XML based programming environments

Supervisor: Jose Paulo Leal <zp@dcc.fc.up.pt>
Research unit: CRACS - http://www.dcc.fc. up.pt/cracs/

January 27, 2008

Context

Programming is an activity that we usually associate with coding in a single programming language - usually our pet language. Although this might be true when learning computer science or writing toy programs, it is increasingly not the case in professional programming. For developing software projects programmers need several languages, not all of them strictly programming languages. A single project could use HTML, CSS, XSLT or JavaScript for web formatting; PHP, JSP or ASP for presentation; Java, C# or Perl for the business logic; SQL for querying data; and several other languages. These other languages are usually configuration languages for components such as: application servers, build managers, logging, among others.

Programmers of larger projects use coding workbenches known as Integrated Development Environments (IDE). An IDE assembles in a single Graphical Users Interface (GUI) a collection of tools that gravitate around program editors. These editors are more then mere text editors and provide many usefully services such as keyword and name completion, incremental compilation, refactoring, and many others. Nevertheless, edition in IDEs is still divided according to languages, and code from different languages is separated in different files. Due to this fact, sometimes the same concept is split or even redundantly stated in several languages, on different files.

Mixing languages in a single document is a basic feature of XML: an XML document can contain elements and attributes from several namespaces and may be typed using several independent schemata (or none, in standalone documents). This feature was required for embedding the new web formatting languages created with XML, such as MathML or SVG, in XHTML documents. It was also necessary for mixing target and transformation languages in XSL, and was decisive for combining data, data types and protocols in web services.

On the other hand, XML establish itself as the formalism of choice for storing data whenever interoperability is the main concern. As part of this trend, not only web and GUI formatting languages where defined in XML, as well as many configuration languages. Even general purpose programming languages, such as Java of C#, have nowadays a standard XML representation.

Goals

The main goal of this PhD plan is to explore the consequences of using XML as core formalism for encoding programs in the context of IDEs. With this approach we expect to have programming units mixing different languages without the need to separate them in different files, to be able to make references across languages, and to implement generic tools that are independent from the programming language.

The goals of this PhD proposal are:

- to study formal definitions of programming languages and other languages used in programming, and its limitations when used for developing programming tools;
- to investigate alternatives to overcome those limitations, relating XML type definition languages with traditional formalisms for specifying languages;
- to implement a prototype IDE with components for editing, refactoring, documenting and navigating, in projects encoded as XML documents.