# Towards a UML Profile for Modeling WAP Applications

**Ricardo Soto De Giorgis**
**Escuela de Ingeniería Informática**
**Pontificia Universidad Católica de Valparaíso**
**Valparaíso, Chile**
**and**
**Mauricio Cámara Joui**
**Escuela de Ingeniería Informática**
**Pontificia Universidad Católica de Valparaíso**
**Valparaíso, Chile**

## ABSTRACT

UML (Unified Modeling Language) is one of the most used languages to specify and document informatics applications. However, UML is a general-purpose language, so it often lacks of elements to model and represent concrete concepts of specifics domains. As a solution, OMG (Object Management Group) has created the profiles, a mechanism to extend the syntax and semantics of UML to express more specific concepts of certain application domains.

In this work we present a UML profile for modeling WAP (Wireless Applications Protocol) applications. The main goal of the proposed profile is to extend UML to provide specifics elements (labeled classes, stereotypes, tagged values and constraints) that allow software developers to model WAP applications.

The expressiveness of the UML diagrams allows modeling important stages of the process of common applications; nevertheless, the modeling process of WAP applications is still a too specific domain that can be hardly dealt with in its entirety without extending the language. In the process exists navigational, design and construction issues that cannot be modeled using the traditional elements of UML. However, by using the specific elements created by the proposed profile, these issues can be completely solved, and even a greater expressiveness can be obtained.

**Keywords:** Software Engineering, UML, UML Profiles, Metamodels, WAP.

## 1. INTRODUCTION

The UML, one of the most used languages to specify and document informatics applications, is a general-purpose language, reason why it is possible to use it to specify different systems and application domains (business world, aeronautical and academic issues, etc.). This UML feature provides a great flexibility at the time systems are modeled. Nevertheless, sometimes it is better to use a more specific modeling language to represent, in a precise way, characteristics of certain domains. This happens, i.e. when the UML syntax does not allow expressing the specific concepts of the domain, or when the software developer needs to restrict and specialize UML constructors, that usually are both too generic and numerous. This lack of expressiveness also exists when we need to model WAP applications, since the traditional elements of UML do not allow supporting important navigational, design and construction issues of the development process. As a solution to this problematic, a new UML profile is proposed, which gives specific elements for the WAP domain.

This paper is organized as follows: Section 2 introduces the MOF (Meta-Object Facility)[1] and meta-modeling activities. Section 3 presents step by step the construction of the proposed UML profile. In Section 4 the profile is applied to a study case. Finally, section 5 gives the conclusions and future research.

## 2. METAMODELS AND MOF

The metamodeling is a mechanism that allows constructing modeling languages, such as UML or CWM (Common Warehouse Metamodel) [2]. The metamodel of a modeling language is an exact definition of all its elements (classes, associations, stereotypes, etc.) by means of concepts and rules of certain metalanguage. In order to understand better the concepts related to metalanguages and metamodels, OMG proposes a four-layer metamodeling architecture [3][4], oriented to standardize all the concepts of modeling, from the most abstract models to the metamodels.

### The four-layer metamodeling architecture

The levels defined by OMG are called M0, M1, M2 and M3:

M0 level (instances): M0 models the real system, and their elements are the data of an information system; i.e. “John Cook”, who lives in “Boulevard Gardens 4934”.

M1 level (system model): The elements of the M1 level are the models of the data. i.e., classes like “person”, “car”, attributes like “name”, “address”, and relations like “selling”, “buying”, etc.

M2 level (metamodel): The elements of the M2 level are the modeling languages, for example, UML. In this case the concepts of the M2 level could be,

"class", "attribute" or "association".

M3 level (meta-metamodel): In this level appears the MOF. This is an abstract language created to define modeling languages such as UML or CWM.

