

EXTENSION OF UML(OPEN) ABSTRACT MODELING TO DEVELOP WEB APPLICATION SUPPORTING PLATFORM AND DEVICE INDEPENDENCE

Manuj Darbari, Assistant Professor, Department of Information Technology, BBDNITM, A-649, Indira Nagar, Lucknow, India.
manujuma@rediffmail.com

Bhaskar Karn, Coordinator, Department of Information Science, B.I.T, Mesra Ranchi, India.

Prof.Dr.P.K. Bansal, Principal, Malout Institute of Management and Information Technology(MIMIT), Malout-Punjab, India-152107,

Rajesh Goel, Associate Professor and Head, Electronics and Communication Engineering Department, Ambala college of engineering and applied Research, Post Office- Sambalkha, Ambala Cantt-133101-Haryana. India

ABSTRACT

Traffic modeling has become a major issue for many cities throughout the world. There are number of simulation tools and websites available which can depict the urban Traffic situations. Our work extends the concept of Traffic Website Design which should be independent of platform as well as device.

Keywords: OPEN(Object Process Environment Notation).

1 INTRODUCTION

In this paper we will be focusing on a web site design specially designated for urban Traffic commuters as well Traffic Management official. To begin with we are focusing integration of platform independent modeling being UML. The concepts that plain UML prescribes for specifying communication between application parts (Objects and components) imply an abstract platform that is based on request response model and on message passing. In UML 2.0 meta-model, *behavioral* operations represent the capability of classifier to receive and to respond to requests. Requests are sent when objects exercise signal instances, which are sent asynchronously by other objects when these execute send signal actions and Broad cast signal actions.

The specialization of UML[8] for defining abstract platform characteristics can be made more manageable and clearly defined through the use of UML profiles. Profiles are language extensions consisting of metamodel elements[1,3] that specialise elements of a reference model. The specialised elements can be given specific semantic variation points.

To further define an abstract platform model, library packages are accommodated in UML 2.0. The abstract platform model library package can be imported by platform Independent Model of the application. Then we will be creating a dependency between the package where platform independence model is defined and the model library package

where the abstract platform is defined. In order to design complex abstract platform, we can use UML 2.0 composite structure to break up a complex design into smaller pieces. Generally for urban traffic modeling we have used state-machine and activity diagrams which encapsulates classifiers to define their behaviour. Since the behaviour of the abstract platform is also described in UML, it may be necessary to continue the explicit and the implicit abstract platform definition approaches. In the next half of the paper we will be concentrating of device independent website design of urban traffic which can provide the information so the common user about traffic conditions on his mobiles, PDA and telephones equipped with speech recognition software's.

During late 90's it was general conception that website means it has to be supported by browser of a personnel computer or a laptop. It is very well shown by Ginage and Murugesan in their paper on web engineering[2] that by the help of XML/XDL we can achieve browser loss web providing in basic flexible infrastructure to independently define content and presentation. We will be using UML modeling technique to develop the device independent website.

2 PLATFORM INDEPENDENT WEB MODELING

Consider an example of Traffic information website in which a user can enquire about the latest Traffic conditions on a particular route through

online query mechanisms with system administration. We can call this abstract platform the Traffic Abstract Platform. In order to define the composition of the Traffic binding object with the application, we use reachable exchange of asynchronous signal exchange with the implicit approach, by defining a UML profile. We can show the relationship between platform independent modeling of the application and the abstract platform.

The main of modeling elements used in Figure 1 are class and association. There are certain additional features to enhance the basic conceptual model.

