Reverse Engineering of Web Applications to Abstract User-Centered Conceptual Models

Mario Luca Bernardi
Department of Engineering, University of Sannio, Italy
mlberna@unisannio.it

Giuseppe Antonio Di Lucca
Department of Engineering, University of Sannio, Italy
dilucca@unisannio.it

Damiano Distante *
Faculty of Economics, Tel.M.A. University, Italy
distante@unitelma.it

Abstract

The Ubiquitous Web Applications (UWA) Hyperbase model is a user-centered conceptual model representing the contents of a Web application, their organization in terms of entities and components, and the semantic associations between entities from which navigation paths are derived. Such a model may provide useful support for the software engineer during maintenance and evolution tasks.

This paper presents a strategy for the semi-automatic abstraction of UWA Hyperbase models from existing Web applications. The results from a case study, involving four applications from real world, carried out to assess the effectiveness of the strategy are also presented and discussed.

Keywords: Reverse engineering, web application evolution, user-centered conceptual models, UWA.

1. Introduction

Several methodologies have been proposed to support the development of high quality Web Applications (WAs) [18, 1, 12, 16, 3]: they usually drive the development through the definition of conceptual models mainly representing the contents, navigation and presentation structure of the application. WAs are characterized by continuous maintenance and evolution operations to meet new functional and non-functional requirements of the evolving context in which they are used. For example, new requirements may derive from the need to meet some new business rules, the need to adopt new technology, as well as the need to implement some ad-hoc functionality. The availability of up-to-date documentation, such as the models describing the application's contents structure, has a key role to successfully maintain/evolve these systems in the short time usually available to accomplish these tasks. Unfortunately, due to development and maintenance processes often constrained by short time-to-market, such documentation is often lacking. This causes maintenance and evolution becoming difficult and risky tasks potentially compromising the effectiveness and correctness of the whole system. In these cases, the usage of techniques and tools to recover models and documentation from the system to be evolved is very useful.

This paper presents an approach for the semi-automatic recovery of user-centered conceptual models from existing WAs. The results from a case study involving four WAs from the real world are also presented. The proposed approach exploits existing reverse engineering techniques and tools to extract, from the analyzed WA, information that is further analyzed and processed to abstract a conceptual user-centered model of the application's contents. The approach is based on the analysis of the client-side pages of the application and it is applicable no matter what the technologies adopted server side are. The recovered model represents the application's contents, their organization and associations, from a user-centered perspective. The recovered content model is defined according to the Ubiquitous Web Application (UWA) design framework [18], a framework comprising a methodology and set of meta-models for the user-centered design of context-aware Web applications. In particular, the recovered model conforms to the UWA Hyperbase model [19] that specifically describes the WA's contents, but any other formalism can be used instead.

This paper extends the work presented in [4] by providing more details on the recovery process and the supporting tool, and by presenting the results of a case study involving four WAs from the real world.

The remainder of the paper is organized as follows. Section 2 briefly describes the UWA Hyperbase model and the main design concepts used in this model to represent the contents of a WA. Section 3 presents the process to recover the UWA Hyperbase models from existing WAs. Section 4 shortly presents the tool supporting the recovery process. Section 5 discusses the results obtained from a case study to validate the approach. A list of related work is reported in Section 6 and conclusions and future work in Section 7.

2. The UWA Hyperbase Model

A view agreed by most of the WAs design methods proposed in the literature (including the Web Modeling Language (WebML) [1], the UML based Web Engineering approach (UWE) [12], the Object-Oriented Hypermedia Design Method (OOHDM) [16] and the Ubiquitous Web Application (UWA) [18]) sees the development of a WA defined by three models: *Contents model*, *Navigation model*, and *Presentation model*. Additional design models may be added to address specific aspects, such as the customization

of the application for different usage contexts or the design of user operations and business processes. The Contents model describes the contents that the application will present to the user in terms of their structure and semantic associations. The Navigation model specifies the units of consumption of the application contents (i.e., navigation nodes), the navigation paths through contents, (i.e., links, indexes, etc.) and the operations each node will enable. The Presentation model defines the pages into which the application will be organized, their layout, the nodes published in each page and the interface objects used to facilitate navigation and user interaction.

Focusing on the Content model, we can observe that all of the above mentioned WA design methods include a model to describe the WA's contents in terms of information concepts (entities, classes, information objects, etc.) and relationships (associations, links, relations, etc.) among them.

The UWA design framework defines the Hyperbase model [19] to represent the contents of an application, their structure and the associations among them. The UWA Hyperbase model makes use of two main design concepts: the concept of *Entity Type* and the concept of S*emantic Association Type*.

Entities Types define the fundamental classes of information the WA delivers to its users. They identify classes of objects or concepts of the "real world" of interest for the user and that will be the application "subjects". An Entity Type is organized into *Components Types* (in the same sense a book is organized into chapters) which in turn are composed of *Slots*, representing the smallest granules of information defined by an UWA Hyperbase model. An *Entity Type diagram* is a stereotyped UML class diagram describing the structure of an Entity in terms of Components and the Slots included in each Component. Moreover, this model defines the hypermedia data type associated to each Slot (e.g., text, image, video, audio, etc.), the cardinality of each Slot and Component, and the min, max and typical number of instances expected for the Entity Type. *Untyped* or *Single Entities* are Entities for which there will be a single instance in the application.