## 3. CONSTRUCTION OF THE PROPOSED UML PROFILE

Before building a WAP application, it is important to use a modeling language that specifies and documents the system in a very precise way. The traditional UML diagrams allow modeling important stages of the process; however, there exists design, navigation and construction issues that can be hardly supported without extending the UML [5]. To solve this, we propose a UML profile for the development of WAP applications. For the creation of this profile, a five stages procedure will be used [6] [7] [8] [9], and it is necessary to pay special attention to the elements that compose the profile, such as stereotypes, tagged values and constraints.
Stereotypes: They are defined by a name and a group of metamodel elements. The stereotypes represent the new features added to the UML metamodel for extending the language.
Tagged Values: They are meta-attributes that are associated to a meta-class of the metamodel extended by a profile. Each tagged value has a name and a type, and it is associated to a certain stereotype.
Constraints: The constraints are associated to the stereotypes, and they impose conditions on the metamodels elements that have been stereotyped. The constraints are written either in natural language or OCL (Object Constraint Language) [10]. OCL is the language defined by OMG as the standard for transcribing constraints and queries.
Next, the five steps to obtain the UML profile:

**Step 1: Building the metamodel**
The first step to create the profile is to define the metamodel of the application domain. In this case the application domain to be represented is both the process of modeling WAP applications, and the elements that will be part of the metamodel, used for programming and implementing WAP applications (cards, decks, forms, links, do, go, etc.)[11][12]. Figure 1 shows the metamodel obtained.

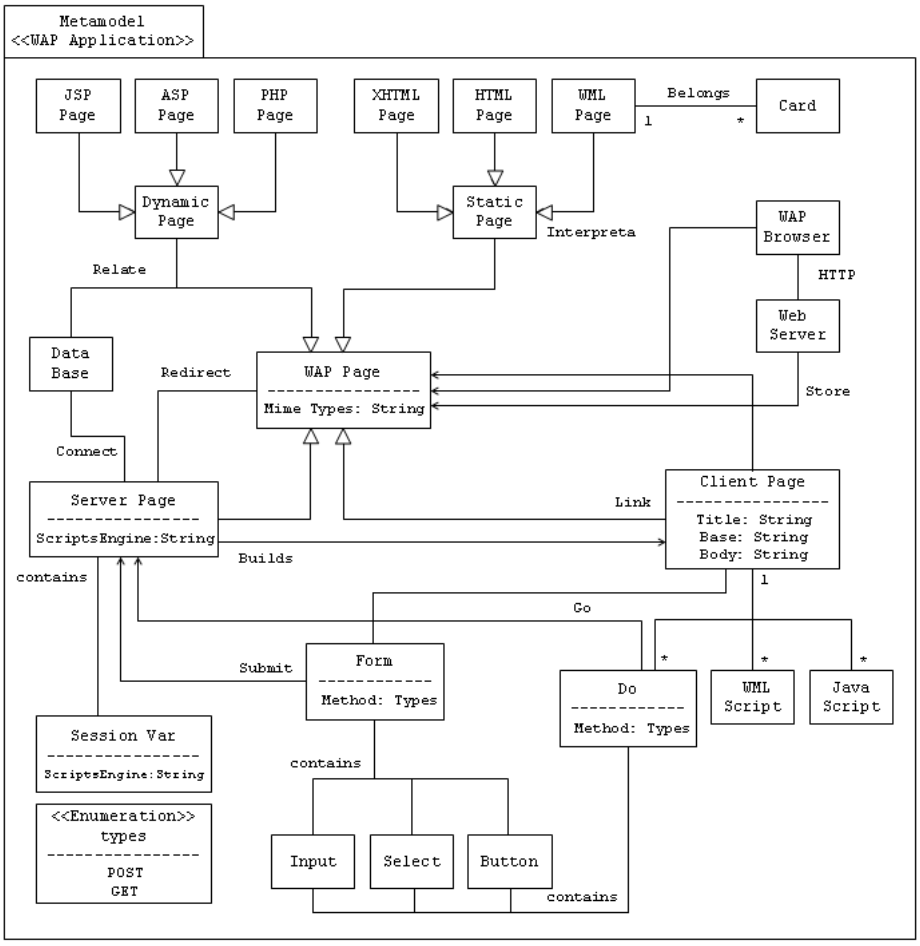


**Fig. 1.** Metamodel

The WAP page is the main element of WAP applications (central class of the metamodel). Static WAP Pages only display information; Dynamic WAP Pages avoid interaction with the end-user (fill forms, login, passwords, read customized data, etc.).

Focusing in the source code of WAP applications, Dynamic WAP Pages can be programmed in JSP, ASP or PHP, and Static WAP Pages can be written in XHTML, WML, JSP, ASP o PHP.

