Title	
Author(s)	,
Citation	
Issue Date	2002-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/920
Rights	
Description	Supervisor: , ,

Japan Advanced Institute of Science and Technology

Lightweight Formal Methods for Component-based Software Development

Michihiro Matsumoto

School of Information Science,
Japan Advanced Institute of Science and Technology

January 10, 2002

Abstract

In the thesis, we discuss (1) lightweight formal methods for component-based software called LFMB and LFME, (2) the component-based software development using LFMB or LFME called CBDL, and (3) the support tools of CBDL. Recently, component-based software development has gained in popularity. The reason for the popularity is that it can increase software productivity. To increase software productivity, components must be reused. The obstacles of component reuse are (a) a lack of a consensus about component usage between component developers and software developers, i.e. component users and (b) an architectural mismatch. We developed CBDL to eliminate the obstacles. CBDL is based on the Catalysis approach. The former obstacle is eliminated by specifying business models and component specifications by using UML diagrams with OCL descriptions and verifying consistency in the UML diagrams. The former technique is an idea of the Catalysis approach. The latter technique, i.e. LFMB and LFME, is a contribution of the thesis. Because LFMB and LFME include automated verification methods of the consistency, component developers and software developers who are not familiar with the verification can get the benefit of the verification by using the support tools of CBDL. The latter obstacle is eliminated by selecting tree architecture that we developed. We can regard the UML diagrams as specifications specified by a language for programming-in-the-large. So, in CBDL, moreover, we generate component-based software from the UML diagrams by combining the connectors specified in the UML diagrams and components of a component library. Note that the above consistency verification guarantees the correctness of the connectors. Because the support tools of CBDL are designed as client-server type software, component developers and software developers can access to the tools through Internet at any place at any time. To summarize, by using the support tools of CBDL, we can increase component reuse and correctness of component-based software.

Key Words: **Lightweight formal methods, component-based software development, algebraic specification, term rewriting system, UML**