Model-based contracts for Java / C# collections

BACHELOR THESIS PROJECT PLAN

Project period: 02.05.2012 - 17.09.2012

Student: Tobias Kiefer

E-mail address: tkiefer@student.ethz.ch

Supervisor: Nadia Polikarpova

1. PROJECT DESCRIPTION

Overview

Complete model based contracts combined with automated contract-based testing have been shown to be able to reveal subtle faults in Eiffel classes. The goal of this project is to apply the same approach to libraries written in other programming languages.

Scope of the work

The scope of the project includes completing an existing port of the Mathematical Model Library (MML) from Eiffel to C# and, with the help of this library, providing complete model-based contracts for C# libraries, such as DSA [3] and .NET collections [4]; as well as conducting contract-based automated tests of the contracted libraries.

Time permitting, the scope of the project can be extended to creating a Java port of the MML library, providing model-based contracts for Java libraries, such as Gnu Trove [7], Jakarta Commons [8] and java.util [9], and conducting automated tests of those libraries.

Intended results

Completing the entire project would result in the following artifacts:

1. C# port of the MML library
2. Selected classes of the DSA library with complete model-based contracts
3. Classes from the System.Collections.Generic namespace of the mscorelib library with complete model-based contracts
4. Java port of the MML library
5. Selected classes from the Gnu Trove and Jakarta Commons libraries with complete model-based contracts
6. Data structure classes from the java.util package with complete model-based contracts
7. Project report

Artifacts 4, 5 and 6 are optional (they correspond to the optional part of the project scope).

2. BACKGROUND MATERIAL

Reading list

Specifying Reusable Components. Nadia Polikarpova, Carlo A. Furia, Bertrand Meyer, *In Proceedings of the 3rd International Conference on Verified Software: Theories, Tools, and Experiments (VSTTE'10) (Gary T. Leavens, Peter O'Hearn, Sriram Rajamani, eds.), Springer, volume 6217, 2010.*
http://se.inf.ethz.ch/people/furia/pubs/vstte10.pdf

Code Contract: User Manual. Microsoft Corporation, January 2012. http://research.microsoft.com/en-us/projects/contracts/userdoc.pdf

Exploiting the Synergy between Automated-Test-Generation and Programming-by-Contract. Michael Barnett, Manuel Fahndrich, Francesco Logozzo, Peli de Halleux, and Nikolai Tillmann, *In Proceedings of the 31st International Conference on Software Engineering (ICSE'2009)*

Pex: tutorials. http://research.microsoft.com/en-us/projects/pex/documentation.aspx

3. PROJECT MANAGEMENT

Objectives and priorities

Project tasks are prioritized as follows:

1. Completing the C# port of the MML library and providing complete model-based contracts for DSA
2. Automated testing for DSA
3. Providing complete model-based contracts for .NET collections
4. Automated testing of .NET collections
5. Creating a Java port of the MML library
6. Providing complete model-based contracts for Gnu Trove and Jakarta Commons
7. Automated testing for Gnu Trove and Jakarta Commons
8. Providing complete model-based contracts for java.util
9. Automated testing for java.util

Method of work

The following specification notations will be used for creating model-based contracts:

- C#: Code Contracts [1]
- Java: JML [5]

The following automated testing tools will be used to conduct contract-based testing:

- C#: Pex [2]
- Java: Yeti [6]

Documentation

The project documentation has to include

1. API documentation for the MML library ports (automatically generated)
2. Summary of the created model-based specifications, in particular: which classes are contracted, what percentage of contracts is incomplete, any design inconsistencies found while creating complete contracts
3. Description of the conducted automated testing sessions, including the parameters of the sessions and discovered faults

4. PLAN WITH MILESTONES

Project steps

1. Familiarize with model-based contracts, the Eiffel MML library, existing C# code, CodeContracts and Pex
2. Complete the C# port of the MML library
3. Write contracts for selected classes from the DSA library
4. Conduct automated testing of DSA using Pex

5. Write contracts for the .NET collection classes
6. Conduct automated testing of .NET collections using Pex
7. Create a Java port of the MML library
8. Write contracts for selected classes from GNU Trove and Jakarta Commons libraries
9. Conduct automatic testing of GNU Trove and Jakarta Commons using Yeti
10. Write contracts for java.util collection classes
11. Conduct automated testing of java.util collection classes using Yeti
12. Write the project report

Steps 7-11 are optional.

Deadline

The deadline for the project is 17.09.2012.

Tentative schedule

	May				June				July				August				September	
Step/Week	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	■	■	■															
2		■	■															
3			■	■														
4				■	■													
5					■	■											■	
6																	■	■
12																	■	■

References

[1] Code Contracts [http://msdn.microsoft.com/en-us/library/dd264808.aspx]

[2] Pex [http://research.microsoft.com/en-us/projects/pex/]

[3] DSA [http://dsa.codeplex.com/]

[4] .Net collections [http://msdn.microsoft.com/en-us/library/ybcx56wz.aspx]

[5] JML [http://www.eecs.ucf.edu/~leavens/JML/]

[6] Yeti [http://www.yetitest.org/]

[7] GNU Trove [http://trove.starlight-systems.com/]

[8] Jakarta Commons [http://commons.apache.org/collections/]

[9] Java.util [http://docs.oracle.com/javase/6/docs/api/java/util/package-summary.html]