Software Tools for Technology Transfer manuscript No.
(will be inserted by the editor)

The RE-UWA Approach to Recover User Centered Conceptual Models from Web Applications

Mario Luca Bernardi[1], **Giuseppe Antonio Di Lucca**[1], **Damiano Distante**[2]

[1] Department of Engineering, University of Sannio, Italy e-mail: mlbernar|dilucca@unisannio.it
[2] Faculty of Economics, Tel.M.A. University, Italy e-mail: distante@unitelma.it

Abstract. Large scale Web Applications, especially those intended to publish contents and provide information to their users, are by their nature subject to continuous and fast changes. This often means fast obsolescence of the design documentation and a lot of effort required to comprehend the application when performing maintenance and evolution tasks. This paper presents a reverse engineering approach for Web Applications enabling the semi−automatic recovery of user−centered conceptual models describing, from a user perspective, key aspects such as the delivered contents and navigational paths. The abstracted models are formalized according to the Ubiquitous Web Applications (UWA) design methodology, but any other design method for Web Applications could be used instead.

The paper describes the recovery process, a tool developed to support the process, and the results from a case study conducted to validate the approach on a set of real world Web Applications.

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

1 Introduction

The success of a software application depends considerably on the level of external quality perceived by its users, i.e., on how well it supports the user in achieving the ultimate goals for which the application has been conceived.

For a Web Application (WA), in particular for information intensive WAs, the contents it provides, the navigation through contents it supports, and the way it presents contents to the user are key aspects influencing such quality:

- Contents and associations between contents have to be of interest for the users to which the application is directed;
- Navigation has to be organized in such a way that finding and accessing contents is easy and effective;
- Presentation has to be conceived in order to offer valuable and attractive views over contents.

The above list represents a set of general requirements which need to be specialized and satisfied for the different classes of users of the application as these might have considerably different goals and related requirements. Contents, navigation and presentation are indeed three layers of design on which basically all of the most known Web engineering approaches organize the design of a WA by devoting to each of them a specific design activity and a specific design model [2,9,20,4].

WAs are also characterized by continuous evolution to meet new functional and non-functional requirements of the changing context in which they are used. For example, new requirements may derive from the need to implement some new business rules, the opportunities provided by new technology, or the need to implement some ad-hoc new functionality. The availability of up to date documentation, such as the models describing the application's contents and navigation structure, has a key role in the successful maintenance and evolution of these systems. Unfortunately, due to development and maintenance processes often constrained by short time-to-market and resources, such documentation is often lacking. This causes maintenance and evolution becoming difficult and risky tasks potentially compromising the correctness and effectiveness of the whole system. In these situations, the usage of techniques and tools enabling the semi-automatic recovery of models and documentation from the system to evolve is useful and necessary.

Several approaches and tools for the reverse engineering of WAs have been proposed in the literature. Some of

them aim at obtaining an architectural view of the WA that depicts WA components (e.g., pages, or inner page components) and their relationships at different levels of detail [3,19,14]. The approach in [13] allows abstracting a description of the functional requirements implemented by the WA which is cast into UML use case diagrams [17]. Some others else recover UML class diagrams of the application business objects and the logical relationships between them [15], or models of the business processes implemented by the WA [7]. Anyway, the most of the existing reverse engineering methods and tools usually model a WA at a low level of abstraction, often in terms of pages and page components, and aren't able to describe the application from the user's point of view. Also the results of these approaches offer partial views on the application (navigation structure, business objects, business process, etc.) and use different representations and meanings.

User-centered conceptual models, by enabling the representation of the application from a user perspective and at a high level of abstraction, can provide effective support for the maintainer when deciding on some change/improvement to be applied to the application, with respect to its external, user-perceived, quality. Conceptual models are also useful when migrating a WA towards different technologies. Indeed, since these models are independent from implementation and technological aspects, they are suitable for being implemented in any possible technology.

The Ubiquitous Web Application (UWA) design framework [22–24] provides a methodology and a set of meta models for the user centered design of context-aware WAs. In particular, the UWA Hyperbase and the Access Structure models are specifically intended to represent the contents of the application, the associations among contents, and the different views (selections) of contents it offers to the user.

This paper presents an approach for the semi-automatic recovery of user-centered conceptual models from existing WAs according to the UWA design methodology. The approach is based on reverse engineering techniques applied to the client−side pages (static or dynamically generated) of the application. As such, the approach is applicable independently from the server-side technologies adopted to implement the WA. The recovered models represent the structure of contents, their semantic associations, and the access structures to contents, as viewed by the final user. The recovered models conform to the UWA Hyperbase and Access Structure models, but any other modeling formalism can be used instead. Additionally to describing in detail the proposed recovery process, the paper also presents a tool developed as an Eclipse IDE to support the process and the results from a case study involving six WAs from the real world.

Compared to our previous work on the subject [1], this paper presents:

1. An extended and revised version of the recovery process which now supports the recovery of the UWA Hyperbase model (entities and semantic associations) and the UWA Access Structures model (collections), and which is less sensitive to the presence of keywords in the analyzed WA.
2. An enhanced version of the prototype tool which is now implemented as an Eclipse IDE and which supports the extended and revised version of the recovery process.
3. An extended case study, which now involves three additional WAs.

The reverse engineering approach proposed in this paper exploits analysis and recovery techniques presented in [15] and [16] which define, respectively, a method to recover a business object model from WAs and to identify duplicated client web pages. These methods have been adapted to the new context and improved to get better results; in particular, the method to identify duplicated client Web pages now allows to identify and analyse also similar pairs of pages and not just perfect cloned ones.

The remainder of the paper is organized as follows. Section 2 briefly describes the UWA Hyperbase and Access Structure models with the main modeling concepts used in them to represent the contents of a WA and views on contents. Section 3 presents the process to recover the UWA Hyperbase and Access Structures 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 and Access Structures Models

Most of the WA engineering approaches available in the literature define the design of a WA by means of three main models: the model of contents (a.k.a., information model, domain model or hyperbase model), the model of navigation (navigation model), and the model of presentation (presentation model). The list of such approaches includes the Object-Oriented Hypermedia Design Method (OOHDM) [20], the UML based Web Engineering approach (UWE)[9], the Web Modeling Language (WebML) [2] and the Ubiquitous Web Application (UWA) design framework [24]). Each of these approaches adopts a more or less different notation to represent the models listed above and some propose additional models for designing specific aspects involved in a WA, such as the supported business processes [5,11] and the customization for the different contexts of usage [23]. The Information Model describes the base contents of the application, their elementary structure and the semantic relationships between different classes of contents. The Navigation Model organizes contents into reusable units of

