Modeling Business Processes in web Applications: An Analysis Framework

Damiano Distante
Research Centre On Software Technology, University of Sannio
Viale Traiano, 1
82100 Benevento, Italy
+39 0824 305528
distante@unisannio.it

Gustavo Rossi
LIFIA – Facultad de Informática, Universidad Nacional de La Plata
Calle 50 y 115
1900 La Plata (Buenos Aires), Argentina
+54 221 4228252
gustavo@sol.info.unlp.edu.ar

Gerardo Canfora
Research Centre On Software Technology, University of Sannio
Viale Traiano, 1
82100 Benevento, Italy
+39 0824 305555
canfora@unisannio.it

ABSTRACT
The addition of business processes to modern web applications entails new challenges to be faced when developing them, hence the need for suitable methodologies to be adopted in the design phase. In response to this need, most of the design methodologies for web application available in the literature include a proper solution. In this paper we propose a framework for analyzing and comparing web application design methodologies with regard to their support for modeling business processes. The analysis framework has proved to be useful for assessing the ability of each considered methodology to deal with the design of business processes in web applications. The framework also provides suggestions on how to possibly enhance a given methodology.

Categories and Subject Descriptors
D.2.1 [**Software Engineering**]: Requirements/ Specifications – *Methodologies.*
D.2.2 [**Software Engineering**]: Design Tools and Techniques.

General Terms
Design, Documentation.

Keywords
Web Applications, Business Processes, Design, Requirements, OOHDM, UWA.

1. INTRODUCTION
Web applications have rapidly evolved over the past few years from brochure-like read-only web sites (*informative web applications*) to complex data- and operation-intensive applications supporting the implementation of business processes (*business web applications*).

End-user and enterprise business web applications, such as online booking systems, e-commerce web sites, e-learning platforms, e-government systems and e-procurement systems, are intended to provide the user with a set of services (book a flight, buy a product, pay taxes, etc.).

SAC'07, March 11-15, 2007, Seoul, Korea.

These kinds of web applications implement business processes via sequences of operations (*functional activities*) and navigation steps (*navigational activities*) through the pages of the application. We refer to such sequences, with their associated execution flows, as to *web transactions*.

In response to the growing interest in web applications that implement business processes, a number of models and methodologies formerly proposed for the design of informative web applications have been extended in order to support the design of business processes. This is the case of the Object-Oriented Hypermedia Design Method (OOHDM) [1][2], the web Site Design Method (WSDM) [3], the Object Oriented Hypermedia (OO-H) and UWE methodologies [4], Web Modelling Language (WebML) [5], and the Ubiquitous Web Applications (UWA) design framework [6].

In this paper we present a framework for analyzing and comparing web application design methodologies, with regard to their approach for designing business processes, highlighting their strengths and their weaknesses. The framework takes its starting point from a similar study comparing approaches to modeling ubiquitous web applications [7] and has its basis on a set of requirements that we identified in the experience of developing business web applications and analyzing the solutions proposed by a set of well known design methodologies. The results of the application of the framework to two well established design methodologies, OOHDM and UWA, are also presented.

The main contributions of the paper are the following:

- We clearly characterize the requirements for a method supporting business processes in web applications.
- As a first consequence we show how a method should be analyzed and evaluated according to the requirements.
- As a side but important contribution we assess two well-known methods and compare them regarding their coverage of business process functionality.

The remainder of the paper is organized as follows. Section 2 presents the list of requirements we identified and contextualizes them with reference to the representative process of flight reservation on an airline company web site. Section 3 presents the analysis framework based on the list of requirements and their classification with respect to three dimensions of design, namely: *business*, *user* and *system*. Section 4 presents the results of the application of the comparison framework to OOHDM and UWA.

Finally, Section 5 summarizes the paper and looks ahead to future work.