Semantic Associations Types connect two or more Entity types of the Hyperbase and represent the relations worthy of interest for the user, between the contents of the application. Semantic Association Types provide the "infrastructure" for possible navigation through the application contents and, by means of the concept of *Semantic Association Center*, define the preview of the target of a navigation link. Similarly to Entities, Semantic Associations can also appear in the form of *Single Semantic Associations* when connecting Single Entities. Semantic Associations are modeled by UML associations between stereotyped classes representing the Entities they connect. Semantic Association Centers are stereotyped UML association classes which represent the contents that will be provided to the user on a target entity instance.

Figures 4 and 5 in Section 5, respectively report an example of an Entity Type diagram and a Semantic Association diagram of one of the WA considered in the case study.

In the remainder of the paper we will refer to the recovery of UWA Entities and Semantic Associations, but, as discussed, these modeling concepts have equivalent ones in most WA design methodology. As such, the proposed approach can be used, by easily and adequately adapting it, to recover the conceptual models of the contents of an application according to other design methods.

3. The Recovery Process

UWA Entities can be recovered from WAs by searching for groups of logically related attributes in the (code of) Web pages of the applications, while Semantic Associations can be recovered identifying attributes shared by different Entities or hyper-textual links connecting portions of a same page or a different pages showing different Entities. Techniques searching the source code for groups of logically related data items are usually based on those language mechanisms for the definition and use of data structures (e.g, records, user data types, state of objects, and table schemas in databases).

We propose a more general approach based on source code analysis of the client-side pages of a WA to find groups of related attributes. In particular the focus is on groups of data items that are: (*i*) involved in the same user input/output operation (e.g. groups of data presented in a form), or, (*ii*) presented in a set of cloned client pages, i.e. client pages having the same layout and reporting the same kind of information but with different values (such as the pages showing the descriptions of products in an e-commerce application). The main motivation for analyzing only client-side pages is because the recovered model is a user-centered conceptual model. Indeed, it provides a representation of the WA at a high level of abstraction and from the user perspective, and, as known, the user accesses the WA through its client-side pages. This has the main advantage of making the approach applicable whatever the WA server-side technologies are.

Usually, labels (e.g., words used as labels of input fields in a form, words included in the table heading of a report, or, in general, labels used to specify the semantics of some data in a Web page) are used to describe the meaning of input/output data items to users; we refer to these labels as to *keywords*. Such keywords typically correspond to the Slots of an UWA Entity. Thus the approach aims to identify related groups of keywords in client pages. Each group of keywords is candidate to be a UWA Entity, while keywords shared by two or more Entities and hyperlinks between two client pages - or portions of a page - showing different Entities are the base for identifying possible UWA Semantic Associations.

Figure 1 depicts the proposed recovery process, which consists of the following four main activities:

1. Automatic extraction of groups of keywords potentially making up an application domain concept, from (HTML) client pages;
2. Validation of the identified groups of keywords;
3. Identification of UWA Entity Types from validated groups of keywords;
4. Identification of UWA Semantic Association Types from Entities with common attributes and hyperlinks between pages or portion of a page.

The process is based on the method to recover business objects models in a WA presented in [7, 8] and exploits two existing tools for the reverse engineering of WAs, namely WARE [5] and Clone Detector [6]. This method and these tools have been evolved and adapted to allow the identification of UWA Entity Types and Semantic Association Types.

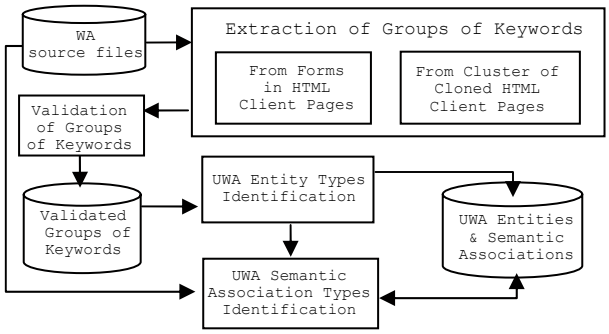

Figure 1. The process to identify UWA Entities & Semantic Associations Types

3.1. Automatic Extraction of Groups of Keywords

The identification of UWA Entities is carried out by searching for groups of related keywords both in (HTML) forms and in sets of cloned client page. Both static and dynamically built client pages are analyzed; the latter are 'captured' on the fly by a Web crawler.

3.1.1 Searching for related groups of data items in forms

A group of keywords involved in the same user input/output operation and included in the same (HTML) form or output report is considered as a possible group of Slots characterizing a Component of an UWA Entity. The rationale behind this assertion is that the set of data items that a user enters into an input form, or that are shown to a user by an output report, usually represents a concept of interest for the user in the domain of the application.

The WARE tool [5] is used to find and analyze forms in HTML pages. For each form, the labels of its input fields are considered and collected into a group of keywords identified by the name of the form and the name of the HTML page. All the groups of keywords extracted from forms are recorded into a list.

3.1.2 Searching for groups of related data items in cloned client pages

A group of cloned (HTML) client pages is characterized by the same control component (i.e., the set of items - such as the HTML code and scripts - determining the page layout, business rule processing, and event management), but a different data component (i.e., the set of items - such as text, images, multimedia objects - determining the information presented to the user). Groups of pages having exactly the same control component will exhibit the same rendering, functional behavior, and kind of content. Thus they can be considered as equivalent pages (i.e. clones), just differing for the data component they contain. Each group of cloned pages is clustered in a set of page-clones. An evolved version of the Clone Detector tool [6] is exploited to identify the set of page-clones.