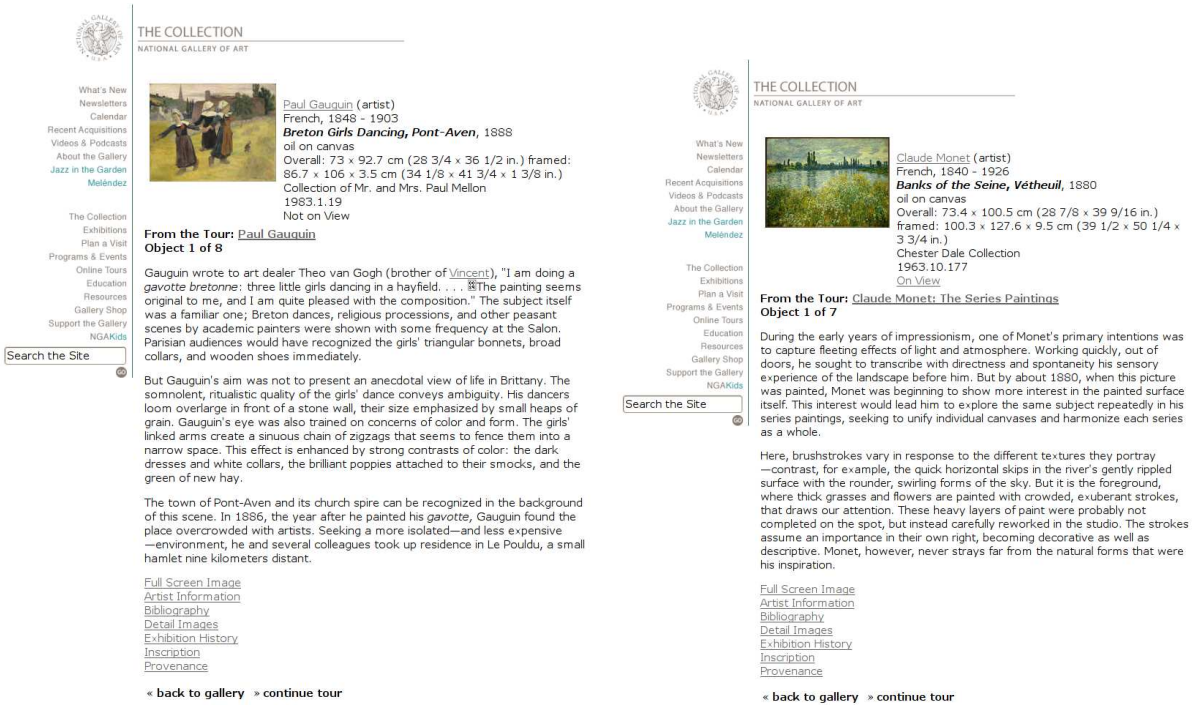

Fig. 1. Two Web pages presenting Work of Arts at NGA.gov

consumption named navigation nodes, defines possible navigation paths through these nodes, and defines the user operations each node will enable. The Presentation Model organizes the application in terms of pages, associates nodes to pages, defines the layout of pages, and specifies which are the interface objects used to facilitate navigation and user interaction.

In the case of the UWA design methodology, the Information Model is composed of two sub-models: the Hyperbase Model and the Access Structures Model [22]. The Hyperbase model describes the base contents of the application, their structure and the semantic associations among them, and makes use of two main design concepts: Entity Type and Semantic Association Type.

UWA Entity Types define the fundamental classes of information the WA delivers to its users. They identify classes of objects of the considered domain which are of interest for the user. An Entity type is structured into Components (in the same sense a book is organized into chapters) which in turn are composed of Slots. Slots have an associated type (e.g., text, image, video, audio, etc.) and represent the smallest granules of information defined by the UWA Hyperbase Model. Entity Types are modeled by means of Entity Type Diagrams which are stereotyped UML class diagrams describing the structure of one or more Entities in terms of their Components and Slots. This model also defines the data types associated to each Slot, the cardinality associated to each Component and Slot, and the min, max and typical number of instances expected for each Entity. Untyped or Single Entities are Entities for which there will be a single instance in the application's Hyperbase[1]. Figure 1 and Figure 2 report, respectively, two Web pages presenting Works of Art of the painter Monet and the page presenting the painter (Artist) from the National Gallery of Art Web site[2]. Related to these pages, Figure 3 reports the UWA Entity Type Diagram modeling the Entities "Work of Art" and "Artist" with their Components and Slots (these models are part of the results of the case study presented in Section 5).

UWA Semantic Associations[3] are unidirectional relationships defined between pairs of Entities (source and target) of the the Hyperbase. Semantic Associations provide the "infrastructure" for possible navigation paths through the contents of the application. A Semantic Association has an associated Semantic Association Center which defines the selection of slots derived from the target Entity, used as its preview. Semantic Associations are modeled with UWA Semantic Association Diagrams in which the source and target Entities are connected by UML associations and a UML association class represents the Association Centers. Similarly to Entities, Semantic Associations can be Single (when they connect Single Entities) and Typed. Figure 5 depicts the Semantic Association Diagram corresponding to the association "is created by" between the Entities "Work of Art"

[1] In e−commerce Web sites, a typical example of such entities is that collecting information on the company running the business, the applied commercial policies, etc., which have a single instance in the whole Hyperbase.

[2] www.nga.gov

[3] The term "Semantic" means that the association represents a relation between two Entities qualified by a meaningful name.

Fig. 2. A Web page presenting an Artist at NGA.gov

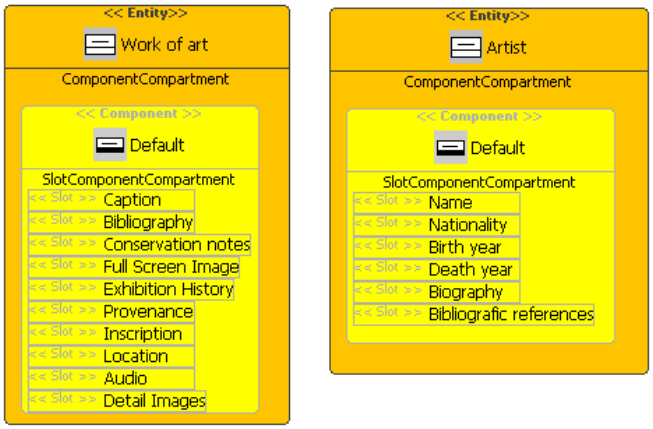

Fig. 3. The UWA Entity Type diagram for the entities Work Of Art and Artist at NGA.gov

Fig. 4. A Web page presenting (and providing access to) a guided tour of Works of Art in NGA.gov

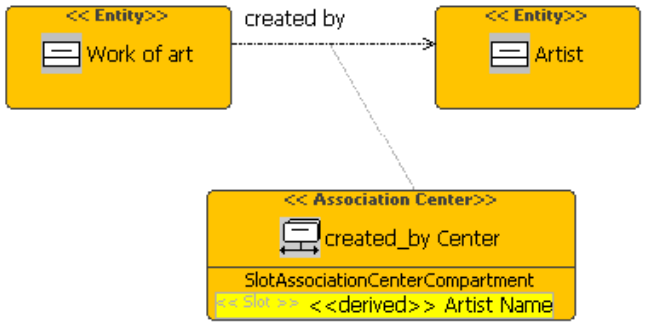

Fig. 5. The UWA Semantic Association diagram for the association WorkOfArt - is created by - Artist at NGA.gov

and "Artist" in NGA.gov. This association models the fact that an "Artist" is the author of a "Work of Art". In this site the name of the Artist is used as preview information for the link connecting the page showing a Work of Art with the page providing information on its Author.

UWA Access Structures (also named Collections) are selections of Entity instances intended to provide the user with interesting and purposely defined views on the contents of the application. A Collection may involve different Entities and Entity instances (members) taking part in the collection that are determined by the selection criterion defined for it. Each Collection has an associated Collection Center which includes information Slots used to present the Collection as a whole. Similarly to Association Centers, information Slots derived from the members of the collection are used as preview of them. Collections provide the base for building navigation paths through contents as well, and can be Single Collections or Collection Types. Collections are modeled by means of Collection Diagrams; in such diagrams a Collection and the involved Entities are represented by stereotyped UML classes while the Collection Center is represented by an association class between the Collection and the involved Entities. Figure 4 reports the screenshot of a page from the NGA.gov Web site presenting a guided tour on a selection of Works of Art from the Artist Monet. The contents shown in this page belong to the Center of the Collection. Figure 6 reports the UWA Collection Type Diagram related to this Collection.

More details on the UWA Hyperbase, Access Structures Models and the notation used in these models are provided in [22]. A MOF meta-model of the UWA Hyperbase can be found in [12].

