Research Plan

Farhang Nemati
Mälardalen University
Farhang.nemati@mdh.se
February, 2007

Assignment 3, Research Planning Course
# Contents

1. Introduction ............................................................................................................. 3

2. Project ....................................................................................................................... 4
   2.1 Extract .................................................................................................................. 4

3. Method ....................................................................................................................... 4
   3.1 Hybrid Model Extraction ..................................................................................... 5
      3.1.1 Static Analysis ............................................................................................. 5
      3.1.2 Dynamic Analysis ....................................................................................... 5
   3.2 Using Control Flow Statistics in HME ................................................................. 6
      3.2.1 Automatic Control Flow Instrumentation .................................................... 6

4. Literature ................................................................................................................... 6

5. Leading Groups ........................................................................................................ 7

6. Conferences ............................................................................................................. 7

7. Future Work ............................................................................................................. 9

8. Research Issues ...................................................................................................... 9
   8.1 Problem .............................................................................................................. 9
   8.2 Basic Conditions ............................................................................................... 10

9. Concrete Research Questions .................................................................................. 10

10. Hypothesis ............................................................................................................. 10

11. Concrete Expected Results ..................................................................................... 10

12. Activities, Milestones and Cooperation ................................................................. 11
   12.1 Planned Publications ....................................................................................... 11
   12.2 Conference Trips ............................................................................................ 12
   12.3 Time Plan ......................................................................................................... 12
   12.4 Cooperation ..................................................................................................... 13
   12.5 Courses ........................................................................................................... 13

13. References ............................................................................................................. 14
1. Introduction

Managing the evolution of software systems is important in order to preserve the large investments associated with the software. Complex embedded software systems, such as industrial control systems or automotive systems, often consist of several million lines of code and are maintained over many years, sometimes decades, during which the software is exposed for changes continuously. Due to the many changes made over the years, the software systems become larger and more complex. As a consequence, the perspicuity of the system decreases, i.e., it becomes increasingly harder to predict how a change, e.g. a new feature, will impact the behavior of the system. Thereby, it becomes harder to maintain the software, which is reflected in development cost and time-to-market. For complex embedded software systems, which typically contain many tasks (processes) with timing requirements, predicting the impact of a change is especially difficult. Non-functional aspects of the change, e.g. increased execution times, may result in errors, for instance time-outs. In worst case, such problems occur only in very specific situations. Errors of this type are very costly, since they are hard to detect and reproduce.

My research area is reengineering of the complex embedded real-time legacy systems. This area concerns how to analyze, model and reuse existing systems and their parts, with the purpose to maintain them or integrate them in future systems together with newly designed parts. The research will focus on development of appropriate modeling frameworks, and will include close collaboration with industrial partners (ABB, Bombardier etc.). The systems that we are working on are complex real-time systems. These systems often consist of several million lines of code and reengineering them is done for the following purposes:

- Maintenance of systems
- Evolution of systems
- Improvement of performance

The reengineering process as illustrated in Figure 1, consists of following parts:

- Reverse engineering
- Restructuring
- Forward engineering

![Figure 1: Reengineering Process](image-url)
As current research I am working on reverse engineering step, and in that topic we are developing methods and tools to extract models from existing systems.

2. Project

2.1. Extract

The aim in this project is to improve the maintenance and evolution of complex real-time systems by developing methods and tools to extract models of the systems. Our aim in this project is to provide the automated model extraction process as far as possible. The extracted models describe the behavior of the systems regarding to the timing attributes and resource usage. The models are used to impact analysis of the changes in the systems, in this way potential problems can be discovered in the early stages and before applying the changes in the real systems.

3. Method

To extract accurate models we try to use both static and dynamic analysis. The static analysis includes source code modeling (control flow modeling, condition modeling etc). In the dynamic modeling we develop methods and tools to record the behavior of the systems during run time and analyze the results offline. Currently we work on hybrid model extraction; in that we intend to combine the results of static and dynamic analyses to obtain a more accurate and reliable model of the complex real-time legacy systems.

![Figure 2: Control Flow Recording in HME](image-url)
3.1. Hybrid Model Extraction

The hybrid model extraction (HME) relies on static and dynamic analysis. In the static analysis the modeling is done based on the source code in which the relevant part of the source code is analyzed and call-graph and relevant state variables are extracted. The skeleton of the model is obtained by using the call-graph and state variables. The dynamic analysis part of HME uses recordings of behavior of the system to obtain the timing information and also the control flow information at run time. We use dynamic analysis for two purposes; extracting timing attributes and control flow statistics. For extracting control flow statistics we have developed automatic instrumentation in which the source code is parsed and obtained Abstract Syntax Trees (AST) are transformed to contain instrumented data (probes). Both static and dynamic analyses are supported by tools that we have been developing. Our aim is to automat the whole model extraction as far it is possible. Currently we are working on control flow recording to use it in the modeling of complex embedded systems; Figure 2 depicts the control flow recording that is a part of HME.

