DSLs in Boo

Domain-Specific Languages in .NET

OREN EINI
WRITING AS AYENDE RAHIEN

MANNING
Greenwich
(74° w. long.)

contents

preface

In 2007, I gave a talk about using Boo to build your own domain-specific languages (DSLs) at JAOO (http://jaoo.dk), a software conference in Denmark. I had been working with Boo and creating DSLs since 2005, but as I prepared for the talk, I was surprised to see just how easy it was to build DSLs with Boo. (I find that teaching something gives you a fresh perspective on it.)

That experience, and the audience's response, convinced me that you don't have to be a compiler expert or a parser wizard to build your own mini-languages. I realized that I needed to formalize the practices I had been using and make them publicly available.

One of the most challenging problems in the industry today is finding a way of clearly expressing intent in a particular domain. A lot of time and effort has been spent tackling that problem. A DSL is usually a good solution, but there is a strong perception in the community that writing your own language for a particular task is an extremely difficult task.

The truth is different from the perception. Creating a language from scratch would be a big task, but you don't need to start from scratch. Today, there are lots of tools and plenty of support for creating languages. When you decide to make an internal DSL—one that is hosted inside an existing programming language (such as Boo)—the cost of building that language drops significantly.

I routinely build new languages during presentations (onstage, within 5 or 10 minutes), because once you understand the basic principles, it is easy. Easy enough that it deserves to be a standard part of your toolset, ready to be used whenever you spot a problem that is suitable for a DSL solution.