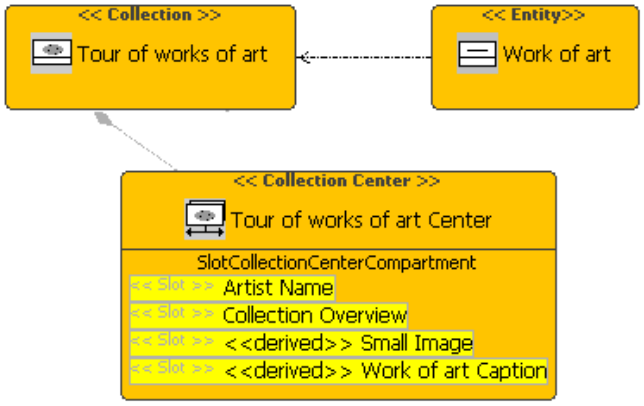

Fig. 6. The UWA Collection Type diagram for the guided tour of Works of Art in NGA.gov

3 The Reverse Engineering Process

This section describes the process that we have defined to recover the UWA Hyperbase (Entities and Semantic Associations) and Access Structures (Collections) models from existing WAs[4].

The process and the underlying analysis techniques arised from the following main considerations:

- UWA Entities can be recovered from WAs by searching for groups of logically related attributes[5] forming an information concept (content type) that the application presents to its users.
- Semantic Associations can be recovered by identifying hyperlinks connecting pages showing instances of different Entities, or by searching for pages showing instances of different Entities.
- Collections can be recovered by identifying pages showing several instances the same Entity, or including a set of hyperlinks towards pages showing instances of the same Entity.

The above considerations suggest that, to reach our goal, the analysis can be limited to the client-side pages of the application, whether they are static or dynamically generated. This is also supported by the fact that UWA conceptual models are intended to describe what the user actually sees of the application and not how it is internally implemented. Starting from this assumption, we have defined a reverse engineering approach that is independent from the technologies used to implement the WA on the server side and that can be applied to any WA having HTML pages as front-end. To this aim, a significant amount of client (HTML) pages of the WA to be analyzed are captured by using a Web Crawler. Of course, among the captured pages, there will be groups of similar pages made up of client pages having the same layout structure (i.e., the same HTML structure) and reporting the same kind of information but with different values (such as the pages showing the descriptions of different products in an e-commerce web site). We call the pages forming such a group "Cloned Client Pages" and use a clone analysis technique to identify them.

Groups of related attributes are identified by performing source code analysis on the client-side pages of a WA. In particular the analysis aims to find 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 a report), or, (*ii*) presented in a set of cloned client pages.

Usually, to describe the meaning of input/output data items to users labels are used. Example of such labels are words used to describe input fields in a form, words included in table heading, or, in general, labels used to specify the semantics of some data in a Web page. We refer to these labels as to keywords. Such keywords characterize a concept of the application domain and correspond to the Slots of a UWA Entity. As a consequence, each group of keywords is candidate to form a UWA Entity.

Pages showing attributes belonging to different Entities and hyperlinks between two client pages showing different Entities are the base for identifying possible UWA Semantic Associations.

A set of instances of a given Entity (e.g., a set of different values associated to the same group of keywords, such as the values in the rows of a table), or a set of hyperlinks pointing to pages showing different instances of the same Entity client page are an indicator for a possible Collection.

The UML activity diagram with object flow reported in Figure 7 shows the different analysis and recovery activities in which the process is organized and the produced artifacts. Most of these activities are executed automatically by the tool presented in Section 4. Of course, this is not true for semi-automatic activities, marked in Figure 7 with the "<<`manualTask`>>" stereotype, which require user intervention. These activities are mostly validation activities.

In the diagram we can distinguish three main phases, corresponding to the three UWA modeling concepts that the process is currently able to identify:

- UWA Entities Abstraction.
- UWA Semantic Associations Abstraction.
- UWA Collections Abstraction.

To start the recovery process for a given WA, a significant amount of HTML pages have to be downloaded from it by means of a Web crawler. The level of depth the crawler has to reach and other rules it has to observe while surfing and downloading the WA have to be care-

[4] In the following of the paper, whenever not differently specified, with the terms of Entity, Association and Collection we refer to the typed version of the corresponding UWA modeling concepts presented in Section 2.

[5] I.e., information items or Slots, using a UWA jargon.

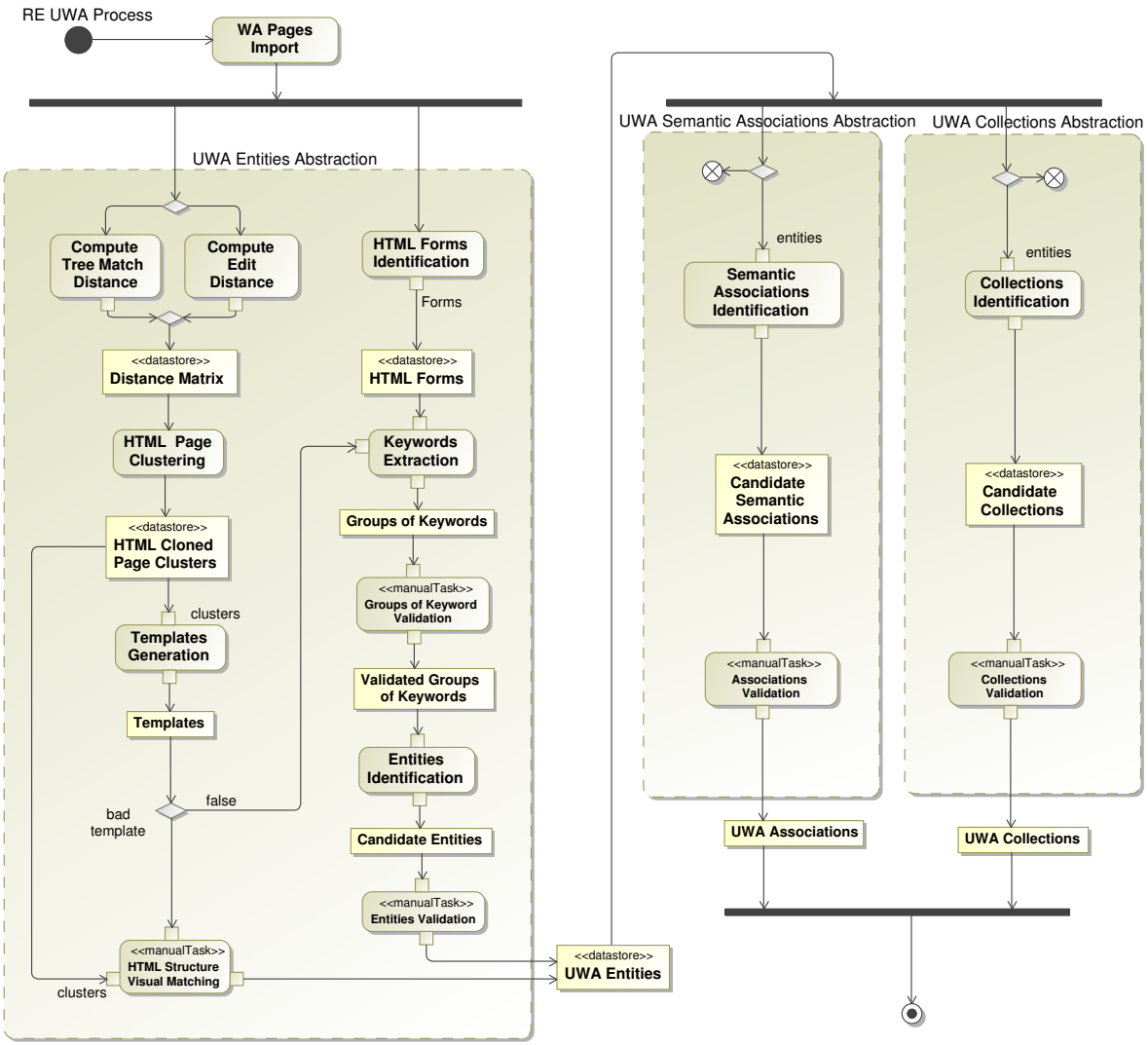

Fig. 7. The process to recover UWA Entities, Semantic Associations and Collections Types

fully defined in order to obtain a dump of all the sections of interest of the WA.

