An action description model for active components in the semantic web

Nicolas Sabouret and **Jean-Paul Sansonnet**
LISMI-CNRS, BP133, 91403 Orsay Cedex, France
email: nico@limsi.fr

1 Active components in the semantic web

The semantic web is an extension of the current web in which software components can perform automatic processing based on the page contents and the semantics of these contents. The current trends propose to make use of *knowledge representation* techniques to enhance web pages information with semantics. The emergence of standards for *structural* knowledge representation, like DAML-OIL and RDF, allowed researchers to offer better services in information retrieval: semantic requests, semantic document indexing, *etc.*

In this context, online services on the Internet led to the notion of **active components**, *i.e.* software components characterised by *actions* and capable of providing services to the users. Active components are not simply used to process or produce information in the web, like most web services do (*e.g.* through CGI-scripts). They are *part of* this information which is no longer only structural but also functional. As a consequence, the semantic web should allow us to represent and manipulate both structural and functionnal knowledge.

In this frame, we take an interest in studying the following issues: 1) Action representation for active components ; 2) Requests about actions of active components ; 3) Reasoning about actions to answer the requests. Our aim is to study and develop active components capable of representing their own actions and reasoning about them in order to answer a large range of questions about their behaviour and execution.

2 Integrating active components into web pages

The aim of our poster is to present VDL, an active component description language, which has the following properties: 1) it is expressive enough to represent a wide class of active components; 2) it allows us to access *at runtime* to a formal description of the component's actions; 3) it can be used to produce explanations about the component's behavior in reply to users questions. As a consequence, every active component described using our formalism has access to its own code. It can reason about its in order to answer questions about its actions.

VDL constitutes a theoretical basis for **integrating** active components into XML pages, called VDL. Nowadays, components are merely *embedded* into web pages as applets. On the contrary, our language allows the programmer to attach functional information to structural data, in order to describe and run active components as web pages.

General principle

The software component's description structure in VDL is a tree whose nodes are character strings called *concepts*. We call this concept tree a *view* (like in Forbus QPT). A view is the formal representation in VDL of all data and actions of the operating world which is embedded in the component.

In VDL, A component's runtime is based on rewriting the view at each *execution step*, like in *Maude* [MESEGUER, 2000]. We defined a function ϕ_{exec} which transforms the view v_t into the view v_{t+1} using the procedural concepts in v_t.

One originality of our formalism is that it integrates *in a single representation* the description of the component's static structure together with the description of its actions, unlike classical action representation formalisms for the semantic web like DAML-S [Coalition, 2002][1]. That is to say that actions, just like data, are subtrees of the component XML tree. Moreover, our language is Turing-complete, linear in time and space wrt a Turing machine and as expressive as Lisp. We totally defined this language's operational semantics and we implemented it in Java 1.1.7. Demos are available on our web page.

References

[Coalition, 2002] DAML-S Coalition. *DAML-S: Web Service Description for the Semantic Web*, 2002.

[MESEGUER, 2000] J. MESEGUER. Rewriting Logic and Maude: Concepts and Applications. In *Proc. RTA 2000*, pages 1–26, 2000.

[1] However, we do not pretend that our representation is as complete as DAML-S is for web service descriptions. VDL is more suitable for describing active components such as the ones we propose on our web page: http://www.limsi.fr/Individu/nico/examples