ABSTRACT

Asserting that it’s absolutely necessary to turn to an organic design phase also for hypermedia and most of all Web applications, we intend to verify the effectiveness of a methodology like W2000 [1] [2], based upon the well-known HDM model [3] but proposing a full conceptual design, by the development of a pilot application. We also want to check if it’s practicable to build a framework, in order to guide the developer along all the development, making the phases as automatic as possible.

In this paper we describe the reasons and the goals of our work, during its working out, as well as the results we already got. We expose the guidelines that helped us making choices and the problems that rose and are still under analysis or made us stop, modify something or roll everything back and start again.

KEY WORDS

Web applications, Methodologies, Hypermedia, W2000, Microsoft .net

1. INTRODUCTION

The production of hypermedia application in general, and of web application in particular, is a very complex process, but often it’s not paid enough attention for it.

The availability of commercial products, able to help the developer in the direct production of what the Market demands, and the particular features of the applications themselves made the design phase less considered, if not judged useless. The Market refused the firm design methodologies of the Software Engineering, considering them not technically suitable for the new media, but also very expensive in resources, time in particular.

As long as people dealt with simple presentation application, small in size, problems didn’t rise in a worrying way. Nevertheless, as they grew in dimensions and complexity, the applications started to have features similar to the ones the traditional software had, combining data-intensive and transactional matters with purely navigational ones.

The consequences of such a situation are showing out and forcing the whole IT community to a reflection. Is therefore plain that new methodologies are needed and these may arise from the traditional ones, but have to be take in account the new requirements.

At the HOC (Politecnico di Milano) and the the SetLab (University of Lecce) has been developed W2000, which tries to reach this goal, allowing an accurate conceptual modeling of the applications, before the implementation phase.

Around W2000, now, we want to build a complete development framework, able to put the methodology nearer to the implementation, so that the precious design effort can quickly become a real application, as much as possible in an automatic way.

W2000

To better understand the topics we are going to talk about is necessary to know the W2000 methodology, at least in its fundamentals.

It assumes that it’s unavoidable a good distinction between the different aspects of the application which have to be observed during the development, in order to make the development itself a structured and easy controllable process.

After the indispensable phase of the Requirement Analysis, made following a goal oriented approach, the methodology suggests a sequence of steps briefly summarizable in:

- **Information Design**: it’s goal is to describe the information the application is going to deal with, giving it a structured organization. Important features of this phase are that, during the construction of the information structure, the user point of view [4] is held as fundamental.
- **Navigation Design**: it makes clearer the most important aspect of hypermedia applications, reconsidering the information and its organization more typically from the viewpoint of its fruition and defining the navigational paths the user can follow.
The features of the application were suitable to fully verify completeness of the methodological support, but also to give a good sense of the problems concerning the pure implementation.  
Our attention, therefore, will dwell upon the following elements:

- to verify which logical steps must be taken to obtain, from the conceptual model, a true implementation.
- to verify the support that the modeling gives, first to the designer of the software architecture and then to the developer.

The architecture
A first choice we had to make has been regarding the architecture at the bottom of the application. We found out two approaches, very far from each other, but good in the same way:

a) We could go directly on the develop of the specific application, using the existing market standards, then a multi level architecture, immediately recognizing the presentation, business and data levels, each one with its own features and functionalities.

b) We could give the specific application a structure that could be already suitable for the framework we intend to base on the methodology., and then produce an engine able to automatically prototype not only the wanted application but also all the applications in the same family.

We decided, in this phase, according to the goals of validating the methodology, to choose the first approach for the following reasons:

1. a development engine would be an immediate result, but it would be also greatly “re-usable” for other applications. This solution, once more, would focus on the implementation problems of the whole model, not only of the pilot application.

2. Must be taken into account the risk that, referring to the design of the pilot application, we loose the generality of engine and calibrate it on the specific problem, without achieving any result (generality and completeness) we resolved upon.

W2000, focusing on the design at a conceptual level, leave complete freedom to choose the particular technology for the implementation. So we decided to use the Microsoft development framework .NET. This choice doesn’t affect the generality of the results we obtained and is perfectly in accordance (as we explain later) with the object-oriented nature of the W2000 model.

Anyway, we are planning to develop another version of the application using the Java One environment [7] to get more information about the portability of our result and to preserve the generality as much as possible.