3.1 UWA Entities Abstraction

The identification of UWA Entities is carried out by searching for groups of related keywords in the client-side HTML pages (static and dynamically generated) of the WA. A group of keywords involved in the same user input or output operation and included in the same (HTML) form or output report is considered as a possible group of Slots characterizing a 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. Thus the recovery of the UWA Entities is based on the extraction of groups of related keywords both from HTML forms and groups of cloned client pages. After the groups of related attributes are identified, they have to be validated by a human expert of the application domain and UWA Entities are associated to the validated groups of keywords. A final Entity validation activity is carried out to validate the identified Entities. The following subsections describe these activities.

3.1.1 Computing Matrix of Distances

To identify groups of cloned client pages, a matrix of similarity distance between pages is computed and analyzed to build clusters of cloned client pages. From each cluster of cloned client pages, a page template representing the common features of the pages in the cluster is derived. The code structure of a client page is made up of a control component (i.e., the set of the HTML tags and scripts determining the page layout), and a data component (i.e., the set of content items determining the information presented to the user, such as text, images,

/div	/td	align	div	height	img	src	td	width
a	b	c	d	e	f	g	h	i

Table 1. HTML Tags and corresponding symbols

multimedia objects). For each pair of downloaded client pages the distance between their HTML control components is computed. Currently, two types of distances are supported: the edit distance (or Levenshtein distance) and a maximum subtree matching distance. The computed distance matrix is used in the following clustering phase to identify groups of pages with a very similar HTML structure.

As an example of Levenshtein distance computation, let us consider the following two HTML code-lines:

```
<td width="20%">
<img src="../images/Nuovo.jpg" width="192"
        height="30"></td>
```

and Table 1, where each HTML tag is associated to a symbol. In the two HTML code lines we can identify the following sequence of HTML tags:

```
(td, width, img, src, width, height, /td)
```

corresponding to the string of symbols: u = `hifgieb`.

Now, let us consider the following HTML code-lines:

```
<td width="75%">
<div align="right">
<img src =" ../pic1.jpg"
        width="155"
        height="17"> </div> </td>
```

which correspond to the following sequence of HTML tags:

```
(td, width, div, align, img, src, width,
 height, /div, /td)
```

and, with reference to the Table 1, to the string of symbols: v = `hidcfgieab`.

The optimal alignment of the strings u and v is:

```
h i d c f g i e a b
h i     f g i e   b
```

and the Levenshtein distance between the two strings u and v is: D(u, v)=3.

For the sake of brevity, we omit details about the computation of the maximum subtree matching distance, that can be found in [10].

3.1.2 Clustering HTML Pages

A group of cloned client pages is characterized by the same control component, but different data components. Groups of pages having exactly the same control component are expected to present their content in the same way. Thus, they can be considered as equivalent pages (i.e. clones), just differing for the data component they contain. The control component of each page is derived from the flattened DOM tree of the page by 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. Each group of cloned pages is clustered in a set of page-clones. Thus, groups of *Perfect Clones*, made up of groups of pages all having a distance equal to zero, can be formed. Groups of *Near Perfect Clones*, made up by groups of pages having a distance in a defined range of values $[d_{min}, d_{max}]$, can be formed too. Different values for d_{min} and d_{max}, empirically selected by the analyst, are used until significative clusters of pages are obtained.

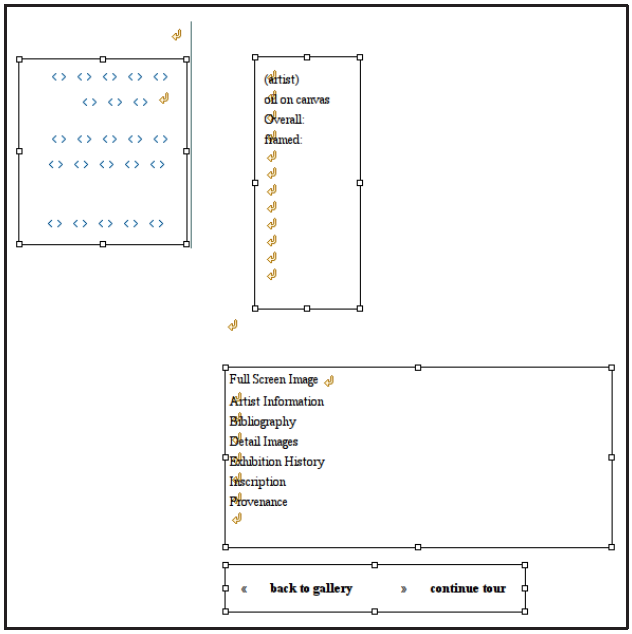

Fig. 8. The Template for the pages in Figure 1

With reference to Figure 1, the two pages are near perfect clones as they have a very similar structure, but different content. This is not true for pages showed in Figure 4 and Figure 2 which have a very different structure.

3.1.3 Generation of HTML Page Templates

From each group of cloned client pages a HTML page template is produced. This template has the same control component characterizing all the pages in the set and 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. The groups of clones are considered for automatically producing the page Templates. Near perfect clones can generate bad templates, i.e. templates presenting no content and thus made up just by the HTML tags and attributes, or templates containing no valid keywords (a keyword is considered valid if it actually represents an attribute of a concept of the application domain).

Bad templates and non valid keywords are identified by a human expert of the application domain carrying out a validation activity. During validation, bad templates undergo a human visual analysis in order to extract valid groups of keywords from them (if any) or to permanently discard the pages contribuiting to generate the templates.

Figure 8 shows the template extracted from the clustered pages reported in Figure 1.

3.1.4 HTML Structure Visual Matching

The pages of clusters generating 'bad' templates can be analyzed in order to generate groups of keywords that otherwise would be missed by the automatic approach. Looking directly at a rendered HTML page, the analyst can mark the page's structures (e.g. tables) that contain Entity instances, and also specify the related Slots. An automatic search is performed to look for the marked structure (i.e. the Entity) in the other pages of the cluster to recover information useful for identifying Associations and Collections. Information on the found Entity instances are then stored in the entities repository.

3.1.5 HTML Forms Identification

The downloaded pages are analyzed to identify HTML forms in them. The labels describing the fields in each form will be considered as "keywords".

3.1.6 Groups of Keywords Extraction

A group of keywords is associated to each group of labels identified in HTML forms. Keywords belonging to the same page template structure, such as HTML tables and divs, are associated to a group of keywords. In the case of 'bad' templates the analyst can directly generate groups of keywords during the 'HTML Structure Visual Matching' activity. Each group of keywords is assigned an identifier (e.g. the name of the form and the HTML page containg it, a number identifying the cluster of cloned pages). All the groups of keywords are recorded into a list.

In the template in Figure 8, four structures are present (the four "boxed" page's areas in the figure): one does not contain any keyword. A group of keywords is associated to each of the other 3 structures:

```
G1=(artist, Overall, framed, Oil on canvas)
G2=(Full Screen Image,Artist Information,
    Bibliography,Detail Images,Exhibition History,
    Inscription,Provenance)
G3=(back to gallery, continue tour)
```

3.1.7 Groups of Keywords Validation

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

1. Identify and solve synonyms (i.e., keywords or 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 of the validated groups of keywords.

For instance, among the groups of keywords generated from the template of Figure 8 the group `G3` is discarded during validation since it contains text from a navigation bar. The other two groups containing valid keywords are retained by the analyst. In this case the analyst assigned the name "Work Of Art" to the groups obtained by merging `G1` and `G2` since they refer to the same domain concept.

3.1.8 Identification and Validation of Abstracted UWA Entities

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 [15] 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 occurrencies of a group of keywords in the HTML pages, 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 or 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 on the number of occurrencies of each group in the code (e.g., the number of times the group is referred in HTML pages), and in ascending order on the arity of each group. An automatic procedure based on the one described in [15] 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 identify Semantic Associations 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 a candidate to make up a Component. A validation of the groups in the `CandEntities` set is to be carried out to discard those ones that do not correspond to a valid concept of the application domain, or re-arranging any others to better match an actual Entity. The validated Entities will make up the list `ValEntities`.