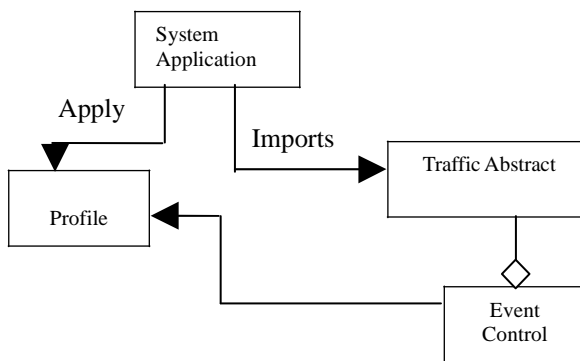


Fig. 1 : Conceptual model of the online Traffic Information

2.1 Transport Abstract Platform

It consists of various signals and interface. It consists of a new user joining the website and existing user leaving the website.

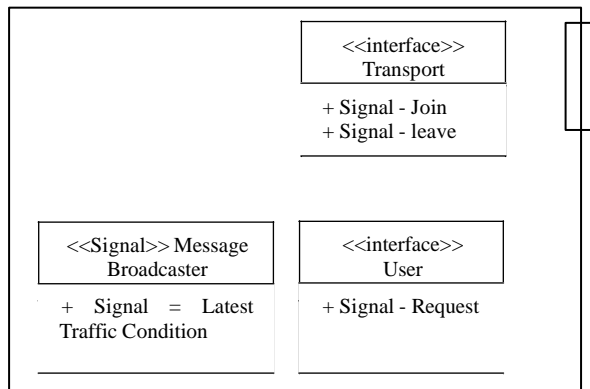


Fig 2. Transport Abstract Platform Components

The interaction point that corresponds to port is of Type Transportation query Port. The query port handles the signals Join and Leave and delegates the handling of signals Message Reg. to the system.

Administrator :

Fig. 3 shows the internal components of Transport Abstract Platform. *Transport Component* instances exchange message signals among each other and *message broadcaster* with interaction.

2.2 Discussion of the above model

From the above UML model we are able to define the behavioural aspects of components taking part in a web model designing. Secondly the if a designer is following some language then he or she may favour directly some platforms over the other. The UML composite structure proposed in Fig. 2 and Fig. 3. The composite structure of transportation website can be created or destroyed along with components to which they are attached if the port connecting to them is destroyed first. This implies that if we want to model our unbounded number of distinct users may use the component through ports : We can further extend this abstract modeling for Device Independent modeling using XML and UML concepts.

3 DEVICE INDEPENDENT ABSTRACT MODELING

In order to explain this let us extend our example of Urban Traffic Modeling where a normal user sends an SMS from his mobile to get the latest information about city bus plying between a fixed origin-destination. The Device independent model consists of XML compiler that can interpret the language and several runtime components that are configured and deployed on the web server.

The components consists of content design how the web application should look like etc. Then comes the implementation stage which is generally done in My XML language which supports XSL stylesheets which is independent of the device.

The deployment part consists of an application logic and web services which are to be extended to various level of users the administrator or used.

The final stage consumer of modification in which the necessary changes can be implemented.

In this section we an approach to model, implement and compose web services[6] into website with device independence. The framework which we are going to model provides support for the separation of layout, content and application logic in websites and automatically generates web since support for Browser less access to the content and functionality.

Consider a detailed device Independent web Model (DIWM) which shows the relationship between various devices and its access to the common transportation server also known as Transportation as transportation Abstract Platform, that supports any query from the user having any device. It applies the features of My XML which provides a framework for loading the content of the website on any device.

The Event Abstract Platform decides the type of event which is to be transported and also supports the session management features.

