

# Migrating to the Web a Legacy Application: The Sinfor Project

D. Distante<sup>\*</sup>, V. Perrone<sup>+</sup>, M. A. Bochicchio<sup>#</sup>

<sup>#</sup>*SET-Lab, University of Lecce, Department of “Ingegneria dell’Innovazione”*

<sup>+</sup>*Politecnico di Milano, Hypermedia Open Center*

<sup>\*</sup>damiano.distante@unile.it, <sup>+</sup>perrone@elet.polimi.it, <sup>#</sup>mario.bochicchio@unile.it

## Abstract

*Various approaches can be used to migrate legacy applications to the Web. In particular, migrating data-intensive legacy applications (e.g. traditional application for business management) needs methodological approach to face the challenges implied by the process.*

*The Ubiquitous Web Applications (UWA) framework is one of the most innovative and complete frameworks for conceptual user centered modelling of a Web application. In this paper we describe the application of UWA to a real experience of reengineering a real legacy application for customer’s order management.*

## 1 Introduction

Recently, more and more companies in the Information and Communication Technologies (ICT) world are attracted by the idea of migrating to the Web the applications they developed in the past, then becoming ASP<sup>1</sup>. Unfortunately, this process is not straightforward, due to the problems and the constraints that the new environment (the Web) implies, and to the inherent complexity of this kind of applications. Two of the most important challenges to accomplish the migration task are:

- changing the interaction paradigm of the application from its current style to the navigational one, more appropriate to the Web;
- choosing the right technologies and the suitable implementation solutions.

These aspects affect both the migration strategy and the design and implementation decisions, so a methodological approach is strongly required to effectively manage the overall migration process.

In this scenario the Sinfor<sup>2</sup> Company, decided to migrate to the Web its legacy “ArcoImpresa” package.

The job was done in collaboration with our SET-Lab.

In the the paper we describe the methodological steps we adopted and the main technical issues we encountered.

## 2 Migration approaches

For companies faced with the challenge of web-enabling their legacy applications, the choice of how to proceed falls generally into one of the following categories [2]:

- *Starting from Scratch* a totally new software application is created.
- *Refacing* and *Remodeling*: additional software is used to enable the interaction with the legacy application via the Web; several commercial solutions and research works [3] exist in this field.
- *Reengineering*: the existing software is adapted and extended to operate according to the ASP paradigm.

The third approach can be limited to the user interface or it can be extended to the whole application. Several techniques have been proposed to reengineer the user interface [4-6], but most of them, based on a simple translation of the legacy user interface to the new front-end, aren’t able to exploit the advantages and the peculiarities of the Web. So, even if all mentioned approaches enable legacy applications to the Web, for the purposes of this paper we feel that the reengineering of the whole application is the right approach. Reengineering, in fact, allows the true fulfilment of Web constraints, by getting benefits of Web peculiarities.

## 3 The Sinfor Project

At present, “ArcoImpresa” is a standalone business management software package based on a non-relational “Btrieve” database, it was developed in Clipper, with an MS-Dos character-based user interface. The Sinfor project

---

<sup>1</sup> Application Service Providers (ASPs) deliver and manage applications and computer services remotely to multiple users via the Internet or private networks [1].

---

<sup>2</sup> Sinfor is an Italian sw house developing business-management packages for commercial enterprises.

is to migrate to the Web the “purchase order management” module of the ArcoImpresa package.

From its first release, built 15 years ago, “ArcoImpresa” was more and more refined and customized to better meet user requirements. Then, we can say that the knowledge acquired by the company during its life is “hardcoded” into its products. We mean that most of the software upgrades were undocumented and known only by personnel working in the company since the first release of the software. So, the *Reengineering* choice is also the right alternative to recover all this know-how [7].