WAP Pages can contain scripts that run in the client layer (Client Pages), or scripts that run in the server layer (Server Pages)[13].

"For" and "Do" Classes get information given by the end-user. JavaScripts and WMLScripts run in the WAP Browser. Finally session vars are stored in the server layer.

**Step 2: Defining the profile**

To define the profile, a stereotype for each metamodel element (WAP Page, Form, Do, JSP Page, HTML Page, etc.) included in the profile must be created. To maintain the relationship between step 1 and 2, stereotypes must have the same name of metamodel elements.
For example, in Figure 2, some classes were selected, creating the corresponding stereotype for each one.

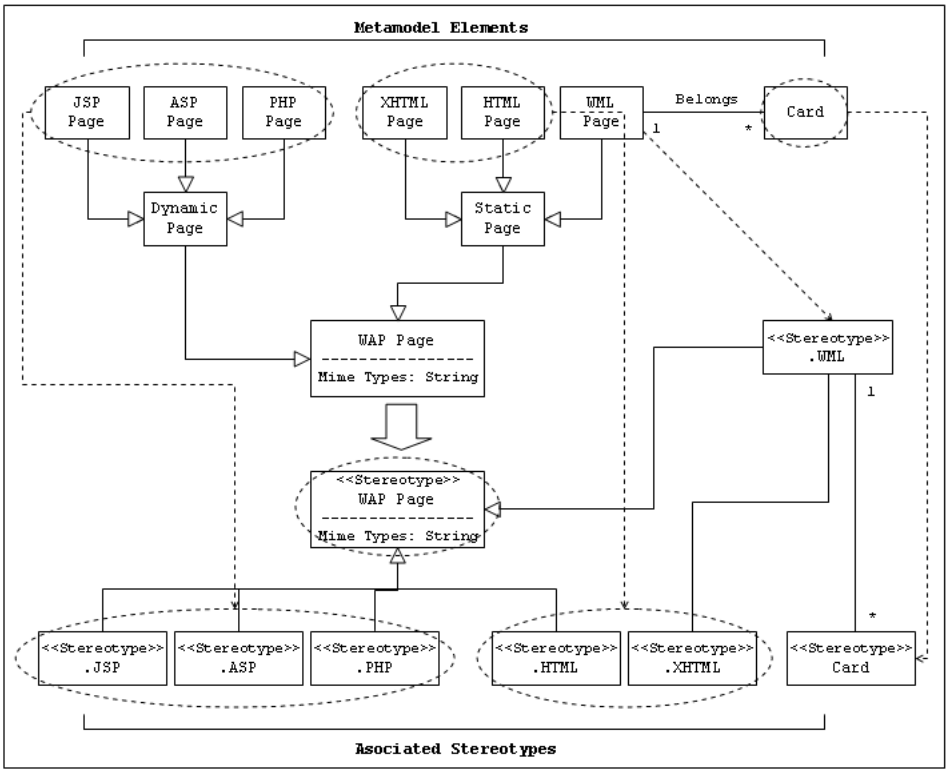


**Fig. 2.** Relationship Metamodel-Stereotypes

**Step 3: Identifying elements to be extended**

Step 3 consists in identifying the UML elements that will be extended with each stereotype. Examples of such elements are: classes, associations, attributes, packages, etc. Figure 3 shows the stereotypes associated to the UML structure called "Class".

**Step 4: Defining Tagged Values**

In this step, tagged values must be added for each metamodel attribute in the profile. Tagged values are additional attributes that are associated to the profile. Figure 4 shows a metamodel class that has an attribute which will be added to the profile like a tagged value.

**Step 5: Defining Constraints**

In the last step, the constraints of the profile must be defined. Constraints will be defined in OCL, language defined by OMG as the standard for transcribing constraints and queries. Next, three constraints of the profile will be showed.
Constraint 1: "Form" class cannot have associated functions. Any operation that interacts with this element must be defined in the "WAP Page" class that contains the "Form" class.

```
Context UML::InfrastructureLibrary::Core::
Constructs::Class Inv:
self.isStereotiped("Form") implies
self.Operations -> isEmpty()
```