3.1.1 Static Analysis

Terminology:
- **Target function** - a user-specified function that explicitly impacts the task scheduling or the utilization of logical resources.
- **Model variable** - a variable that is selected for explicit representation in the model.
- **Model-relevant function** is a function that contains at least one model event.
- **A model event** is a program statement matching one of three descriptions:
  - A call to a target function,
  - A call to a model-relevant function, or
  - An assignment of a model variable.

As illustrated in Figure 2 the source code modeling process of HME consists of two activities. The call-graph search produces a list of model-relevant functions, based on sets of target functions and model variables. Initially, the set of model variables is empty. The model-relevant functions are analyzed in the next step, conditions modeling, in which model variables are identified and modeled. The identification is performed by manual analysis of the control-flow conditions that directly impact the execution of the model events. The modeling process terminates when no new model variables are found during the conditions modeling, as no new model-relevant functions will be discovered in the next call-graph search.

For the call-graph search we use a commercial tool, *Understand for C++* from Scientific Toolworks [16]. This tool generates a database describing all functions, variables and data types in the analyzed source code, and their relations. The call-graph search is implemented using a Perl script that interfaces the database, using a Perl API provided by the tool. The output is a list of model-relevant functions.

3.1.2 Dynamic Analysis

The model resulted from the static modeling describes the behavior from a functional view, but does not contain timing information, such as execution times and task inter-arrival times. This information is instead obtained in a later activity, using dynamic analysis. This use of
dynamic analysis is not to be confused with the control-flow recording proposed in this paper. The recording of timing information requires that the system can provide a call-back connected to the context-switch events. This is however possible with commercial, commonly used real-time operating systems such as VxWorks [14] and OSE [15]. This callback is used to record each context-switch together with an accurate time stamp. This recording needs to be permanently added to the system in order to avoid a probe effect.

3.2. Using Control Flow Statistics in HME

During the conditions modeling activity of HME, the user has to decide how to model each model-relevant condition, i.e. what abstractions to make. How these abstractions are made is the core issue when using HME. To facilitate the decisions concerning these abstractions, we use control-flow recording in order to obtain statistics regarding the runtime behavior of model-relevant control-flow statements.

By recording control-flow statistics, the probabilities can be obtained automatically, when probabilistic modeling is desired. The recording is achieved by adding code instrumentation (probes) to the system, which records the execution of selected control-flow statements, i.e. the model-relevant control-flow statements. The user needs to build the system using the instrumented code and execute the reference scenarios. Based on the resulting recordings, the modeling tool presents statistics for each control flow statement.

3.2.1 Automatic Control Flow Instrumentation

Control flow recording requires fairly much instrumentation of the source code. In large industrial systems, doing the instrumentation manually is not realistic, it would be too time-consuming and error prone. Therefore, we try to use automatic code instrumentation for this purpose. We are implementing this in our modeling tool, which already contains a C parser generated by ANTLR [17]. This C parser has been extended to instrument all control-flow statements containing model events.

4. Literature

Not much research has been done in the legacy systems area especially within the embedded systems; although in the industry it is extremely needed..

The following are some basic books in the reengineering area:


In the Reengineering area the following papers are more like seminal papers:

5. Leading Groups

To my knowledge there are few groups who are doing research in this area. The similarity between our research and there is that they also model the legacy systems by combining static and dynamic analyses:

- **E-CARES:**
  E-CARES (Ericsson Communication ARchitecture for Embedded Systems) research project is a cooperation between Ericsson Eurolab Deutschland GmbH (EED) and Department of Computer Science 3, RWTH Aachen University.

  The aim of the project is to provide a flexible and interactive reengineering environment by using static and dynamic analyses. In their research they focus on Ericsson’s Mobile-service Switching Center software (AXE10) which contains more than 10 million lines of code and is written in Ericsson’s in-house programming language PLEX (Programming Language for EXchanges).

- **STAR:**
  The Software Technologies Applied Research (STAR) laboratory is a practice-oriented software engineering research laboratory, located in the Department of Electrical and Computer Engineering, University of Waterloo, Canada.

  Their research activities reflect two complementary themes: a scientific approach to understand complex software systems, and an engineering focus on the ability to control, modify, and design such software systems.

  They have done research in the modeling of legacy systems based on static and dynamic analyses, but the context of their modeling is not real-time systems.

