

I. Overview

Project Title

Porting BETA to ROTOR/sscli

Contributors Names

Ole Lehrmann Madsen, Peter Andersen, plus possibly student programmers.

Abstract

The goal of this project is to make a full port of the BETA programming language including a bootstrapped compiler and porting central BETA libraries to ROTOR/sscli as well as .NET/CLR.

II. Contact Details

Institution Details

Department of Computer Science
University of Aarhus
IT-parken, Aabogade 34
DK-8200 Aarhus N, Denmark

Phone: +45 8942 5600

Fax: +45 8942 5601

Computer Science Department WWW: <http://www.daimi.au.dk>

Object-Oriented Language Interoperability: <http://www.daimi.au.dk/~beta/ooli>

Contributor Details

Ole Lehrmann Madsen, Professor,
Email: olm@daimi.au.dk
URL: <http://www.daimi.au.dk/~olm>

Peter Andersen, M.Sc., Academic Researcher,
Email: datpete@daimi.au.dk
URL: <http://www.daimi.au.dk/~datpete>

III. Project Description

Background

Over the last two years a project has been targeting porting the BETA programming language [1] to the Microsoft .NET platform for Windows. The project was partly funded by a grant from Microsoft Denmark.

The project has been part of a research project on Language Interoperability where the goals have been to study platforms like .NETA/CLR and Java/JVM with respect to language interoperability involving a general object-oriented language like BETA. In addition language interoperability issues arising from implementing support for BETA on Visual Studio and Eclipse have been investigated. The results of these projects are presented in [4, 5], which are papers in progress.

Porting BETA to CLR and/or JVM has a number of challenges. BETA is more general than languages like C# and Java:

- In BETA abstraction mechanisms like class and method are unified into one common abstraction mechanism, called pattern.
- BETA supports generic classes in the form of virtual classes (patterns).
- In BETA it is possible to jump out of a nested structure of method calls.
- BETA's concurrency mechanism is based on Simula-style coroutines as a basic mechanism for supporting higher-level concurrency abstractions like monitor, Ada-like rendezvous, futures, etc. [1, chapter 14].

The above list are examples of BETA constructs that were not straight-forward to implement on .NET.

In order to support true language interoperability, it should be possible to use, say C# libraries from BETA – with a port of BETA to CLR this should be possible. It should, however, also be possible to use BETA libraries from say C#. This means that the mapping of BETA to CLR should be '*readable*' when seen from a C# view.

The status of that project is that most parts of the BETA language are now supported by the .NET compiler. A few of the BETA class libraries have been reimplemented on top of the .NET framework classes. The BETA compiler for .NET has not been bootstrapped - a native WIN32 based compiler is still used.

Project Details

The goal of this project is to make a full port of BETA including a bootstrapped compiler and porting central BETA libraries to Rotor as well as .NET/CLR. Below are listed the main activities of the project. **The activities 1-3 are expected to be completed during**

the project. The activities 4 and 5 will be investigated in case there is time left when 1-3 are completed.

1. Implementation of complete BETA interface to CLR.

There are elements in CLR that are not supported by BETA. These include:

- a. Value types
- b. Enumerations
- c. Throw and handling of CLR exceptions
- d. Support for static fields
- e. Support for custom attributes
- f. Leave/restart over coroutine border
- g. Support for multiple interfaces
- h. 64 bit arithmetic

Some of these features must be supported in order to port essential BETA libraries to Rotor and .NET.

2. Porting essential BETA libraries to Rotor and .NET.

By essential BETA libraries are meant the BETA libraries necessary to bootstrap a full BETA compiler to ROTOR and .NET.

3. Bootstrapping the BETA compiler to Rotor and .NET.

4. Develop a GUI framework on top of ROTOR and .NET.

It is possible to port the current BETA GUI framework to .NET, but the binding would be dependent on the `System.Windows.Forms` and `System.Drawing` namespaces of the .NET Framework libraries. But as these are not supported by the ROTOR libraries, a different approach would have to be used here. *Views* [2] is an alternative, but the approach is far too simple for realistic use scenarios. A far more advanced graphics library called Octopus [3] is being developed at our university; we would like to investigate in the possibilities of porting this to run on top of .NET and ROTOR.

5. Support for Simula/BETA-style coroutines. The current implementation of coroutines in the .NET port of BETA is based on CLR-threads. Since coroutines are much more light-weight than threads and intended to be a primitive for implementing threads, it imposes a significant overhead to use threads for implementing coroutines. In this activity we will investigate the possibility of modifying the ROTOR CLR runtime runtime (bytecodes, jitter, GC, ...?) to support Simula-like coroutines and use these to implement higher order concurrency abstractions.

Experience

Please see “Contributor Details” URLs and “Object-Oriented Language Interoperability” URL above, and “References” below.

IV. Project Plan

Deliverables/Milestones

- M1—end of month 1: A simple BETA program (Hello World) can be compiled and the resulting assemblies can be executed on ROTOR
- M2 – end of month 3: The BETA compiler test suite can be compiled and the assemblies can be executed on ROTOR
- M3 – end of month 6: Complete bootstrap of a full BETA compiler to ROTOR and .NET.

Intellectual Property

We expect to submit academic articles to relevant conferences on language interoperability in general and .NET/ROTOR in particular.

The tools and other software resulting from this project will be made freely available for download from the University web.

V. Supporting Information

Costing

The estimated costs are:

Salaries:	EUR	22.000
Travel expenses:	EUR	3.000
Total:	EUR	25.000

References

- [1] Ole Lehrmann Madsen, Birger Møller-Pedersen, Kristen Nygaard: *Object-Oriented Programming in the BETA Programming Language*. Addison-Wesley, June 1993, ISBN 0-201-62430-3, 350 pages.
<ftp://ftp.mjolner.com/BETA/BOOK/betabook.pdf>. See also
<http://www.daimi.au.dk/~beta>
- [2] <http://www.cs.up.ac.za/rotor>
- [3] <http://www.daimi.au.dk/~beta/octopus>
- [4] Ole Lehrmann Madsen, Peter Andersen, Mads Brøgger Enevoldsen: *Integration of BETA with Eclipse – an exercise in language interoperability*. January 2004.
Submitted for *eclipse Technology eXchange*. Draft can be obtained upon request from datpete@daimi.au.dk or olm@daimi.au.dk
- [5] Ole Lehrmann Madsen, Peter Andersen: *Language Interoperability with BETA on .NET and Java Virtual Machines* (in progress).