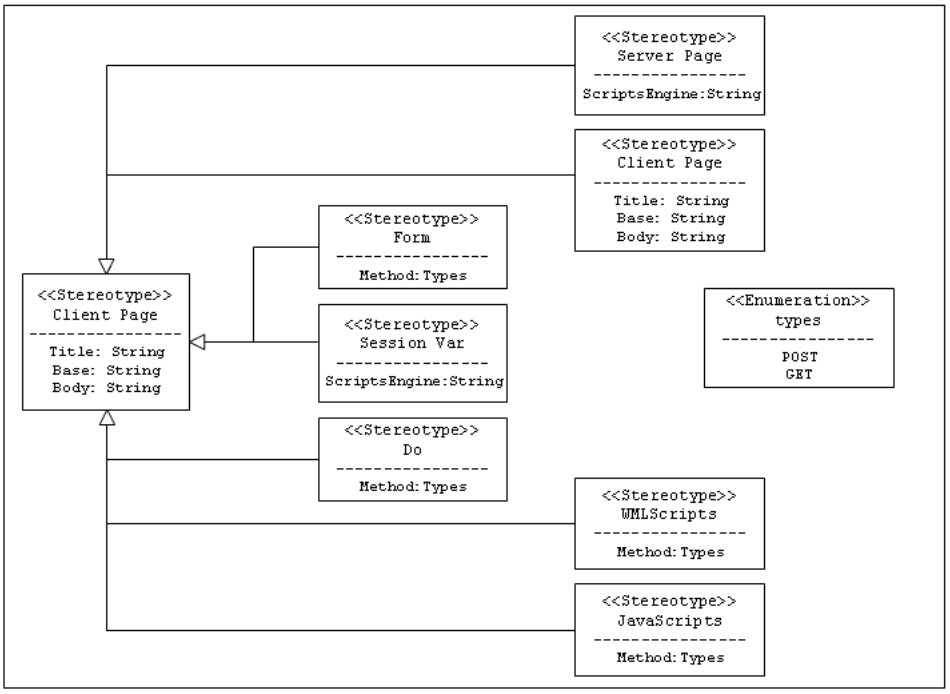


**Fig. 3.** UML elements to extend

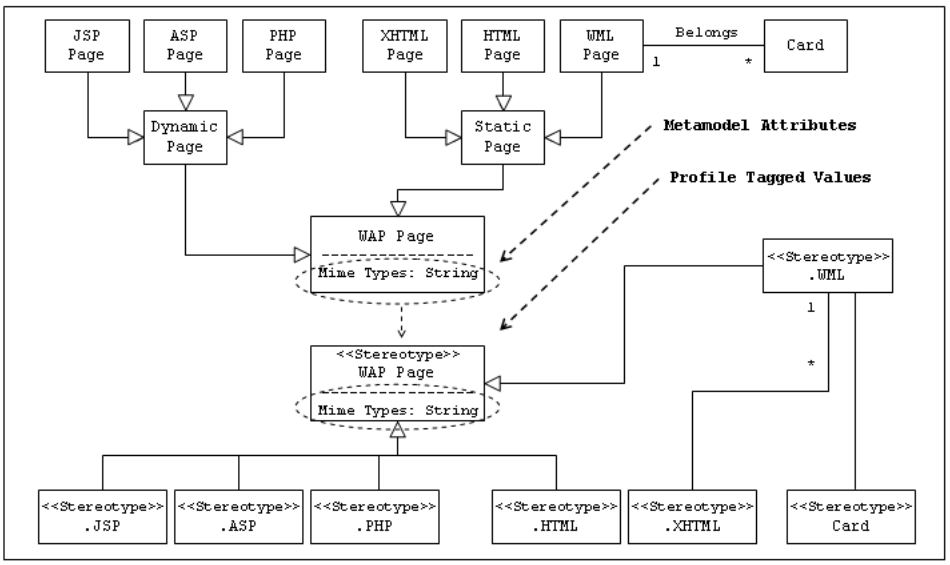


**Fig. 4.** Profile Tagged Values

Constraint 2: Class "Form" cannot have associated functions. Any operation that interacts with this element must be defined in the "WAP Page" class that contains the "Do" class.

```
Context UML::InfrastructureLibrary::Core::
Constructs::Class Inv:
self.isStereotiped("Do") implies
self.Operations -> isEmpty()
```