6. Conferences

- **European Conference on Software Maintenance and Reengineering (CSMR)**

  *Software Evolution in Complex Software Intensive Systems*

  “The European Conference on Software Maintenance and Reengineering (CSMR), the premier European conference on the theory and practice of maintenance, reengineering and evolution of software systems, promotes discussion and interaction among researchers and practitioners about the development of maintainable systems, and the evolution, migration and reengineering of existing ones.”, CSMR website.

- **IEEE International Conference on Program Comprehension (ICPC)**
“ICPC provides an opportunity for researchers and industry practitioners to present and discuss both the state-of-the-art and the state-of-the-practice in the general area of program comprehension.”, ICPC website.

- **International Conference on Software Maintenance (ICSM)**

  “Since its start in 1983, ICSM has grown and developed into an international forum for software maintenance researchers and practitioners to examine key issues facing the software maintenance community. Participants from academia, government, and industry share ideas and experiences solving critical software maintenance problems.”, ICSM website.

- **International Workshop on Program Comprehension (PCODA)**

  “The goals of this workshop are to catalogue existing techniques in the field of program comprehension that use dynamic analysis.”, PCODA website.

- **Working Conference on Reverse Engineering (WCRE)**

  “The Working Conference on Reverse Engineering (WCRE) is the premier research conference on the theory and practice of recovering information from existing software and systems. WCRE explores innovative methods of extracting the many kinds of information that can be recovered from software, software engineering documents, and systems artifacts, and examines innovative ways of using this information in system renovation and program understanding.”, WCRE website.

- **IEEE Real-Time and Embedded Technology and Application Symposium (RTAS)**

  “The scope of RTAS 2007 consists of the traditional core area of real-time and embedded systems infrastructure and theory, as well as three additional areas of special emphasis: embedded applications; development, verification, and debug tools for real-time and embedded systems; and embedded systems hardware/software interaction/co-design.”, RTAS website.

- **IEEE Real-Time System Symposium (RTSS)**

  “RTSS provides a forum for the presentation of high-quality, original research covering all aspects of real-time systems design, analysis, implementation, evaluation, and case-studies.”, RTSS website.

- **Euromicro Conference on Real-Time Systems (ECRTS)**
“The conference has very strong roots within the Real-Time Laboratories and research groups in both Academia and Industry throughout Europe and across the world.”, ECRTS website.

- **IEEE International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA)**

  “The conference has the following goals: to investigate advances in embedded and real-time systems and ubiquitous computing applications; to promote interaction among the areas of embedded computing, real-time computing and ubiquitous computing; to evaluate the maturity and directions of embedded and real-time system and ubiquitous computing technology.”, RTCSA website.

### 7. Future Work

Industrial evaluating the methods and tools that we have been developing with our industrial partners and investigate the results with them is of my future works. In order to make the extracted models of legacy embedded real time systems more confident, we have to validate the models, therefore my future research also will be in the model validation area. We will study the current methods and possibly improve them or develop new methods to validate models. The confidence in the extracted models can be improved as the adaption between the models and the real systems increases.

### 8. Research Issues

#### 8.1. Problem

Maintenance of complex legacy systems becomes harder and more complex over time by adding new functionalities. The challenge in this area is how to recover the design of real-time legacy systems by extracting information from implementation to higher level of abstraction and to organize the extracted information into models. The modeling implies: Visualization of function call (call graph), source code analysis and dynamic analysis. For the static analysis we extract the skeleton of the related parts of the source code but the problem is that modeling the system only on based on the source code can’t be accurate, especially non functional attributes can’t be obtained from code, therefore dynamic analysis should be done to extract information during run time. For some aspects of dynamic analysis the source code should be instrumented (probes), the problem in this case is that the probes may affect the functionality of the system by affecting the timing behavior of the system, therefore evaluations should be done to measure the affect of the probes and some times it is necessary to let the probes remain in the system.

Another problem to be considered is to adapt the extracted model with the real system (model validation). In [13] an automatic model validation has been proposed. The solution should be evaluated and if possible improved. The challenge in this area is how accurate a validation can be and what parameters and what tolerance limits should be taken into mind when validating a model and the limitations.
**8.2. Basic Conditions**

The HME modeling process assumes a software system based on a real time operating systems (RTOS) with the possibility to add monitoring and recording in the task scheduling, i.e. the time and tasks of each context switch. This possibility is offered by several commercial operating systems such as VxWorks [14] and OSE [15]. Our approach for performance modeling targets complex industrial software systems, e.g. industrial robot controllers, telecom systems and automotive control systems.

