A METHODOLOGICAL APPROACH FOR THE MIGRATION OF LEGACY DATA INTENSIVE APPLICATIONS TO THE WEB

Distante Damiano
SET-Lab – Dipartimento di Ingegneria dell'Innovazione
Università degli Studi di Lecce
Via per Monteroni
73100 - Lecce, Italy
tel. +39 0832 320229 fax +39 0832 320279
damiano.distante@unile.it

Vito Perrone
Hypermedia Open Center
Politecnico di Milano
Via Ponzio 34/5
20133 Milano - Italy
tel. +39 02 23993557 fax +39 02 23993411
perrone@elet.polimi.it

TRACK: MIGRATION OF LEGACY APPLICATIONS TO THE WEB

Abstract

The research we present in this paper is the result of the conjunct work of the authors in their own research areas. In the first part we analyze and compare different approaches for the problem of the migration of legacy data-intensive applications to the web and towards Application Service Providing (ASP) paradigm, and in the second part we illustrate the approach we have elaborated and experimented in a real project. The approach handles the migration problem by applying a framework for technological solutions selection and a customised approach to web application design based on a methodology for web application design named W2000. In addition, the most relevant concepts we show in this paper have been tested in a tangible experience in a research project dealing with the migration of a legacy application used for customer management software (CMS), providing a useful feedback for future improvements.

KEYWORDS: LEGACY APPLICATIONS, WORLD-WIDE WEB AND DATABASES, WEB DESIGN, TECHNOLOGY CHOOSING, APPLICATION SERVICE PROVIDING, WEB APPLICATION, MIGRATION.

1. Introduction

With the advantages it implies for the companies that intend to adopt the "ASP" model to obtain software resources useful for their activities, the "ASP" model, is going to develop and its use will become widespread all over the whole world in the few next years[1]. More and more companies in the ICT world are attracted by the idea of opening to the new market model the applications they had already developed in the past, interpreting the role of an ASP company[1]. In particular, there are many software houses in Italy and Europe whose main business is to sell management packages to a certain number of existing customers. Much of this software was developed some years ago and used now obsolete technologies. Moreover the market shows a scarcity of both ASP operators, that integrate skills working at various levels of the business architecture model, and of application ASP native, i.e. developed for the web and multiple access[3]. On the other hand, one aspect that makes the adoption of this new model of application particularly interesting, is that these companies have developed a high level of knowledge of the needs of the customers they are addressing, giving the highest level of completeness to their own applications. The prospects of the ASP market growing up, together with the richness of management applications, let us comprehend that there is a strong interest and incentive for these firms to adopt the ASP model transferred onto web applications inherited or planning and producing new applications spread through Internet.

2. Migration strategies and problem description

Among possible strategies for migrating its own legacy software to the Web and towards ASP paradigm two appear the most used:

1) Usage of middleware software to enable the use of these applications (running) completely on a remote server through Internet;
2) Apply reverse engineering to the legacy application in order to create an ASP-native web application by preserving the same data structures, business rules and offering the same user facilities and interface.

Last strategy is intended as reverse engineering of the whole legacy application; anyway rengineering could be applied partially to the application, for example just to the user inteface[13][14].

Some of the advantages and drawbacks of the two approaches are shown in the table below.

[1] Application Service Providers (ASPs) deliver and manage applications and computer services from remote data centers to multiple users via the Internet or a private network[2].

Strategy	*Use of enabling middleware software*	*Reverse engineering and creation of ASP-Native applications*
Advantages	• Initial low efforts and investment. • Reduced Time-to-Market.	• User interface inconsistent with the web environment and generally with low usability. • Heavier delivery system. • Link to narrow technology privately owned commercial architectures • Use of privately owned client non standard like the web browser
Drawbacks	• Totally new user interface, suitable for the web and user-friendly. • Faster delivery system without middleware software between client and Application Server. • Company innovation and increasing personnel know-how. • Autonomy from privately owned commercial technologies. • Access to the application through a standard web browser.	• Initial high efforts and investment. • Long Time-to-Market.

