# Redesigning Legacy Applications for the Web with UWAT+: A Case Study

Damiano Distante
Research Centre on Software Technology
University of Sannio, Italy
distante@unisannio.it

Scott Tilley
Department of Computer Sciences
Florida Institute of Technology, USA
stilley@cs.fit.edu

Gerardo Canfora
Research Centre on Software Technology
University of Sannio, Italy
canfora@unisannio.it

Shihong Huang
Department of Computer Science & Engineering
Florida Atlantic University, USA
shihong@cse.fau.edu

## ABSTRACT
This paper reports on a case study of redesigning a legacy application for the Web using the Ubiquitous Web Applications Design Framework with an extended version of its Transaction Design Model (UWAT+). Web application design methodologies hold the promise of engineering high-quality and long-lived Web systems and rich Internet applications. However, many such techniques focus solely on green-field development, and do not properly address the situation of leveraging the value locked in legacy systems. The redesign process supported by UWAT+ holistically blends design recovery technologies for capturing the know-how embedded in the legacy application with forward design methods particularly well suited for Web-based systems. The case study highlights some of the benefits of using UWAT+ in this context, as well as identifying possible areas for improvement in the redesign process and opportunities for tool automation to support it.

## Categories and Subject Descriptors
D.2.2 [**Design Tools and Techniques**]: *Evolutionary prototyping; User interfaces.* D.2.7 [**Software Engineering**]: Distribution, Maintenance, and Enhancement – *Enhancement; Restructuring, reverse engineering, and reengineering*

## General Terms
Design, Human Factors, Performance

## Keywords
Redesign, Legacy Systems, Migration, Reengineering, Web, UWA, UWAT+, Experience, Case Study


*ICSE'06*, May 20–28, 2006, Shanghai, China.


## 1. INTRODUCTION
Web applications design methodologies hold the promise of engineering high-quality and long-lived Web-based systems and rich Internet applications. Such methodologies borrow from established principles of software design. They also incorporate best practice from related areas, such as user interface principles, usability guidelines for online systems, and lessons learned from hypermedia development.

However, many such methodologies typically focus on new development, and do not properly address the situation of leveraging the value locked in legacy systems. There is considerable corporate knowledge embedded in the business processes that are implemented in the code base of such systems. A design methodology that does not attempt to capture this valuable knowledge is bound to be of less strategic use than one that leverages this important asset.

This paper reports on a case study of redesigning a legacy application for the Web using the Ubiquitous Web Applications (UWA) Design Framework [16][17] and an extended version of its Transaction Design Model (UWAT+) [4][5]. The UWAT+ approach blends design recovery technologies for capturing the know-how embedded in the legacy application with forward design methods particularly well suited for Web-based systems. The result is a more complete design migration, with a user interface that reflects modern principles yet still retains the unique aspects of the original system.

The next section of the paper outlines some of the salient issues of redesigning legacy systems for use on the Web, including a brief overview of UWA and UWAT+, the redesign process supported by UWAT+, and related work in the area. Section 3 details the redesign case study, which followed the UWAT+ process in migrating a legacy procurement system to the Web. Section 4 discusses some of the important lessons learned in executing the case study. Finally, Section 5 summarizes the paper and outlines possible avenues of further work.

## 2. REDESIGNING FOR THE WEB
In the past, attempts to migrate legacy applications to the Web have paid close attention to changing the user interface from an

old “green screen” to a collection of HTML-based Web pages [23]. The assumption is that the design inherent in the original panel-driven system is suitable for a nearly literal translation to a series of forms viewed in a browser. There is no doubt that such a modernization approach has proven reasonably successful (at least in the short term). However, this type of translation does not fully exploit new capabilities on the more modern platform.

This section provides an overview of UWA and UWAT+ and the redesign process that these frameworks support. Related work is also briefly discussed, in order to compare and contrast the UWAT+ redesign process with other types of system migration techniques.

## 2.1 UWA and UWAT+ at a Glance

The holistic approach to redesigning legacy applications for the Web presented in the next section relies in part on the Ubiquitous Web Applications (UWA) Design Framework and an extended version of its Transaction Design Model (UWAT+). This section provides an overview of these two design frameworks, focusing on their use in the redesign process.

### *2.1.1 UWA*

The Ubiquitous Web Application (UWA) design framework offers the designer a set of methodologies, meta-models, and tools for the user-centered design of data and operation intensive ubiquitous (i.e., multi-channel, multi-user and generally context-aware) Web applications. A major strength of UWA is that it addresses the design of Web applications by adopting separation of design concerns into four main activities:

1. *Requirements Elicitation* [18]. Using a goal-oriented approach [1] that produces UML use case diagrams, this activity is intended to identify the application stakeholders, their needs, and the requirements for the application to design and implement.
2. *Hypermedia and Operation Design* [20]. This activity is accomplished by using a user-centered methodology named W2000 [2], composed of four phases, each of which is concerned with specific aspects of the Web application:
    a. *Information Design*, during which the Hyperbase (the contents, their structure, and the semantic relations among them) and the Access Structures (ways of accessing the contents, that is view on the hyperbase) of the application are defined.
    b. *Navigation Design*, during which the atomic information units (named Nodes) that will be delivered to the user are defined and organized into contexts (named Clusters).
    c. *Publishing Design*, during which the actual pages of the application are defined, combining navigational aspects inherited from the previous phase and purely publishing aspects.
    d. *Operation Design*, during which the user and system operations are defined.
3. *Transaction Design* [21]. This activity is intended to design the user activities and the system transactions that will implement the business process the Web application aims to support. Generally a transaction is defined in terms of the navigational and operational capabilities of the user.
4. *Customization design* [22]. This design phase is specifically intended to support the ubiquity of Web applications designed with UWA. This design activity defines how the application will adapt to different contexts characterized by various dimensions (user type, user profile, device, time, location, etc.). During this activity the relevant context variables have to be recognized and the customization strategy has to be defined.