Constraint 3: The attribute "ScriptsEngine" of the "session var" class must be equal to the "ScriptsEngine" of the related class "Server Page".

```
Context UML::InfrastructureLibrary::Core::
Constructs::Class Inv:
self.isStereotiped("Session Var") implies
(self.isStereotiped.("Server Page").
ScriptsEngine = self.isStereotiped.
("Session Var").ScriptsEngine)
```

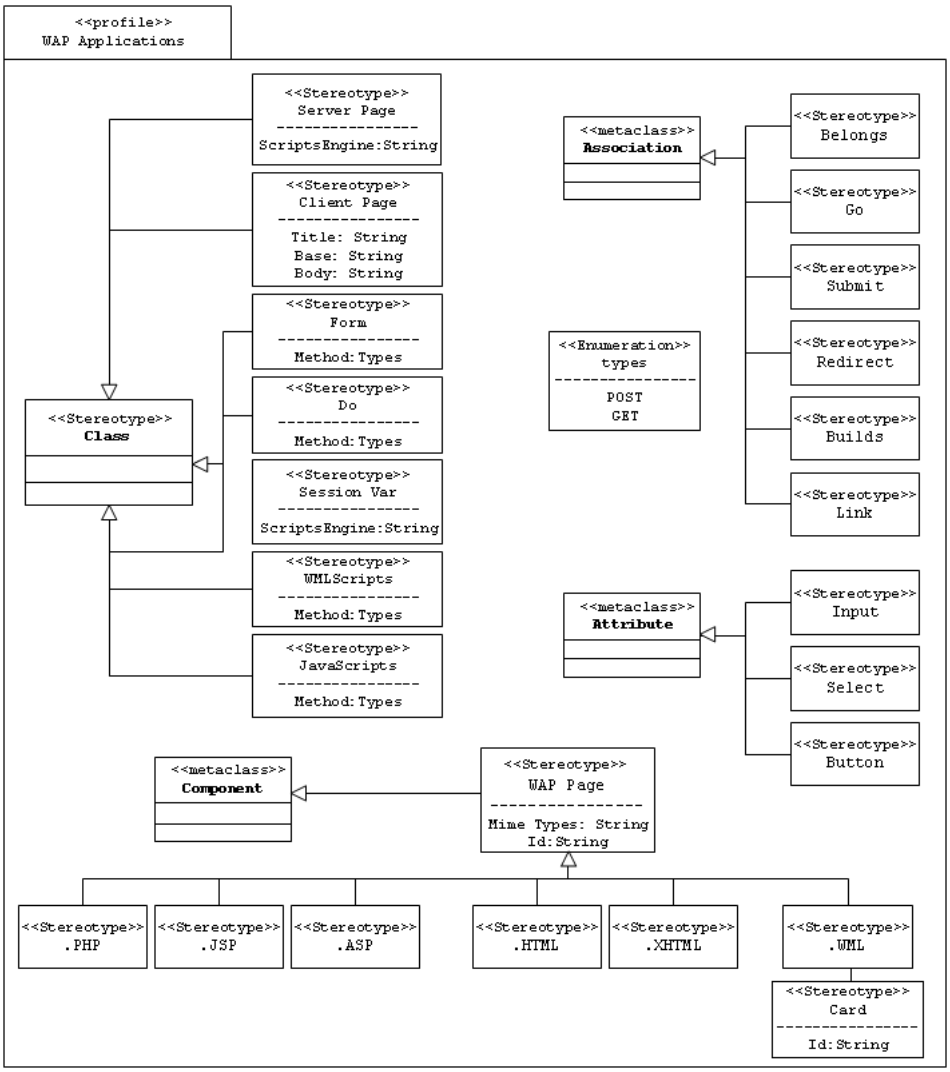


**Fig. 5.** UML Profile obtained

Figure 6 shows the UML profile finished. All the metamodel elements added to UML appear in the profile; each element appears with its respective tagged value. In addition, the UML structures that will be extended are specified.

## 4. USING THE PROFILE

The inclusion of new characteristics to the UML models is one of the main goals that profiles must achieve. In order to present the new characteristics, and the way a profile is applied, we will use the proposed profile by applying it in a real system. The real system that was used will be an online bookstore. This application has two great functionalities. The first is doing a book sale through a cell phone, and the second is maintaining the site (by adding, deleting and modifying books).
The following diagrams will show the extension made by the proposed profile, and its use for the study case.