Figure 1: *Two possible migrating approaches: Advantages and Drawbacks*

Whilst there are more established commercial solutions to conduct migrations according to the first strategy (an example is the Citrix Metaframe product used jointly to Terminal Server on Windows platform [4]), there are no, in the authors' knowledge and experience, methodical approaches for the development of ASP-Native data-intensive application, from a legacy application. On the other hand the latter one appears to be more advantageous compared to the previous one.

The accomplished migration highlights some of the problems that a migration process presents to the designer and developer:

All the business rules implemented in the legacy application should be preserved and so with the data dealt within (data and information contents appear mostly fixed);

The structural and navigation information problem of the web applications should be handled;

The access to the application should be allowed even with low bandwidth and round trip delay due to traffic towards the server;

Application performance must be guaranteed even with an increasing number of users accessing and using the application;

The usability and interactive level of the new web application must be at least equal to the desktop version of the application. (Normally, software company has a reasonable number of existing clients who already use the previous version of the application) by limiting the differences between web application and traditional window application.

The management of several data processing operations handled within a single transaction is required. This activity has to cope with two issues:

The web is connectionless and a solution to "state application tracking" is required;

Often a high amount of data is involved causing the application to run slowly.

3. A methodological approach

Our approach for the solution of the problems deriving from migrating through reverse engineering is mainly based, on one side, on the use of established methodology for designing web application with a suitable customised design approach; on the other, to the actual definition of a framework able to offer implementation-technological solutions to the arising problems.

FRAMEWORK FOR CHOOSING IMPLEMENTATION / TECHNOLOGICAL SOLUTIONS

In carrying out the migration of this type of application towards ASP through reverse engineering we have to know that the proper requirements of this application are known and so are the solutions applied. A certain number of requirements of the migration operation should be identified and satisfied. An example given by the constraints for which the web application resulting from migration process must only be used by users with a low bandwidth.

The use of frameworks for the solution of a set of problems is a generally adopted practice in various information sectors. The basic principles of framework definition are common in various situations, the differences are found only in the primitive that constituted the framework and usage methods. The framework examples, which inspired us, are the ones from the field of Object Orientation [5] and Web applications modelling [6].

In order to answer this demand, we decided to adopt the approach for the analysis of the requirements based on goal [7] commonly used in requirements engineering field. It is important to point out that in our approach we utilise the goals methodology to analyse and model the migration requirements of this kind of application and not the actual application requirements dictated in a binding way by the starting legacy application.

The following steps could describe the design of a framework:

1) The goals of the migration process of this application and through subsequent refinement of these, the requirements, are identified. In this step, by following the goal approach, possible stakeholders and requirements are identified leaving the value property unspecified of each individual one. Through the study of various application cases, our aim is to identify and define a set of stakeholders, goals and requirements covering all possible migration problems towards the Web and ASP model of this type of application.

2) Once defined, various requirements, possible approaches and implementation solutions are identified using a process similar to the pattern definition one [8] although without requiring the same effort of the pattern definition. Obviously for

the definition of these implementation solutions starting from the requirements, we followed an interactive approach. This lead to the search for compromises for the re-definition of some requirements. For this reason we have also supplied, for each single requirement, different possible solutions in order to be able to select the most suitable one depending on the particular application, taking advantage of the value mechanism.

3) A number of possible enabling technologies, which allow the creation of previously recognised implementation solutions, are identified. In this phase we also proceed with the identification of possible requirements within the totality which can create constraints in the technological selection.

The separation of the approaches and implementation solutions from the selection of the enabling technologies for their same solutions confers to the framework the ability to update itself and to remain accurate, as the technologies develop notoriously fast in the Web field. For each specific migration case of the kind of application considered herein, we proceed with "cutting out" the goal tree of the framework, specialising the basis of the goals and requirements foreseen therein and confirming the property value of the various goals on the basis of the stakeholder specific needs. From this we obtain the set of implementation solutions and consequently the actual technological solutions which create them better.