Figure 1 gives an overview of the usual process of designing a Web application with UWA, showing the order in which the design activities are carried out and the models resulting from each of them.

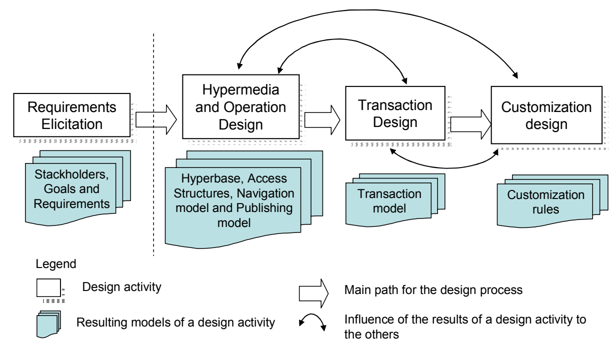


**Figure 1. The UWA Web application design process.**

### *2.1.2 UWAT+*

UWAT+ is a revised and extended version of the UWA Transaction Design method that reserves the design of business processes a first place in the UWA overall design approach. Compared with its original version, UWAT+ highlights a user-centered approach, is capable of designing the interplay of content navigation in process execution, and to specify which contents and which navigation links should be provided to the user when executing each of the activities of a business process. UWAT+’s abilities rely on a conceptual model (shown in Figure 2) that takes into consideration the user-centered nature of business processes on the Web and enables strong integration with the other UWA conceptual models (in particular, the Information Model and the Navigation model).

UWAT+ targets the objective of designing business processes in Web applications basically in two steps. In the first step a business process is modeled by means of one or more *Web transactions*. In the UWAT+ jargon, a Web transaction is a set of business process activities the user has to carry out by means of the Web application and that will permit him to execute the business process. In the second step each activity included in the defined Web transaction is associated with a *Navigation node* and a *Navigation cluster*. Navigation nodes and Navigation clusters are the way an activity will be presented to the user.

The first step produces for each Web transaction two models: an Organization model and an Execution model. The Organization model is a customized version of the UML class diagram describing properties and relations of the Web transaction components activities. The Execution model is a customization of the UML activity diagram [11] that represents a Web transaction from a dynamic point of view. Its activities and sub-activities of the Web transaction are represented by states (ovals), and the execution flow between them is represented by state transition (arcs).

The second design step of UWAT+ enriches the UWA Navigation model of the Web application producing the Activity nodes and the Activity clusters associated with each of the activities included in the Web transactions defined with the first design step. An activity cluster is a design concept introduced in UWAT+ to design the possible navigation between a node and its “adjacent” and the interaction between activity execution and content navigation.

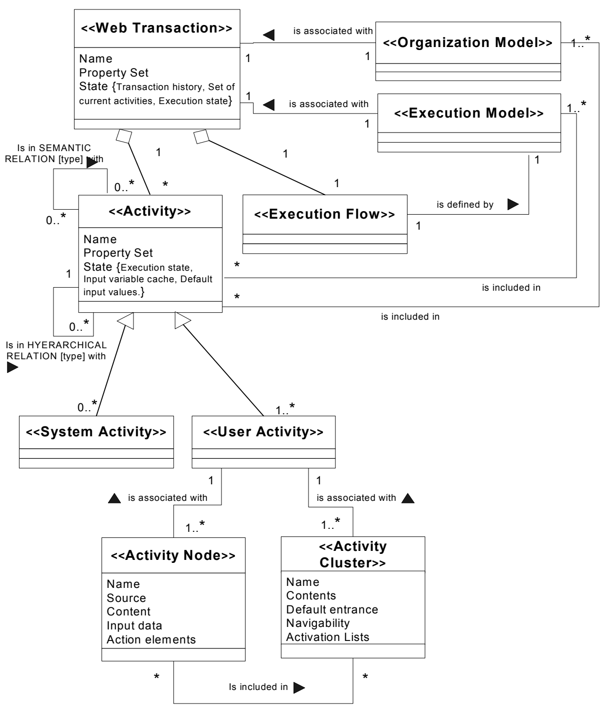


**Figure 2. The UWAT+ conceptual model.**

## 2.2 The Redesign Process

The redesign process is intended to produce the UWA conceptual design of the new Web-based version of the legacy application. The process consists of three phases [7]: (1) *Requirements Elicitation,* which is aimed at formalizing the goal of the redesign process and the requirements for the Web based version of the legacy application; (2) *Reverse Engineering*, which is aimed at abstracting information from the legacy application and formalizing them by means of some of the UWA models, thereby sketching a first draft of the new application design; and (3) *Forward Design*, which uses the requirements defined the first phase to refine the design draft defined at the second phase, thus producing the final UWA and UWAT+ design of the new version of the application.

Figure 3 provides an overview of the whole redesign process by means of the IDEF0 [10] notation. The diagram provides summary information on Input (documentation, source code, application front end, etc.), Controls (methods, meta-models, etc.), Resources (users, developers, designer) and Output (documentation, software, etc.) for each of the phases of the redesign process. Each phase is discussed in more detail in the rest of this section.

