Construction of a Tool for Dynamic WAP Applications Editing

E. Carrillo¹, J.J. Samper², J.J Martínez Durá², V.Ramón Tomás²

¹Facultad de Ingeniería de Sistemas
Universidad Autónoma de Bucaramanga
COLOMBIA

²Instituto de Robótica
Universidad de Valencia
SPAIN

Abstract: In this paper, a summary of the work undertaken in the construction of a grammar set and a finite automata which give support to the development of an editor/compiler of dynamic WAP pages is presented. The compiler provided support for WML and WMLScript, as well as the possibility of using different technologies for interaction with databases on the server side, such as ASP (Active Server Pages), JSP (Java Server Pages), and PHP (Hypertext Processor), allowing the utilization of the defined elements in recent WML 2.0 specification.

KeyWords: WML, XML, ASP, PHP, JSP

1 Introduction

Internet Markup Languages have evolved from the SGML (Standard Generalized Markup Language), developed in the 60’s, to the introduction of HTML (Hypertext Standard Generalized Markup Language) at the end of the 80’s. XML (Extensible Markup Language) was developed as a subset of SGML, with the aim of facilitating the interaction of information contained in different portals, and with the possibility of separating content from presentation.

XML has its own set of standards associated with both the organization of data and presentation of same data. The documents should be well formed and comply with the definitions and rules of validity defined in those DTDs (Document Type Definitions). The format with which the data is visualized may be defined via XSL (Extensible Stylesheet Language).

On the other side, WAP (Wireless Application Protocol) defines a series of standard components which allows the communication between mobile terminals and information servers which include support for the WML (Wireless Markup Language). WML is similar to HTML but optimized for the inherent limitations found in mobile devices. For the client-side interactions there is a script language, denominated WMLScript.

The recent WML 2.0 specification has been designed in order to extend the syntax used in XHTML (Extensible Hipertext Markup Language), which, in turn, has arised as a proposed reformulation of HTML based on XML (Extensible Markup Language).

WML 2.0 attempts to achieve the following goals:

- Backward compatibility.
- Convergence with actual standards and the evolution of the internet.
- Optimization for the access realized from small devices.
- Allowing the creation of interfaces adaptable to different user agents.
- Internationalization of the architecture.

This work outlines the steps undertaken in the development of a compiler which facilitates the process of a correct WML code generation, allowing for the creation of dynamic and interactive pages with support for the standard WML 2.0.
2 Development

A lexical, syntactic and semantic WML code was designed, this allows for error detection in the edition with a detailed description which is important in the revision of documents developed manually as well as in the revision of documents previously created.

Additionally a translator was developed that allows the transformation of written pages in older versions of WML and WML 2.0. The traslator is useful in the creation of a grammar and finite automata with which to analyze the code.

The final software interface facilitated the editing of WML code and the inclusion of the tags presented in the technologies for the integration of Web applications and databases: ASP, JSP and PHP. The software prototype has the ability to simultaneously edit multiple files. [2]

For the prototype development, object orientation methodologies were used and the design was specified with UML (Unified Modeling Language) using C++ as the program language. [1,3]

The main classes implemented were:
- Tsearcher, is used for analysis and a translation of the code.
- TAnalizer, for the lexical, syntactic and semantic analysis of the tags.
- TGlobalAnalizer, which directs the analysis of the code in global format.
- TConverter class, which controls the conversion of exiting WML code into WML 2.0 code.

The form in which the tags were defined made it impossible to find a regular grammar which could recognize a tag (the attributes may be found in any order and should not be repeated). This was due to the fact that they could not “remember” what transpired previously.

Additionally, a grammar independent of context would be able to recognize them.

3 Grammar for the analysis of the WML code

A grammar set is a finite group of exact rules for the construction of sentences of a given language, each sentence of that given language may be constructed in a finite number of steps. If, starting from the initial production, we apply a series of steps by which we construct a chain, then this will pertain to the defined language for the grammar set.

In order to recognize the WML language we have defined three grammar sets, one for each of the existing versions. The tool that has been developed allows for the assistance of code generation by means of access to the graphic elements which characterize a WML interface.

4 Conclusion

This article describes the principal elements used in the design and implementation of a grammar set. Also a lexical, syntactic and semantic analyzer present in an integrated environment for the analysis of dynamic WAP pages based on WML 2.0 has been characterized.

References:


This project has been presented with the sponsor of the Instituto de Robotica and International Relations Direction of Universidad de Valencia, and Colciencias in Colombia.