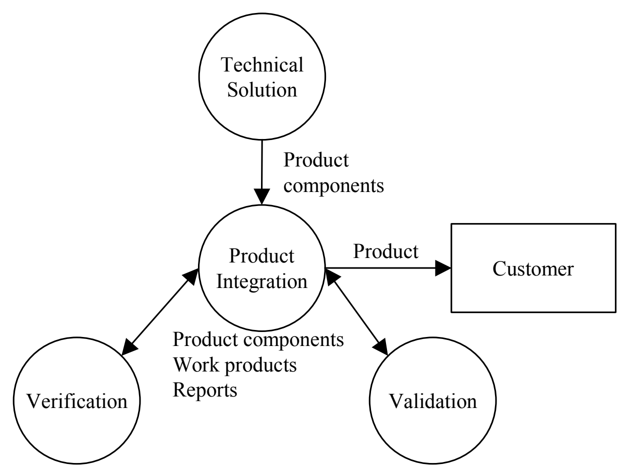

Figure 1. Processes related to product integration

The interactions between the processes rely on clear rules and agreements between the project members. Although good practices have been described [6][7], we have seen serious friction between engineers conducting the development, integration and verification tasks. One common problem is that a component has not been tested in the right environment. This indicates that additional knowledge and support for the relations are needed to achieve efficiency and effectiveness.

Besides well working process, efficient integration of software requires that the parts to be integrated are well defined. Software development within the industry today uses components that are only rudimentary defined. For example, the Common Object Model (COM) specification from Microsoft defines a component as "a piece of compiled software, which is offering a service" [4]. The lack of precise definitions leads to tedious checks at integration time, for example that the right interfaces are used or that the right prerequisites for the use of a component are fulfilled. It also makes automated checks and tests cumbersome.

Effective integration will take place when errors are easily found and components are verified before the integration. The ability to confirm the qualities in a component depends on what attributes are available in the interface definition. Solutions currently used in the industry do not include significant information of this type.

Both the efficiency and effectiveness thus relies on the ability to clearly specify the components. More complete specifications of components display properties that follow the implications from the definition made by Szyperski. [5]. These include that the component should be independent from its environment, have clearly specified interfaces and that the component development should be independent from the integration. However, current models have little or no provision for the non-functional parts such as test coverage.

1.2. Research goal

The main goal with the proposed research is to suggest and evaluate improved integration processes for systems built on well-defined components. The focus is on industrial systems with specific requirements such as reliability, maintainability and availability. The main research questions are related to the particular characteristics of the integration process and the requirements for the component development.

The hypothesis is that an efficient and effective integration process requires components with additional capabilities and characteristics. The characteristics must include both process-oriented attributes such as review coverage and product oriented attributes such as performance. These are needed to secure that the assembly of parts results a product or systems with the expected functional and non-functional characteristics.

1.3. Related work

Descriptions of the product integration process are available from many sources. The Software Engineering Institute lines out the requirements on the process in the Capability Maturity Model Integration (CMMI) [6]. Jacobson et al [7] and Rational [8] provides more direct technical descriptions.

Additional references will be one of the results from the research.

2. Research method

The research is divided into four phases, and care will be taken to evaluate each step to improve the methodology and results of the next step.

2.1. Research phases

The first phase is to describe current problems in the integration process for software products. In addition to mapping the state-of-the-art, experiences will be collected from different product development organizations working with technologies that are component based in varying degrees. The type of data will include statistics from the integration and test processes as well as interview material. We will examine the use of software as a part of the solution in both consumer and industrial products. The questions that need an answer include the following:

- Is there an integration process that can be observed as a specific process?
- What are the problems in the current integration process?
- Is the integration a part of the product development project or treated as a separate project?
- When in the development lifecycle is the integration prepared and started?
- How are components handed over to integration? What types of reviews and tests are performed before the components are delivered to integration?
- Who drives the integration? Is there a push from the development to the integration or is there a pull from the integration?
- How are the results from the integration process verified?
- How is the effectiveness in the integration process measured? Does the metrics collected include data regarding reviews made, errors found and success rates both from the steps before, under and after the integration process?

The second phase is to examine current research in Computer Based Software Engineering, concentrating on covering the product integration process. The efforts will be focused on examining results from experiments and experiences from implementations, but will also cover theoretical model specifications. The results will be compared to the problems found in the industry data to identify practices that can be easily transferred to the industry.

The third phase is to propose improvements to the process for integration of component-based systems. Based on the improvement opportunities identified in the state-of-the-practice study, the current solutions for component based software systems will be extended. This effort aims at describing the additional requirements on components as well as the integration process.