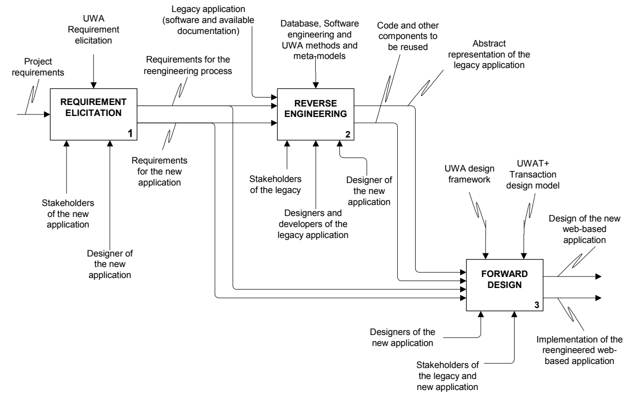


**Figure 3. The UWAT+ redesign process.**

### 2.2.1 The Requirements Elicitation Phase

The first phase of the redesign process is intended to elicit the requirements that will determine what the reengineered application will be like. The goal-oriented approach adopted in the UWA requirements elicitation activity is used. The stakeholders, their goals, and the related requirements are defined with regard to both the redesign process itself and the desired new version of the application.

Requirements that apply to the redesign process are mostly constraints to be satisfied and to be taken into account throughout the process. Often they are constraints on which features (broadly intended) of the legacy application must be conserved in the new version of the application. Another typical constraint concerns the preservation of all or a portion of the business rules implemented in the legacy application.

In contrast, requirements for the new application will mostly impact the forward engineering phase and, in particular, will include the requirements that are new compared to those of the legacy application. The designers executing the redesign process should carry out the requirement elicitation phase in close collaboration with the process stakeholders.

### 2.2.2 The Reverse Engineering Phase

The second phase has a double intent of recovering all of the valuable information held by the legacy application and abstracting this information by means of UWA models that will constitute the basis for designing the new application in the forward design phase.

The software components that can/must be reused in the new version of the application are also identified in this phase. In deciding which portions of the legacy application to analyze and which information to look for, the requirements defined in the requirements elicitation phase are considered as a driver. Since the UWA framework will be used for designing the new version of the application, where possible the UWA meta-models are used to abstract and formalize the information drawn in this phase from the legacy application. Otherwise, other well-known models of the software engineering and database practices are used.

The source of information drawn from the legacy application include the following: related documentation; the set of stakeholders and types of users of the application, their roles in the system, their goals and the requirements for the application; the business process models (or portions of them) implemented by the application; the logical and/or conceptual model of the database used by the application; a draft of the application hyperbase, navigation and publishing models; the functionalities implemented by the application; the business rules and constraints implemented by the application; the architecture of the legacy application and the software components in which it is decomposed; and the user interfaces enabling the user in executing the set of operations he can access. In particular, business processes are reverse modeled by means of the UWAT+ Transaction design models. More details on the usage of UWAT+ for reverse engineering Web transactions are available in [6].

Currently, the reverse engineering phase is largely carried out by the direct inspection of the legacy application, both from its front end and the source code/database. However, there is no reason why this process cannot be supported by appropriate automation tools, existing or tailor-made. Details on needs and benefit of tool supporting the UWAT+ method to reverse designing transaction implementing business processes in Web applications are discussed in [15].

### *2.2.3 The Forward Design Phase*

The third phase builds on the results of the previous phases and produces the design of the new Web-based version of the legacy application. The requirements defined in the requirement elicitation phase and the information drawn from the legacy application in the reverse engineering phase are used to accomplish the UWA/UWAT+-led design of the new application. The approach followed, outlined in Figure 4, is a customized version of the design methodology proposed by the UWA framework where the UWAT+ design method is used instead of the UWA transaction design activity.

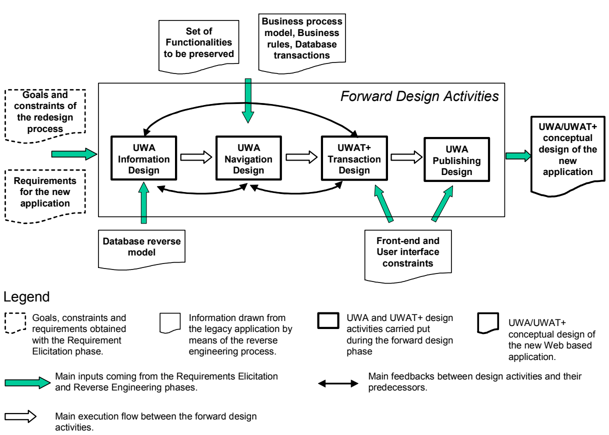


**Figure 4. The forward design phase.**

In the *UWA Information design* step, most of the Web application's content is inherited from the database of the legacy application; other information may be derived from potential new requirements. In the *UWA Navigation design* step (perhaps the most delicate of the application design process), the navigational dynamic of the application is designed. In the *UWAT+ Transaction design* step, all the non-atomic legacy user activities are considered together with the business process model to define the Web transactions that the new application will implement. In the *UWA Publishing design* step, the design of the application pages is achieved by following the traditional procedure set out in the UWA framework.

In this phase it is important to comply with the cognitive characteristics of the legacy application from which the Web application derives (User Interface Constraints). To avoid presenting the user with a drastically different application, it is important to preserve the old (established) mode of operation as far as possible. This does not contradict what was said regarding Navigation design since we are only talking here of guaranteeing that the association of information and operations on the individual page reflects the user's operating choices in the legacy application.

