Research

Table of contents

1 Research Topics........................................................................................................... 2
2 Current Projects............................................................................................................ 2
2.1 Formal Specification and Verification on Software................................................ 3
2.2 UML/OCL Based Formal Developing Method for Distributed Real-time Systems guarantying timeliness QoS............................................................... 4
2.3 Model Driven Development of Software............................................................... 7
3 Software....................................................................................................................... 7
1. Research Topics

My main research interests are:
- Formal Specification and Verification on Software
- Model Checking and Formal Verification of Timeliness QoS in Multimedia Systems using Timed Automata
- Design Techniques for Object-Oriented Programming in UML/OCL
- Module Based Programming in Functional Programming Languages
- High Reliable Distributed Systems

Other topics also I'm interested in are:
- Application of FDT to Software Engineering
- Assertion Based Design for Software and Hardware
- TRS and its application
- Real-time Systems

2. Current Projects
2.1. Formal Specification and Verification on Software

One of essential approaches to ensure high-quality in software is to establish formal description and verification techniques for software specification. Our research group has studied on formal specification and formal verification techniques. Our studies have an originality in exploiting a decision procedure for Presburger sentences. A Presburger sentences is closed form of a logical expression consisting of inequalities of integers allowing addition and subtract operators.

\#x\#y\{(x\#0 \# y> 0) \# (y=0 \# x\#) \# y>0\}

In general, a verification problem of correctness of a program against its specification can be reduced into a decision problem on a kind of logical expression. We interprets the expression as a Presburger sentence by regarding a term not belonging to Integer type nor Boolean type an interpretation free term. We have developed a verification support system for a functional programming language SML in OCaml. It also exploits several useful results in TRS as well as a fast decision routine for a general class of Presburger Sentences. Through a case-study on a book management system, we have found that correctness proof can be performed in reasonable time.

keywords: Formal Specification, Verification, Presburger Sentences, ML, TRS
2.2. UML/OCL Based Formal Developing Method for Distributed Real-time Systems guarantying timeliness QoS
Real-time Systems (image)

Formal Specification of Quality of Service (QoS) properties is an important part of the component based design of distributed systems. To produce a correct specification, it is crucial to study the effect of QoS statements of the components on the overall behavior of the system. We have studied a method of verifying timeliness QoS properties such as Jitter, Throughput and Latency. Suppose that the functional behavior of the system is modeled via a network of timed automata. Conformance Verification is performed on UPPAAL using Test Automaton which is derived from timeliness QoS specification and NTA which models behavior of a component. We reduce possibility of state explosion by introducing another step where testing of required timeliness QoS is satisfied under every provided timeliness QoS of each component and network configuration. We formalized the testing problem and present a method reducing the problem into an infeasibility problem on linear inequalities on reals. It is also important that end-user can easily use our method, therefore, we use UML/OCL to specify the abstract level of specification for the whole system. We also give automatic derivation program that creates a Java program that behaves temporal actions specified in UML state-charts extended with clocks. Timeliness QoS is checked via the model checker **UPPAAL**.

keywords: UML/OCL, Component Based Design, timeliness QoS, timed automaton, Java Program, UPPAAL
Time-Action-Lock Checker

2.3. Model Driven Development of Software

We also study how to apply formal method into real software developing environments such as MDD (Model Driven Development). The key issue is how to provide useful methods and support systems based FDT. One of such approaches is to develop a UML based support system which also provides formal verification engines. We consider effective use of JML and SPIN like verifier. It is an important and challenging aim that the provided techniques should be automatic verification as well as how to feed-back errors and failures detected in the system.

keywords: UML/OCL, JML, MDA, MDD, SPIN

3. Software

Timed Action Lock Checker (A checker for detecting Timed-Action-Lock in a network of Timed Automata) (2003-)