Component Diagram: The component diagram is used for modeling the static view of a system. It shows both the organization and the dependencies between a set of components. The component diagram (Fig. 6) shows WAP pages, and the associations between the different pages of the administration module. Obvious associations have been omitted for a better visualization of the diagram.

In the system, each page is represented by a component. The design classes that define the WAP page behavior are inside the components, and each one has been obtained from the profile. Figure 6 shows the previously defined stereotypes (card, PHP Page, WML Page).

The following model shows classes, relations and attributes defined in the profile and used in the study case. In the design classes, diagram WAP pages can be either client pages or server pages. Server pages need server access, and client pages only need local access. Another element that appears in the model

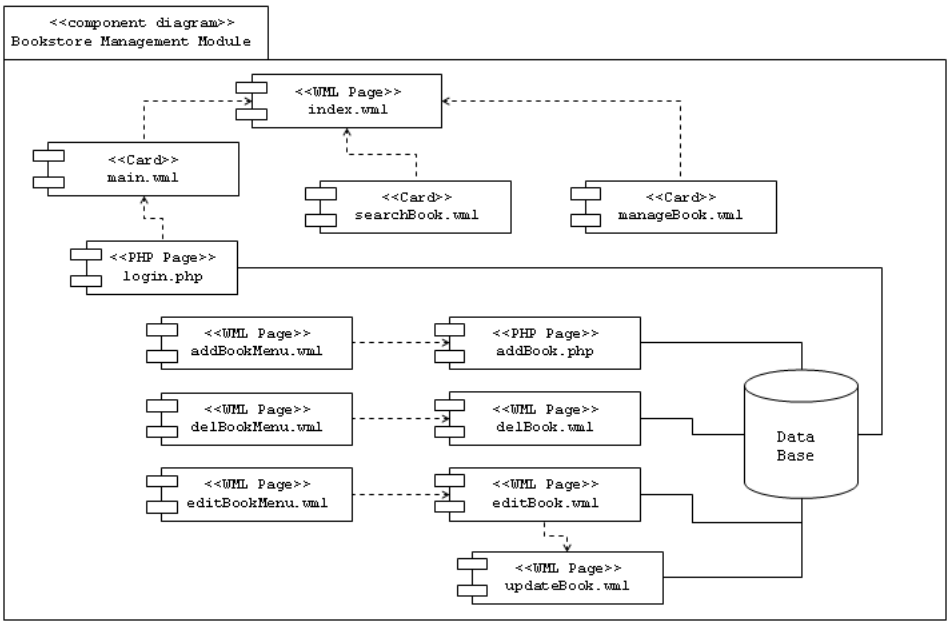


**Fig. 6.** Component Diagram

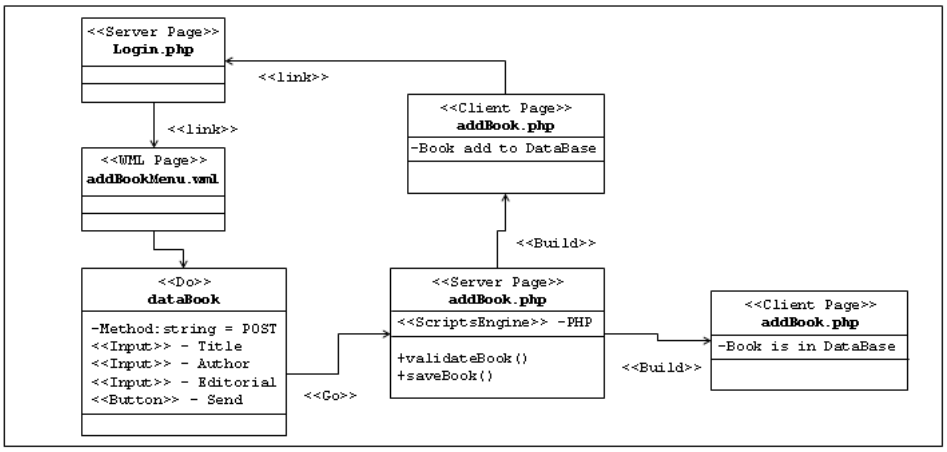


**Fig. 7.** Design Class Diagram

is the stereotype "Do", which indicates the existence of a form inside a client page. For relationships between each WAP page, stereotypes "Go", "Link", "Submit", "Redirect" can be used. Through stereotypes names can be understood what each one represents.