The two entities "Artist" and "Work Of Art" are the result of this step with reference to the pages in Figure 1 and Figure 2.

3.2 Abstraction of UWA Semantic Associations

This phase of the process is aimed at recovering UWA Semantic Associations. A candidate Semantic Association is identified between each pair of Entities having some Slots in common. 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 mainly when a Slot is used as an anchor to set the hyperlink. Indeed UWA Semantic Associations are the base for defining navigation paths through different content types. Similarly to candidate Entities, Associations found in this step have to be validated by a human expert knowledgeable of the application domain. The expert will discard the associations that do not correspond to valid ones in the application domain, according to her/his expertise. With reference to the NGA.gov Web site, the Semantic Association "created by" was identified between the Entities "Artist" and "Work Of Art". It is due to the hyperlink connecting the page showing a "Work Of Art" to the page providing information about the Artist that made it.

3.3 UWA Collections Abstraction

The identification of UWA Collections is based on the ways they are usually implemented in a WA. These include: (*i*) the usage of a table where each row reports a different instance of a given Entity or Association; (*ii*) a list of hyperlinks pointing to pages showing different instances of an Entity; (*iii*) forms reporting a list of fields with data related to the attributes of an Entity or the Entities involved in an Association. The HTML code of client pages is analyzed to identify Collections and Collection Centers.

As for Entities and Semantic Associations, the automatically recovered UWA Collections will undergo to a validation phase conducted by a human expert knowledgeable of the application domain with the support of the tool.

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

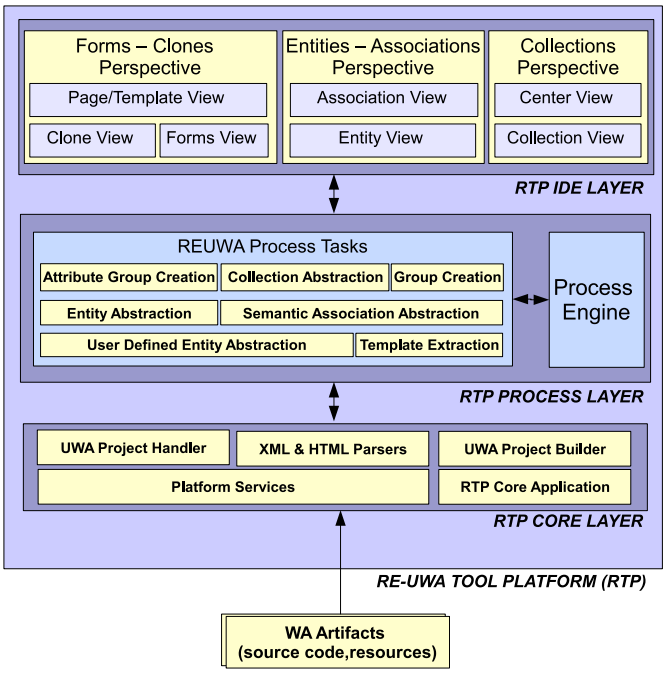

Fig. 9. The architecture of RE-UWA Tool Platform

4 Tool Support

To support the RE-UWA approach the RE-UWA Tool Platform (RTP) prototype has been developed.

4.1 RE-UWA Tool Platform Architecture

At the lowest level of the architecture is the RTP Core layer that introduces project integration providing builders aware of UWA resources and a project nature enabling RE-UWA process workflow for Eclipse WTP projects. In this layer there are also basic services for the entire RTP: HTML/XML parsers, along with similarity distance calculators between HTML documents and core platform services.

The platform services implement the logic to import WA pages into a UWA project. Such import phase extracts structural information about: (i) the downloaded client pages; (ii) the inner components of each page (e.g. forms, scripts module, frame, applet, etc.); (iii) the hyperlinks connecting the pages.

The extracted information are stored into a repository located in the analyzed project. A 'clone detector' module is used to perform static analysis on the HTML client Web pages of the application to identify pages that are clones. The clone detector component traverses HTML/XML DOMs to generate distance matrices for the HTML pages under analysis. This component can be configured by independent modules that gain access to the DOMs of WA pages to calculate the distance matrix with several distance algorithms. Currently, as discussed in Section 3, two distances are supported: the edit distance and a maximum sub-tree matching distance. The data extracted are made accessible to the entire RTP environment.

The RTP Process layer implements the process logic: it's based on a workflow engine that follows the RE-UWA process specification. For each step of the process there is a component implementing it. The engine takes the process instance and transfer the control between the steps as specified in the process definition.

The process is structured as a direct graph in which there are several kind of nodes and edges. Nodes can be simple nodes or composite ones with an inner structure. Simple nodes can be process (executing recovering process logic) or predicate nodes (to structure the control and data flow). Composite nodes can be of several types depending on the policy of execution of inner nodes (i.e. all nodes must be executed; only one must be executed; any of inner nodes can be executed). Edges are of different types according to the needs for interaction on the transition and on the routing policies (auto or manual routing). The software components participate to the framework by inheritance and composition: they can be added, removed or modified in flexible ways. Each process has a customizable configuration phase where control and data flow dependencies, among the steps involved in the process, can be specified.

4.2 RTP IDE Layer

This layer implements the presentation layer that allows the interaction with users to drive the process execution. It is structured as a set of Eclipse editors and views that interact with the engine and the concrete components. It allows the analyst to execute the step logic providing the needed and related information to support her/his choices. The RTP IDE layer introduces the following perspective each one related to the recovering of a well defined portion of the UWA model:

- **Forms and Clones Perspective** - This perspective contains all the views related to HTML page clustering, templates generation and group of keywords extraction and validation. In this perspective there are also some editors defined to handle the recovered elements.
- **Entities and Associations Perspective** - This perspective contains all the views related to UWA Entities and Semantic Associations. Entities are added to this view from: (*i*) the output of the algorithm discussed in Section 3; (*ii*) the Entities specified by performing the semi-automatic analysis using the Web Page Designer (WPD) editor embedded in Eclipse. The validated and refined Entities and Associations can be finally saved into the internal repository.
- **Collections Perspective** - The Collections Perspective groups together the views used to drive the recovering of collections and their centers. For each identified Collection, the Collection view shows the pages containing it. By using the collection editor, the analyst can validate the identified Collections and refine them.

Figure 10 shows a screenshot of the RTP tool in the perspective for the recovery of UWA Entities and Associations captured during the analysis of the Web site NGA.gov.

5 Case Study

In this section we present and discuss the results obtained in a case study involving six real world WAs in order to validate the approach and assess its effectiveness.

The experimentation was mainly performed to assess whether:

- the groups of keywords extracted by the approach included all the actual Candidate UWA Entities;
- the Candidate UWA Entities corresponded to actual Entities;
- actual UWA Entities were left undetected by the approach;
- the Candidate UWA Semantic Associations identified by the approach corresponded to actual UWA Semantic Associations;

