Chapter V

Negotiating Early Reuse of Components – A Model-Based Analysis

J. A. Sykes
Swinburne University of Technology, Australia

ABSTRACT

Unless existing components are considered during formulation of a system specification, the amount of component reuse that is possible may be limited. In order to increase the amount of reuse, it may be necessary to alter the functionality or performance of the system from that originally envisioned. Tension between stakeholders thus exists. Reuse of components also significantly changes the specification activity because it must now deal with component specifications as input models, which is not necessarily the case when reuse is not the goal. These issues are investigated using a modeling framework based on semiotic theory. The nature of modeling abstractions that could support the negotiation between stakeholders is also explored. Two scenarios are examined: one based on the idea of functional abstractions that can be composed and the other one using structural abstractions of the kind available in the UML as the basis of component composition. Even though at this stage, there are no good examples of functional abstractions that can be composed, it is concluded that functional abstractions are the best prospect for supporting collaboration and negotiation.

INTRODUCTION

Reuse of existing software components is becoming more attractive to system developers as a way of shortening development time, reducing costs, or producing software of higher quality or greater complexity.
However, as Wallnau et al. (2001) have noted, the decision to reuse components constrains the choices open to system designers and alters the nature of the design process. Instead of being able to specify subsystems in a relatively free way, designers must choose from the limited, although possibly large, number of ways that those components could be combined. Furthermore, in order to make such choices, designers must be able to rationalize the ways that different combinations of components would behave. This differs from the top-down, problem-solving approach to design that is possible when subsystems can be specified in a less constrained way.

The possibility that decisions to reuse components could be made quite early in the development process has been discussed previously. Herzum and Sims (2000) envision maturity of the component industry as meaning that components of large granularity will be reusable at the enterprise level by developers with limited technical skills. These components will be supported by suitable software tools to manage deployment, configuration, etc. How such components should be represented, in order to enable their composition by domain experts rather than software experts, then becomes an issue. This point has been explored by Sykes and Gupta (2001), who discussed the need for functional abstractions that support the composition of components during requirements specification. The authors also noted the lack of suitable abstractions at present in the Unified Modeling Language (UML).

A related issue, which is addressed in this chapter, is tension. Taking component behavior into account during the formulation and specification of system requirements introduces a tension between the desire to specify a system that precisely meets some user’s requirement and the possibility of implementing the system sooner or more inexpensively if some component that is already available is used, even if it does not exactly meet the requirement.

The approach adopted is to regard the formulation of the specification for a system that will incorporate existing components as a negotiation between stakeholders, who will make use of models of the system and the candidate components through their collaboration.

The modeling aspects are discussed using the modeling framework described in the next section. The framework, which is based on semiotic theory, is an extension of that used in Sykes and Gupta (2001). Here it has been extended with concepts about interactive systems in order to deal with the nature of components as subsystems.

The modeling framework is used to examine the role of specifications in a development process that reuses existing components. Terminology from the Unified System Development Process (USDP) has been used for this part of the work, mainly because the USDP is well-documented (Jacobson et al., 1999) and is now quite widely known.

Other characteristics of the USDP that make it an appropriate context for the work, are first, its architectural focus on subsystems, which makes it compatible with the use of components. Second is the fact that it is model-based, which makes it possible to use the modeling framework for studying it.

The process through which the use of existing components can influence the formulation of a system specification is investigated using the modeling framework. This extends an earlier discussion in Sykes and Gupta (2001) and also puts it in a more general context (i.e., in terms of the participants involved in the collaboration).