RIT Department of Computer Science
MSc Project Pre-Proposal:
An Aspect Oriented Programming Framework for C#

Wei Liao

June 30, 2012

Chair: Prof. James Heliotis

Procedural or Object Oriented Programming are not sufficient to clearly capture some important software design decisions [1]. While OOP has allowed programmers to nicely abstract properties and behaviors of real world objects, usage of those objects could still be scattered throughout different modules of the program. Overtime, these tangled cross-cutting concerns [1] can be difficult and expensive to maintain. Aspect Oriented Programming (AOP) technique [1] can be used to cleanly separate such concerns from the program.

Few languages have built-in AOP support, so frameworks implemented with different techniques and flavors are available to add its capabilities to various languages, including C# [2, 3, 4]. Among them, a commercial product called PostSharp has generated lots of interest in the .NET community. For its ease of use, flexibility and powerful features [2].

The idea of this project is to attempt to provide a free open source alternative to PostSharp. I have looked at and compared several different approaches on how to implement the framework. The technique I plan on using is called Compile Time Weaving. It roughly works as follows: it uses C# attribute as the aspect, so functions defined in the attribute contain the advice code. By applying the attribute to a function, class or assembly, it marks them as possible join points. At the post compilation phase, AOP Framework will be invoked if specified by the developer. The framework takes apart the assembly using reflection. An aspect weaver [1, 2] is employed to identify and stitch together the join points and aspects, finally it modifies the bytecode with MSIL rewriting [5], producing a modified assembly.

To evaluate the project, I will show that programs implemented with the framework will have cleaner code; and that developers will also have lesser codes to write; cross-cutting concerns will be clearly separated from the program when compared to the program implemented without using the framework.

My project deliverable will be in the form of a setup executable. Upon installation it will integrate itself with the Microsoft Build system used in the Visual Studio IDE. As opposed to most other frameworks, a secondary project goal is to have zero configuration on the part of the developer, to provide a seamless integrated experience with the IDE.

References

[1] et al. Gregor Kiczales, John Lamping. Aspect-Oriented Programming. In *Proc. of European Conference on Object-Oriented Programming (ECOOP), Finland*, 1997.

[2] Gael Fraiteur. User-friendly aspects with compile-time imperative semantics in .NET: an overview of PostSharp. In *International Conference on Aspect-Oriented Software Development*, 2008.

[3] Howard Kim. AspectC#: An AOSD implementation for C#. Master's thesis, Trinity College Dublin, 2002.

[4] M. Devi Prasad and B. D. Chaudhary. AOP Support for C#. In *Proc. of Second International Conference on Aspect-Oriented Software Development*, pages 49–53, 2003.

[5] Aleksandr Mikunov. Rewrite MSIL Code on the Fly with the .NET Framework Proling API. *MSDN Magazine*, 2003.