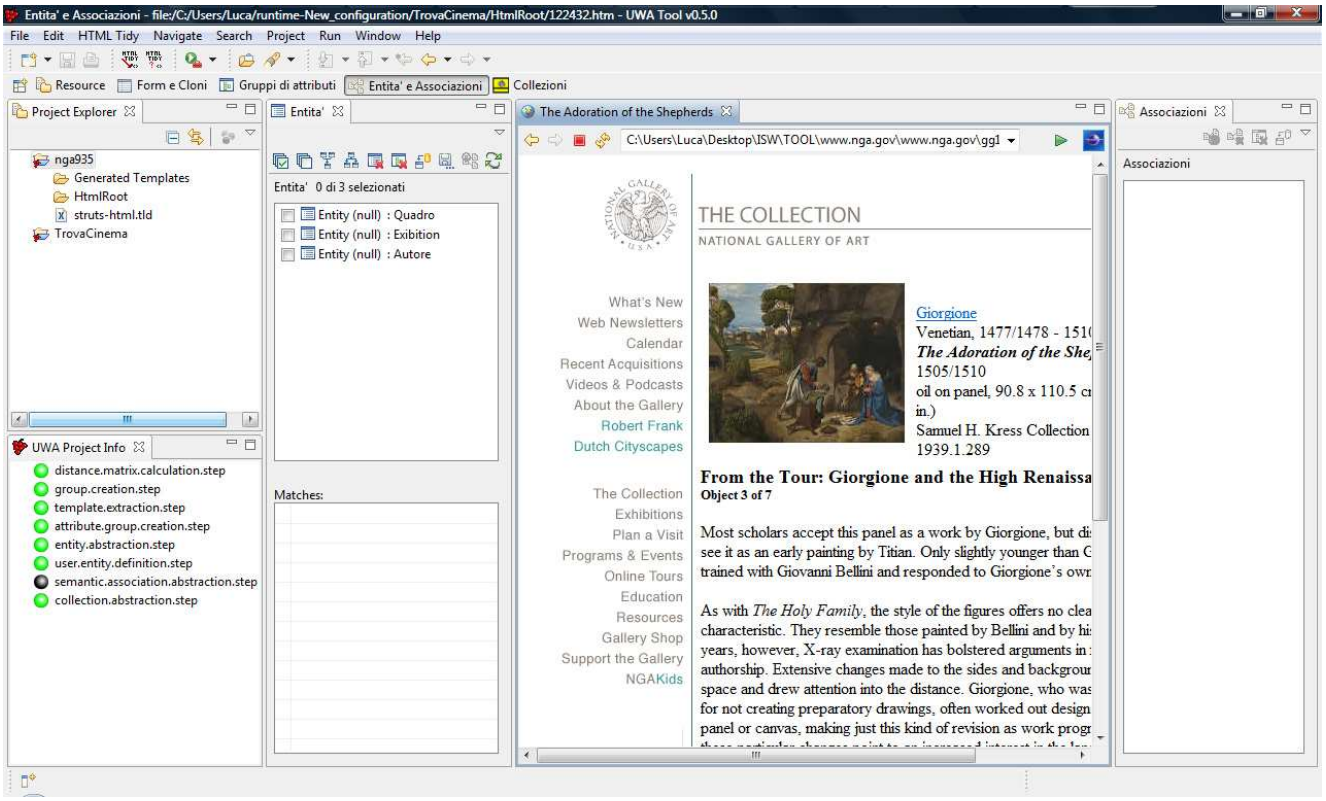

Fig. 10. A screenshot of the RE-UWA tool showing the UWA Entity and Association Perspective and a page from NGA.gov

WAs	#Downloaded Pages	#Clusters of Perfect Clones	#Groups of Keywords	#Discarded Groups	Precision
NGA	882	68	26	6	.76
eBay	935	133	10	3	.70
CHL	235	53	295	262	.11
FilmUp	1300	197	81	17	.79
TrovaCinema	550	123	65	11	.83

Table 2. Summary of recovered model elements for CHL, FilmUp, TrovaCinema and NGA.

WAs	#Imported Forms	#Groups of Keywords	#Discarded Groups	Precision
CourseNet	49	49	25	0.48

Table 3. Summary of recovered model elements for CourseNet WA.

	#Entities			#Associations			#Collections		
WAs	Tool	Expert	Recall	Tool	Expert	Recall	Tool	Expert	Recall
NGA	6	7	.85	5	6	.83	3	3	1
eBay	5	8	.62	4	4	1	3	5	.6
CHL	5	6	.83	3	3	1	3	4	.75
FilmUp	13	16	.81	6	5	.83	5	5	1
Trovacinema	9	10	.88	14	8	.85	7	7	1
CourseNet	8	8	1	17	17	1	4	5	.8

Table 4. Recall for identified Entities, Associations and Collections.

– no actual UWA Semantic Association was left undetected by the approach;
– the Candidate UWA Collections identified by the approach corresponded to actual UWA Collection;
– no actual UWA Collection was left undetected by the approach.

To reduce the risks of biasing the results, the analysts that conducted the case study were software engineers not involved in the definition of the approach but knowledgeable of the UWA design methodology. Five 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 five WAs are suitable cases of WAs with a large number of cloned HTML pages. These WAs were:

- NGA http://www.nga.gov
- eBay http://www.ebay.com
- CHL http://ww.chl.it
- FilmUp http://www.filmup.it
- TrovaCinema http://www.trovacinema.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 maximize the number of pages including Entities/Associations which were relevant for the application domain.

The sixth analyzed WA was 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 was characterized by several HTML forms. Both the downloaded pages of the first five WAs, as well as the client pages of the sixth one, were statically analyzed by the RTP tool and data extracted was the input for the recovery process described in Section 3.

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

Table 2 reports a summary of the results obtained from the analysis of the first five analyzed WAs (excluding CourseNet that is form-oriented WA). The first column of this table reports the names of the five WAs. The second column reports the number of HTML pages downloaded from each WA.

They were analyzed to identify clusters of cloned pages. For each cluster, a HTML template was generated as specified in Section 3.1.3 and each template was analyzed to extract groups of keywords to be validated by the analyst and subject to the algorithm described in Section 3.1.8 to identify UWA Entities. The third and the fourth columns of the Table 2 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. The fifth column of the table reports the number of the groups discarded by the analyst during the validation phase and the sixth one the computed precision.

Entity	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

Table 6. Entity types identified in CourseNet.

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 3.1.7. 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 five WAs were submitted to the algorithm described in Section 3.1.8 to generate the list of Candidate UWA Entities which were finally subject to the analyst validation. Table 5 shows the list of the UWA Entities that were finally identified using the RTP tool and their Slots for the first five WAs. 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. The Slots common to more Entities were assigned to just one Entity. The list of identified Associations for the analyzed WAs is reported in Table 7.

WA	Entity	Slots
eBay	**Bid**	Bidder, Bid Amount, Date Of Bid
	Feedback	Comment, Date, Time, From, Item
	eBay	About Ebay, Announcements, Security Center, Policies, Help, Privacy Policy, SiteMap
	Item	Payment method, Preferred/Accepted, Buyer protection on eBay
		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
		Starting time, Duration, Payment methods
		Description
		Shipping Item Cost, Additiona Item Cost, Destination, Shipping service, Insurance
		Return Policy
	Member	MemberID, Feedback score, Positive feedback, Member since, Members who left a positive, Members who left a negative, All positive feedback received, #Bid retraction
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	Organization, Sponsor, Schedule, Passes
	Feedback	Full name, Email Address, Question or Comment
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
		General Description
		Technical specifications, Composition
	Comment	Author, Text, Data
	CHL	Shipping centers, News from Popitt
		Item Warranty, PC Warranty
		Shipping Costs, Delivery Costs, Handling costs
		Virtual Money Box, Forum, Votes, Comments
		Withdrawal right, Erroneous Shipping, Handling Not Working Items, Incomplete Items, Items damaged during shipping, Decree Low
		CHL in Short, Buying on CHL, Join in CHL, How to find Products, Technical Sheet, Shopping Cart, Assembled PC , Product Order, After buying
FilmUp	**Movie**	Title, Original title, Nation, Year, Genre, Duration, Director, Official Site, Cast Produiction, Distribution, Release Date, Plot, Trailer, Details, Review, Poster
	Soundtrack	Movie Title, Album Title, Music Author, Performer, Year, Edition, Distribution, Discount, Tracks Outline
	Review	Review Author Name, Review Text
	Trailer	Duration, Type, Details, Movie Title
	Poster	Movie name, Image
	Opinion	Opition Author Name, Opinion Text
	News	News Title, Description, Image, Date
	Biography	Actor Name, Image, Biography Autor, Last Update Date, Biography Text
	Filmography	Movie Title, Character
	PhotoGallery	Movie Title, Image
	Curiosity	Title, Description, Image, Date
	Cinema	Cinema Name, Cinema Address, Cinema Telephone ID, Cinema Web Site
TrovaCinema	**Your Review**	Author, Title, Review Date
	Expert Review	Review Author, Journal, Review Image, Review Text
	Poll	Title, Poll Result, #Votes
	Cinema	Cinema Name, Cinema Description, Movie Name, Address, City, Telephone ID, Price Table, Cinema Web Site, Annotations, Parking Information, #Movie Halls, Timetable
	Festival	Title, Expert Review, Calendar, Jury, Awards, History
	Movie	Title, Director, Actors, Year, Expert Reviews, Reviews, Poster, Trailer, Fotogallery, Plot, Expert, Average Rating, Original Title, Country, Duration, #Votes, Multimedia
	Poster	Nome Film, Image, Poster description
	Actor	Nome Attore, Actor birth date , Actor biography, Curiosity, Multimedia
	News	News Title, News Date, News Description, Video, Image

Table 5. Entity types identified in CHL, eBay, NGA, TrovaCinema and FilmUp.

