Components and Services

Ivica Crnkovic[*], Vikram Jamwal[#]

[*]*Mälardalen University, Department of Computer Science and Electronics*
PO Box 883, SE-721 23 Västerås, Sweden
ivica.crnkovic@mdh.se
[#]*Kanwal Rekhi School of Information Technology,*
Kanwal Rekhi School of Information Technology, IIT Bombay, Mumbai. India
vikram@it.iitb.ac.in

1. Introduction

This paper shortly summarise the working session "Components and Services" being held at WICSA 2005 (Working International Conference on Software Engineering). Both components and services are strongly related to software architecture and with increased focus in research and practitioners' communities. Both component-based and service-oriented development use the same technologies and the same or similar principles in architecting of software systems. Still there have not been many interactions between the "components" and "services" communities. For this reason the events like this working session are very important to bring the communities together, to increase the mutual understanding, to exchange experience and to share solutions.

The intention of this working session was highlight certain problems, challenges and possibly solutions from these two areas.

The session was organised in the following way. First the selected papers have been shortly presented, followed by short discussion session. After this starting phase, the participants have in a discussion proposed different topics of interest for further discussion. The proposed topics have then been prioritised and the most interesting ones have been selected. The session continued with discussions of the selected topics, but also often referred to other topics. This paper gives a summary of the discussion, which have generated many new questions important for further research and practicesin these areas.

The rest of the paper is organised as follows. Section two summarises the presentations of the papres. Section three lists the topics and questions proposed by the participants. Section four gives a summary of the discussions of the selected topics. Finally section five concludes the paper.

2. Session Presentations

The working session starts with the presentations of the following papers.

- Rikard Land, Laurens, Stig Larsson, Ivica Crnkovic, Mälardalen University, Sweden and Eindhoven University of Technology: *Architectural Concerns When Selecting an In-House Integration Strategy Experiences from Industry*

  Abstract. The paper considers merging of existing software systems which have been developed in-house, for different purposes. Over time, the systems have been evolved to contain more functionality, until a point where there is some overlap in functionality and purpose. A new system combining the functionality of the existing systems would improve the situation both from an economical and maintenance point of view, and from the point of view of users, marketing and customers. To investigate this problem and challenges of in-house integration, we carried out a multiple case study, consisting of nine integration projects in six organizations from different domains. The present paper investigates issues of importance to an industrial and focuses on how to select a high-level integration strategy.

- Massimo Tivoli and David Garlan, University of L'Aquila and Carnegie Mellon University: *Adaptor Synthesis for Protocol-enhanced Component Based Architectures*

  Abstract. Correct assembly of software components is an important issue in Component Based Software Engineering. Composing a system from reusable components often introduces a set of problems related to communication and compatibility. In particular, one of the main

problems in component assembly is that components may have incompatible interaction behaviour. In this paper, we address this problem using an architecture-based approach that can detect integration mismatches, and semi-automatically synthesize a suitable adaptor, or glue code, to bridge them.

- Frank Lüders, Daniel Flemström andAnders Wall, Mälardalen University and ABB Corporate Research: *Software Component Services for Embedded Real-Time Systems*
  Abstract. Component models usually define basic standards for component naming, interfacing, binding, etc., in addition to standardized sets of run-time services oriented towards the application domains they target. Unlike for desktop applications and distributed information systems, there has been no widespread use of software component models in the domain of realtime and embedded systems. The purpose of this position paper is to lay the groundwork for a software component model for embedded real-time systems, using the basic concepts of COM as the starting point. Our vision is to make component-based software development an attractive option for embedded real-time systems by extending the basic model with services of general use for this application domain, much like COM+ extends COM with services for distributed information systems.

- M. Hepner and R. Gamble, University of Tulsa, *Establishing Connectors as Integration Services.*
  Summary. The basis of the presented research is to manipulate and organize connectors to determine what goes inside the service boundary in the form of an integration service and what does not. The goals of minimizing integration service functions, limiting redundancy, and requiring consistency lead to enhancing the workflow specification of the business.

The presentation session was concluded with discussions related to topics from the presentation.

3. Topics of interest

The following topics of interest have been crystallized in the discussion:

- What are the characteristics of SOA and Web-Services for Mobile and Distributed devices?
- How do we extend the principles (functional and non-functional aspects) of components to embedded systems?
- How to predict the consequences of 3-party component integration at the system level?
- What are the implications of connectors on the system properties?
- How do we generate an initial component configuration for Dynamic behaviour?
- Related question - How do we get a software architecture that guarantees certain properties of a dynamic behaviour?
- What are the similarities and differences in a Service Oriented and Component bases approaches?
- How is software evolution affected in component based systems?
- How are the binding mechanisms in software architecture and component implementations related?
- How to translate legacy systems to Component Based System?

4. Selected discussion topics

4.1 Components for Embedded systems

The main question was: How do we extend the principles (functional and non-functional aspects) of components to embedded systems?

This main question has been break down into several subquestions:

- How can we redefine aspects of embedded systems to fit those of components?
- How do we include components with known functional and non-functional properties in embedded systems?
- How does the component functional overhead influence performance and predictability of embedded systems?
- How can we ensure some properties with dynamically uploading components?
- How much dynamism can we allow while not loosing too much of predictability?
- How much dynamism (which types) can be implemented with acceptable runtime overhead?
- Could we use aspect oriented programming to separate functional and non-functional properties in case of embedded systems.

- Did we need to be able to ad and remove dynamically non-functional services in case of RTES.
- What impact does the variety of platforms for mobile devices have and how components can be used in those embedded systems?
- How to balance the flexibility of CBSE with the dependability in realtime embedded systems?
- How do we use component-based principles in combination with existing approach that target NF aspect?
- What dynamism to use that allows good enough predictability? One approach can be to start from something that is predictable and build more and more dynamism. This way it might be easier task.
- How do take knowledge from other systems and extend them to embedded component systems? General quality aspects can be applied to ES. Given the constraint on resources we need to do that quality evaluation / tradeoff analysis. Other aspect is reusability.
- What techniques in other dynamic system can be extended or applied here (e.g. in agent systems)? The openness of communication in e.g. say multi-agents, where the forces are combined a very unpredictable manner. Formalization of interaction protocols and goal-oriented approaches can provide the clue perhaps.

4.2 Relation between system and component properties

We want all dependencies to be known and we want to maintain black-box nature of component. Statically it is difficult. But we can observe dynamic behaviour. Component models specify for a component - what is required and what it delivers (provided). Perhaps prohibiting certain behaviour would be a good thing, but would it decrease dynamisms. One possible way it to list all interactions and then provide some sort of contracts. Component based representation might not be the right view for all kind of useful analysis. We also need to construct a view for analysis.

4.3 Discussion on Component Binding

In software architecture, it means connecting one component to another and has one non-ambiguous meaning here. Binding in known component systems is used more in sense of picking up the requisite component. The use of term 'binding' is not very clear and not very often used. We perhaps use the term 'attach' more often. In component models it is much more permissive than that is allowed in architectural models.

Can we have run-time architectures? What would be interesting is to look at extension of present component models like com to include the architectural elements. We may need several component models for different domains. Some commercial systems like COM have done very good lower level jobs like component life-cycles.
Different position: Basic concepts in component models are and should be same. Only extensions are required for different domains.

5. Acknowledgements

We would like to thank all participants being involved in the discussions. Their contribution in pinpointing the problems, analysing the challenges and proposing the solutions made the working session not only exciting but also a source of inspiration for further research in these closely related areas.