For each pair of client pages the Levenshtein edit distance [6] between their control components is computed. Based on the distance value, the pages may be grouped together. Thus groups of Perfect Clones, made up by groups of pages all showing a distance equal to zero, can be formed, as well as groups of Near Perfect Clones, made up by groups of pages showing a distance in a defined range of values [dmin,dmax], may be formed. The control component of each page is derived from the flattened DOM tree of the page taking into account only the structure of the page (i.e., the HTML tags with their attributes) and filtering out values and contents within nodes.

From each group of page-clones a HTML page Template is produced. This template will have the same control component characterizing all the pages in the set and only the portion of the data component that is common to all the pages in the set (i.e., the 'cloned' portion of the data component). This common portion includes the keywords corresponding to candidate UWA Slots. Keywords belonging to the same page Template 'structure', such as HTML tables and divs, are grouped together and recorded into a list. At the present just the groups of Perfect Clones are considered for producing the page Templates.

3.2 Validation of Identified Groups of Keywords

The extracted groups of keywords have to be validated in order to:

1. Identify and solve *synonyms* (i.e., keywords or group identifiers with different names but same meaning) and *homonyms* (i.e., keywords or identifiers with the same name but different meanings),
2. Discard 'spurious' keywords eventually collected in any group (e.g., keywords extracted from forms or page-clones that have no actual associated data item, such as

labels 'Previous' and 'Next' intended to enable navigation between a set of linked pages).

3. Discard those groups of keywords that do not correspond to any application domain concept (e.g., keywords making up a menu or a navigation bar).

This step requires the intervention of an analyst knowledgeable of the application domain, supported by the tool described in Section 4.

A meaningful name is to be assigned to each validated group to describe the concept it represents. The result of this step is a list with validated groups of keywords in it.

3.3 Identification of UWA Entity Types

The validated groups of keywords are arranged into a list (`ValGrpList`) that is automatically analyzed to produce a set of candidate UWA Entities (or of Components making up an Entity). The approach to identify business objects in WAs defined in [7, 8] is exploited to this aim by adapting it to the new context. Here we recall the main points on which the approach is based.

The analysis of the list `ValGrpList` is based on two heuristic rules: (*i*) the more the references to a group of keywords in the code, the greater the likelihood that it represents a concept (application content type) of interest for the user; (*ii*) groups with small cardinality may represent more simple and atomic concepts than larger groups, and larger groups may represent more complex concepts made up of joined smaller groups. Considering these heuristics rules, the `ValGrpList` is preliminarily arranged in descending order with regard to the number of references of each group in the code (e.g., the number of times the group is referred in HTML pages), and in ascending order with regard to the arity of each group. An automatic procedure based on the one described in [7] analyzes the ordered list `OrdValGrpList` and produces the set `CandEntities` of candidate UWA Entities.

Starting from the top group in `OrdValGrpList`, the procedure analyzes each group and, if a group comprises at least a new keyword not yet included in any other group of the list `CandEntities`, it will be added to it. `OrdValGrpList` is examined until it includes at least a group, or until the union set of all the keywords of the candidate Entities in `CandEntities` and the union set of all the data items of the groups in `ValGrpList` are equal.

When a group *h* from `OrdValGrpList` includes all the keywords making up one or more groups C_i in `CandEntities`, only the *k* keywords in *h* that are not yet included in any group of `CandEntities` are added to the C_i groups whose elements are all included in *h*. The reason is that the group *h* is likely to represent a composite concept produced by a logical link among the C_i groups. The attributes that are added to the C_i groups are necessary to record this link, which will be used to deduce Semantic Association between Entities.

Each group in the `CandEntities` list will have to be assigned a meaningful name describing the concept it represents, i.e. the UWA Entity (or Component). Each keyword will correspond to a Slot of an Entity and each sub-group of keywords will be candidate to make up a Component. A validation of the groups in the `CandEntities` set is to be carried out by discarding those ones that do not correspond to a valid concept in the application domain, or re-arranging any others to better match an actual Entity. The validated Entities will make up the list `ValEntities`.

The lists `OrdValGrpList`, `CandEntities` and `ValEntities` are stored in a repository in order to be able to trace each validated Entity to the groups of keywords it derives from and the HTML pages referring it.

3.4 Identification of UWA Semantic Association Types

An UWA Semantic Association is identified for each group of validated Entities having Slots in common. The Entities with common Slots will be linked by an UWA Semantic Association. Each common Slots will be assigned to just one of the Entities in the group. The analyst intervention may be required to establish the correct assignment of the common Slots to an Entity.

Moreover, Semantic Associations are identified by analyzing the content of forms, tables, and other reports displayed to the user: if Slots from different Entities are required by a form or displayed in a report, or if Slots from different Entities are shown in the same HTML page, an Association will be considered to exist between those Entities.

Semantic Associations are also derived from hyperlinks connecting pages showing different Entities, or from 'anchors' connecting portions of the same page showing attributes of different Entities. Similarly to candidate Entities, Associations found in this step have to be validated by an human expert knowledge of the application domain.

At the end of the whole recovery process, the information on recovered UWA Entities and Associations will have been stored in a repository together with information allowing to trace the HTML pages in which each identified Entity/Association (i.e., their Slots) was found. By querying the repository, a cross reference list can be obtained showing: (*i*) the names of the identified Entities and Associations, (*ii*) their Slots, (*iii*) the name of the pages where each Entity/Association is referred, and (*iv*) the name of the Slots referred in each page.

4. Tool Support

In order to support the presented recovery process and a more general methodology that aims to recover, from a WA, the UWA conceptual models, the RE-UWA environment has

been designed and partially implemented. In this section, the Hyperbase Model Abstractor module, the module intended to support the recovering of UWA Hyperbase models, is described.