Finally, the proposed process improvements will be verified in industry and lab environments and be the foundation for the conclusions on how the process can be integrated in current practices. The verification will include tests of the additions to current component models as well as to the processes. The phase will also include preparation of proposals for how the results from this research can be disseminated.

2.2. Research challenges

The first challenge is the gap between solutions available in the research community and the methods used in the industry. The current State-of-the-Practice for product development includes only to a very limited degree components that build on State-of-the-Art specifications. This results in two possible obstacles for the outlined research.

Experiences that can help in defining additional requirements may be difficult to find. Problems found may be of basic nature and hide problems in the areas of interest for the research area.

The verification of the additional value of an extended set of requirements on the component specifications may be cumbersome. Systems, projects and organizations that can be used to test and verify the research results may be difficult to find or the proposed solutions difficult to implement in industrial systems.

However, as the problems in current projects will be used to model the requirements on the component specifications and the component models, we anticipate that the experiences are possible to interpret and extrapolate to product development using modern component specifications. In addition to this, we have the intention to run experiments with State-of-the-Art technologies. This will also increase the validity of the findings.

Another challenge is to restrict the research to areas that lead to the goals. Different levels of the integration process must be covered and examined. Figure 2 shows a model with different research areas that constitute the product integration process. All three areas need to be described to give a balanced view of the process.

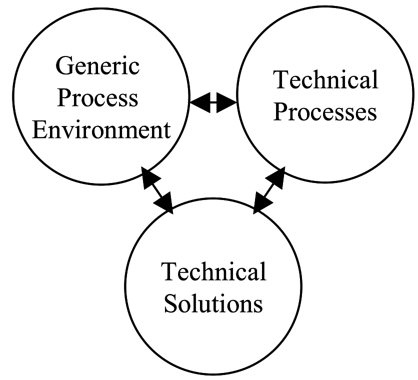

Figure 2. Research areas for the integration process

The generic process area includes the high-level descriptions of other parts of the product development process. One example where an investigation is needed is the fact that the basis for a well functioning integration process can be found already when requirements are developed for the product. The information included in the requirement specification reveals the relative importance between the functions and performance requirements, and can thus be used to govern the implementation of the integration.

The integration process planning and execution are also depending on the selected development project life cycle. However, a well working process for integration adds to the knowledge about status in the project and the quality of the product, information that is needed for well-grounded decisions regarding the project.

The more technical processes define how the work in the product integration is performed. This includes among other things the activities to specify components, describe component interfaces, define and perform tests, review of components delivered to integration and the collection of metrics to monitor the performance. The activities are performed in different ways depending on the life cycle model or development model selected for the project. Daily builds [9] will, as an example, require the developers to deliver tested components at specific times during the day, while eXtreme Programming [10] will require the developers to run regression tests before the integration.

Finally the technical solutions will put requirements on the component specifications. Typically, automated tests and automated reviews of interfaces require well-defined models with extensive information regarding the behavior of the component.

3. Expected research results

The results from the research will be the following.

- Experience reports from the different product development organizations examined
- State-of-the-Art and State-of-the-Practice descriptions of the software product integration area.
- Proposals what should be added in existing component models
- Proposals for an efficient and effective process for the product integration of systems built on well defined components
- Proposals for changes in the current practices for software development based on traditional components
- Proposals for how the results can be disseminated

4. References

[1] Crnkovic, I., and M Larsson, *Building reliable component-based software systems*, Artech House, Boston, 2002.

[2] Brown, A. W., *Large-Scale Component-Based Developments*, Prentice Hall, Upper Saddle River, NJ, 2000.

[3] Yacoub, S., H. Ammar and A. Mili, Characterizing a Software Component, *Proc. Int. Workshop on Component-Based Engineering,* Kyoto, Japan, 1999, http://www.sei.cmu.edu/cbs/icse99.

[4] Microsoft, The Component Object Model Specification, Report Vol. 99, Microsoft Standards, Redmond, WA, 1996.

[5] Szyperski, C. *Component Software -- Beyond Object-Oriented Programming*, Addison-Wesley, Reading, MA, 1998.

[6] Chrissis, M.B., M. Konrad, S. Shrum, *CMMI,* Addison-Wesley, Boston, MA, 2003

[7] Jacobson, I., G. Booch and J. Rumbaugh, *The Unified Software Development Process*, Addison-Wesley-Longman, Reading, MA, 1999

[8] Kruchten, P. *The Rational Unified Process An Introduction*, Addison-Wesley, Reading, MA, 2000

[9] Cusumano, M.A., R. W. Selby, *Microsoft Secrets*, Touchstone, New York, NY, 1995

[10] Beck, K., *Extreme Programming Explained*, Addison-Wesley, Reading, MA, 1999.