REASONS FOR INTEGRATING SOFTWARE COMPONENT SPECIFICATIONS IN BUSINESS PROCESS MODELS

Marie-Therese Christiansson
Division for Information Technology, Information Analysis,
651 88 Karlstad, Sweden

Benneth Christiansson
Division for Information Technology, Information Analysis,
651 88 Karlstad, Sweden

Keywords: Software Component Specification, Business Process Modelling, Acquisition

Abstract: Organisations, business processes and co-workers are in a ”never-ending change-mode”. It is therefore unrealistic to expect any definitive requirements for computer based information systems. In this paper we argue for the need of bridging the gap between business process modelling and software component specification. By using a core business process model that integrates both essential knowledge concerning business processes and their possible improvements and also integrates software component requirements in the form of software component specifications; IS professionals should be able to judge the potential, development and management of component-based information systems. This implies the need of an “informal” software component specification that is grounded in business processes and created for the people who are best suited to model requirements, i.e. people who run and perform business. This “close to business” specification can be expressed on a high-level and in an informal manner due to the fact that the specification does not have to serve as software development requirements because the software component already exists, the difficulty is acquisition. With an integrated core business process model we have the possibility to perform modelling more effective and achieve more benefits and use it as a foundation for software component acquisition. We can focus on business actions and their constant changes and at the same time identify the corresponding changes in software requirements.

1 INTRODUCTION

Langefors (1995, p. 142) describes the development of software systems as finding the solutions to: *”Two fundamental problems with information systems were pinpointed at the outset: (1) The "infological” problem of how to define the information to be made available to the information system user, and how to design data that may represent the information to the user; and (2) The ”datalogical” problem of how to organize the set of data and the hardware so as to implement the information system.”.* We believe that emphasis during software development needs to be on both these problem areas. In today’s software engineering community the focus is on the “datalogical” side. We believe that component-based systems development delivers an opportunity to focus more on the “infological” part of software systems development. This is due to the fact that the software already is developed. Software components already exist, we do not need to express requirement in such details that we can construct the software based on them, we only need enough details to enable acquisition. This situation is more similar to acquisition of application packages rather than traditional software development. We argue for a software component specification in the ‘infological’ side. This specification should be grounded in business processes that captures the “what, why and for whom” the system is developed. We also believe this approach to be an informal strategy that captures requirements based on business practice and

Christiansson M. and Christiansson B. (2004).
REASONS FOR INTEGRATING SOFTWARE COMPONENT SPECIFICATIONS IN BUSINESS PROCESS MODELS.
In *Proceedings of the Sixth International Conference on Enterprise Information Systems*, pages 647-650
DOI: 10.5220/0002615906470650

describe them in such a way that people, who are best suited to model requirements, find them usable. Our approach for specifying software components is based on the integration of results from two research fields; 1) Component Based Software Engineering and 2) Business Process Modelling.

A study of 20 approaches in software component specifications shows that the main focus in software component specification is towards the 'datalogical' side mainly concerned with construction and somewhat regarding assembly (Christiansson & Christiansson, 2003). This does not imply any "integration" effort. We argue for the need of specification strategies that takes advantage of the new possibilities and handles some of the challenges such as enabling acquisition of software components.

2 PROBLEMS WITH COMPONENT-BASED SOFTWARE DEVELOPMENT

Component-based software development should mean that software systems are created through the assembly of more or less standardized software components into unique software solutions. *"Although each bought component is a standardized product, with all the attached advantages, the process of component assembly allows the opportunity for significant customization."* (Szyperski, 2002, p. 6). A key reason for the interest in component-based software development is the possibility to reuse. In many approaches, reusability is not inherent in the development process. One problem is time and effort required for development of components. Reusability requires generality this requires increased time and effort to develop. This also implies that to be widely reusable, a component must be sufficiently general, scalable, and adaptable; it will therefore be more complex (and thus more complicated to use) (Steel, 1996). Another problem is unclear and ambiguous requirements (Beck, 2000; Heineman & Councill, 2001). In general, requirements management is an important and complex phase in the development process, its main objective being to define consistent and complete component requirements. Yet another problem is component maintenance costs. Although application maintenance costs can be lowered, component maintenance costs can be very high since the component must be able to respond to the different requirements of different applications running in different environment (Steel, 1996). How do we express requirements? Where do we find requirements? How do we know if a software component fulfils requirements? These are all problems that occur when acquisition is at hand. By using an integrated core business process model we enable the possibility to have a source of knowledge to be used when:

- Software systems need to be integrated or at least connected with each other to enable structural changes within business practices.
- Software systems overlap with in-house systems and create a problem regarding from which system the functionality should be used.
- We need accurate, correct and usable documentation other than the source-code in itself.
- Predicting the feasibility of the final implementation.