Under the choice we made, the “Debit Card” application has been developed using the three levels architecture, such as:

- **Data access**: it has access to the application data.
- **Business**: it holds all the business logic behind the application; it is concerned in the support of the multi-user access, making clearer the roles of the users and the relative behaviors of the application.
- **Presentation**: it’s the level which, using the data supplied by the data access level, filtered according to business logic, adjusts the
presentation for the various devices, composing the pages for the Web, PDA, WAP clients, etc.

Data Access and database structure
As the model suggests, all the information content of the application must be described during the Information Design. As a result we have a series of diagrams, in which the reality, as it’s perceived by the various users, is represented in terms or primitives (entity types, component types, slots, etc.).

Turning from the conceptual model of the information into the implementation in a database, we found out two possible solutions:

1. to create a database which representing the information structures of the model in a completely general way, describing the primitives themselves and the links among them. The specific application data would fill the database up, according to their classification in the model.

2. to create a classic Entity-Relationship schema, starting from the study of the application domain, using the model as it was an highly detailed document for the requirement analysis.

In other words we had to decide if, to turn from the model to the implementation of the data, it’s more suitable to describe into the E-R the container of the information (the structures) or to make a re-engineering effort and describe the information itself.

The first method allows to obtain a data structure, completely independent from the application domain (the database describes a meta-model of the data) and does not require any effort to the developer. He can avoid any design of the data (for example in an E-R schema).

Therefore, we preferred the second method, because, in spite of its being bound to the specific application, it makes the database much more easy to read; moreover the effort needed to obtain a classic E-R from a W2000 Information schema is not so big: the diagrams resulting from the Information Design are easy to be turned into E-R, removing the redundant links.

When the data structure is well defined, using stored procedures and appropriate views, all the other structures of the model can be reproduced, keeping a good degree of flexibility when accessing the data.

Business
Certainly, a peculiar feature of the modern applications is that they are getting even more multi-device and multi-user. Thanks to the Object oriented approach of the chosen development framework and to the property of polymorphism [8] of the objects, it’s possible to implement and get, with a little effort, different behaviors, depending on the role of the user who asks for a service.

Presentation
According to the model resulting from the Publishing Design and to the kind of implementation we chose, it’s possible to make an accurate mapping of the model structures on the .NET framework ones.

The Publishing Model allows the production of pages moving from page templates. Then the pages are modeled as union of Sections and these as aggregation of Publishing Units, in a hierarchical structure, which suitably describes the page.

In a similar way, .NET, using ASP.NET allows to create UserControl classes (already provided with rendering interfaces (HTML, WML, etc.), we can compose a presentation page with.

Pages are nothing but container for classes, which correspond to the Publishing Unit of the model.

In order to make the concepts clearer, we can take as example the structure of the home page for the pilot application:

![figure 1 - publishing model of the homepage for the pilot application](image)

Correctly explaining the UML diagram of the structure for the page, we are able to make an exact mapping inside the framework .NET and to obtain the web page.
In Figure 2, can be remarked that the various UserControl inside the page are instances of the class below; this is made plain using appropriate names: CatalogueIndex1.ascx, in fact is an instance of the class (UserControl) CatalogueIndex. These features of the .NET framework perfectly agree with the goals of the W2000 model, giving us a good feedback on the choices we made.

3. CONCLUSION
The idea of creating a framework for the direct prototyping of families of applications can make use of the results of the experience we did developing the pilot application, but implies new problems related with the generality of the product we want to get. The framework must be able to set the contents free from their representation: this result, already obtained for the pilot application, thanks to its development with .NET, must be engineered into a peculiarity of the framework itself. The creation of a "content engine" which provides only the right contents for the particular application in a format independent from the selected device, using, for example XML, may be able to make us reach our goal. The way to go is finally open, but the work to do is still a lot, also considering a series of open issues still to be solved. The W2000 model showed some limits and difficulties in the customization phase for different devices and user roles. We must verify the data representation with more data-intensive case studies. The one we adopted, as we said, is only one of the possible choices: we are not sure it's the best in terms of effectiveness and efficiency. In the mapping from the Publishing model and the .NET framework, we are not completely sure of the role the Sections play in our solution. We must be sure that in extreme devices, like WAP clients, this situation does not lead to difficulties in modeling and then in publishing.

REFERENCES