

Automatic Verification of Eiffel Agents

MASTER THESIS PROJECT PLAN

Project period: October 20th 2008 – April 19th 2009
Student name: Julian Tschannen
Status: 9th Semester
Email address: juliant@ethz.ch
Supervisor name: Martin Nordio

1. PROJECT DESCRIPTION

Overview

Eiffel - as well as other object-oriented languages - have a built-in support for higher-order implementations through function objects. These are called *agents* in Eiffel. As with other functions, these can have pre- and postconditions.

Although the problem of static verification of function objects has been solved for functional programming languages, these solutions cannot be applied to object-oriented languages due the use of the heap and side effects. Agents are therefore difficult to prove.

In *Reasoning about Function Objects* [3], a novel approach is described which uses side effect free (pure) routines to specify the pre- and postconditions of agents. To specify routines that take agents as arguments, these pure routines are used.

Scope of the work

The master thesis focuses on the techniques described in *Reasoning about Function Objects* [3]. This verification methodology for agents will be implemented in *Ballet* [4] to allow static verification of agents. Also, the methodology will be extended to allow a generic argument count and return values.

Instead of having *Ballet* as a standalone modification of EiffelStudio [6], it will be integrated in the *ETH Verification Environment* (EVE) [7].

As optional parts of the thesis, the specification language will be extended with modifies clauses, and non-interference will be implemented.

As *Ballet* is based on *Boogie* [5] which currently only runs on Windows, the verification will for the moment only run on Windows.

Intended results

At the end of the master thesis, the automatic verification system of *Ballet* should be able to verify agents and should be integrated in EVE. That is, a user should be able to statically verify his code which includes agents using EVE.

2. BACKGROUND MATERIAL

Reading list

- **Reasoning about Function Objects.** M. Nordio, C. Calcagno, B. Meyer, and P. Müller

- **Specification and verification challenges for sequential object-oriented programs.** Gary T. Leavens and K. Rustan M. Leino and Peter Müller
- **A Modular Verification Methodology for C# Delegates.** Peter Müller and Joseph N. Ruskiewicz
- **BoogiePL: A typed procedural language for checking object-oriented programs.** Robert DeLine; K. Rustan M. Leino, March 2005
- **Boogie: A Modular Reusable Verifier for Object-Oriented Programs.** Mike Barnett, Bor-Yuh Evan Chang, Robert DeLine, Bart Jacobs, and K. Rustan M. Leino. In FMCO 2005, LNCS vol. 4111, Springer, 2006
- **Making classes provable through contracts, models and frames, Chapter 8.** Bernd Schoeller, dissertation thesis, Departement Informatik, ETH Zurich, 2007

3. PROJECT MANAGEMENT

Objectives and priorities

- Updating Ballet for EiffelStudio 6.3
- Verification of agents with zero or one open argument
- Verification of agents with return values or closed arguments
- Implementation of non-interference
- Integration into EVE

Criteria for success

The master thesis is a success, if it is possible to statically verify a set of examples illustrating common application of agents. These examples include the formatter example and the archive example described by Nordio et al [3].

If an example is entered which does not satisfy its specification, the tool will not prove it. The tool will only prove correct programs and is expected to be sound, although soundness of the tool is out of scope for this thesis.

Method of work

Meetings with the supervisor will be held on a regular basis.

Quality management

Documentation

A thesis report will be written continuously throughout the project.

Validation steps

Continuous feedback from the supervisor will guide the development process. Multiple examples should be created for the various steps of the thesis which can be used as a test suite.

4. PLAN WITH MILESTONES

Project steps

1. Updating *Ballet* – 4 weeks
 - a) Compiling *Ballet* for EiffelStudio 5.7
 - b) Evaluating *Ballet* as a verifier
 - c) Updating *Ballet* for EiffelStudio 6.3

- d) Testing *Ballet*
- 2. Verifying Agents – 5 weeks
 - a) Transforming agent calls without arguments
 - b) Transforming agent calls with one open argument
 - c) Transforming agent calls with closed arguments
 - d) Generalising argument count
- 3. Extending the verification methodology with functions – 4 weeks
- 4. Optional: Extending the specification language with modifies clauses – 3 weeks
- 5. Optional: Implementing non-interference – 3 weeks
- 6. Integration into EVE – 3 weeks
- 7. Writing thesis report – 4 weeks

Deadline

April 19th 2009

Tentative schedule

Task	Duration	Weeks	43	44	45	46	47	48	49	50	51	52	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	
1. <i>Update Ballet</i> for ES 6.3	20 days	4																											
2. Verifying agents	25 days	5																											
3. Extending to functions	20 days	4																											
4. Extending with modifies	15 days	3																											
5. Non-interference	15 days	3																											
6. EVE integration	15 days	3																											
7. Thesis report	20 days	4																											

REFERENCES

- [1] Chair of Software Engineering: *Semester-/Diplomarbeiten*; Online at: <http://se.inf.ethz.ch/projects/index.html>, consulted in October 2002.
- [2] Bertrand Meyer: *Object-Oriented Software Construction, 2nd edition*, Prentice Hall, 1997.
- [3] M. Nordio, C. Calcagno, B. Meyer, and P. Müller: *Reasoning about Function Objects*.
- [4] Bernd Schoeller: *Making classes provable through contracts, models and frames*, dissertation thesis, Departement Informatik, ETH Zurich
- [5] Spec#/Boogie: <http://research.microsoft.com/specsharp/>
- [6] EiffelStudio: <http://eiffelstudio.origo.ethz.ch/>
- [7] ETH Verification Environment: <http://eve.origo.ethz.ch/>