3 A SOFTWARE COMPONENT SPECIFICATION - CLOSE TO BUSINESS

Information systems and information technology can be viewed as the backbone of the modern enterprise and as such crucial to its supporting and providing new possibilities of running business operations. Information technology should, like Hammer (1990) states, be used as an enabler to perform innovations which makes a difference to business customers/clients. Business modelling is the work of reconstructing and describing how business operations have been run, are being run, may or should be run to achieve a basis for observing and comprehending business operations (Tolis & Nilsson, 1996).

We need an approach to specify requirements on software components to identify actual needs. Communications patterns need to be clarified assessed and developed, to achieve efficient business operations, internally and in co-operation with others. Business actions and their conditions and requirements of co-workers, knowledge, machinery; flows of information, materials and payments; must be distributed and coordinated. Part of this work is to establish the roles and responsibilities needed to perform business operations (Christiansson M-T., 2001). Another part is to determine if business actions should be supported or run by a software system, to efficiently provide people with information that is to be used in performance of operational actions (Langefors, 1973). Business models are founded on two basic needs, namely *"... trying to understand and trying to change"* business operations in organizations (Tolis & Nilsson, 1996, p. 10). Understanding and changing organizations signifies that the starting point should be what is actually carried out in these. Taylor (1993, p. 112) underlines this as follows:*"... communication (and therefore organization) is grounded in action, not in information transmission, nor even in the transfer of knowledge."* In other words, the focus in system development should be on the way we carry out tasks and on the communication pursued as prerequisites for, and results of, these activities. A "software component specification – close to business" needs a process framework to indicate which essential matters of business operations and software components to describe in an integrated core model (Christiansson & Christiansson, 2004 b).

4 A SOFTWARE COMPONENT SPECIFICATION - CLOSE TO BUSINESS CHANGE

With an established motive with modelling namely software component acquisition, you don't want to or need to describe all possible variants of business processes at present ("is-processes"). Focus can be on desired business processes with the software components as enablers ("should-processes"). Like Hammer (1990, p.104) states; *"use computers to redesign - not just automate existing processes... companies tend to use technology to mechanize old ways of doing business. They leave the existing processes intact and use computers simply to speed them up."* The purpose is not to explain all kinds of performance in a business process, rather to describe desired business processes in more focused terms of "to what result and for whom". If we can use the same business process model to capture the business situation at the present time (the "Is"-process) and at the same time use the model to capture requirements for the desired business situation (the "Should"-process) we can get to the systems development faster and produce fewer documents along the way, This will in the end lead to an increased possibility to reach the "should"-process description at all, which in today's business modelling is far from always the situation.

Our approach does not rest on the notion that it is possible to define and describe all system requirements and software features beforehand, or at least very early in the development process. (Christiansson & Christiansson, 2003). We believe that software requirements should be defined in the description of the actual business actions it is intended to support. Thereby we enable the possibility to cope with the problems of missing and/or inaccurate requirements as well as coping with constant business change. By using one core integrated model changes can be updated and tested in one type of model without the time consuming job of updating several related documents. Beck (2000, p. 3) illustrates this problem as *"Business misunderstood – the software is put into production, but it doesn't solve the business problem that was originally posed. Business changes – the software is put into production, but the business problem it was designed to solve was replaced six months ago by another, more pressing, business problem."*

By using an integrated core business process model we enable the possibility to:

- Be able to capture requirements based on the desired business processes expressed by people who run and perform business operations.
- Handle that people like doing things the same "old" ways by visualizing and comparing the "Is-processes" with the "Should-processes", i.e. improve business processes supported by software system.
- Be able to gather knowledge of business operations and their supports of software components to enable changes, tests and maintenance.
- Handle the rapid changes in business practices that results in new and changed requirements on software, as well as further reorganizations of business processes.
- Enable an ongoing business process improvement which handles new or changed software requirements.

