

Evolving a Multi-language Object-Oriented Framework: Lessons from .NET

Jim Miller

Microsoft Corporation,
jsmiller@microsoft.com

Abstract. In 2001 Microsoft shipped the first public version of its Common Language Runtime (CLR) and the associated object-oriented .NET Framework. This Framework was designed for use by multiple languages through adherence to a Common Language Specification (CLS). The CLR, the CLS, and the basic level of the .NET Framework are all part of International Standard ISO/IEC 23271. Over 20 programming languages have been implemented on top of the CLR, all providing access to the same .Net Framework, and over 20,000,000 copies have been downloaded since its initial release.

As a commercial software vendor, Microsoft is deeply concerned with evolving this system. Innovation is required to address new needs, new ideas, and new applications. But backwards compatibility is equally important to give existing customers the confidence that they can build on a stable base even as it evolves over time. This is a hard problem in general, it is made harder by the common use of virtual methods and public state, and harder still by a desire to make the programming model simple.

This talk will describe the architectural ramifications of combining ease-of-use with system evolution and modularity. These ramifications extend widely throughout the system infrastructure, ranging from the underlying binding mechanism of the virtual machine, through program language syntax extensions, and into the programming environment.