## 2.3 Related Work

Several approaches to reverse engineering and reengineering legacy applications can be found in the literature, but none of them, in the best of our knowledge, features all of the following characteristics:

- It works at a conceptual level, and thus is independent of the technology/architecture/platform used by the legacy application and gives the user point of view a first place in the reengineering process;
- It is specific to redesign a Web-based version of the legacy application;
- It is based on a comprehensive and robust framework for the conceptual design of data and transaction intensive Web applications, exploiting its methodologies, meta-models and tools for conducting the forward design phase and for driving the reverse engineering phase.

Among considered reverse engineering and reengineering approaches are the Iterative Reengineering of Legacy Systems (IRLS) [24], the Reverse Engineering Environment Framework (REEF) [25], the Hybrid Re-engineering (HR) [26], and Options Analysis for Reengineering (OAR) [27]. The major common shortcomings we found in the examined reverse engineering and reengineering approaches are 1) the low level of abstraction they work at and 2) the fact that they do not propose any model specific for Web application design to formalize the information abstracted during the reverse engineering process.

The literature of Web applications design methodologies offers several possibilities alternative to UWA. Among them are the Araneus Data Model (ADM) [28], the Relationship Management Design Methodology (RMM) [29], the Object Oriented Hypermedia Design Method (OOHDM) [29][31], OO-H and UWE [32], and the Web Site Design Model (WSDM) [33].

## 3. REDESIGNING GPA TO GPAWEB

This section provides an overview of the application of the redesign process presented in Section 2 to a representative legacy system. The activities related to each of the three phases of the redesign are described. The resultant Web-based system is also discussed.

## 3.1 The GPA Legacy System

The legacy system that underwent the redesign process is called GPA.[1] It is a procurement system developed in the 1990's by Biosal S.c.r.l., an ICT company based Lecce, Italy, to support the process of "call for tenders" in the procurement process of a Local Public Health Company (AUSL) in Italy. GPA is a Windows stand-alone application developed with Microsoft Visual Basic 5, using a Microsoft SQL database and a functional MDI graphical user interface.

Table 1 illustrates the complexity of the GPA software by reporting the value for size metrics such as lines of code (LOC), software components (Forms, Modules, Reports), database tables and queries. Figure 5 is a screenshot of GPA in use. In particular, the figure shows the form that enables the user to access/modify the data regarding an advertisement of a specific call for tenders.

**Table 1. Size metrics of the GPA application**

| METRIC | VALUE |
|---|---|
| # Procedure | 11.230 |
| # Form (user interface) | 68 |
| # Modules | 11 |
| # LOC (approximation by defect considering an average of 10 LOC per procedure) | 112.300 |
| # Report (printing template) | 50 |
| # Database table | 53 |
| # Database stored procedures | 18 |

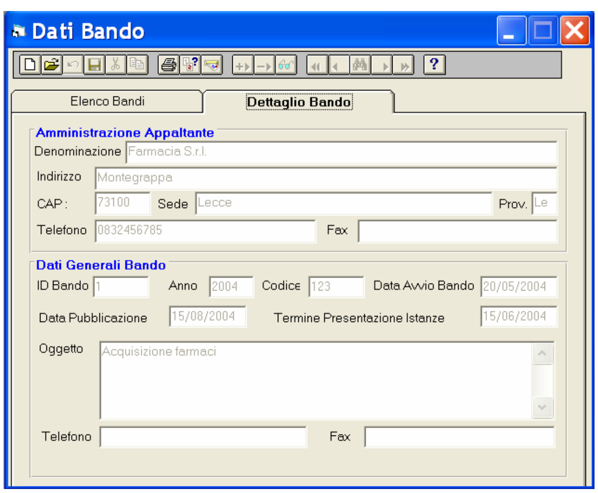


**Figure 5. One of the forms of the GPA legacy application.**

[1] GPA stands for "Gestione Procedure d'Acquito", which in Italian means "Management of the Procurement Processes".

The major motivations underlying the redesign of GPA included the high cost of maintenance and customer service the company had to deal with, changes in the requirements because of changes in the National and European regulations regarding the procurement procedures in Public Administrations (which the AULSs are), and market competition from companies offering more innovative Web-based procurement solutions.

## 3.2 The Redesign Process

The redesign of GPA to GPAWeb closely followed the redesign process described in Section 2.2. There was a requirements elicitation phase, followed by a reverse engineering phase, and then a forward design phase.

### *3.2.1 The requirements elicitation phase*

As described in Section 2.2, the first phase of the redesign process is intended to elicit the requirements that will determine the characteristics of the new system. Besides addressing the motivations underlying the redesign project mentioned above, a basic requirement was the creation of a Web application that would be easy to learn and effective to use for the end users. This requirement induced a constraint on the redesign process of preserving all the functionality of the legacy application and, where possible, of its user interface (in terms of data grouping and arrangement in each form and possible navigation between forms) because of the user habits in using the legacy system.

To properly address this constraint imposed by current users of the legacy system, the requirements elicitation phase identified key stakeholders of the new system. Among the stakeholders identified were the following:

- *Competitive tendering manager*: this person follows the entire procedure of management of a competitive tender;
- *Health unit director and Pharmacy manager*: this stakeholder asks for goods and services to be acquired by the AUSL;
- *Supplier*: someone who is interested in participating in the call for tenders, formulating and submitting its offer;
- *Technical commission*: a person in charge of the technical evaluation of the offers submitted by providers participating at a call for tenders;
- *Economic commission*: a collective in charge of the economic evaluation of the offers submitted by providers participating at a call for tendering;
- *Citizen or Generic User*: a user who accesses general information on the calls for tenders and the e-procurement process ongoing in the AUSL.