Figure 2 depicts the layered architecture of this module.

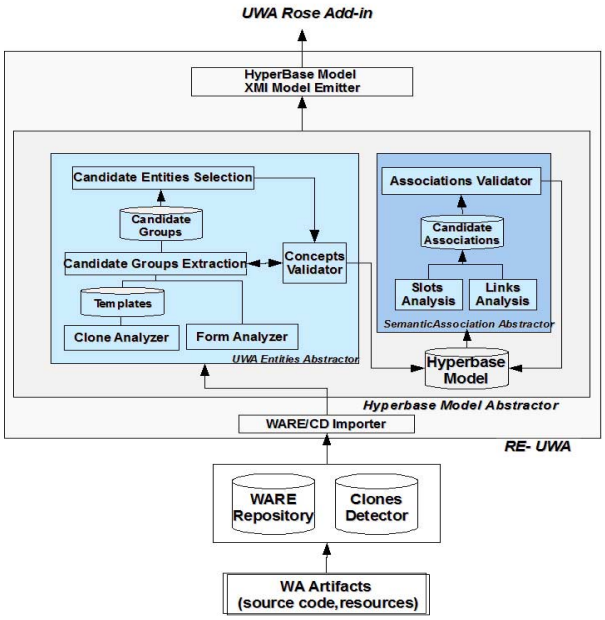

Figure 2. The architecture of the Hyperbase Model Abstractor module of the RE-UWA environment

At the lowest level of the architecture are the WARE and Clone Detector tools. WARE [5] is used to perform static analysis of the WA client-side static or dynamically generated pages to extract structural information about: (*i*) the pages making up the WA; (*ii*) the inner components of each page (e.g., forms, scripts module, frame, applet, etc.); (*iii*) the different types of hyperlinks connecting the pages (e.g., link, build, submit, redirect), etc. The extracted information is stored into the WARE Repository. Clone Detector [6] is used to perform static analysis on the client-side Web pages of the application to identify pages that are clones. The data extracted by the WARE and Clone Detector tools are made accessible to the entire RE-UWA environment and, in particular, to the Hyperbase Model Abstractor module, by the WARE/CD Importer API.

The Hyperbase Model Abstractor module comprises two main components: (*i*) the Entities Abstractor and (*ii*) the Semantic Associations Abstractor. The first is intended to support the process of recovering UWA Entities discussed in Sections 3.1-3.3; the latter is responsible for identifying UWA Semantic Associations between Entities by implementing the techniques described in Section 3.4.

The Clone Analyzer component builds up a HTML template from each set of cloned client-side pages detected by the Clone Detector tool. HTML templates are then parsed to extract groups of keywords possibly representing, after validation, UWA Entities. The Form Analyzer parses HTML forms, identified by the WARE tool, with the same objective. The Candidate Entities Selector implements the procedure described in Section 3.3 to produce the set of candidate UWA Entities.

The Concepts Validator component supports the analyst in the phase of validation of the groups of keywords extracted from HTML forms and HTML templates, as well as of the validation of the Candidate Entities. It provides a user interface enabling the browsing of extracted groups of keywords and supports the software engineer to decide if to accept, make some modification before accepting, or reject the recovered groups of data. The list of groups can be adequately sorted in order to facilitate the task of the analyst. For example, the list can be sorted on the arity of the groups, because it is likely that groups with the same arity can be synonyms or homonyms or contain synonyms/homonyms keywords.

The validated recovered UWA Entities are stored into the UWA HyperBase Model Repository.

Similar considerations can be made for the Semantic Association Abstractor. Indeed, the Slots Analysis identifies candidate Semantic Associations due to the presence of the same keywords in different Entities, while the Link Analysis identifies candidate associations due to hyperlinks linking pages, or areas of a same page, showing different UWA Entities. The Association Validator works in a way similar to the Concepts Validator.

At the top of the RE-UWA architecture is the UWA modeling tool, implemented as an add-in of the Rational Rose suite. This tool is able to create and manage UWA models, including the UWA Entity Type diagrams and Semantic Association diagrams described in Section 2, and to import/export UWA models as XMI files.

Figure 3 shows a screenshot of the RE-UWA tool used for conducting the case study presented in Section 5. In particular, it depicts the tool during the phase of validating the recovered groups of keywords for the CHL.it WA.

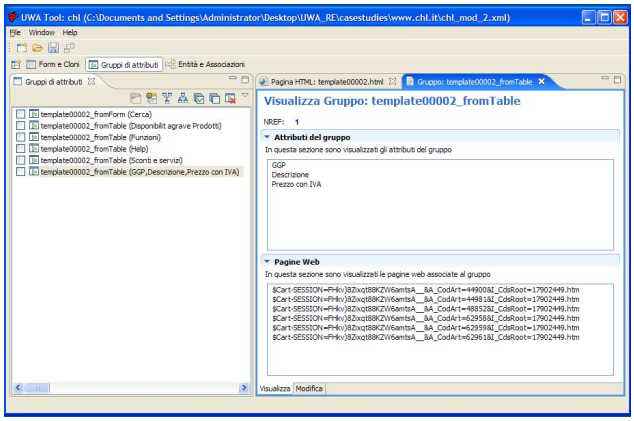

Figure 3. A screenshot of the RE-UWA tool.