5 A SOFTWARE COMPONENT SPECIFICATION - CLOSE TO PEOPLE

Our opinion is that we need to delimit the information amount in a software requirement specification to bare essentials with a pragmatically language in an integrated core model which different stakeholders can relate to. *"The various stakeholders ought to be allowed to specify desires or requirements in a language that they could understand... this language ought to be efficient for their talking about their own reality."* (Langefors, 1995, p. 71). We do agree on the fact that it is important to let the people who know the business state their requirements in their own business language. This may though lead to communication problems as Nellborn (1999, p. 201) states it*; "...business development and information systems development are seen as separate issues performed by different people and not as two components of the same solution to a problem."* A rewarding strategy is to be able to capture requirements based on the desired business processes expressed by people who run and perform business practice in an integrated model. A more focused modelling might give more value in fewer models. This assumption is based on the nature of development processes. A system development is dependent on knowledge from different stakeholders who want to spend as little time as possible in a useful manner. Christiansson & Christiansson, 2004 a)

By using an integrated core business process model we enable the possibility to:

- Ease the difficulty to state and to communicate requirements by found those in daily practice.
- Handle the natural uncertainty and lack of understanding regarding software requirements by found those in daily practice.

6 CONCLUSIONS

In the increasingly competitive software industry the need for new innovative techniques to deliver satisfying software systems is greater than ever. This may in some sense explain the great belief and adaptation of new techniques, strategies and/or tools that are made out to be the great solution to the different problems that the software industry is facing. We believe that by combining business process modelling with software component specifications, we have a solution that addresses and handle many fundamental problems with software development. This core model can be used as an enabler of acquisition. The barriers between business analysis and systems development can be mended by embedding focus on software components within the business process model. By using a business model that integrates essential knowledge concerning actions in business processes and software components ability to support, gives new innovative ways to perform business actions, and new ways to handle management of component-based information systems. With our approach we avoid treating business development and information systems development as separate issues, and instead regard them as one integrated task. We believe that using an integrated core model we can capture essential knowledge concerning software component requirements based on the desired business processes expressed by people who run and perform business practice. We can also facilitate the fact that software requirements change at the same time as the business change and a core business process model can be used without the time consuming job to update a lot of related documents. A core business process model may serve as a basis to enter deeply in related models to capture knowledge but with different motives, perspectives and purposes than software component acquisition.

REFERENCES

Beck, K. (2000) Extreme Programming Explained, Addison Wesley Longman, Inc., California, Menlo Park, USA.

Christiansson, M-T. (2001) Business modeling in inter-organisational cooperation – what to describe, why and how? in Nilsson, A. G. & Pettersson, J. S. (Eds.) On Methods for Systems Development in Professional Organisations – The Karlstad University Approach to Information Systems and its Role in Society, Studentlitteratur, Lund.

Christiansson, B. & Christiansson, M-T. (2003) The Missing Approach for Component Specification, in Assar, S., Semmak, F. & Barkhi, R. (Eds.) Proceedings of 1:st International Workshop on Component-Based Business Information Systems Engineering (CBBISE'03), 2 September, Geneva.

Christiansson, M-T. Christiansson, B. (2004 a) Extreme Modeling – Less is More, Proceedings of IRMA 2004, 23-26 May, New Orleans (forthcoming).

Christiansson, M-T. Christiansson, B. (2004 b) A Generic Framework for a Software Component Specification – founded in Business Processes, (submitted paper).

Hammer, M. (1990) "Reengineering Work: Don't Automate, Obliterate", Harvard Business Review, Vol. 59, No. 4, pp.104-112.

Heineman, G. T. Councill, W. T. (2001) Component-Based Software Engineering Putting the Pieces Together, Addison Wesley Longman, Inc., California, Menlo Park, USA.

Langefors, B. (1973) Theoretical Analysis of Information Systems, 4:th ed., Studentlitteratur, Lund.

Langefors, B. (1995) Essays on Infology – Summing up and Planning for the Future, Studentlitteratur, Lund.

Nellborn, C. (1999) Business and Systems Development: Opportunities for an Integrated Way-of-Working, in Nilsson, A G. Tolis, C. Nellborn, C. (Eds.) Perspectives on Business Modelling – Understanding and Changing Organisations, Springer-Verlag, Berlin.

Steel, J. (1996) Component Technology Part I An IDC White Paper, Proc. Int. Data Corporation, London.

Szyperski, C. (2002) Component Software Beyond Object-Oriented Programming, Addison Wesley Longman, Inc., California, Menlo Park, USA.

Taylor, J. R. (1993) Rethinking the Theory of Organizational Communication: How to Read an Organization, Ablex Publishing, Norwood, New Jersey.

Tolis, C. Nilsson, A. G. (1996) Using Business Models in Process Orientation, in: Lundeberg, M. & Sundgren B. (eds.). Advancing Your Business – People and Information Systems in Concert, Studentlitteratur, Lund.