Towards a Holistic Approach to Redesigning Legacy Applications for the Web with UWAT+

Damiano Distante[1], Scott Tilley[2] and Gerardo Canfora[3]
[1,3]*Research Centre on Software Technology, University of Sannio, Italy*
[2]*Department of Computer Sciences, Florida Institute of Technology, USA*
distante@unisannio.it, stilley@cs.fit.edu, canfora@unisannio.it

Abstract

Web applications design methodologies hold the promise of engineering high-quality and long-lived Web systems and rich Internet applications. However, many such methodologies focus solely on green-field development, and do not properly address the situation of leveraging the value locked in legacy systems. This paper proposes a holistic approach to redesigning legacy applications for the Web using the Ubiquitous Web Applications Design Framework (UWA) and an extended version of its Transaction Design Model (UWAT+). The 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.

Keywords: legacy systems, redesign, migration, reengineering, Web, UWA, UWAT+

1. Introduction

Web application 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 a user interface principles, usability guidelines for online systems, and lessons learned from hypermedia development.

However, many such methodologies typically focus on green-field 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.

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. 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 paper proposes a more holistic approach to redesigning legacy applications for the Web using the Ubiquitous Web Applications (UWA) Design Framework [1][2] and an extended version of its Transaction Design Model (UWAT+) [3][4]. The 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 provides a brief overview of the UWA and UWAT+. Section 3 outlines the process underlying the holistic redesign approach. Section 4 discusses use of the approach in an on-going case study. Finally, Section 5 summarizes the paper and discusses possible avenues for future work.

2. UWA and UWAT+ at a Glance

The 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

0-7695-2536-9/06 $20.00 © 2006 IEEE

applications by adopting separation of design concerns into four main activities:

1. *Requirements Elicitation* [5]: identifying the application stakeholders, their needs, and, from these, the application requirements;
2. *Hypermedia and Operation Design* [6]: defining the application contents and their association (*hyperbase*), the possible ways for the user to access and navigate them (*access structures* and *navigation model*), the organization of contents into pages (*publishing model*) and finally the operations the user will be able to execute in the application; the hyperbase and the access structures form the *Information model* of the application.
3. *Transaction Design* [7]: designing the user activities and the system transactions that will implement the business process the Web application is supposed to support;
4. *Customization Design* [8]: defining the customizations rules enabling application ubiquitousness.

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 1) that takes into consideration the user-centered nature of business processes on the Web, and enables strong integration with the other UWA conceptual models (e.g., Information Model and Navigation Model).

The implementation of a business process in a Web application is designed by UWAT+ by means of the concept of "Web transaction". A Web transaction is defined as *a set of activities with an associated execution flow (both derived from the business process model) the user will execute to carry out a task of his/her in the considered business process*. Depending on its complexity, a business process may be designed by means of a single Web transaction or a set of Web transactions associated to a number of sub-processes.

Each Web transaction is designed with UWAT+ in two steps: the first step focuses on capturing the business process- related constraints and produces the Organization Model and the Execution Model of the Web transaction (see Figure 1). The second design step addresses the information and navigation issues related to the introduction of business processes in the Web applications. This step creates the Navigation Model associated with the Web transaction, a model composed of the Activity Nodes and Activity Clusters associated with the user activities included in the transaction. More details on the UWAT+ model and design method can be found in [3].