Example of framework application
In order to explain the use of the goal approach and the contents of our framework we produce the following example:
Phase 1:
Let us assume that one of the goals is that the resulting web application should be used satisfactorily even under the constraint of a low bandwidth. One of the requirements resulting from this goal is:
To reduce interactions between client and server.
Phase 2:
The framework will reply to this requirement with the following implementation solution: Manipulation of the data from the client side.
Phase 3:
A possible enabling technological solution in the Microsoft environment is joint use of RDS technology. With scripting of the client side which allows you to have a remote recordset on the client and to act through remote control on the data from the server side without the need of requesting and updating the pages [9].
Examples of this type can be easily extracted from the framework, but due to limited space we described the above with the purpose of showing how the frameworks has been identified.

CUSTOMISED APPROACH FOR DESIGNING WITH W2000

In respect of the second point, we proceeded with the customisation of the approach used in a design methodology of a native web application. The chosen methodology is W2000 [10] (descending from HDM [11]) currently involved in a European project [12] in which it has been extended for supporting actual characteristics of third generation web applications. W2000 methodology structures web *application design* into three activities:

- *Information Design:* structuring the content and its organisation in terms of access structures.
- *Navigation Design:* structuring the content in order to provide effective navigation and structuring its navigation paths.
- *Publishing Design:* defining how the application is organised into pages.
- *Operations Design*: defining which operations are provided to the users.

If the first three activities are "typical" of Hypermedia design, the fourth activity is a new one and its definition and integration with the rest is one of the main contributions of UWA.

Using the concepts (provided by the models) and the corresponding notations, a designer can write down an *application schema*, i.e. a description of what the application will be (not how it will be implemented). The application schema should be the result of a *design activity* that takes into consideration the overall *application requirements* (both typical business and migration requirements).

For each design activity, schemas can be specified at two levels of depth and precision: *in-the-large*, to describe general aspects, sometimes informally, without many details, and *in-the-small,* to describes the various aspects in fine detail.

Breaking design of web applications into several activities and multiple steps, does not add complexity to the job, as it may appear, but it helps in obtaining a *separation of concerns* (a well known software engineering principle), and has many advantages:

- It helps in mastering the complexity of designing a sophisticated web application. The result of each design activity, in fact, encapsulates a set of clear design decisions focused on a specific aspect of the application.
- It improves *team communication*, since the different aspects can be communicated and analysed more systematically among the members of the team.
- It allows designers to create better-organised *design documentation* as discussed in [12] annex A.
- It facilitates *maintainability* of the design, since it will be easier to trace the design features that are affected by changing requirements, and to identify which design specifications must be modified.
- It promotes *design reuse*, since different portions of an application design can be easily extracted and re-used for different applications, or different versions of the same application.

- It supports design *independence*, since different portions of the design specifications can be modified without necessarily affecting the others. Designers, for example, could easily modify some presentation structures without necessarily changing navigation and information design, or, if they have to, being able to immediately spot what should be changed.
- The modularity of the design activity is also very relevant bearing in mind the need to integrate hypermedia-operation design with customisation and transaction design.

On the basis of the aforementioned considerations on W2000 and taking into account that the characteristics, which this type of application lays down, together with the typical market demands, required a special definition and customisation of the approach generally used in the design of a web application with W2000. This represents an element of the research work submitted in this paper, partly still under refinement and validation.

The usual approach we use for designing a web application through W2000 could be sketched as in the following figure:

