Chapter V

UML Modeling Support for Early Reuse Decisions in Component-Based Development

J. A. Sykes
Swinburne University of Technology, Melbourne, Australia

P. Gupta
Central Queensland University, Melbourne, Australia

ABSTRACT
Component-based development is the software industry’s latest answer to some long-standing problems in software development. Its aim is to make actual reuse of existing software units (components) a widespread reality. We argue that significant reuse decisions can occur in the early stages of system development. Using a theory of the modeling process, we show that choice of abstractions and notations is critical. We investigate the kinds of models that would support early reuse decisions. We show that easily composable, business-oriented abstractions about software behaviour and a functional mental model are necessary. Evaluation of the UML in these terms emphasizes its bias towards structural mental models and abstractions derived from the software domain.

INTRODUCTION
Component-based development (CBD) is the software industry’s latest answer to some long-standing problems in software development (Allen & Frost, 1998; Szyperski, 1998). Building on lessons learned from earlier software-engineering innovations, CBD focuses on the goal of actual reuse of software (rather than reusability, which is simply the potential for reuse). This is seen as the key to a number of desirable economic outcomes, such as rapid development, clarification of developer roles and amortization of software development costs.

If components are thought of simply as units of software, it might be concluded that reuse is mainly a matter for the design and construction phases of development, but we argue
that significant reuse decisions can, and must, occur earlier in the development process. This means that modelling has an important role to play in enabling developers to evaluate different reuse decisions prior to actual implementation.

Software development is a human-activity system that involves many people with different backgrounds, interests and skills. Effective communication between the participants is essential for success. Increasingly, software development is seen as a model-based activity, which means that various kinds of models are used as the chief formal technique for communication during the development process. As a result, the process of modelling has now itself become an important subject for research.

Some of the participants in the early stages of development of software for business and information systems are not software experts. The ways that they think about business systems and software systems can differ in significant respects from those preferred by software designers and programmers. Therefore modelling languages that are based predominantly on concepts appropriate to the software domain, as is the case with the Unified Modelling Language (UML), present obstacles to effective communication.

In this chapter we investigate the modelling process and the kinds of thinking it involves. We do so using a theoretical framework for models and modelling based on the semiotic (meaning) triangle and some ideas about mental models from the field of human-computer interaction.

We examine the interactions between the typical developer roles of Client and Analyst in an assumed component-based modelling context. The framework enables us to identify the characteristics of abstractions and notations suitable for supporting the modelling process. Several UML diagrams that have been put forward as suitable for CBD are evaluated against these characteristics.

**CHARACTERIZING MODELS AND MODELLING**

The process of modelling is a purposeful, subjective human activity. It involves perception (of the thing to be modelled), conceptualization (thinking about the thing to be modelled in a particular way) and action (actual creation of the model).

One way to study the modelling process is via the semiotic triangle. Based on this, a so-called modelling triangle was used by Brinkkemper (1990) to study formalization of information system modelling. More recently, it was used to investigate the conceptual modelling process (Gupta & Sykes, in press). This approach has now been extended with ideas about mental models and the system development notion of modelling purpose, to create a general theoretical framework for analyzing information systems modelling and development processes.

**The Semiotic Triangle**

The semiotic triangle (also called the meaning triangle) deals with the process by which a human observer is able to claim that some sign or symbol

![Semiotic Triangle Diagram]

symbolizes

stands for

refers to

concept

referent

sign

Figure 1. Meaning triangle
Modeling and Programming by Commitment Rules in Agent Factory
Rem Collier and Gregory M.P. O’Hare (2009). Handbook of Research on Emerging Rule-Based Languages and Technologies: Open Solutions and Approaches (pp. 393-421).
www.igi-global.com/chapter/modeling-programming-commitment-rules-agent/35868?camid=4v1a