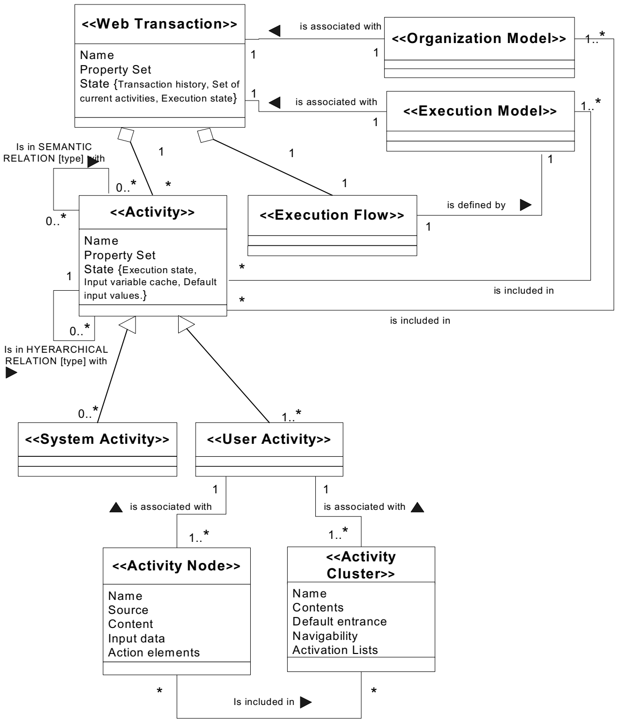

Figure 1. The UWAT+ conceptual model

3. The Redesign Process

The proposed redesign process is intended to produce the UWA/UWAT+ conceptual design of the new Web-based version of the legacy application. The process consists of tree phases [3]:

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 2 provides an overview of the whole redesign process by means of the IDEF0 [9] 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 described in more detail in the rest of this Section.

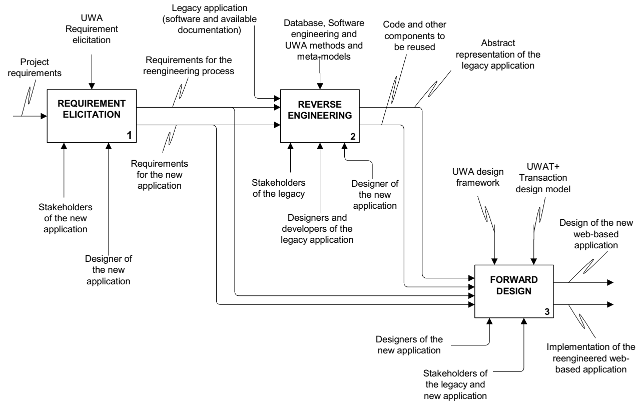
UWA Requirement elicitation
Legacy application (software and available documentation)
Database, Software engineering and UWA methods and meta-models
Code and other components to be reused
Abstract representation of the legacy application
Project requirements
Requirements for the reengineering process
REQUIREMENT ELICITATION
1
REVERSE ENGINEERING
2
Requirements for the new application
Stakeholders of the new application
Designer of the new application
Stakeholders of the legacy
Designers and developers of the legacy application
Designer of the new application
UWA design framework
UWAT+ Transaction design model
Design of the new web-based application
FORWARD DESIGN
3
Designers of the new application
Stakeholders of the legacy and new application
Implementation of the reengineered web-based application

Figure 2. An overview of the redesign process

3.1. 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 and the stakeholders of the new application.

3.2. 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 implemented by the application (or portions of them); the logical and/or conceptual model of the database used by the application; a draft of the application hyperbase, navigation and publishing models; the set of functionalities offered 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 functionalities s/he can access. In particular, the hyperbase, navigation and publishing models drafts are obtained by using the corresponding UWA meta-models. Similarly, UWAT+ meta-models are used to reverse model the business processes implemented by the application. More details on the usage of UWAT+ for reverse engineering business processes in Web applications are available in [10].

Currently, the reverse engineering phase is largely carried out by direct inspection of the legacy application, both from its front-end (user perspective) and its source code and 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 Web transactions implementing business processes in Web applications are discussed in [11].

3.3. The Forward Design Phase

The third phase of the redesign process 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 3, 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.