5. Case Study

Some real world WAs were selected and analyzed according to the process described in Section 3, in order to validate the efficiency and the effectiveness of the approach in recovering the UWA Hyperbase model. In particular, the main aim of the case study was to verify whether:

- the groups of keywords extracted by the approach included all the actual Candidate UWA Entities;
- the Candidate UWA Entities corresponded to actual Entities;
- no actual UWA Entity was left undetected by the approach;
- the Candidate UWA Semantic Associations identified by the approach corresponded to actual UWA Semantic Associations;
- no actual UWA Semantic Association was left undetected by the approach.

In this section we present and discuss the results obtained from applying the approach to the analysis of four WAs. To reduce the risks of biasing the results, the analysis was conducted by sw engineers not involved in the definition of the approach. Three of the considered WAs are characterized by having few forms and being mainly devoted to presenting contents (lists of data items) to the user. These three WAs are suitable cases of WAs with a large number of cloned HTML pages. These WAs were:

- NGA.gov (National Gallery of Arts, www.nga.gov)
- eBay.com (http://www.ebay.com)
- CHL.it (http://ww.chl.it)

Since the approach is based on the analysis of only the client-side pages, the pages to analyze were downloaded with a Web crawler. A first, 'by hand', analysis was performed on the WAs to recognize and select sections of interest and to define the level of depth to use in the download. The sections were selected to avoid missing pages including Entities/Associations which were relevant for the application domain. The Web crawler was instructed with the parameters defined in the above way and the download of the client-side pages of each application started.

The fourth WA is an application, named CourseNet, used to support the activities related to the undergraduate courses offered by a Department of Computer Engineering of an Italian University, such as setting dates for student tutoring, or exam timetables. This WA is characterized by several HTML forms.

5.1 Results from NGA.gov, eBay.com and CHL.it

Table 6, at the end of the paper, reports a summary of the results obtained from the analysis of the first three WAs. The first column of this table reports the names of the three WAs. The second column reports the number of HTML pages downloaded from each WA. Since no forms were included in the downloaded pages, they were analyzed only by the Clone Detector tool to identify clusters of cloned pages. For each cluster, a HTML template was generated as specified in Section 3.1.2 and each template was analyzed to extract groups of keywords to be validated by the analyst and subject to the algorithm described in Section 3.3 to identify UWA Entities. The third and the fourth columns of the table report the number of clusters of perfect clones (i.e., the number of templates) and the number of groups of keywords recovered for each application, respectively. Finally, the fifth column of the table reports the number of the groups discarded by the analyst during the validation phase and the sixth the computed precision.

We can note that some of the extracted groups of keywords were discarded in the validation phase The high percentage of discarded groups for the CHL WA was mainly due to a very large number of synonyms found for this application and discarded as specified in Section 3.2. Several other groups of keywords were discarded because obtained from labels from menu and navigation bars, which did not represent any valid domain concept.

The validated groups of keywords obtained for the three WAs were submitted to the algorithm described in Section 3.3 to generate the list of Candidate UWA Entities which were finally subject to the analyst validation.

Tables 1 to 3 show the list of the UWA Entities that were finally identified using the REUWA tool and their Slots organized into Entity Components for the three WAs.

Table 1. Entities and slots identified by REUWA for eBay.com

WA	Entity / Entity_Component	Slots
eBay	**Bid**	Bidder, Bid Amount, DateOfBid
	Feedback	Comment, DateTime, From, Item
	Ebay	AboutEbay, Announcements, Security Center, Policies, Help, PrivacyPolicy, SiteMap
	Item_Overview	Name, Item number, Buy It Now price, Current bid, End time, Shipping costs, Shipping Service, Service to, Ships to, Item location, History, High bidder, Larger picture
	Item_Listing and Payments Details	Starting time, Duration, Payment methods
	Item_Description	Description
	Item_Shipping and Handling	Shipping cost for a single item, Cost for each additional item, Destination, Shipping service, Shipping insurance
	Item_Return Policy	Return Policy
	Item_Payment Details	Payment method, Preferred/Accepted, Buyer protection on eBay
	Member	MemberID, Feedback score, Positive feedback, Member since, Members who left a positive, Members who left a negative, All positive feedback received, # Bid retraction

Table 2. Entities and slots identified by REUWA for NGA.gov

WA	Entity / **Entity**_Component	Slots
NGA	**Work of art**	Full Screen Image, Bibliography, Conservation Notes, Detail Images, Exhibition History, Provenance, Inscription, Location, Audio
	Artist	Name, Nationality, Birth year, Death year, Biography, Bibliografic references
	School_Tour Request	Contact person, Title, Name of school, School address, City, State, County, Zip code, School fax, Home telephone, E-mail address
	Current exhibition	Organization, Sponsor, Schedule, Passes
	Past exhibition	Title, Overview, Web Site, Location, Attendance, Catalogue, Other venues
	Feedback	Full name, Email Address, Question or Comment

Table 3. Entities and slots identified by REUWA for CHL.it

WA	Entity / **Entity**_Component	Slots
CHL	**AuthenticationData**	Username, Password
	Item	GGP,Price with Taxes, Quantity, Description
	Product	Payment e Handling, CHL Price, Productor Warranty, Average rate, CHL Warranty, Printable report, Shipping cost, CHL Promotion, Discount, Code
	Product_Sheet	General Description
	Product_Details	Technical specifications, Composition
	Comment	Author, Text, Data
	CHL_Location	Shipping centers, News from Popitt
	CHL_Warranty	Components Warranty, Warranty for Assembled PC
	CHL_Payment and Delivery	Shipping Costs, Delivery Costs, Handling costs
	CHL_The Community	Virtual Money Box, Forum, Products Votes, Comments
	CHL_After-buying Services	Withdrawal right, Erroneous Shipping, Handling Not Working Items, Incomplete Items, Items damaged during shipping, Decree Low n°185
	CHL_In Short	CHL in Short, Buying on CHL, Join in CHL, How to find Products, Technical Sheet, Shopping Cart, Assembled PC , Product Order, After buying

The next step of the analysis was the identification of the Semantic Associations among the recovered Entities. Both the presence of common Slots among Entities, the presence of attributes of different Entities in the same page (i.e. template), and the presence of hyperlinks between pages presenting different Entities were used to identify Associations. Slots common to more Entities were assigned to just one Entity. The list of identified Associations for the three analyzed WAs is reported in Table 4.