Figure 7 shows the interaction between different classes of the profile. All these classes are necessary for adding a new book to the Data Base. The complete modeling process of the WAP site implies having a design class diagram for each one of the site pages; each site page must be incorporated as a component in the component diagram.

The proposed profile allows visualizing the elements that each page has clearly. It also allows understanding easily both the interaction and behavior of all the participating elements in the system.

## 5. CONCLUSION AND FUTURE RESEARCH

The creation of profiles and UML extensions is necessary when the traditional models do not give the expressiveness required to represent the specific characteristics of particular domains, as it happens with the WAP sites modeling process.

In this work, a UML profile for modeling WAP applications was presented, which was obtained from a five steps process. This process gave -as a result - a UML profile, constituted essentially by the most used elements to design and implement WAP sites. Then, the profile was applied in a study case. The profile allowed modeling in a clear, simple and expressive way the aspects of navigation, design and construction that could be hardly modeled with the traditional elements of UML. In addition, we offer

a new modeling tool that allows software developers, by its expressiveness, to design WAP applications without omitting fundamental aspects from design and implementation stages. It is very useful for the software developers to have procedures, specific constraints and precise nomenclature for the domain that is intended to be modeled. Following this way, diagrams have greater definition and expressiveness, obtaining a reduction both in time and resources used in the modeling steps.
The Profile proposed is the beginning of a complete study of the development process of WAP sites, which will be applied in the development of different types of systems for mobile equipment. The next goal is to get more experience to build patterns, and a CASE tool that allows simplifying even more the entire building process of WAP software.

## 6. REFERENCES

[1] Object Management Group: MOF 2.0 Core Specification (2004), http://www.omg.org/cgi-bin/doc?ptc/2004-10-15 http://www.omg.org/cgi-bin/doc?ptc/2004-10-15

[2] Object Management Group: CWM 1.1 (2002) http://www.omg.org/technology/documents/formal/cwm.htm

[3] Object Management Group: UML 2.0 Infrastructure Specification (2003), http://www.omg.org/cgi-bin/doc?ptc/2003-09-15.

[4] Object Management Group: UML 2.0 Superstructure Specification, (2004) http://www.omg.org/cgi-bin/doc?ptc/2004-10-02.

[5] Soto Ricardo & Rodríguez Nibaldo, "New UML 2.0 based models to design WAP Applications". Seventh International Conference on UML Modeling Languages and Applications, UML 2004. Lisboa, Portugal.

[6] Blankenhorn Kai & Jeckle Mario, "A UML Profile for GUI Layout", NODe 2004, LNCS 3263, pp.110-121, 2004 Springer-Verlag Berlin Heidelberg 2004.

[7] L. Fuentes, J. M. Troya, A. Vallecillo. "Using UML Profiles for Documenting Web-based Application FrameworkS". Annals of Software Engineering, Vol. 13, pp. 249-264, June 2002.

[8] Vincenzo Grassi, Raffaela Mirandola and Antonino Sabetta, "A UML Profile to Model Mobile Systems". Seventh International Conference on UML Modeling Languages and Applications, UML 2004. Lisboa, Portugal.

[9] Tewfik Ziadi, Loïc Héloüet, and Jean-Marc Jézéquel, "Towards a UML Profile for Software Product Lines", PFE 2003, LNCS 3014, pp. 129-139, 2004 Springer-Verlag Berlin Heidelberg 2004.

[10] Object Management Group: UML 2.0 OCL (Object Constraint Language Specification) (2003), http://www.omg.org/cgi-bin/doc?ptc/2003-10-14.

[11] Open Mobile Alliance: Wireless Markup Language v2.0 http://www.openmobilealliance.org/tech/affiliates/wap/wap-238-wml-20010911-a.pdf

[12] Open Mobile Alliance: Technical Section http://www.openmobilealliance.org/tech/affiliates/index.html

[13] J. Conallen. "Building Web Applications with UML", Addison Wesley, 1999