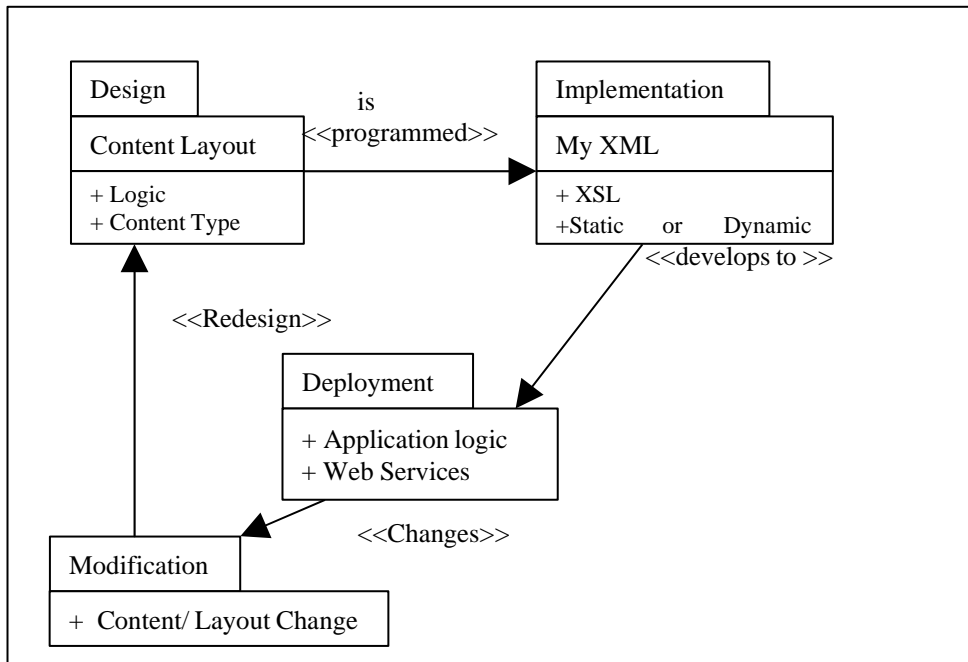


Fig. 3 : Device independent Web Design Components

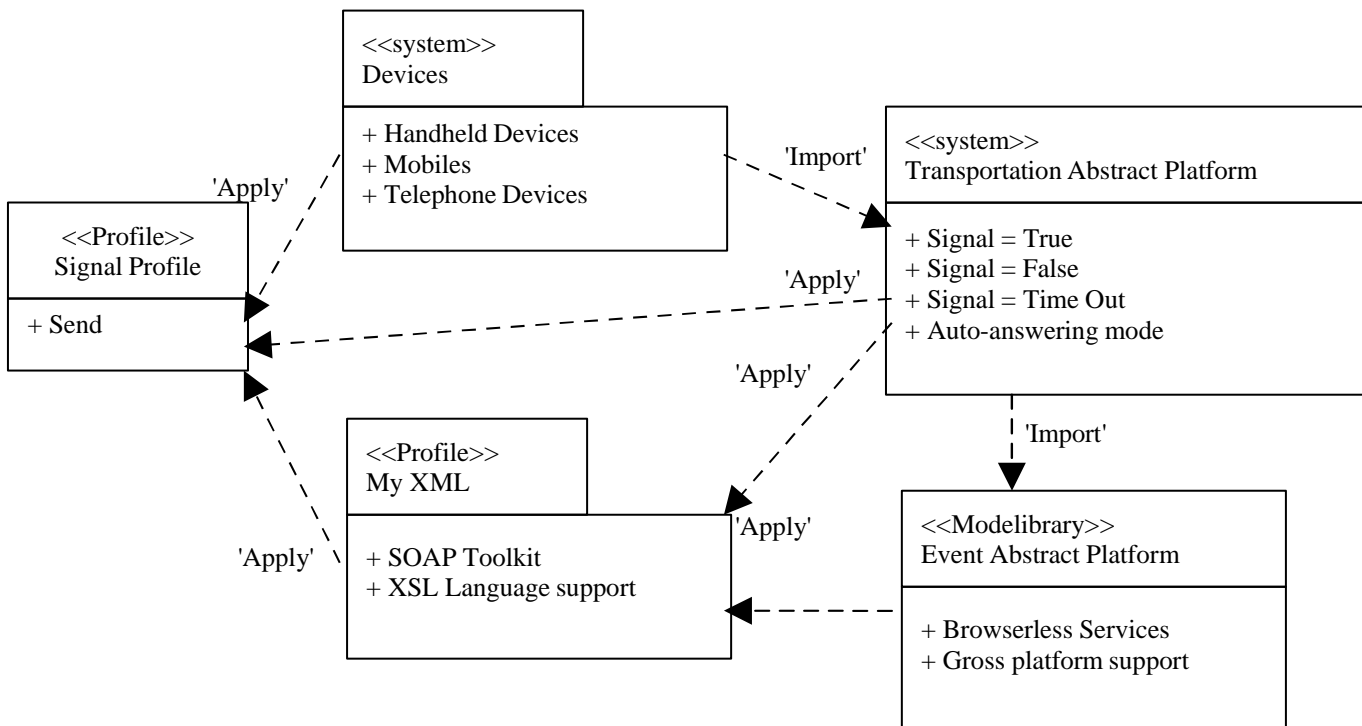


Fig. 4 : Relationship between DIWM and Abstract Platform

3.1 General Description of DIWM components

3.1.1 Devices

Devices means any device which has the accessibility of web content it can have windows and Java platform

3.1.2 Signal

The signal profile specializes the exchange of asynchronous message sending between the two systems

Transport Abstract platforms : It consists of interfaces and signals . It is same as explained in the previous section with an additional feature of

autoreply mode about the latest traffic conditions based on origin-destination

3.1.3 Event Abstract Platform

It deals with providing a common platform interface with features like browser less content generation and cross platform interface

3.1.4 My XMK Profile

It contains SOAP toolkit[7] which is a simple object Access protocol. In this case it is used with combination of HTTP. It provides the framework which supports for the separation of layout, content and application logic in websites and automatically generates web service support for the browser-less access to the content and functionality. My XML language compiler integrates the layout and generates static content embedded in HTML and XML. For every page that is created with MY XM compiler , a corresponding Apache SOAP deployment descriptor in XML is generated service is always of the form <Generated Page name> SOAP

3.4 Discussion on the above model

The abstract model of DIWM is an upcoming concept. We have used the concept of UML for integration and composition of web services and then later on integrating and composing of web services into websites.

We have extended the concept of abstract modeling in viewing the web service as functionality that complements the typical functionality provided by a website and focus on the web engineering problem of using web services.

4 CONCLUSION AND EXTENSION