## 4 Our migration approach

To migrate “ArcoImpresa” to the Web we chose the UWA methodology [8] for three main reasons:

- it represents one of the most innovative and complete frameworks for conceptual user centered modelling of Web applications. In this field, other approaches and/or methodologies for Web application design, like in [11] [12], don’t include all the UWA features.. Moreover other known works tackle the problem of the Web application design at lower level. For example, Conallen [11] propose a workable solution, but it privileges client-server interactions and page presentation, underestimating, in our opinion, the logical vs. physical design of both information and navigation structures.
- in the UWA project a set of UML-based tools to design, to support and to fast prototype Web applications are being developed and made available.

To better understand the adopted approach, a short description of UWA is given in the next section.

### 4.1 UWA Framework in a nutshell

UWA methodology provides a complete framework particularly suitable for the last generation of Web application that are multi-channel, multi-user and generally context-aware. The overall framework includes and integrates four aspects:

- *Requirements Elicitation*: it is a version of the wide spread goal approach [14] suitably tailored for addressing the characteristics of a Web application and the integration needs of the framework.
- *Hypermedia and Operation Design*: this activity, based on the W2000 [9] user centered methodology, relies on four design phases:
  - *Information Design*: during which the contents of the application and the way of accessing them are defined.
  - *Navigation Design*: during which the atomic information units (named *Nodes*) delivered to the user are defined and organized in contexts.

- *Publishing Design*: during which the actual pages of the application are defined combining navigational aspects, inherited by the previous phase, and pure publishing aspects.
- *Operation Design*: in this phase the user and system operations are defined. The operation can impact on the business state of the application but can also change the understanding of the application from the user point of view.
- *Transaction Design*: consists in defining user activities, system transactions and how operations are involved in. Generally, transaction are defined in terms of navigational and operational capabilities of the user.
- *Customisation design*: during this activity the relevant context variables have to be recognized and the customisation strategy has to be defined.

At the end of the previous activities the designer produces various application schemas to take into account the overall *application requirements*, that in such a case come from both the legacy application and the migration proces.

### 4.2 Redesigning “Arcoimpresa” with UWA

The first step of our reengineering process consisted in defining stakeholders, goals and requirements, starting from legacy software analysis in close collaboration with Sinfor designers.

A set of issues implied by the migration process of this kind of application have been gathered. We gave solutions to them by means of both appropriate design choices and suitable implementation technologies.

Two type of goals have been identified:

- goals related to the adopted migration process;
- goals related to the new version of the software.

The first type of goals, not strictly related to the specific application, generate both requirements (desired) and constraints (not desired but imposed). They have to be considered during the entire design process, up to the implementation phase. Some of the requirements we found in our project and we put in this category are presented here. The solutions we developed to deal with them are also presented (after the “→” symbol, in the list):

1. All the business rules and all data implemented in the legacy application had to be preserved (data and information contents of the new application are mostly fixed). → User-types, user-needs and functionalities of the legacy application have to be analysed in depth.
2. The new application should be accessible via Web and fast to use even with poor bandwidth. → a) Data transferred between client and server have to be limited to the minimum; b) Round trip delays towards the server should be minimized; c) Navigation design

of the application have to be functional to the operation the user needs to do.

3. Application performance must be guaranteed even with an high number of users. → True scalability of the designed system is required and tools for monitoring system performance are needed;
4. The usability and the interactivity level of the new Web application must be at least equal to the legacy version of the application (the company has a reasonable number of existing clients who already use the previous version of the application) by limiting the differences between the Web style and the Window style. → High quality user centered design and techniques to measure user usability of the application are needed; comfortable user interface (like Windows-based user interface) are desirable.
5. The management of several data processing operations handled within a single transaction is required. This activity has to cope with two issues: a) the web is connectionless and stateless; b) high amount of data are needed to describe the application state.
6. The structural and navigational information problem of the web applications should be handled.

Second type of goals come into application goal category. They comes from the reverse engineering of the legacy application, with its stakeholders, its needs and requests. The stakeholders we detected are: “Sinfor Management” representing the “Sinfor” company (it is interested in the migration process and in the production of the new application but doesn’t represent a real final user of the software), “Company Manager”, “Sales Agent”, “Order Administrator” and “System and database Administrator”, (they are real final users of the application.)