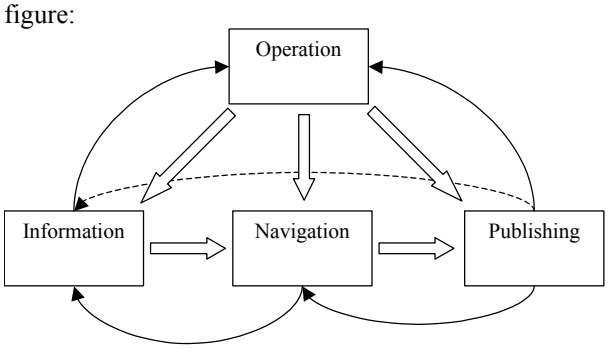

Figure 2: Web Application Design Approach for W2000

Here is a brief explanation of what the above figure means. Designing a web application through W2000 approach involves achieving the above design activities following the order specified by the bordered arrows. As it can be observed the phases of Information, Navigation, and Publishing Design follow a specific order (as shown by the thick arrows). First of all the designer models the information and the access structures of the application, then navigation and finally the pages. However, from our experience as designers, accomplishing these three steps, doesn't mean finishing one step and never going back to it for revising or updating the produced model. Designing a web application with W2000 does not adopt a traditional cascade or waterfall lifecycle. We agree that this old-fashioned and unnatural methodological approach will not result in a reusable, flexible and high-quality software product. Designing with W2000 involves various returns to the previous steps as shown by the thin arrows. We use a full line or a broken line meaning respectively a greater or less influence. So generally some decisions taken in a following step may involve some changes in previous steps. For example if designing some pages we realise that some nodes defined in the navigation model contain too much information (heavy nodes vs. light nodes) or some navigation features are unrealisable in certain pages, the designer should get back to the navigation model and introduce some changes.

Furthermore from the sketch it can be observed how the modelling of the operations constitutes a transversal activity to the others. We should briefly recall another very important aspect of W2000 not easily perceivable from the picture. The designing of the application is firstly carried out in-the-large dealing with the four identifies activities, we then proceed through subsequent detailing works the "in-the-small" version. Thanks to the dual levels of granularity in the modelling, the methodology is suitable for various purposes, such as in a general design which can be created in short time by tackling the "in-the-small" level only.
These are the characteristics, which make W2000 particularly suitable to tackle the design problem of the applications dealt with in this paper. To give you an idea of how W2000 and its various aspects can be used for these applications, in the following picture we added some information and customised the general approach of W2000. What we present in this article is a possible approach to the use of W2000 for designing these types of applications. In this approach we used the strategy of lightening the approach, emphasising the aspects considered more important. This became necessary in order keep in consideration the requirement of a short time-to-market strongly felt by this sector of the market. This does not mean that the total approach of W2000 in the whole could be applied to this kind of application design, within the previously described constraints.

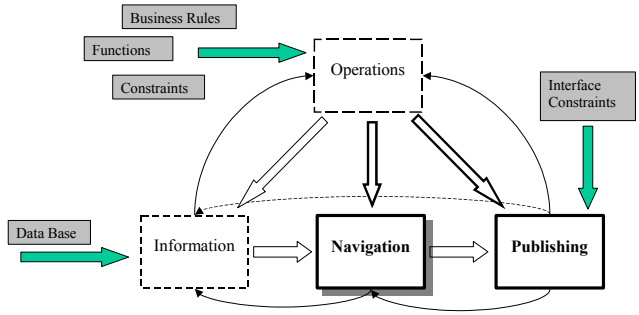

Figure 3: Customised Approach for W2000

The previous picture shows the emphasis of our approach in respect of the various phases of W2000. In fact, we used different styles for the box, introducing what should be kept in consideration of the legacy application. We hereafter briefly describe the adopted approach step by step.

- **Information:** the majority of the applications of web information are inherited from the basic data of the legacy application. Apart from these, other information deriving from potential new possibilities offered by ASP paradigm should be taken into account. From our experience, in this phase it is advisable to make a design of the hyperbase "in-the-large", always starting from the information obtained from the data base, and subsequently emphasising the design of the access structures which represent the actual new thing introduced at ICT level.

- **Navigation:** we consider this phase the most delicate one of the application design process, in which we are executing the migration. In fact it is in this phase that we re-design the navigational dynamic of the application moving from an environment typically organised in a functional way to a navigational one. Here the sophisticated navigational model of W2000 is of fundamental importance. Furthermore it is in this phase that we can increase the usage facilities of the application and re-design the user experience increasing in this way the level of satisfaction in the use of the application. The guidelines defined in W2000 offer all the necessary tools for the designing of this phase.
- **Publishing:** The design of the application pages is also particularly important even if the dynamic of the application is already defined by the navigational model. In this phase it is important, as far as possible, to comply with the cognitive characteristics of the legacy application from which the web application derives. (*Interface Constraints*). In fact it is important to avoid to the user having a drastic transition to the new application by preserving the old (established) operative way on the new system. Although this could appear to be in contrast with the previous statements, it is not true since the only need is to guarantee a correct association of information and operations on the same page, which recall the operative way of the user on the legacy application.
- **Operation:** the application design of the operations is strongly bound by the functional characteristics and business rules inherited by the legacy application. In this case it would have been possible to proceed in two different ways: intensely re-designing all functionality or re-proposing in a suitable way the operative methods of the current ones. Due to the previously discussed needs, we adopted the second route, which essentially lead to the integration of the operations of the information, navigational and publishing aspects by using a sub-set of W2000. In fact, as can be observed from the picture, the main emphasis in this case was given to the impact of the operations on navigation and publishing.