In this paper we have fully exploited the concept of EDOC component collaboration architecture in which application part interaction are decomposed into a synchronous messages that are exchanged through various ports.

In both the examples (PIM and DIWM) we shown the concept of abstract platform in standard UML, through both implicit and explicit abstract platform defination. Explicit abstract platform definition is comparable to the definition of

connectors in ADLs (Architecture Description Language) can also be implemented in this approach for composite structures both for defining abstract platform from external and internal prospective.

The limitation of behavioral aspect of Abstract components cannot be shown in our proposed model which could be the picture extension of the work. The further extension could be modularisation of abstract platform definitions where the designer should be able to compose an abstract platform from the abstract platform modules developed previously.

REFERENCES

- [1] Lucknow, D.J., Man : Specification and Analysis of System Architecture using RAPID, IEEE Transaction on software engineering, Vol. 21, No. 4, 1995.
- [2] Lucktiman, D.: An Event Based Architecture Ddefinition language,. IEEE Transaction on software engineering vol. 21, No. 9, 1995.
- [3] Allen, R : A formal Basis for Architectural Connection, ACM Transaction on software engineering and methodology, Vol 6, No. 3, 1997.
- [4] C. Abatleir, A model-Driven Architecture Design space modeling, Technical report, Ankara, Turkey, June 2003.
- [5] J. Warmer, W.Bast, MDA Explained, The model-driven Architecture : practice and Promise, Addition-Wesley, 2003.
- [6] MDG Guide Version 1.0 edited by Jo aquin miller and Jishnu Mukerji.
- [7] E.D. Willink : UMLX : A graphical language for MDA : In proc. of model driven Architecture : Foundations and applications, pp 13-24, University of Twente, Netherlands, 2003.
- [8] G. Booch, J, Rumbaugh : The unified modeling language user guide, Addison -Wesley, 1999.