Ph.D. Thesis Proposals (3) on formal Language Processing (MAPi)

Pedro Rangel Henriques
LP@CCTC – DI/Universidade do Minho

scholar year 2010/11

Inference of Component Annotations for Safety Reuse and Testing

Supervisor: Pedro Rangel Henriques
Keywords: Design-by-Contract, Reuse, Program Testing, Verification, Slicing, Inference

Abstract:
The *Design by Contract* (DbC) approach to software development (Meyer, 1992) facilitates modular verification and certified code reuse. The *contract* for a component (a procedure)—a set of preconditions, postconditions and invariants that *annotate* the beginning or end of each procedure, or *annotate* each cycle in the component body—can be regarded as a form of enriched software documentation that fully specifies the behavior of that component. The development and large adoption of *annotation languages* for the most popular programming languages (like Java Modeling Language (JML) (Burdy et al, 2003) for Java, or Spec# (Barnett et al, 2004) for C#) reinforce the importance of using DbC principles in the development of programs.

In the context of Gama project (daCruz, 2010a/2010d) Daniela da Cruz et al combine Verification Condition Generators, Theorem-Provers and Program Slicers to study (and visualize) the intra- and inter-procedural program behavior, and verify software systems. For that purpose the authors introduced two new concepts, *assertion-based slicing* (daCruz, 2010b) and *contract-base slicing* (daCruz2010f), both implemented in GamaSlicer tool.

Later on the authors also introduced another concept, *caller-based slicing* (Areias, 2010e) as a way of certifying that the integration of a component annotated with a contract into a system will preserve the correct behavior of the former, avoiding malfunctioning after integration. Once again, that concept was implemented in another tool, GamaPolarSlicer (Areias, 2010g).

In these ways, the authors address the problem of reusing annotated components proposing a rigorous approach to assure the quality of the application under construction.

However, annotating programs to formalize the contracts is not a trivial task for most software engineers who are used to code directly their algorithms in traditional programming languages. So, and despite of the recognized importance of DbC principle, many valuable SW components, available for reuse from SW archives, are not annotated.

To take profit of Gama tools (GamaSlicer to optimize reused code, or GamaPolarSlicer to verify the consistency of the calls) we aim, with this Ph.D. thesis proposal, at designing and implementing a tool to analyze a component and *infer an annotation* for it. After analyzing the source code to infer the annotation, we also envisage to generate a set of test cases to validate dynamically the component behavior.

The Ph.D. is planned to be divided into the following tasks: analyze and write the state of the art, studying Hoare Logic, Design-by-Contract, Component Annotation, Contract-based Slicing and Call-based Slicing; propose a way to infer the pre- and postconditions for SW components; propose a way to create tests for annotated SW components; design the architecture of a system to automatize the inference and testing processes, and implement it; extend the previous system with Contract-based and Caller-based Slicing to verify the system; choose case studies and make the system experimental validation.

Conferences (in ranks A to C, according to ERA2010) where the intermediate or final results should be published: ATVA, ISSTA, SEFM, PASTE, VinSE