WA	Associations
eBay	Member-Feedback Member-Item Item-Bid Works of art-Artist
CHL	Product-Comment Product-Product (2)
NGA	Works of art-Artist Artist-Works of art (4)
FilmUp	Movie-Poster, Movie-Review, Movie-Trailer, Movie-Soundtrack, Movie-Opinion
TrovaCinema	Cinema-Movie, Festival-Movie, (Expert) Review-Movie, Poll-Movie, Poll-Festival Movie-Actor, Movie-Trailer
CourseNet	Student-Tutoring Request, Student-Exam Reservation, Student-Tutoring, Student-Course, Student-Exam, Tutoring-Tutoring Request, Tutoring-Course, Tutoring-News, Teacher-Tutoring, Teacher-Tutoring Request, Teacher-Course, Teacher-News, Course-News, Course-Exam Reservation, Course-Exam, Exam-Exam Reservation, Exam-News

Table 7. Associations identified in all analyzed WAs.

WA	Collections
eBay	Items of a Seller, Products, User Feebacks, Bid History, Favorite Sellers
CHL	Products, Productors, Orders, Comments
NGA	Tours, Exibitions, Pictures
FilmUp	Biographies, Movies, Posters, Trailers, Opinions
TrovaCinema	Cinema, News, Movies, Expert Reviews, Biographies, Posters, Actors
CourseNet	Exam Session List, Course List, Tutoring List, Tutoring Request List, News List

Table 8. Collections identified in all analyzed WAs.

All the Candidate Entities proposed by the tool at the end of the first step of the recovery process were validated by the analyst, while some candidate Semantic Associations were discarded because redundant. In the tables each Entity and Slot is identified by a name. The Slots are those resulting after the step of identifying Semantic Association. Figures 3 and 5 show, respectively, an excerpt of the UWA Entity Type and Semantic Associations Type diagrams recovered for the NGA.gov WA.

5.2 Results from the CourseNet WA

Table 3 reports the information regarding the analysis of CourseNet. This WA was characterized by having a high number of forms used for input/output operations. No client-side cloned pages were identified in it and so we extracted groups of keywords just from forms. The analysis retrieved 49 groups of keywords. The synonyms and homonyms analysis revealed that some forms referred the same group of keywords, even if they had been assigned different names, i.e. they were synonyms. At the end of synonyms and homonyms analysis, we had 24 groups of keywords. 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. Table 6 reports the list of the identified Entities. In the next step of the process, 17 candidate Associations were identified among the Entities. All the 17 identified Associations were considered to be valid by the UWA expert. Finally, 5 collections was found as reported in Table 8. The identified and validated associations for CourseNet are reported in Table 7.

5.3 Discussion

To verify the efficiency and the effectiveness of the proposed approach, six analysts, one for each WA, expert of the application domain and of the UWA methodology, were asked to analyze (without the support of the RTP tool) the WAs to identify UWA Entities and Semantic Associations. Thus, each expert analyzed the pages, 'captured' by the crawler, 'manually' just using a browser. For the eBay WA the expert identified two more entities than the tool and this was the worst result in term of recall (62%). For the NGA and CHL WAs the experts found in both two cases one more Entity and the recall was always greater than 80%. In all those 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 Associations are 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 RTP 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. For the FilmUp WA, the tool was able to recover 13 Entities, while the expert recovered 16 Entities. Looking at the three missing entities (Quiz, PhotoGallery and TV Guide) the reason for that was clear. For Quiz and TV Guide it was related to missing keywords in the pages containing such Entities (for such cases visual entity identification approach discussed in section 3.1.4 based on the analysis of the pages associated to the bad template is a viable alternative to recover the Entities). For the Entity "PhotoGallery" the reason was related to the kind of the data type: the only attribute was the image data that is not yet supported in the keywords automated extraction approach. The tool recovered 6 Associations but 2 of them were false positive (and were discarded) and one recovered by the expert was missed (because of a link to a page not downloaded by the crawler). For the WA TrovaCinema the tool identified 14 Associations but only 8 of them were actual Associations identified also by the expert. The other were false positives introduced by misleading common attributes having the same name.

For what regards the Collections, Table 8 reports the ones identified for the six WAs considered in this case study. The Collections identified were the same produced by the experts. However the experts were able to provide a more detailed model in which they distinguished between different Collections towards the same Entities (showing different Centers). The tool was not able to recover such level of detail identifying a single Center with all recovered attributes. However, this can be addressed focusing, during the collections validation phase, on the pages containing Collections instances and searching for the different sets of slots (in order to create a Collection Center for each set).

To further validate the approach, a comparison among the two WAs FilmUp and TrovaCinema, related to the same application domain (they are two well known Italian portals for movies), was performed. The aim was to verify if similar Entities, Associations and Collections were found, and what main differences were in the recovered models mirroring the differences in WAs usage. The analysis revealed the same main abstractions of the domain (film, actors, biographies), but the models also highlighted different approaches and functionalities specific to each WA. In particular they highlighted that while TrovaCinema has a richer model related to movies and theaters providing much detailed information on them, FilmUp has a more accurate model for the movies information and the related multimedia content (such as richer image and video galleries and soundtrack information). This actually reflects the targeted audience: FilmUp is mainly a movie information portal while TrovaCinema is more targeted at finding the nearest and cheapest best cinema or theater for the user. It has been interesting to verify that the approach was able to provide models explicitly showing such different choices made at modeling time (for each WA).

The results from the presented case study suggest some considerations about the proposed approach. As previously mentioned, in some cases the generated templates gave origin to synonyms, duplicated and not useful groups of keywords (as in the case of content belonging to page navigational structure). To improve the quality of groups extracted from templates, some techniques from the information retrieval field can be used. These include the use of: (*i*) stop word lists to avoid taking into consideration the usual words in menus and navigation bars; (*ii*) techniques allowing the identification of synonym groups, thus reducing the number of groups to validate.

Finally, the total effort to execute the entire process depends on the manual tasks to be performed. Manual tasks are mainly due to the generation of bad templates and then by the presence/absence of meaningful keywords in the WA pages. WAs characterized by a large number of pages with no keyword require a major effort.

6 Related Works

Several approaches for the reverse engineering of a WA have been proposed in the last 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 [14,8,19] 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 [13], an approach for abstracting a description of the functional requirements implemented by the WA is proposed. UML use case diagrams are used to represent the abstracted functional requirements. 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 [21]. The VAQUISTA [25] 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 [18] 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 [8] 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 al-

ternative 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.

The reverse engineering approach proposed in this paper differs from the works cited in this section, and others proposed in literature, mainly because it 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 HTML pages as front-end, regardless of the technologies used server-side.

7 Conclusions and Future Work

In this paper we have presented an approach to recover user-centered conceptual models from an existing WA. In particular, the approach is able to abstract a conceptual model representing the WA's contents, relationships between contents and views on contents, as perceived by the final users of the application. The models are formalized according to the Ubiquitous Web Application design framework in terms of Entities, Semantic Associations, and Collections, but being these modeling concepts representative of those adopted by other well known WA design methodologies, the approach can be adapted to recover the models proposed by other methodologies. A tool developed as an Eclipse IDE has been developed to support all the phases of the process.