All the stakeholders identified are also real users of the system, in the sense that they use it directly for carrying out the procurement process. The Citizen is a type of user new to the Web-based version of the GPA software. The need for transparency in e-governance suggests that the Citizen be included in the list of stakeholders to be taken into account when designing the Web-based version of GPA. The Citizen will basically be allowed by GPAWeb to obtain overall information about the e-procurement process of a AUSL and the ongoing calls for tenders.

As indicated by the UWA requirement elicitation approach, for each of the identified stakeholders their high-level goals were identified. Each high-level goal was then refined in sub-goals, which in turn were further refined into sub-goals related to GPAWeb requirements. Figure 6 shows the high-level goals of the competitive tendering manager. Figure 7 illustrates a portion of the sub-goals and requirements obtained by the refinement process.

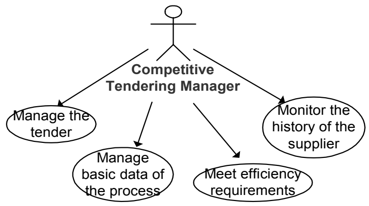


**Figure 6. High-level goals for the Competitive Tendering Manager.**

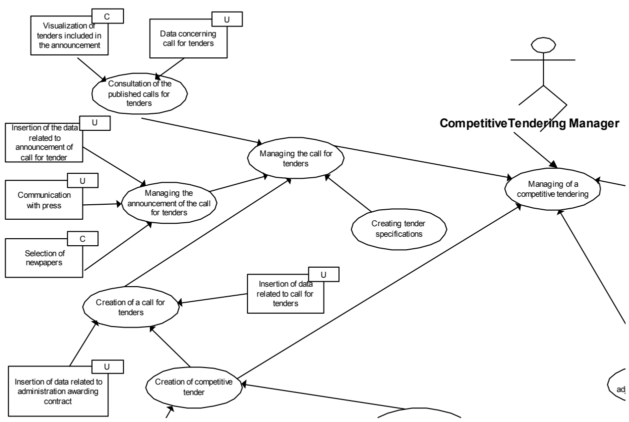


**Figure 7. Sub-goals and requirements for the Competitive Tendering Manager.**

### *3.2.2 The reverse engineering phase*

After the requirements elicitation phase, the second phase of the redesign process began: reverse engineering of the original GPA system to recover information and represent it by means of UWA models. These models serve as the basis for designing the Web-based version of the application in the third phase of the redesign process.

Data was gathered from the GPA system by defining and running a number of hypothetical call for tender operations. All the roles corresponding to the available type of stakeholders were interpreted. Particular attention was placed on the analysis and reverse modeling of the application user interface (forms and menus). The database was reverse engineered by first recovering the relational model using Microsoft Visio Enterprise edition and its database reverse engineering tools. Then the E-R model was recreated from the relational model of the database for which no documentation was available.

Table 2 schematically represents the activities of the reverse engineering phase. The activities are organized in the order in which they were carried out, showing the inputs and outputs of each activity. Inputs are basically aspects and features (broadly intended) of the legacy application analyzed. Outputs are information and models used to formalize the results of the analysis.

**Table 2. Inputs and outputs of the reverse engineering phase applied to the GPA legacy application**

| ACTIVITY # | INPUT | OUTPUT |
|---|---|---|
| 1 | Business process | IDEF0 model of the business process |
| 2 | Business process model (IDEF0), application transactions, application functionalities and business rules | UWAT+ Organization and Execution model of each of the identified user activities in the business process |
| 3 | Database and application user interfaces | Relational model, E-R diagram |
| 4 | Database E-R model and application user interface | UWA hyperbase |
| 5 | UWA hyperbase and application user interface | UWA navigation model and UWA publishing model |

It is important to note that some of the outputs of one of the reverse engineering activities are used as inputs in a following reverse engineering activity. For example, Figure 8 shows one of the Navigation clusters defined during the reverse engineering phase that refers to the GPA form reported in Figure 5. The cluster defines three navigation nodes (aggregation of data and information that is presented to the user in a single form) and the possible navigation the user can execute among them.

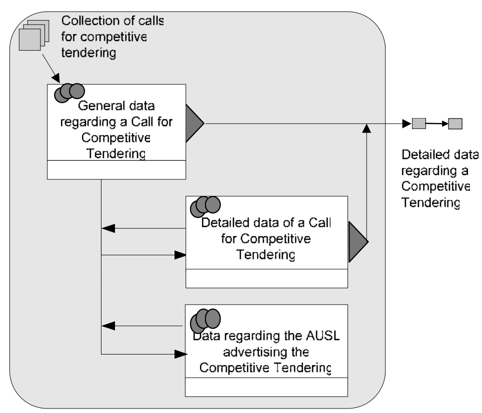


**Figure 8. The UWA Navigation Cluster derived from the GPA user interface.**

### *3.2.3 Forward Design*

The forward design phase builds upon the results of the requirements elicitation and reverse engineering phases to create the UWA and UWAT+ design of GPAWeb, which is the new Web-based version of GPA. According the process in Section 2.2 and represented in Figure 4, several design activities were carried out: information design, UWA Navigation design, UWAT+ Transaction design, and UWA Publishing design.

### *3.2.3.1 Information design*

Information design uses the draft models created in the reverse engineering phase to conduct this UWA design activity. The reverse design of the original database is also included as input to this activity. The result is an enhanced information model for GPAWeb.

