



Chapter VI

A Rigorous Model for RAISE Specifications Reusability

Laura Felice

Universidad Nacional del Centro de la Provincia de Buenos Aires, Argentina

Daniel Riesco

Universidad Nacional de San Luis, Argentina

ABSTRACT

During the RAISE specification development process, a variety of components and infrastructures are built. All of these components are not independent, but they are related to each other, especially when we specify different systems in the same infrastructure. The RAISE method is based on the idea that software development is a stepwise, evolutionary process of applying semantics-preserving transitions. So, the reuse process is crucial in all stages of the development, but there is not explicit reference to the specification reusability in this development process.

This chapter presents a rigorous process for reusability for RAISE Specification Language (RSL) components. We provide the mechanism to select a reusable component in order to guide RAISE developers in software specification and construction.

INTRODUCTION

Software components are typically very rich in information, making the task to characterize them and capture their relevant properties difficult. However this is not the only reason that makes software reuse difficult.

Information retrieval methods based on analyses of natural-language documentation have been proposed for constructing software libraries (Helm & Maarek, 1991; Maarek, Berry & Kaiser, 1991). Software components represented by natural language can make the retrieval process a task with ambiguity, incompleteness and inconsistency. All of these problems can be minimized by using a rigorous method in the retrieval of a component.

The RAISE method (D. Bjorner, lecture notes, Technical University of Denmark, 2000) is based on the idea that software development is a stepwise, evolutionary process of applying semantics-preserving transitions.

Based on this observation, we propose to introduce a Reusable Component (RC) model for the definition of the reusable component structure into RAISE.

In this work we propose the RC model for the definition of the structure of a reusable component that integrates specifications in RSL (George, Haff, Havelund, Haxthausen, Milne, Nielson, et al., 1992) and object-oriented code.

The RC model describes object-oriented classes at different levels of abstraction:

- Specialization — hierarchies of RSL implicit specifications related by formal specialization relation;
- Realization — hierarchies of RSL complete algebraic specifications related by realization relations;
- Code — hierarchies of imperative RSL schemes related by implementation relations and linked to object-oriented code.

We define a rigorous process for reusability of RC components. Its manipulation, by means of specification building operators (Rename, Extend, Combine, Hide), is the basis for the reusability.

Our approach allows that the properties of components formally specified can be characterized by giving a functional (RSL specification) description. Therefore they may be useful to someone searching for a particular component.

Different possible classes of existing RC components may be retrieved using a formal reasoning technique: an exact match to the query specification, a component more general than the query, or a component more specific than the query. An illustrative example presents a reusable component classification and a specification matching.

RELATED WORKS

Different approaches to specifying reusable components functionalities have

17 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage: www.igi-global.com/chapter/rigorous-model-raise-specifications-reusability/28111

Related Content

Interconnection Issues in CPS

(2015). *Challenges, Opportunities, and Dimensions of Cyber-Physical Systems* (pp. 61-75).

www.irma-international.org/chapter/interconnection-issues-in-cps/121250

An Empirical Study on the Migration to OpenOffice.org in a Public Administration

Bruno Rossi, M. Scotto, A. Sillittiand Giancarlo Succi (2009). *Software Applications: Concepts, Methodologies, Tools, and Applications* (pp. 1592-1607).

www.irma-international.org/chapter/empirical-study-migration-openoffice-org/29466

Self-Boosted With Dynamic Semi-Supervised Clustering Method for Imbalanced Big Data Classification

Akkala Abhilashaand Annan Naidu P. (2022). *International Journal of Software Innovation* (pp. 1-24).

www.irma-international.org/article/self-boosted-with-dynamic-semi-supervised-clustering-method-for-imbalanced-big-data-classification/297990

The Effects of the State of Union Address of USA Presidents to Their Performance Ratings

Carl Lee (2015). *International Journal of Software Innovation* (pp. 1-12).

www.irma-international.org/article/the-effects-of-the-state-of-union-address-of-usa-presidents-to-their-performance-ratings/122788

A Survey of Software Architecture Approaches

Kendra M.L. Cooper, Lirong Dai, Renee Steinerand Rym Zalila Mili (2009). *Designing Software-Intensive Systems: Methods and Principles* (pp. 256-288).

www.irma-international.org/chapter/survey-software-architecture-approaches/8239