### 4.3 Hypermedia Design

Through this activity, the application contents have been extracted by reverse engineering from the legacy database. Content organization in terms of *semantic associations* between them and *access structures*, *navigation paths* and content organization into *publishing pages* have then defined. Legacy database has been restructured and redesigned into a relational database during implementation phase.

To develop the Web prototype, we just considered a single type of device (the PC) and a single type of media (the Web) with no customization.

Other information deriving from the new possibilities offered by ASP paradigm has been detected and taken into account. The design of the *hyperbase* “in-the-large” [9] was then performed, together with the access structure design.

With hypermedia design activity we also defined the most relevant navigation path of the application: this is the most delicate phase of the whole application design

process. In fact it is in this phase that we re-design the navigational dynamic of the application moving from the original (functional) style to the new (navigational) one. Here the sophisticated navigational model of W2000 is of fundamental importance. Furthermore it is in this phase that we increase the user friendliness of the application and we re-design the “user experience”, so increasing the level of satisfaction of the users. The guidelines defined in W2000 offer all the conceptual tools for this design phase.

### 4.4 Operation, Publishing and Transaction design

User operations of the new application are strongly bound to the business rules and to the functional characteristics inherited by the legacy application. In our case it would have been possible to proceed in two different ways: by completely re-designing all functionalities or by adapting the legacy functionalities to the Web. Due to the previously discussed needs, we adopted the second solution.

*Transaction Design* is of fundamental importance for the category of application “ArcoImpresa” belong to. Business management software, in fact, is characterized by a high number of transactions operating on large amount of data. With this design step we defined all the legacy non-atomic user activities (e.g. a new purchase order creation done by a Sales agent). This design phase was performed by adopting UML sequence diagrams.

The design of the application pages was also particularly important, even if the dynamic of the application is already defined by the navigational model. In this phase we felt important 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 interface 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.

### 4.5 Implementation Issues and resulting application prototype

Referring to the solutions listed in section 4.2, and in particular to requirements a) and b), the prototype we developed was based on Microsoft platform (MS IIS, SQL Server, Internet Explorer, ASP dynamic pages) and on RDS technology [15] for client-side caching of data. With RDS, client-side applications have the option of working directly with an OLE DB provider on the remote process/machine that uses ADO recordsets as input/output parameters or return values. Equivalent solutions exist also in other developing platform. In Java environment, for example, we

can adopt a combination of Java applet and Javascript code on the client and a servlet to the server [17]. Both solutions results in reduction of data transfer between client and server to the minimum required to accomplish each given user operations.

Referring to the required “Windows like” user interface, with menus, contextual menus, drag&drop features and other operations invoked by mouse, we extensively adopted DHTML and CSS [18].

Fig. 2 reproduces a screen shot of the prototype of the Web enabled version of “ArcoImpresa”. It is subdivided into two main section: the left ssection of the page is used to shows collections of application contents (collection of Products, Customers, Orders etc.) while the right section is used to show detailed information, for example about a new order. The activity of compiling a new order for a customer is accomplished by dragging the item selected on left side of the page and dropping it to the order form on the right side of the page. Usability and intuitiveness of user interface is easy to prove.

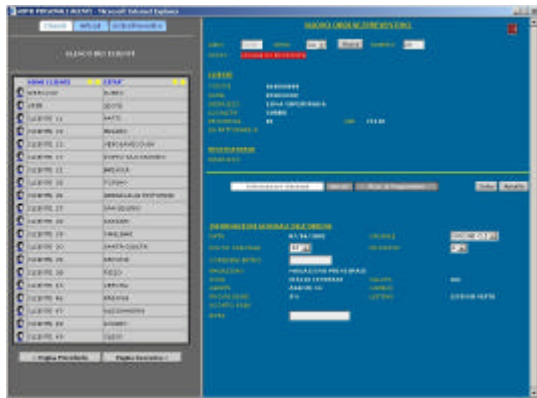


Fig. 2 - Creating a new purchase order with the new version of “ArcoImpresa”

## 5 Project results and conclusions

Usage of a methodology for designing and implementing a web-application requires high start-up investments and will cost more than just starting implementing the solution. This practice agrees to a production-oriented approach that reflects a commercial development strategy guided by cost and time -to-market reduction [19]. Unfortunately, this widely used approach is very poor in terms of software reuse, technology independence and sw maintainability.

