RETR: Reverse Engineering To Requirements

Yijun Yu, John Mylopoulos,
Yiqiao Wang, Sotirios Liaskos,
Alexei Lapouchnian,
University of Toronto
{yijun,jm,yw,liaskos,alexei}
@cs.toronto.edu

Ying Zou
Queen's University
ying.zou@queensu.ca

Marin Litiou
IBM Canada
marin@ca.ibm.com

Julio C.S.P. Leite
PUC-Rio
julio@inf.puc-rio.br

Abstract

Reverse engineering aims at extracting many kinds of information from existing software and using this information for system renovation and program understanding. The goal of this full day WCRE'05 workshop is to identify methods and techniques for Reverse Engineering from software to Requirements (RETR).

1 Introduction

Reverse engineering aims at *extracting many kinds of information from existing software, such as requirements specifications, design documents,* and *system artifacts, and using this information in system renovation and program understanding* [1].

Existing *reverse engineering* methods focus on recovering architecture and design of software products that are often represented in standard formats such as UML, GXL or ADL. However, few methods *recover requirements* such as goals of the various stakeholders, non-functional requirements, early aspects, variability tradeoffs and dynamic/emmergent behavior of autonomic systems.

Therefore a forum is needed to discuss the issues related to recovering requirements from the software. It can enable the reverse engineered software systems to continuously adapt to the evolving functional requirements, and to be reengineered to meet the non-functional requirements.

2 Topics of interest

Topics of interest include, but are not limited to:

```
+ Early Requirements
  - Aspects
  - Goals
  - Scenarios
  - Use cases
  - Variabilities
  - Viewpoints
+ Non-functional Requirements and Qualities
  - Understandability and Maintainability
  - Performance, Usability and Reliability
  - Security and Privacy
  - Interoperability
+ Evolution of Software Requirements
  - Mining and Clustering Software Repositories
  - Reconciliation of Requirements and Implementation
  - Adaptive Software for Autonomic Computing
  - Emergent Behavior in Software Integrations
+ Traceability
  - Establishing Traceability
  - Maintaining Traceability
+ Verification and Validation
  - Design Rationale and Impact Analysis
  - Requirements Testing
  - Empirical Case Studies
```

3 Objectives of the workshop

The goal of this full day WCRE'05 workshop is to work on methods and techniques for Reverse Engineering from software to Requirements (RETR). The objective of the workshop is to sketch the state-of-the-art of the RETR practice and to identify current trends and fields of interest, possible paths of collaboration and points of future research directions.

References

[1] E. J. Chikofsky and J. H. Cross II. Reverse engineering and design recovery: A taxonomy. *IEEE Software*, 7(1):13–17, 1990.

Proceedings of the 12th Working Conference on Reverse Engineering (WCRE'05)
1095-1350/05 $20.00 © 2005 **IEEE**