All the Candidate Entities proposed by the tool at the end of first step of the recovery process were validated by the analyst, while some candidate Semantic Associations were discarded because redundant. Each Entity and Slot in Tables 1 to 3 is identified by the name given to them by the analyst that carried out the analysis. The Slots are those resulting after the step of identifying Semantic Association, i.e. Slots common to more Entities have been assigned to just one Entity.

Table 4. Associations identified by for eBay, NGA and CHL

WA	Associations
eBay	Receives(Member, Feedback)
	Offers (Member, Item)
	Receives (Item, Bid)
NGA	CreatedBy (Works of art, Artist)
	IsAuthorOf (Artist,Works of art)
	AssociateArtists (Artist, Works of art)
	HasWorksAfter (Artist, Works of art)
CHL	Is Associated To (Product, Comment)
	SimilarTo (Product, Product)
	SamePriceOf (Product, Product)

Figures 4 and 5 show, respectively, the UWA Entity Type diagram and Semantic Associations Type diagram drawn by the expert for the eBay.com WA. In drawing the diagrams, the expert structured the Entities into Components, each one corresponding to an Entity identified by the proposed approach.

To verify the efficiency and the effectiveness of the proposed approach, three analysts, one for each WA, knowledgeable of the application domain and expert of UWA, were asked to analyze (without the support of the RE-UWA tool) the NGA.gov, eBay.com and CHL.it WAs to identify UWA Entities and Semantic Associations. Thus, each expert was provided with the WA pages 'captured' by the crawler and analyzed the pages 'manually' just using a browser.

For the NGA WA, the experts found the same Entities proposed by the tool and validated by the analyst, while for the eBay and CHL WAs the experts found in both two cases one more Entity. In both cases an Entity identified by the tool and validated by the analyst was decomposed in two smaller Entities by the experts. Thus no Entity was lost by the proposed approach but just a different granularity of aggregation was used. Moreover, some differences in the names given to some Entities, Slots and Associations were found. As far as the Semantic Association is concerned, the experts just identified the actual Semantic Associations existing among the Entities, i.e., they did not considered the redundant Associations identified when using the tool in the semi-automated recovery process.

However, the experts spent a larger amount of time than the analyst who used the RE-UWA tool to get the, almost, same results. They needed an average of about 34% of more time, but this time is expected to increase as the dimension of the analyzed applications increases.

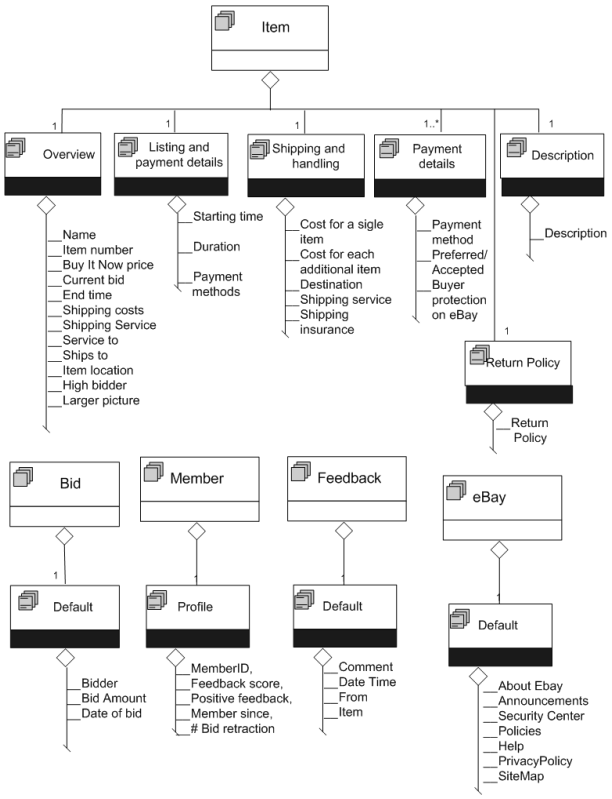

Figure 4 – Entities identified by RE-UWA for eBay.com

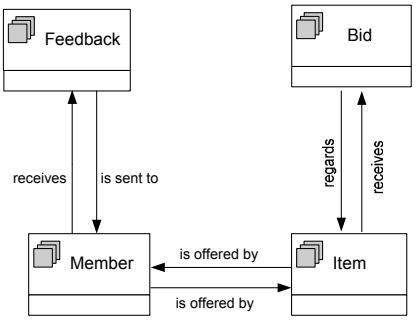

Figure 5 – Associations identified by REUWA for eBay.com

5.2 Results from the CourseNet WA

This WA is characterized to have a high number of forms used for input/output operations. No client-side cloned pages were identified in it and so the groups of keywords were extracted just from analyzing the forms. The analysis retrieved 46 groups of keywords. The validation phase revealed that some forms referred the same group of keywords, even if they had been assigned different names by the tool, i.e. they were synonyms. This is due to the fact that the name the tool assigns to a group includes the name of the page including it, thus also for forms with a same name but included in different pages the complete names are different, producing synonyms. Some more few groups of keywords were discarded because associated to forms that just allowed the user to make some selections, i.e. they acted as menu, then these groups did not correspond to any concept of the application domain. At the end of the validation phase, 24 groups of keywords were found as valid ones. The resulting groups were submitted to the procedure which identifies the candidate UWA Entities. Eight candidate Entities were identified and all proved to be valid by the analyst. Table 4 reports the list of the identified Entities.