### 3.2.3.2 UWA Navigation design

In accordance with the specifications of the UWA framework, this activity involves the definition of atomic units of information supplied to the user (called nodes) and the way in which they are organized in navigational contexts (called clusters). Considering the GPAWeb requirements and the model obtained with the Information design activity, the navigation model draft reverse engineered from the legacy application was refined to obtain the UWA Navigation model for GPAWeb. By reusing the navigation model abstracted from the legacy application, the constraint of preserving, where possible, the legacy application user interface was satisfied.

### 3.2.3.3 UWAT+ Transaction design

To carry out this activity, the model of the business process, the business rules and the draft of the UWAT+ transaction model recovered by the legacy application were used as input. First the recovered business process model and business rules were updated to meet the changes in the laws regulating the call for tender process in the Public Administrations in Italy. Then, the draft of the UWAT+ transaction model (Organization and Execution models designing the Web transactions that will implement the business process) was refined according to the updated business process model and the requirements for GPAWeb identified during the requirements elicitation phase. Dring this design activity, the activity nodes and activity clusters associated with each of the activities of the designed Web transactions were defined and added to the UWA Navigation model of GPAWeb obtained with the Navigation design step.

### 3.2.3.4 UWA Publishing design

The final step of the forward design of GPAWeb was Publishing Design. During this activity, navigation and activity nodes (as well as navigation and activity clusters) were arranged into pages that define the organization into informative/operative spaces of each of the pages that will be viewed by the GPAWeb users (not their layout and graphics). By preserving the cognitive characteristics of the legacy application user interface in the navigation model, the approach helps to minimize the side effects of the user dealing with a drastically new application.

Figure 9 shows a portion of the publishing model for the page of "Assign Score" that is intended to enable the Economic commission stakeholder to assign a score to each of the bids received from different suppliers for a given call for competitive tenders.

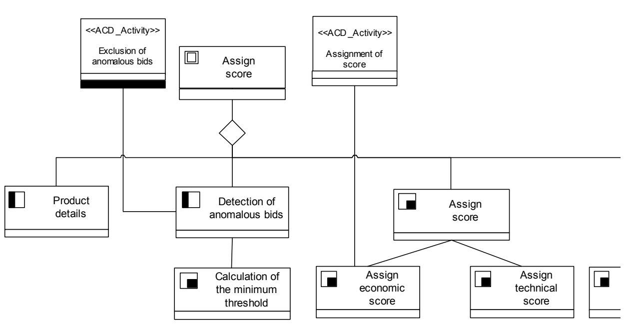


**Figure 9. Publishing model for the "Assign Score" Web page of GPAWeb.**

## 3.3 The Redesigned GPAWeb System

The redesign process described above produced the UWA/UWAT+ conceptual user-centered design of the new GPAWeb system. UWA design tools were used to assist with the creation and update of the models constituting the design. The whole redesign process required one senior analyst working part-time and two junior engineers working full-time six months. The senior analyst supervised the entire project; he had deep knowledge of (and expertise in using) UWA and UWAT+ and certain knowledge of the competitive tendering process. The junior engineers were knowledgeable in the use of UWA and the UML, but they had no prior knowledge of the Rational Rose modeling tool that was used.

The project personnel worked in close relation with Biosal for the duration of the redesign. One expert of GPA (an employee of Biosal) was part of the project. So there were two competitive manager experts of two different AUSLs were interviewed 3-4 times during the requirements elicitation phase as to who was presented the prototype that was developed to validate the design. Lastly, the project team included an expert of the competitive tendering process who was informed of the new regulation about e-procurement in Public Administrations (lows at national and European level).

Implementing a prototype of GPAWeb using Java Server Pages technology validated the new design. A screenshot of the prototype is shown in Figure 10. The final application is currently near completion in the Biosal company. It has been developed using the JavaServerFaces framework [9] to implement the Model View Controller [3][14] software architecture. The adopted DBMS is PostgreSQL [8] and Eclipse [12] IDE.

The successful creation of GPAWeb speaks to the strength of the redesign approach. Taken as a whole, the three phases of the redesign were able to recover the know-how "hard-coded" in the legacy application and transfer it in the conceptual user-centered model of the new truly Web-based version of the original system.

The redesign was successful due in part to combining the capabilities of reverse engineering with those of a robust and complete Web application design methodology codified in UWAT+, the formalization of the information abstracted by the reverse engineering process by means of UWA models, and to reusing these models in the following forward design phase.

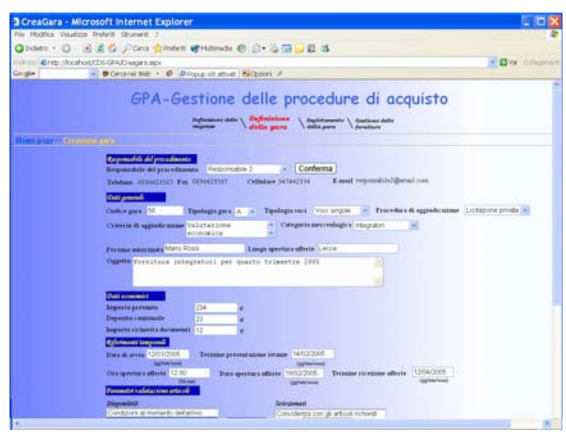


**Figure 10. A Screenshot of GPAWeb.**

# 4. DISCUSSION

There is little doubt that the effort and time necessary for conducting this redesign were minimal compared to what would have been required to design the Web application from scratch. Moreover, without a reverse engineering phase, the know-how embedded in the GPA legacy software would not have been recovered and used to design the new version of the system. Nevertheless, the case study did identify possible areas for improvement in the redesign process, as well as opportunities for tool automation to support it. This section discusses three such areas that could benefit from a further refinement of the redesign process: reducing the effort required to adopt UWAT+, leveraging the opportunities offered by the redesign process, and introducing a logical design phase that would implement the UWA conceptual model.