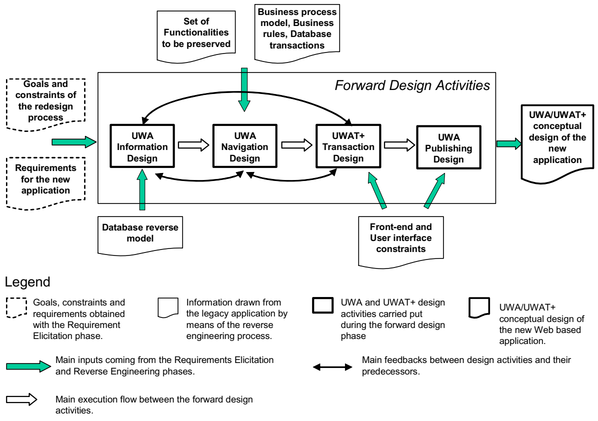
Set of Functionalities to be preserved
Business process model, Business rules, Database transactions
Goals and constraints of the redesign process
Requirements for the new application
Forward Design Activities
UWA Information Design
UWA Navigation Design
UWAT+ Transaction Design
UWA Publishing Design
UWA/UWAT+ conceptual design of the new application
Database reverse model
Front-end and User interface constraints
Legend
Goals, constraints and requirements obtained with the Requirement Elicitation phase.
Information drawn from the legacy application by means of the reverse engineering process.
UWA and UWAT+ design activities carried put during the forward design phase
UWA/UWAT+ conceptual design of the new Web based application.
Main inputs coming from the Requirements Elicitation and Reverse Engineering phases.
Main feedbacks between design activities and their predecessors.
Main execution flow between the forward design activities.

Figure 3. The forward design phase

In the *UWA Information design* activity, 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* activity (perhaps the most delicate of the application design process), the navigational dynamic of the application is designed. In the *UWAT+ Transaction design* activity, 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* activity, the design of the application pages is achieved by following the traditional procedure set out in the UWA framework. Each of the design activities listed above takes advantage from considering the draft of the UWA and UWAT+ models of the legacy applications obtained in the second phase of the redesign process.

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.

4. Experiences Using the Approach

The approach has been successfully used in the reengineering to the Web of a legacy procurement system named GPA[1]. GPA is a software system developed in the 90's by Biosal S.c.r.l., an ICT company based Lecce, Italy, and intended to support the process of "call for tenders" in the procurement process of Italian Local Public Health Companies (AUSLs). GPA is a Windows stand-alone application developed with Microsoft Visual Basic 5, using a Microsoft SQL database and adopting a functional MDI graphical user interface.

The major motivations underlying the reengineering process included high cost for maintenance and customer service the company has 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 high market competition by companies offering more innovative Web based e-procurement solutions.

The main goals of the reengineering process, besides satisfying the above motivations, also included the creation of a Web application that easy to learn and effective to use for the final users. This implied that the constraints of preserving all the functionalities 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 application.

The new Web-based version of the GPA system is currently in the final stages of its implementation. A detailed description of the redesign process conducted with the approach presented in this paper is reported in [12].

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

5. Summary

This paper presented work towards a holistic approach to redesigning legacy applications for the Web using the Ubiquitous Web Applications Design Framework (UWA) and an extended version of its Transaction Design Model named UWAT+. The 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 approach is in fact able both to recover the know-how "hard-coded" in a legacy application and transfer it in the conceptual user-centered model of a new truly Web-based version of the legacy application.

Due to the formalization of the information abstracted by the reverse engineering process by means of UWA/UWAT+ models, and to reusing these models in the following forward design phase, the effort and time necessary for conducting this phase are minimized. This is a significant savings when compared to designing the Web application from the scratch. This is particularly true when one considers that designing from the scratch would mean loosing the legacy application value.

One of the areas of future work that we are exploring is more automated tool support in some of the tasks involved in each of the three phases of the redesign process. There are substantial potential benefits through the use of codified best practice to reduce the level of skills needed by the software engineer performing the redesign of the legacy system for the Web.

6. References

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

[2] UWA Consortium, Deliverable D2: General Definition of the UWA Framework, (2001).

[3] 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.

[4] Distante, D. and 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). Los Alamitos, CA: IEEE CS Press, 2005.

[5] UWA Consortium: Deliverable D6: Requirements Elicitation: Model, Notation and Tool Architecture, (2001).

[6] UWA Consortium: Deliverable D7: Hypermedia and Operation design: model and tool architecture, (2001).

[7] UWA Consortium: Deliverable D8: Transaction design, (2001).

[8] UWA Consortium: Deliverable D9: Customization design, (2001).

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

[10] 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.

[11] 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).

[12] Distante, D.; Tilley, S.; Canfora, G.; and Huang, S. "Redesigning Legacy Applications for the Web with UWAT+: A Case Study." To appear In *Proceedings of The 28th International Conference on Software Engineering* (ICSE 2006: May 20-28, 2006; Shanghai, China) Los Alamitos, CA: IEEE CS Press, 2006.