Reengineering of Legacy Systems
“From Legacy Systems to Layered Architectures and Services”
Leg2Net

Leicester
June, 2008

Carlos Matos
Dep. of Computer Science
University of Leicester
cmm22@mcs.le.ac.uk
Outline

• Introduction
• Reengineering towards SOA
• Methodology
• Future work
Introduction

• The adoption of Service-Oriented Architectures (SOAs) has been growing over the last years
• According to a Gartner study, around 65% of SOA projects are implemented via wrapping or reengineering of existing applications
• A well defined methodology is necessary to support a systematic way of addressing migration of existing systems to SOA
Reengineering towards SOA

• Some of the properties that derive from service-orientation principles have a direct impact on legacy systems:
  1. The separation of business logic from presentation logic (technological dimension)
  2. The loosely coupled relationship between services (functional dimension)
  3. The coarse-grained nature of services (functional dimension)

• In our approach we address these separately in two dimensions: technological and functional
General Methodology

1. Code Annotation
2. Reverse Engineering
3. Redesign
4. Forward Engineering

- Source Code
- Annotated Source Code
- Target Graph Model
- Annotated Target Code
- Metamodel
- Target Constraints

R1: <<instantiates>>
R2: <<conforms-to>>
R3: <<conforms-to>>
The annotated source code is obtained through an iteration of manual input and the application of automated code categorisation rules, based on the categories defined in the metamodel. These rules are applied over the Abstract Syntax Tree (AST). We are using L-CARE, a tool from ATX, to implement the code annotation.
Reverse Engineering

- The reverse engineering step abstracts the annotated source code into a graph representation.
- This graph is an instance of the type graph defined in the architectural and technology paradigm.
- The graph’s level of detail depends on the annotation process.
- This representation will be used later for the transformation process.
- A relation between the graph and the original code is kept to support traceability.
Redesign

• The transformation of the source graph model to reflect the intended target architecture is called redesign.
• Graph transformation rules are used to specify this code category-driven transformation.
• These rules are applied to the source graph model to achieve the target one, complying to the intended architecture.
• Additional target constraints are given to specify the success criteria of the transformation.
Forward Engineering

- The target code can be obtained using two alternative strategies:
  - Logging the transformation execution to be replicated at code level using tool support
  - Generate the code from the target graph model and the links to the original AST
Future Work

• Completion of the reverse engineering and forward engineering steps
• Apply the methodology to a large size case study
Questions

• Thank you
• Questions?