The recovery process can be applied to any WA producing HTML pages as front-end and can be beneficially adopted for re-documentation, comprehension and evolution purposes. The abstracted models, indeed, by providing an up to date and user-centered representation of the WA, can be used to reason about possible evolution tasks aimed at satisfying new user requirements or to better meet the user's expectations. Additionally, the recovered models can also be used as a starting point of a forward engineering process aimed at migrating the application towards new technologies and implementation frameworks. In this sense we are currently extending our tool in order to use the recovered models as an input in the UWA model driven development process presented in [6].

The case study carried out showed that the approach is feasible and valid, and highlighted some possibilities of improvement. Indeed, for all the analyzed WAs, the approach was able to correctly identify the same UWA Entities, Associations, and Collections 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.

Moreover the expertise and domain knowledge of the analysts affects more the identification of Entities than the identification of Semantic Associations.

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.

Future work will also consider the extension of the RE-UWA approach and the RTP tool. In particular the tool will be extended with:

- **New resource analyzers** - Components providing new analyses for both WA artifacts and UWA resources can be easily integrated into the environment. As instance, future work in this area will be related to the definition and validation of new structural distance algorithms among HTML documents as well as the analysis of JavaScript code elements, in order to recover UWA models also from AJAX applications.
- **New processes and tasks** - In particular in this area work will be devoted to extend the process with the logic needed to abstract the Navigation Model (identifying the mapping of nodes to pages) and the Transaction Model (extracting a business Transaction Model from the WA).
- **New perspectives, editors and views** - As more complete models will be recovered, new perspectives will be added to drive the steps for their identification. Such activities will be implemented by means of components within the RTP framework. A new GMF[6]−based UWA Editor is also being integrated into RTP in order to directly edit and refine recovered models by means of a graphical notation editor that supports UWA ecore instance of the UWA MOF/EMF Metamodel.

References

1. M. L. Bernardi, G. A. Di Lucca, and D. Distante. Reverse engineering of web applications to abstract user-centered conceptual models. In *Proceedings of the 10th International Symposium on Web Site Evolution (WSE2008)*, pages 101–110, Beijing, China, 2008. IEEE.
2. S. Ceri, P. Fraternali, and A. Bongio. Web modeling language (webml): a modeling language for designing web sites. *Comput. Netw.*, 33(1-6):137–157, 2000.
3. S. Chung and Y.-S. Lee. Reverse software engineering with uml for web site maintenance. In *WISE '00: Proceedings of the First International Conference on Web*

[6] GMF stands for Graphical Modeling Framework, more information are available on http://www.eclipse.org/modeling/

Information Systems Engineering (WISE'00)-Volume 2, page 2157, Washington, DC, USA, 2000. IEEE Computer Society.

4. J. Conallen. *Building Web Applications with Uml*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2002.
5. D. Distante, R. Gustavo, C. Gerardo, and T. Scott. A comprehensive design model for integrating business processes in web applications. *Int. J. Web Eng. Technol.*, 3(1):43–72, 2007.
6. D. Distante, P. Pedone, G. Rossi, and G. Canfora. Model-driven development of web applications with uwa, mvc and javaserver faces. In *Proceedings of the 7th International Conference on Web Engineering (ICWE2007)*, pages 457–472, Como, Italy, 2007. Springer Berlin / Heidelberg.
7. D. Distante, S. Tilley, and S. Huang. Documenting software systems with views iv: documenting web transaction design with UWAT+. In *SIGDOC '04: Proceedings of the 22nd annual international conference on Design of communication*, pages 33–40, New York, NY, USA, 2004. ACM.
8. F. Estivenart, A. Francois, J. Henrard, and J.-L. Hainaut. A tool-supported method to extract data and schema from web sites. In *WSE '03: Proceedings of the IEEE International Workshop on Web Site Evolution (WSE)*, page 3, Los Alamitos, CA, USA, 2003. IEEE Computer Society.
9. N. Koch and A. Kraus. The expressive power of uml-based web engineering. In *IWWOST'2002 : Proceedings of 2nd International Workshop on Web Oriented Software Technology*. Springer Verlag, 2002.
10. T. Lindholm. A three-way merge for xml documents. In *DocEng '04: Proceedings of the 2004 ACM symposium on Document engineering*, pages 1–10, New York, NY, USA, 2004. ACM.
11. B. Marco, C. Stefano, F. Piero, and M. Ioana. Process modeling in web applications. *ACM Trans. Softw. Eng. Methodol.*, 15(4):360–409, 2006.
12. B. G. Maritati, L. Baresi, F. Garzotto, and M. Maritati. W2000 as a mof metamodel. In *Proc. World Multiconference on Systemics, Cybernetics and Informatics - Web Engineering track*, page 2002, 2002.
13. G. A. Di Lucca, A. R. Fasolino, F. Pace, P. Tramontana, and U. D. Carlini. Comprehending web applications by a clustering based approach. In *IWPC '02: Proceedings of the 10th International Workshop on Program Comprehension*, page 261, Washington, DC, USA, 2002. IEEE Computer Society.
14. G. A. Di Lucca, A. R. Fasolino, and P. Tramontana. Reverse engineering web applications: the ware approach. In *Journal of Software Maintenance and Evolution*, volume 16, pages 71–101, New York, NY, USA, 2004. John Wiley & Sons, Inc.
15. G. A. Di Lucca, A. R. Fasolino, P. Tramontana, and U. D. Carlini. Recovering a business object model from web applications. In *COMPSAC '03: Proceedings of the 27th Annual International Conference on Computer Software and Applications*, page 348, Washington, DC, USA, 2003. IEEE Computer Society.
16. G. A. Di Lucca, M. D. Penta, and A. R. Fasolino. An approach to identify duplicated web pages. In *COMPSAC '02: Proceedings of the 26th International Computer Software and Applications Conference on Prolonging Software Life: Development and Redevelopment*, pages 481–486, Washington, DC, USA, 2002. IEEE Computer Society.
17. O. M. G. (OMG). *Unified Language Modeling Specification (Version 2.0)*. On-line at www.omg.org., 2002.
18. L. Paganelli and F. Paterno. Automatic reconstruction of the underlying interaction design of web applications. In *SEKE '02: Proceedings of the 14th international conference on Software engineering and knowledge engineering*, pages 439–445, New York, NY, USA, 2002. ACM.
19. F. Ricca and P. Tonella. Understanding and restructuring web sites with reweb. In *IEEE MultiMedia*, volume 8, pages 40–51, Los Alamitos, CA, USA, 2001. IEEE Computer Society.
20. D. Schwabe and G. Rossi. An object oriented approach to web-based applications design. In *Theor. Pract. Object Syst.*, volume 4, pages 207–225, New York, NY, USA, 1998. John Wiley & Sons, Inc.
21. S. Tilley, D. Distante, and S. Huang. Web site evolution via transaction reengineering. In *WSE '04: Proceedings of the IEEE International Workshop on Web Site Evolution (WSE)*, pages 31–40, Los Alamitos, CA, USA, 2004. IEEE Computer Society.
22. UWA Project Consortium. *Deliverable D7: Hypermedia and Operation design: model and tool architecture*. UWA Project Consortium, 2001.
23. UWA Project Consortium. *Deliverable D9: Deliverable D9. Customization Design Model, Notation and Tool Architecture*. UWA Project Consortium, 2001.
24. UWA Project Consortium. Ubiquitous web applications. In *eBusiness and eWork Conference 2002*, 2002.
25. J. Vanderdonckt, L. Bouillon, and N. Souchon. Flexible reverse engineering of web pages with vaquista. In *WCRE '01: Proceedings of the Eighth Working Conference on Reverse Engineering (WCRE'01)*, page 241, Washington, DC, USA, 2001. IEEE Computer Society.