The UWA design framework can be very helpful to support the reverse engineering and the reengineering of legacy applications and to achieve high usability, good quality documentation, high maintainability and true Web-native and ASP ready application.

We believe that the approach outlined in this paper is rather general and that it can be reused in other similar projects. The next research directions are: the application of the framework to other migration projects, to

consolidate the methodology, and the definition of guidelines for choosing implementation/technological solutions to reduce the start-up costs of each migration project.

## 6 References

- [1] ASP Industry Consortium, <http://www.allaboutasp.org/>.
- [2] L. Akers, “*Web-enabling Legacy Applications, an Overview for VSE Users*”, SoftTouch Systems Inc., Jan 2002.
- [3] H.-A. Jacobsen , O. Günther, “*Middleware for Software Leasing over the Internet*”, Proceedings of the first ACM conference on Electronic commerce, Nov. 1999.
- [4] Melody M. Moore, Lilia Moshkina, “*Migrating Legacy System to the Internet: Shifting Dialogue Initiative*”, Proc. of Working Conference on Reverse Engineering (WCRE 2000).
- [5] Ken C. K. Law, Horace H. S. Ip, Fang Wei, “*Web-Enabling Legacy Applications*”, Proc. of the International Conference on Parallel and Distributed System Reverse Engineering (1998).
- [6] E. Stroulia, M. El-Ramly, L. Kong, P. Sorenson, B. Matichuk, “*Reverse Engineering Legacy Interfaces: An Interaction-Driven Approach*”, Proc. of the 6th Working conference on Reverse Engineering (WCRE’99) Oct -’99.
- [7] Renè R. Klösch, “*Reverse Engineering: Why and How to Reverse Engineer Software*”, in Proceedings of the California Software Symposium (CSS), 1996.
- [8] UWA (Ubiquitous Web Applications) Project: <http://www.uwaproject.org>.
- [9] L. Baresi, F. Garzotto, and P. Paolini, “*Extending UML for Modeling Web Applications*” In Proceedings of 34th Annual Hawaii International Conference on System Sciences (HICSS-34). IEEE Computer Society, 2001.
- [10] Unified Modeling Language: UML™, <http://www.uml.org/>.
- [11] J. Conallen: “*Building Web Applications with UML*”, Addison-Wesley, 2000.
- [12] D. Schwabe, G. Rossi, OOHDM web site, <http://www.telemidia.puc-rio.br/oohdm/oohdm.html>.
- [13] S. Ceri, P. Fraternali, A. Bongio, S. Comai, *Web Modeling Language (WebML)*, <http://webml.elet.polimi.it/webml/>.
- [14] A. Dardenne, A. van Lamsweerde, and S. Fickas, “*Goal-directed Requirements Acquisition*”, *Science of Computer Programming*; Vol. 20, 1993.
- [15] J. Papa et al. “*Professional ADO 2.5 RDS Programming with ASP 3.0*” 2nd Ed (Jan 2000) Wrox Press Ltd; ISBN: 1861003242.
- [16] Kamaljit Bath, “*Using the Customization Handler Feature in RDS 2.0*”, April 1998, Microsoft Corporation
- [17] Donald Bales, “*Dynamic Database Access from Client-Side JavaScript*”, O’Reilly Network: <http://www.onjava.com/lpt/a/onjava/2002/01/23/javascript.html>, Jan 2002.
- [18] DHTML examples, [http://www.w3schools.com/dhtml/dhtml\\_examples.asp](http://www.w3schools.com/dhtml/dhtml_examples.asp), W3Cschools.com.
- [19] Chris Barry, Michael Lang, “*A survey of multimedia and web development techniques and methodology usage*”, IEEE Multimedia Vol. 8, No. 2; April-June 2001, pp. 52-60.