**9. Concrete Research Questions**

- How efficiently fight against complexity of evolution of complex legacy systems?
- How to extract efficient models of a complex real-time legacy system?
- How far the extraction of models can be automated?
- How to extend the modeling of legacy systems to different contexts?
- What methods are there for model validation to support the modeling of complex legacy systems? If any, can they be improved? How? Can new methods be developed? How far can the model validation be automated?
- How to transform traditional software such as modular software to new technologies like component based software engineering (CBSE)?
- How to use reengineering techniques in emerging technologies like CBSE?

**10. Hypothesis**

Our hypothesis:
- “Reliability of evolution of a real-time legacy system can be improved by model based impact analysis of changes”
- “Accuracy of extracted models of a legacy system can be improved by automatically combining information of static and dynamic analyses as far as possible”
- “An automated model validation improves the efficiency of modeling of a complex legacy system”

**11. Concrete Expected Results**

The followings are some expected results:

- Reliable model extraction methods
• Classification of model validation methods and develop new methods

• Improving existing tools and developing new tools to support the model extraction process also model validation

• Case studies to evaluate methods and tools

12. Activities, Milestones and Cooperation

12.1. Planned Publications

“Using Control Flow Recording in Modeling of Complex Embedded Systems”

Abstract:

We aim to enable model based impact analysis with respect to performance properties in maintenance of complex embedded systems. By using this analysis, potential performance problems of a proposed change can be identified at an early stage, which makes the maintenance more efficient and predictable.

This paper extends a previously presented approach for tool-assisted performance modeling with the presentation of recorded control flow statistics. This improves the accuracy of the resulting model as program comprehension is facilitated during the manual modeling of the process.

The paper presents the extended modeling process, an implementation of the extension in our modeling tool and an evaluation on a complex industrial software system, a control system for industrial robot developed by ABB. The evaluation investigates the usability of the control-flow statistics and the performance penalty of the necessary recording.

We will submit it on March 31st to IEEE International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA 2007) in South Korea.

Or we will submit it on June 13th to Working Conference on Reverse Engineering (WCRE 2007) in Canada.

Results:
The control-flow recording facilitates the manual modeling of conditions but also introduces new manual activities in the modeling process; building the instrumented system and executing the reference scenarios. In the modeling process proposed, these activities are necessary in each iteration of the process, which is far from optimal as these activities are rather time consuming.

We have therefore outlined an improved process for Hybrid Model Extraction, better suited for the use of control flow recording. As a bonus, a higher degree of automation is obtained. This improved process is however not yet implemented.

12.2. Conference Trips

- If our paper is accepted by RTCSA, we will have a trip to South Korea, August 21-24 2007.
- If it is accepted by WCRE, we will go to Canada, October 29-31 2007.
- The conference trip for my second paper is not yet decided

12.3. Time Plan

- **2006 August-December**
  - Study the research area and investigate the work that has been done in the legacy system area of the real-time context.
  - Develop a tool to automatically instrument the source code for control flow recording. This tool is a part of analysis tool (MASS) developed by Johan kraft at Mälardalen University to assist modeling process.
  - Write paper.
  - Research Methodology Course
  - PhD School Course
  - Progress Course
  - Artes Summer School in Stockholm
  - Progress trip to Santander, Spain

- **2007 April-May**
  Empirical study on the methods proposed in our paper.

- **2007 April-June**
  Investigate and Study existing model validation techniques

- **2007 August-October**
  Prepare a paper on model validation.

- **2007 November-December**
  Do more work on model validation.

- **2007 spring**
Look for possibilities to cooperate with other universities. Investigate possible cooperation with other areas in our research center, MRTC, and may work on some joined papers, e.g. reengineering of component based software in real time area or in common.

- **2008 August**
  Prepare for licentiate thesis.

- **2009**
  To be planned.

### 12.4. Cooperation

The research is done within Progress project at Mälardalen Real-Time Research Center (MRTC) at Mälardalen University. Professor Christer Norström is the leader of this research area.

For our research we have close cooperation with our industrial partner, ABB. The research is mainly done on ABB’s robotic control system. We are also establishing cooperation with Bombardier Transportation and hopefully we will be able to evaluate our methods and tools on their systems. We are also hoping to establish cooperation with Ericsson; in this case we will get the opportunity to extend our research context to the telecommunication systems.

### 12.5. Courses

**Taken:**

- Research Methodology in Computer Science
- PhD School
- Progress
- Legacy Systems
- Research Planning

**Planned to Take:**

- ARTIST2 Graduate Course on Embedded Control Systems (in Lund university)
- Real Time Systems, Advanced Course
- Advanced Topics in Component Based Software Engineering
- Design of Embedded Real Time Systems
- Formal Methods in Real-Time Systems
- Distributed Software Development
- Software Architecture for Industrial Systems
- Program Analysis
13. References


