Extensions to UML Using Stereotypes

Daniel Riesco
Universidad Nacional de San Luis and Universidad Nacional de Rio Cuarto, Argentina

Marcela Daniele
Universidad Nacional de Rio Cuarto, Argentina

Daniel Romero
Universidad Nacional de Rio Cuarto, Argentina

German Montejano
Universidad Nacional de San Luis, Argentina

INTRODUCTION

The Unified Modeling Language (UML) allows to visualize, to specify, to build and to document the devices of a system that involves a great quantity of software. It provides a standard form for writing the models of a system, covering so much of the conceptual aspects (such as processes of the business and functions of the system) as the concrete ones (such as the classes written in a specific programming language, schemas of databases and software components).

In 1997, UML 1.1 was approved by the OMG becoming the standard notation for the analysis and the design oriented to objects. UML is the first language of modelling in which a metamodel in its own notation has been published. It is a strict subset called Core. It is a self-referential metamodel.

It is a very expressive language that covers all of the necessary views to develop and to deploy systems. UML is a language that provides three extension mechanisms (Booch, Rumbaugh, & Jacobson, 1999): stereotypes, tag values, and constrains. The stereotypes allow to create new types of elements of model based on the elements that form the metamodel UML extending the semantics of the same one, the tag values are an extension of the properties of an element of UML, allowing to add new information to the specification of the same one, and the constrains are an extension of the semantics of UML that allow to add new rules or to modify the existent ones.

The organization of this overview is given in the following way: first, we present the stereotypes according to the standard of OMG; second, we expose the analysis of works that extend UML using stereotypes in diverse real domains; third, we make an analysis of the stereotypes of UML; and we finish giving a general conclusion where we focus ourselves in the distinction of the works according to their inclusion or not of the created stereotypes in the metamodel of UML.

STEREOTYPE ACCORDING TO THE STANDARD OF OMG

A stereotype provides a form of classifying elements in such a way that they work in some aspects as if they were instances of a new constructor of the "virtual"

Attributes	
BaseClass	This specifies the names of one or more elements from UML modeling to which the stereotype is applied, such as classes, associations.
Icon	This is the geometric description of the icon that will be used to present an image of the element of the marked model with the stereotype.

metamodel. A stereotype could also be used to indicate a meaning or different use between two elements with identical structure. A stereotype can also specify a geometric icon to be used to present elements with the stereotype.

USING STEREOTYPES IN DIVERSE REAL DOMAINS

UML adapts to any technique, because it has extension mechanisms that don't need to redefine the nucleus UML, allowing to obtain a modeling more appropriate to the different particular domains. All of the extensions should follow the standard proposed by the OMG (2001).

Modeling of Business with UML

UML was initially designed to describe aspects of a software system. For the modeling of business, UML needed to be extended to identify and to visualize resources, processes, objectives and rules more clearly. These are the primary concepts that define a business system. The Eriksson-Penker Business Extensions (Eriksson & Penker, 1999) provide new stereotypes for

their model. In a diagram of class of UML, they represent a process through a specific symbol that corresponds to an activity stereotyped in an activity diagram. The resources used by the process are modeled with a stereotyped dependence *«supply»* and the resources controlled by the process are modeled with a stereotyped dependence *«control»*.

Modeling of Web Applications with UML

In Baresi, Garzotto, and Paoloni (2001), they propose a framework denominated W2000, for the design of Web applications. They combine the use of UML and HDM (Hypermedia Model). The Web applications require the integration of two different, but interrelated, activities: the design hypermedia that is focused in the navigation way and the structure of the information, and the functional design that is focused in the operations. Among their main purposes is the extension to the standard of UML of the dynamic diagrams.

It uses *«node type»* like stereotype of UML to define a node type that allows to reach different structures of information and defines the symbol "@" to indicate that a node stereotyped with *«node type»* will be the node for defect where all the users begin to navigate.

They propose a symbol called "index" that allows the users of a navigation to select one of the elements from a list of indexes.

The "collection links" define how the users can navigate between the core and the members of a collection, and they add a symbol to graphically represent the pattern "Index + Guided Tour".

For the functional design, they define the diagrams of scenarios, and these are represented through an extended sequence of diagrams of UML. The extensions refer as much to objects as to the step of messages. The objects are organized in entities (components and nodes), semantic associations and collect. The "free navigation" is represented with dotted lines and the "constrained navigation" is represented with a line with a diamond in it.

In 1999, Jim Conallen, Principal Consultant of Conallen Inc., Object Oriented Application Development in Conallen Inc., presented in their paper an extension to UML, in a formal way, to model applications Web.

The extension was presented at several other conferences in 1999, including the Rational Users Conference in Seattle (July 1999), and two Wrox Press ASP conferences in Washington, DC (September 1999) and in London (November 1999).

Various summaries and introductions to the extension have or will appear in the *Communications of the ACM* (ASPToday, *http://www.asptoday.com/articles/19990517.htm*), and in the UML Resources Web site at Rational Software. A full explanation of this work is currently being prepared for the book, *Building Web Applications with UML* (Conallen, 2002), published in the Object Technology Series of Addison-Wesley Longman.

This article presents an extension of UML for Web application designs. Part of the extension mechanism of UML is the ability to assign different icons to stereotyped classes. A list of prototype icons for the most common class stereotypes can be found as an appendix. It defines two new stereotypes to model the difference between the executed methods in the server and the executed functions in the client. In a page, a method that executes on the server will be stereotyped as *«server method»* and functions that run on the client *«client method»*. This solves the problem of distinguishing attributes and methods of a page object. It proposes the modeling of a page with two stereotyped classes, *«server page»* and *«client page»*. They define several stereotypes to represent the associations, such as: *«builds»* that is modeled with an unidirectional association from the server page to client page, *«redirects»* to model the redirection to other *«server page»*, *«links»* for defined associations between pages clients and other pages (client or server). Also, they define stereotypes to model components, *«server component»* and *«client component»*, for Forms *«form»*, for Framesets *«frameset»*. Other defined stereotypes, *«scriplet»* for cached client page, and *«xml»* for a hierarchical data object that can be passed to and from a Web server and client browser.

In Gorshkova and Novikov (2001), a UML extension capable to refine the design of the client part of Web application is defined. Several new diagrams are specified which provide a precise definition of the content of Web pages and navigation between them. The composition diagram is a special case of class diagram. We use it to express the structure of the Web pages and identify their content: how they are connected together and what data is carried from one page to another. The main notion of the composition diagram is the page, defined as an autonomous block of screen. Each screen in the navigation diagram is mapped into several pages in the composition diagram. The tool may provide links from pages to screens and vice versa to show their relationship. A page is modeled in the composition diagram as a class stereotyped *«page»*. A page may play the role of container for other pages. Nested pages are modeled as aggregated classes. The page has elements like buttons, links and input fields. They are modeled as attributes of the corresponding page. The *«form»* stereotype is a child of *«page»*. It is used to model HTML forms. The navigable association between source and target pages is stereotyped *«link»*. Each *«link»* has a tag context with expression as value.