Chapter 9
Enhancing a Rigorous Reuse Process with Natural Language Requirement Specifications

Laura Felice, Carmen Leonardi, Liliana Favre, and Maria Virginia Mauco
Universidad Nacional del Centro de la Pcia. de Buenos Aires, Argentina

Reusability is the ability to use the same software elements for constructing many different applications. Formal specifications can help to semiautomatic design processes based on reusable components. However, during the first stages of development, when the interaction with the stakeholders is crucial, the use of client-oriented requirements engineering techniques seems to be necessary in order to enhance the communication between the stakeholders and the software engineers. In this chapter, we propose a systematic reuse approach that integrates natural language requirement specifications with formal specifications in RSL (RAISE Specification Language). On the one hand, some heuristics are described to develop a formal specification in RSL starting from models belonging to the Requirements Baseline. On the other hand, we have defined a reusable component model that integrates RSL specifications at different levels of abstraction, as well as presented a process with reuse based on the model.

INTRODUCTION
The challenge of the software engineering is to satisfy the increasing demand of software systems in an economic and rapid way. Reusability software techniques based on component library provide a great potential to face it.

This chapter appears in the book, Successful Software Reengineering by Sal Valenti. Copyright © 2002, IRM Press, an imprint of Idea Group Inc.
The main problems associated with reusability techniques are:

- How to define reusable components library
- How to identify reusable components in a library
- How to integrate “implementation pieces” in a consistent system implementation

Our work hypothesis is that the formal specification of reusable components and the development of rigorous methods for their systematic reuse can help building “correct” and efficient software. “If, instead of being developed for just one project, a software element has the potential of serving again and again for many projects, it becomes economically attractive to submit it to the best possible quality techniques, such as formal specifications of components” (Meyer, 1997). There are many works which prove that software reusability can be addressed from formal descriptions (Krueger, 1992; Mili et al., 1995; Zaremski & Wing, 1997). Besides, formal descriptions are only accessible to specialists. If we want to construct a new software system we need other kind of techniques to represent the domain in which the software will be inserted. Those representations may be familiar to the stakeholder, whose participation in the first stages of development is crucial.

This work integrates and extends previous results from our research (Favre et al., 2000; Mauco, 2000). We propose a reuse strategy that integrates informal specifications with a reusable component library. In particular we use natural language-oriented models belonging to Requirements Baseline (Leite et al., 1997). These models are used to produce incomplete algebraic specifications in RSL (George et al., 1992), the formal specification language used in RAISE method. Those specifications are the input for the second part of the strategy, the reuse process, whose final result is a complete imperative specification in RSL, directly connected to code through the RAISE method. The reuse process is based on the RC model which integrates specifications at different levels of abstraction. The manipulation of RC components by means of reuse operators is the basis for the reusability. An essential step in the reuse process is component identification, not only because of its complexity, but also because it is the key to the success of the overall process.

**THE REQUIREMENTS BASELINE**

The Requirements Baseline (Leite, 1997) is a structure which incorporates descriptions about a desired system in a given macrosystem. It is composed of five views, but in this paper we will deal only with the Lexicon Model View and the Scenario View.

**The Lexicon Model View**

It is implemented by the LEL (Language Extended Lexicon). The LEL is a structure that allows the representation of significant terms in the studied
12 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the product's webpage:
www.igi-global.com/chapter/enhancing-rigorous-reuse-process-natural/29973?camid=4v1

This title is available in InfoSci-Books, InfoSci-Select, InfoSci-Select, InfoSci-Select. Recommend this product to your librarian:
www.igi-global.com/e-resources/library-recommendation/?id=1

Related Content

Dynamically Reconfigurable Architectures: An Evaluation of Approaches for Preventing Architectural Violations
www.igi-global.com/chapter/dynamically-reconfigurable-architectures/115422?camid=4v1a

The Impact of Regulatory Compliance on Agile Software Processes with a Focus on the FDA Guidelines for Medical Device Software
www.igi-global.com/article/impact-regulatory-compliance-agile-software/53206?camid=4v1a

A Look Behind Conceptual Modeling Constructs in Information System Analysis and Design
www.igi-global.com/article/look-behind-conceptual-modeling-constructs/40954?camid=4v1a
Open Source Software Systems: Understanding Bug Prediction and Software Developer Roles
www.igi-global.com/chapter/open-source-software-systems/37047?camid=4v1a