## 4.1 Reducing UWAT+ Adoption Costs

Because of its completeness and its richness in terms of the numbers of meta-models it includes, the application of UWA and UWAT+ “”as is” – executing all of its design activities and for all the application portions – can require considerable effort. Indeed, the effort increases with the complexity of the Web application being designed.

Nevertheless, the larger and more complex the application to design, the more the usage of a robust and well tested design methodology, such as UWAT+, is desirable, if not mandatory, in order to dominate the problem complexity and guarantee high quality of the resulting design. In this context, the term “high quality” can be interpreted from at least two points of view: quality from the user point of view (i.e., meetings user requirements and improving user experience), and quality from the client point of view (i.e., reducing cost and time to market, improving application maintainability, and so on).

Lesson learned in developing the presented case study suggest that the efforts required for the adoption of the UWAT+ methodology, both in the reverse engineering and forward design phase, may be reduced by two different means. The first is selecting which of the UWA/UWAT+ design activities to apply, and at which level of detail to which portion of the problem (business process). The second is using modeling tools to ease the creation, update, and synchronization of the UWA models. (The case study did employ a special UWA plug-in for IBM Rational ROSE [13].) In this regard, a similar extension of the UWA Rational Rose plug-in to support the UWAT+ models would be helpful to run the UWAT+ reverse engineering and forward design phases.

## 4.2 Leveraging the Redesign Process

One of the risks faced in developing the redesign project was to be too constrained by the restrictions of the legacy application, to the point of missing possible opportunities offered by the new “deploy platform” (the Web), its related technologies, and the redesign process as a whole. This risk was minimized in two ways. Firstly, by executing the requirements elicitation phase prior to the reverse engineering phase, as prescribed by the proposed redesign process. Secondly, by executing the requirements elicitation phase in tight collaboration with both the final user of the application and with experts in the business processes in question (two competitive managers of two different AUSLs were involved in the presented case study).

Mitigating the risk in this manner allowed the project to leverage the full strengths offered by the redesign process. The project personnel were able to model the “to be” version of the business process without being constrained by what the original version considered in designing GPA were. Defining the “to be” business process model and the requirements for the new version of the application prior analyzing the legacy system enable the analyst to have a reference point when choosing which parts of the legacy system to investigate, and, perhaps even more important, what to conserve and what to change. It also let Biosal pursue innovative opportunities unique to the Web and the redesign process.

As an example, in the original GPA system the suppliers were requested to install on their PC a stand-alone Windows program to prepare their offers and submit them by sending a floppy disk to the AUSL. The Web offered the opportunity for the supplier application to be Web-based (as with all of the GPAWeb system). This migration allowed their offers to be submitted online by using proper security technologies (e.g., digital signature, asymmetric coding, secure transmission protocols, and time stamps).

## 4.3 Introducing a Logical Design Phase

After carrying out the redesign process and obtaining the UWA/UWAT+ design of GPAWeb it became apparent that executing another design step prior to implementing the application might prove beneficial. What is needed is a means of translating the UWA/UWAT+ conceptual model of the application to a design model closer to the implementation.

A number of mapping rules were experimented with that allowed one to define the UML design (basically class diagrams and sequence diagrams) of the software components to be developed starting from the UWA/UWAT+ models of the application. In defining the mapping rules the MVC design pattern and the JavaServerFaces framework were chosen.

The advantages of introducing a logical design phase are considerable. By doing so, the UWA/UWAT+ design is really implemented and not only used as an abstract reference while implementing the application. It also can provide traceability between the design (at all levels, from conceptual to logical) and the implemented software. This makes it more likely maintain alignment between application documentation (the design models) and the software during all phases of the application life cycle.

However, introducing an extra phase into the process does carry some overhead. The logical design phase requires effort just by virtue of being an extra phase during redesign. There may also be higher complexity to manage in the forward design process; it is here that tool support would again prove useful.

# 5. SUMMARY

This paper reported on a case study of redesigning a legacy application for the Web using the Ubiquitous Web Applications Design Framework with an extended version of its Transaction Design Model (UWAT+). The redesign process supported by UWAT+ holistically blends design recovery technologies for capturing the know-how embedded in the legacy application with forward design methods particularly well suited for Web-based systems. The case study highlighted some of the benefits of using UWAT+ in this context, as well as identifying possible areas for improvement in the redesign process and opportunities for tool automation to support it.

As discussed in Section 4, there are a number of possible avenues of future work in this area. Incorporating more automated tool support into the redesign process, in particular in the reverse engineering phase, would help ease the adoption of the approach by making the recovery process more feasible for non-experts. It is doubtful that full automated of this process would be possible, due in part because of the relatively high level of abstraction (user perceptions and point of view) that characterizes the UWA models to be recovered.

If the proposed logical design phase was incorporated into the redesign process, a further refinement of the mapping rules would be beneficial. The use of tools in this activity might also address the adoption issue, perhaps by using generation technology (along with the aforementioned mapping rules) to derive some of the lower-level design models from their high-level UAW/UWAT+ counterparts.

## REFERENCES


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

[2] Baresi, L., Garzotto, F., Paolini, P.: "Extending UML for Modeling Web Applications". *Proceedings of the 34th Annual Hawaii International Conference on System Sciences* (HICSS-34). IEEE Computer Society, 2001.

[3] Buschmann, F., Meunier, R., Rohnert, H., Sornmerlad, P., Stal, M. *Pattern-Oriented Software Architecture - A System of Pattern*s (Volume 1). Wiley, 2001.

