The Development of a Computer Application that Identifies Reusable Components through Formal Specifications

Francisco Moreira Couto
Dept. of Informatics Engineering, Technical University of Lisbon
Av. Rovisco Pais, 1049-001 Lisboa Codex, Portugal
E-mail: fcouto@math.ist.utl.pt

Abstract

Software reuse is certainly a way for increasing software productivity and his quality, because we don't need to implement and verify it again. We propose to use Formal Methods to develop one computer application, which will be able to automate the process of software reuse. We use single-sort algebraic specifications to specify the components functionality, to then apply the mathematical framework of category theory to develop matching mechanisms, which identify reusable components.

This work presents one solution to the problem of software reuse. The method, proposed here, uses formal specifications in the classification and retrieval of software components, with the ultimate goal of identifying reusable components. More precisely, the solution presented here consists of a computer application with the functionality represented in the figure 1. Generically, the user identifies the component's algebraic specification [TM87], which is given to the application. Afterwards, the application creates the respective categorical representation [Wal91, Pie93]. Finally, the application will try to identify reusable components using the categorical representations that are in the repository.

We selected algebraic specifications in the component's classification because the algebraic specification languages are closer to human reasoning than the category theory, so is more difficult to the user classify the components using the category theory. We selected category theory in the components retrieval because category theory is a powerful, language independent, mathematical framework and can be efficiently implemented in matching algorithms because of his simple syntax.

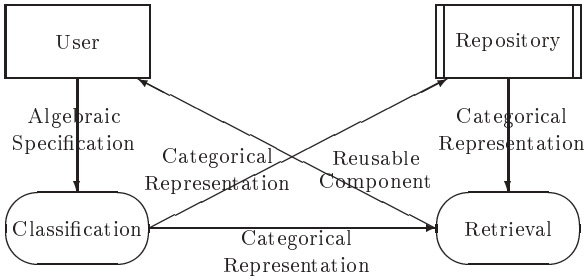

Figure 1: The generic architecture of the application proposed

The implementation of this application will be split in three phases. The first phase develops the process Classification of the figure 1, which translates components' algebraic specifications into their respective categorical representations. The second phase develops the process Retrievel of the figure 1, which uses matching algorithms to identify categorical representations that represent the same functionality behaviour. In the third phase will be studied solutions to the limitations of the algebraic specifications used in the application produced in the previous phases.

1 First phase

This phase will be made in two stages. The first stage will be the revision and the feasibility study of the theory proposed in the literature [Cre98, Lai76]. The goal is to obtain a set of algorithms that can be efficiently implemented in a computer application. The second stage consists in implementing the al-

gorithms identified in the first stage. The result of this stage will be a computer application that classifies automatically the software components in the category theory.

2 Second phase

This phase consists in retrieving reusable components through their categorical representation. Here, it'll be revised the methods presented in the paper [Cre98], which determine if two categorical representations match one with each other. The goal is to obtain a set of efficiently algorithms that retrieve reusable components through isomorphic and composite matching methods. The main objective of this phase is to develop a computer application that implement the matching algorithms identified. This application will automatically identify possible reusable components.

3 Third phase

In this stage will be necessary to make some extensions to the originals methods, so they can be implemented in a feasible computer application. These extensions will consist in: developing new methods for manipulate quantifiers; extending the structure of algebraic specifications; using infinite sorts in the algebraic specifications; identifying methods to manipulate predicates of arity more than one.

4 Complexity

During the developing of this project it'll be created some component's algebraic specifications to test the functionality of the computer application developed. Therefore, it'll be possible to make studies about the complexity of the algorithms developed, and then see if they are practicable.

5 Programming language

The algorithms developed in this project will be expressed in ML [RB98], a functional programming language. The principal reason for this choice is because functional languages are closer to mathematical notation and the computer application will manipulate two types of specifications based in mathematics.

6 Work Done

The first phase and the implementation of the isomorphic matching method had already been concluded. The deadline to the conclusion of this project is July of 2001.

Keywords:

algebraic specification, category theory, software reuse, specification matching.

References

[Cre98] Rui Gustavo Crespo. Matching single-sort algebraic specifications for software reuse. *International Journal of Software an Knowledge Engineering*, 8(3):401–425, 1998.

[Lai76] Luis M. Laita. Un estudio de la lógica algebraica desde el punto de vista de la teoría de categorías. *Notre Dame Journal of Formal Logic*, 17(1):89–118, January 1976.

[Pie93] Benjamin C. Pierce. *Basic Category Theory for Computer Scientists*. MIT Press, 1993.

[RB98] David E. Rydeheard and Rod M. Burstall. *Computational Category Theory*. Prentice Hall, 1998.

[TM87] W. M. Turski and T. S. E. Maibaum. *The Specification of Computer Programs*. Addison-Wesley, 1987.

[Wal91] R. F. C. Walters. *Categories and Computer Science*. Cambridge University Press, 1991.