The lack of space prevents us from exhibiting a full design example of the application with the above-described approach.

4. Conclusions and Future Works

In this paper we saw the problem of migrating a legacy data-intensive application to the Web environment towards an ASP paradigm. Furthermore we aim to point out the relevance of this kind of problem in the field of European SMEs. As a solution for the above problem, we introduced a methodological approach for the migration of legacy applications through a reverse engineering process. Our approach tackles this problem from both an implementation, technological and design point of view. In response of the first issue we proposed a *framework for Choosing Implementation – Technological Solutions* and for the second one we proposed a *customised approach for designing web applications with W2000*. The illustrated solution has been applied to a real case study where we migrated a management software package, getting a good response. Benefits essentially came from the reuse of the knowledge "stored" in the implementation-technological framework and the high quality of the designed application. Moreover we believe that applying our solution should involve a cost reduction, thanks to re-use, and a quality product thanks to using an improved design methodology.
The solution we have explained is currently "work in progress", in fact we intend to complete and extend our approach with:

- Completing the framework of implementation-technological solution choosing.
- Refining customised W2000 methodology we used for the design.
- Testing the entire approach on a large number of case studies of migrations and evaluating the results.

References

[1] Source: IDC may 2001.

[2] ASP Industry Consortium, http://www.allaboutasp.org/.

[3] Some guys from Accenture Italia, Asp Industry Consortium and others in a local focus magazine (September 2001)

[4] http://www.citrix.com/products/metaframe/default.asp

[5] W. Pree: "Design Patterns for object-oriented software", Addison Wesley, 1994.

[6] D. Schwabe, G. Rossi, L. Esmeraldo, F. Lyardet "Web Design Frameworks: An Approach to Improve Reuse in Web Applications", Lecture Notes in Computer Science (Hot Topics) - Proc. of the Second International Workshop on Web Engineering, WWW Conference , Springer Verlag, 2000.

[7] A. Dardenne, A. van Lamsweerde, and S. Fickas, "Goal-directed Requirements Acquisition", *Science of Computer Programming*; Vol. 20, 1993.

[8] F. Garzotto. P. Paolini, and S. Valenti, "Modeling-by-Patterns of Web Applications". In P. P. Chen,D. W. Embley, and S. W. Liddle eds, "Advances in Conceptual Modeling" (Proc. WWWCM099 -ER99Int. Workshop on the World Wide Web and Conceptual Modeling) Paris, Nov. 1999.Springer LNCS 1727**.**

[9] J. Papa et al. "Professional ADO 2.5 RDS Programming with ASP 3.0**"** 2nd Ed (1 January, 2000) Wrox Press Ltd; ISBN: 1861003242.

[10] L. Baresi, F. Garzotto, P. Paolini, and S. Valenti: "HDM2000: The HDM Hypertext Design Model Revisited", Tech. Report, Politecnico di Milano, Jan. 2000.

[11] F. Garzotto, P. Paolini, D. Schwabe, "HDM - A Model-Based Approach to Hypertext Application Design". In ACM Transactions on Information Systems 11(1), 1993.

[12] UWA (Ubiquitous Web Applications) Project: Deliverables D2 (General definition of the UWA framework), D7 (Hypermedia and Operation Design: Model, Notation, and Tool Architecture).

[13] Ken C. K. Law, Horace H. S. Ip, Fang Wei, "Web-Enabling Legacy Applications", in Proceedings of the International Conference on Parallel and Distribuited System Reverse Engineering (1998).

[14] E. Stroulia, M. El-Ramly, L. Kong, P. sorenson, B. Matichuk, "Reverse Engineering Legacy Interfaces: An Interaction-Driven Approach", in the Proceedings of the 6th Working conference on Reverse Engineering (WCRE'99) October '99.