DB
1712
2000

A Formal Approach to Software Composition in Component Based Software Development

筑波大学審査学位論文　(博士)

2000

新川芳行

筑波大学大学院

経営・政策科学研究科　企業科学専攻

寄	贈
新川芳行 氏	平成 年 月 日

01301699

A Formal Approach to Software Composition in Component Based Software Development

PhD. Thesis

2000

Yoshiyuki Shinkawa

Graduate School of
Management Science and Public Policy Studies
Major in Management Science

The University of Tsukuba

Abstract

Complicated and sophisticated requirements to today's huge and complex software systems cause the following serious problems.

1. Difficulties in developing software systems reflecting many requirements which are mutually interrelated. (There could be various *gaps* between the requirements and the implementations.)

2. Difficulties in maintaining large-scale systems caused by additional and ad-hoc changes.

Those difficulties lower the productivity of software development, especially in industrial applications or *enterprise back-office* applications which are usually based on large-scale and vague requirements.

Two emerging technologies in software engineering for this decade have been tried to cope with such problems as the difficulties in implementation and maintenance of software systems. They are enterprise modeling and Component Based Software Development (CBSD). Enterprise modeling mainly deals with requirements, and helps us to build accurate enterprise models which would be a part of the requirements definition for software systems. On the other hand, CBSD mainly focuses on software systems, and provides us with convenient ways to develop software systems using reusable components, with the assistance of standardized component specifications, inter-component communications, and system assembly methods.

Those technologies provide us with various practical methods to construct software systems based on requirements models by reusing components, and contribute to productivity improvement of software system development. However, due to insufficient formalism in those technologies, we often have to make considerable efforts for applying them to real applications.

This thesis proposes a formal approach to construct software systems in CBSD environments, which covers from requirements modeling to software system evaluation. The following four kinds of formal methods are used in this thesis, that is:

1. Rough Set Theory (RST)
   RST is a theory for dealing with knowledge and concepts based on equivalence relations in standard set theory.

2. Colored Petri Nets (CPN)
   CPN are one of the enhancements of ordinary Petri nets. In addition to the modeling capability of ordinary Petri nets, CPN provide us with a way to express functionality or semantics of the models.

3. Σ algebra
   Σ algebra is an interpretation for the signature $\Sigma = (S, \Omega)$, where S is a set of sorts , and Ω is an $S \times S^*$ sorted set of operation names. It specifies computational properties of systems, with the capability of evaluating equivalency between two different Σ algebras.

4. A Calculus of Communicating Systems (CCS)
   CCS is one of *process algebras* which express the behavioral aspects of transition systems in algebraic forms. CCS has excellent capability of evaluating behavioral equivalency between two systems.

The proposed approach in this thesis is organized by three methodologies.

The first is a domain modeling methodology for accurate requirements analysis and definition, from vague domain knowledge. RST is used to coordinate pieces of such vague knowledge that is provided by various domain-experts, and CPN are used for making requirements expression accurate, from functional and behavioral viewpoints.

The second is a methodology for component mining and software composition, which is used to compose adaptable software systems to the requirements. The component mining is based on algebraic equivalency between requirements and components. Σ algebra is used for evaluating the algebraic equivalency. On the other hand, the software composition is based on decision tables extracted from the CPN models constructed by the first methodology. RST is used in both the mining and the composition, mainly for optimization.

The third is a verification methodology, which evaluates adaptability of the composed software systems to the requirements obtained by the first and second methodologies. Two classes of adaptability are defined in the methodology, that is, functional adaptability and behavioral adaptability. Σ algebra is used for the functional adaptability verification in the similar way to the first methodology. On the other hand, CCS is used for behavioral adaptability evaluation, which expresses the dynamic aspects of the requirements and the software systems.

This approach is formal enough to compose the adaptable software systems uniquely by available pieces of knowledge in a problem domain and given components. Therefore it will contribute to software reuse, which is one of the major purposes of CBSD. In addition, we can apply each methodology of this approach independently to a particular stage in software development projects.

Acknowledgements

I wish to thank my supervisors, Prof. Masao J. Matsumoto (chair), Prof. Takeo Terano, Assoc. Prof. Yasushi Kuno, and ,Assoc. Prof. Yasufumi Saruwatari for their encouragement and guidance throughout this study. I also would like to thank the members of SWIM TG(SoftWare Interprise Modeling Technical Group) in IEICE (The Institute of Electronics, Information and Communication Engineers) for suggestion and advice on this study. I received many practical hints and tips from the members of Systems Laboratory and other organizations in IBM Japan Ltd. Finally, I would like to appreciate the financial support from IBM Japan, Ltd.

Contents