

Domain-Specific Languages with Custom Operators*

Áron Baráth, Zoltán Porkoláb

Eötvös Loránd University
{baratharon,gsd}@caesar.elte.hu

Abstract

Embedded domain-specific languages have become more popular due to their expressive power. The tasks are composed at the level of the problem domain. This paper shows the key elements and the benefits of implementing embedded domain-specific languages in strongly typed imperative programming languages. These ideas are useful for those people who use or implement embedded domain-specific languages but they are not professionals in functional programming and prefer imperative programming languages instead.

The most common host language of the domain-specific languages is a functional programming language, because of the freedom to introduce custom operators. The investigation is based on a strongly typed imperative language which allows to define such operators. Domain-specific languages increase abstraction level and hide details. This paper explores other aspects of domain-specific languages for functional programmers.

The key language element is in functional languages to define custom operators. These language elements must be found in the host language for implementing an efficient embedded domain-specific language. In this paper there is an example language to demonstrate the expressive power. The investigation concludes that the benefits of using an imperative language to implement an embedded domain-specific language are the increased expressiveness by eliminating redundant parentheses' and unnecessary verbose function names from the source code. Based on the strong static type system the custom types preserve the type-correctness.

This paper shows ideas and recommendations about how to implement a domain-specific language in a strongly typed imperative language.

Keywords: custom operator, domain-specific language, static type-system, imperative language

MSC: 68N15 (Programming Languages)

*The project was supported by Ericsson Hungary.