In the last step of the recovery process, 14 candidate Associations were identified among the Entities, by considering common attributes between Entities and hyperlinks between pages including different Entities. All the 14 identified Associations were considered to be valid by the analyst.

Also in this case an expert of the application domain and UWA was asked to 'manually' analyze the client side pages of the WA and identify the UWA Entities and Semantic Association. In this case the expert identified the same Entities and Associations recognized by the proposed approach, i.e. no Entity or Association was found left undetected by the approach, no more Entity or Association was found by the approach also in this case.

Table 5. Entity types identified in CourseNet

Entity/ Component	Slots
Student	Student name, Student surname, Student code, Student email, Student phone number, Student password
Teacher	Teacher name, Teacher surname, Teacher email, Teacher phone number, Teacher password, Teacher code
Exam Session	Course code, Exam date, Exam time, Exam classroom
Tutoring	Tutoring date, Tutoring start time, Tutoring end time, Course code, Course name, Student code, Teacher surname
Course	Course code, Course name, Course academic year
Tutoring Request	Student name, Student surname, Student code, Tutoring request date, Teacher surname, Teacher name
News	Course code, News text, News number, News date, Teacher code
Exam Reservation	Student code, Student name, Student surname, Course code, Exam date, Exam reservation date

5.3 Discussion

The results from the presented case study and from some other similar WAs' analyses suggest some considerations about the proposed approach.

The identification of groups of keywords from forms proved to produce better results than those ones from cloned pages. This is because, often, the groups of keywords identified in HTML templates do not actually represent a concept of the application domain but they derive from page structures such as menu, navigation bars or other elements which are useful for navigation across the application pages but which have no meaning with regard to the application domain.

Furthermore, forms are usually used to request input data related to some concept of the application domain. Moreover the number of data groups extracted from templates was larger than those extracted from forms.

As previously mentioned, this was mainly due to templates generated from non-perfect page-clones which gave origin to synonyms and duplicated groups of keywords.

To improve the quality of groups extracted from templates, some techniques from the information retrieval field can be used. These might include the use of: (*i*) stop word lists to avoid taking into consideration the usual words in menus and navigation bars; (*ii*) techniques permitting the identification of synonym groups, thus reducing the number of groups to validate.

6. Related Work

Several approaches for the reverse engineering of WAs have been proposed in recent years. They differ in the aspects they focus on, the level of abstraction of the recovered information and the formalism they adopt to represent it. The works presented in [5, 11, 15] focus on recovering an architectural view of the WA depicting its components (i.e., pages, page components, etc.) and the relationships among them at different levels of detail. In [9], an approach for abstracting a description of the functional requirements implemented by the WA is proposed. UML use case diagrams [13] are used to represent abstracted requirement information. A technique and an approach for reverse engineering UWA Web Transactions models representing the business processes implemented by a WA from a user centered perspective are presented in [10] and [17]. The VAQUISTA [20] system by Vanderdonckt et al. allows the presentation model of a web page to be reverse engineered, in order to migrate it to another environment. The TERESA tool presented in [14] produces a task-oriented model of a WA by source code static analysis, where each task represents single page functions triggered by user requests. The resulting model is suitable for assessing WA usability, or for tracing the profile of the users of the analyzed WA. In [11] Estievenart et al. propose a tool-supported method to reengineer static web sites. The tool analyzes the pages of the site, trying to identify Web site concepts and alternative layouts for their presentation. The abstracted information is stored in XML schemas that can be used to build the database of a new version of the site. In contrast to works cited in this section, and others proposed in literature, our reverse engineering approach refers to a robust and complete methodology, specific for the conceptual design of WAs to abstract models which features a user-centered perspective on the application. No other work, to the best of our knowledge, deals with the recovering of such user-centered conceptual models. Moreover, being that our approach is based on client-side source code analysis, it is applicable to any WA producing as front-end HTML pages, regardless of the technologies used server-side.

7. Conclusion and Future Work

This paper presented an approach to recover user-centered conceptual models from an existing WAs. In particular the approach is able to abstract from the analyzed application, the model representing its contents, as perceived by the user. The model is formalized according to the Ubiquitous Web Application design framework in terms of Entities and Semantic Associations, which are modeling concepts representative of those adopted in the content model of any other WA design methodology. The recovery process can be used for re-documentation, maintenance and evolution purposes. The abstracted models, indeed, can offer a useful support to the maintainer during maintenance and evolution tasks by providing a representation of the application contents, their structure and semantic associations, as presented to the user. The maintainer can use these models to reason about possible evolution interventions to enable the application meet new requirements and user expectation. The case study we carried out showed that the approach is feasible and valid, and highlighted some possibilities of improvement, both for the process and the supporting tool. Indeed, for all the analyzed WAs, the approach was able to correctly identify the same UWA Entities and Associations that were identified by a human expert conducting the analysis manually.