[4] Distante, D. "Reengineering Legacy Applications and Web Transactions: An extended version of the UWA Transaction Design Model." Ph.D. Dissertation, University of Lecce, Italy. June 2004.

[5] Distante, D., Tilley, S. "Conceptual Modeling of Web Application Transactions: Towards a Revised and Extended Version of the UWA Transaction Design Model". In *Proceedings of the 11th IEEE International Multi-Media Modelling Conference* (MMM 2005: Jan. 2-14, 2005; Melbourne, Australia).

[6] Distante, D., Parveen, T., and Tilley, S. "Towards a Technique for Reverse Engineering Web Transactions from a User's Perspective." In *Proceedings of the 12th IEEE International Workshop on Program Comprehension* (IWPC 2004: June 24-26, 2004; Bari, Italy). Los Alamitos, CA: IEEE CS Press, 2004.

[7] Distante, D., Tilley, S., and Canfora, G. "Towards a Holistic Approach to Redesigning Legacy Applications for the Web with UWAT+." To appear in *Proceedings of the 10th European Conference on Software Maintenance and Reengineering* (CSMR 2006: March 22-24, 2006; Bari, Italy). October 2005.

[8] Eclipse.org, http://www.eclipse.org.

[9] Geary, M., and Horstmann, C. S., "Core JavaServerFaces". Sun Microsystems Press Java Series, 2004.

[10] IDEF-0, Integrated-Computer-Aided Manufacturing Definition, Activity Model, Integration Definition for Function Modeling, NIST FIPS pub 183, 31 December 1993.

[11] Object Management Group (OMG): "Unified Language Modeling Specification (Version 2.0)". Online at www.omg.org, 2004.

[12] PostgreSQL.org, http://www.postgresql.org/

[13] ROSE. IBM Rational Software. Online at www.ibm.com/rational.

[14] Singh I., Stearns B., and Johnson M., Enterprise Team "Designing Enterprise Applications with the J2EE Platform, Second Edition". Sun Microsystems Press Java Series, 2004. Available online at http://java.sun.com/blueprints/guidelines/designing_enterprise_applications_2e/

[15] Tilley, S., and Distante, D. "On the Adoption of an Approach to Reengineering Web Application Transactions." Proceedings of the *4th International Workshop on Adoption-Centric Software Engineering* (ACSE 2004: May 25, 2004; Edinburgh, Scotland, UK).

[16] UWA Consortium, "Deliverable D2: General Definition of the UWA Framework." 2001.

[17] UWA Consortium, "Ubiquitous Web Applications", in *Proceedings of the eBusiness and eWork Conference 2002*, (e2002: 16-18 October 2002; Prague, Czech Republic).

[18] UWA Consortium, "Deliverable D6: Requirements Elicitation: Model, Notation and Tool Architecture." 2001.

[19] UWA Consortium, "Deliverable D6: Requirements Elicitation: Model, Notation and Tool Architecture." 2001.

[20] UWA Consortium, "Deliverable D7: Hypermedia and Operation design: model and tool architecture." 2001.

[21] UWA Consortium, "Deliverable D8: Transaction design." 2001.

[22] UWA Consortium, "Deliverable D9: Customization design." 2001.

[23] Weiss, C., and Bruner, E. "Understanding the IBM WebFacing Tool: A Guided Tour". IBM Press, 2004.

[24] Bianchi, A., Caivano, D., Marengo, V., and Visaggio, G., "Iterative Engineering of Legacy Systems", *IEEE Transaction on Software Engineering, Vol.29, NO. 3*, Mar. 2003, Page(s):225 – 241.

[25] Tilley, S. "A Reverse-Engineering Environment Framework" (CMU/SEI-98- TR-005). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, April 1998.

[26] Rosenberg, L., H., and Hyatt, L., E. "Hybrid Re-Engineering", Presented at the *3rd IEEE International Symposium on Requirements Engineering*, Jan 1997.

[27] Bergey, L., O'Brein, L., and Smith, D. "Options Analysis for Reengineering (OAR): A Method for Mining Legacy Assets", The Software Engineering Institute, Carnegie Mellon University, Jun. 2001.

[28] Atzeni, P., and Parente, A. "Specification of Web Application with ADM-2." Araneus Project, Università Roma Tre, *1.er Workshop Internactional de Tecnologia Software Orientada a Ambiente Web*, Valencia, Spain, 2001.

[29] IsaKoviwitz, F., T. , Stohr, E,. A., and Balasubramanian, P. "RMM: a methodology for structured hypermedia design". *Communications of the ACM Vol. 38, N. 8,* Aug.1995.

[30] Schwabe, D. and Rossi, G. "An Object-Oriented Approach to Web-Based Application Design." *Theory and Practice of Object Systems (TAPOS)*, Vol 4 (1998) 207-225.

[31] Schwabe, D. and Rossi, G. “Web Application Models are more than Conceptual Models.” In *Proceedings of the International Workshop on Conceptual Modelling and the Web* (1999).

[32] Koch, N., and Kraus, A. “Towards a Common Metamodel for the Development of Web Applications.” In *Proceedings of the 3rd International Conference on Web Engineering* (ICWE 2003) Cueva, J., Gonzalez, B., Joyanes, L., Labra, J. and Paule, M. (Eds) LNCS 2722, ©Springer Verlag (July 2003) 497-506.

[33] De Troyer, O. “Audience-driven web design”, In *Information modelling in the new millennium*, Eds. Matt Rossi & Keng Siau, Publ. IDEA Group Publishing (2001).