A first consideration is that the process is sensitive to the presence, within forms and pages, of structures reporting explicit labels. Improvements are needed mainly in the extraction of groups of related keywords from clusters of HTML page-clones. Information retrieval techniques can provide useful support to this aim. A further consideration that emerged from the study is that the identification of Semantic Associations is less sensitive than that of Entities to the expertise and domain knowledge of the analyst. Possible improvements may also be reached by complementing the current recovery process with the analysis of the WA's server side code. In particular, the identification and analysis of SQL queries could provide useful and more precise information to Entity/Association identification. Of course, this would require the availability of the entire WA's source code, but it would enable correlating the abstracted models with the server-side application components, thus offering better support for maintenance and evolution tasks.

Future work will also consider the extension of the approach and of the RE-UWA tool to recover the other UWA models. These will include: (*i*) the UWA Navigation Model (which defines the units of information delivered as a whole by the application to the user and the navigation path among them), (*ii*) the UWA Publishing Model (which describes how the WA is organized into pages and which information is presented in which page) and (*iii*) the UWA Operation and Transaction Models (which describe, respectively, the user operation and the business processes implemented by the WA).

Table 6. Summary of the results obtained for the NGA, eBay.com and CHL WAs

WA	# Pages	# Clusters of perfect clones	# Groups of keywords	# Groups of keywords discarded during validation	Precision
NGA	82	17	26	6	76,92%
eBay	935	133	10	3	70,00%
CHL	273	53	295	262	11,19%

References

[1] Ceri, S., Fraternali, P., Bongio, A. "Web Modeling Language (WebML): a Modeling Language for Designing Web Sites". in Computer Networks, 33 (2000).

[2] Chung, S., Lee, Y.S. "Reverse software engineering with UML for web site maintenance". In Proceedings of 1st International Conference on Web Information Systems Engineering, 2001, IEEE CS Press, Los Alamitos, CA.

[3] Conallen, J., "Building Web Applications with UML – 2nd Edition". Addison Wesley Publishing Company: Reading, MA, 2002.

[4] Di Lucca, G. A., Distante, D., Bernardi, M. L. "Recovering Conceptual Models from Web Applications." In Proceedings of the 24th International Conference on Design of Communication, October 25-27, 2006, Myrtle Beach, South Carolina, USA). ACM Press: New York, NY, 2006.

[5] Di Lucca, G. A., Fasolino, A. R., Tramontana, P. "Reverse Engineering Web Applications: the WARE Approach", Journal of Software Maintenance and Evolution: Research and Practice, John Wiley & Sons, Ltd., Chichester, England, vol. 16, Jan. 2004.

[6] G. A. Di Lucca, M. Di Penta, A.R. Fasolino, An approach to identify Duplicated Web Pages, *Proc. of 26th COMPSAC,* IEEE Computer Society Press, 2002.

[7] Di Lucca, G. A., Fasolino, A.R., De Carlini, U., Tramontana, P. "Recovering a Business Object Model from Web Applications". In Proceedings of 27th IEEE Annual International Computer Software and Applications Conference, Dallas, USA, November, 2003, IEEE Comp. Soc. Press, Los Alamitos, California.

[8] Di Lucca, G. A., Fasolino, A.R., De Carlini, U., Tramontana, P. "Abstracting Business Level UML Diagrams from Web Applications". In Proceedings of the 5th IEEE International Workshop on Web Site Evolution. Amsterdam, The Netherlands, 22 Sept. 2003, IEEE Comp. Soc. Press, Los Alamitos, California.

[9] Di Lucca, G. A., Fasolino, A.R., De Carlini, U., Pace, F., Tramontana, P. "Comprehending Web Applications by a Clustering Based Approach". In Proceedings of 10th IEEE Workshop on Program Comprehension, IEEE CS Press.

[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 International Workshop on Program Comprehension 2004: June 24-26, 2004; Bari, Italy, Los Alamitos, CA: IEEE Computer Society Press, 2004.

[11] Estievenart, F., Francois, A. "A tool-supported method to extract data and schema from web sites", In Proceedings of the 5th International Workshop on Web Site Evolution, Amsterdam, The Netherlands, 2002.

[12] Koch, N., Kraus, A. "The Expressive Power of UML-based Web Engineering". In Proceedings of 2nd International Workshop on Web Oriented Software Technology at ECOOP´02. June 10, 2002. Málaga, Spain.

[13] Object Management Group (OMG). Unified Language Modeling Specification (Version 2.0). Online at www.omg.org.

[14] Paganelli, L., Paterno, F. "Automatic Reconstruction of the Underlying Interaction Design of Web Applications". In Proceedings of the 14th International Conference on Software Engineering and Knowledge Engineering 2002. ACM Press: NY, 2002.

[15] Ricca, F., Tonella, P. "Understanding and Restructuring Web Sites with ReWeb", IEEE Multimedia, 2001, 8(2): 40-51.

[16] Schwabe, D., Rossi, G. "An Object Oriented Approach to Web-Based Application Design". Theory and Practice of Object Systems 4(4), 1998. Wiley and Sons, New York.

[17] Tilley, S., Distante, D., and Huang, S. "Design Recovery of Web Application Transactions." In Advances in Software Evolution with UML and XML (Editor: Hongji Yang). Hershey, PA: Idea Group Publishing, May 2005.

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

[19] UWA Project Consortium. Deliverable D7: Hypermedia and Operation design: model and tool architecture. 2001.

[20] Vanderdonckt, J., Bouillon, L., Souchon, N. "Flexible reverse engineering of web pages with VAQUISTA." In Proceedings of Eighth Working Conference on Reverse Engineering - 2001. IEEE Computer Society Press, Los Alamitos, CA, 2001.