2. REQUIREMENTS FOR DESIGNING BUSINESS PROCESSES IN WEB APPLICATIONS

Moving from the experiences gathered in a number of case studies of reverse and forward design of web applications implementing business processes [10][11][12][13][14] and from the analysis and comparison of the approaches adopted by a number of design methodologies such as OOHDM [1][2], WSDM [3], OO-H, UWE [4] and UWA [5], we built a representative set of requirements for a suitable methodology and related meta-models to design business processes in web applications. The list of requirements is reported in Table 1 and contextualized in the following of this section in relation to the typical process of booking a flight on an airline company web site.

The process of flight reservation on an airline company web site can be thought as composed of six main activities: a_1) *Search for a flight*; a_2) *Select a flight*; a_3) *Authenticate*; a_4) *Choose on-board options*; a_5) *Confirm reservation*; a_6) *Pay for the ticket* (this resulted from a reverse engineering process applied to the Alitalia.it web site [8] presented in [11]). Some of these activities can be further decomposed in simpler activities and dependencies can be identified between them. As an example the *authenticate* activity can be thought as composed of the *login* and *register* sub-activities, one excluding the other. Activities a_1-a_3 and a_5 can be defined as mandatory for completing the reservation process while the activity a_4 might be optional (the user may skip this activity and go for the default options). The need for defining these activities, their relations and their properties is represented by Req1.

The *search for a flight* activity is the first one executed by the user and it may be usually repeated several times, with different parameters (departure and arrival day, time and location) until a suitable flight is found. Then a flight is selected and, after the user authenticates himself, he is requested to confirm the reservation. The workflow just described is the one suggested by the user-experience, but other possibilities may be made available. For example the user could first authenticate himself and then proceed with the search, select and confirm activities. The needs for designing the possible workflows that will be available for the user is represented by Req2.

In order to book a flight, the user must be authenticated. If the user has authenticated himself at some point of the transaction, the information he provided should be stored for future use during the transaction and the condition of “user-authenticated” maintained. To do so, a state has to be associated with the designing transaction and the way it changes during the execution of the transaction modeled. These needs are taken into account by requirements Req3.

Once the user has confirmed the reservation, he is requested to proceed with the Payment. This activity may be “suspendable”, i.e. the user may be allowed to buy the ticket for the reserved flight in another session, by a certain time. The possibility to define suspendable activities in a process is represented by Req4.

At least two different actors can be identified in our hypothetical flight reservation process. The client that wants to book a flight and the airline back-office employee (which often corresponds to the system) which interact with the client confirming the reservation via email, send a receipt of the payment and (if the paper format is selected) send the ticket to the address specified by the client. Requirement Req5 takes into account the need/opportunity for the designer of the web application to represent how the client and the company back-office operator collaborate to have the client by a ticket for a flight.

The execution of each of the activities involved in a web transaction is usually associated with one or more navigation steps depending on the possible results of the activity execution. As an example, when confirming the flight reservation, the system could bring the user to a page where the summary of the reservation data is presented, while when failing the login activity the system may bring the user into a page that request the user to retry the login. Reciprocally, the navigation followed by the user can start, suspend, complete, abort or resume a transaction. As an example, the designer could want to specify that following a link to the “special offer” section of the site (while the user is requested to confirm his reservation) should temporarily suspend the ongoing transaction or, free the locked seat and abort the transaction. The execution of an activity itself affects the state of an ongoing transaction. When successfully executing the login activity, the state of the transaction takes care of storing the condition of “user-authenticated” for the remainder of the session. The requirement Req6 attends the needs just exemplified.

Which information should be provided to the user to let him pick one of the flights resulting from the search activity? The flight number? The flight fare? The departure and arrival time? The list of intermediate stops between departure and arrival? A section that allows him to retry the search with different parameters without going back to the home page of the web site? The requirement Req7 requires a suitable web transaction design methodology to support the designer to answer questions as the above mentioned.

Table 1. A set of requirements for a methodology to design business processes in web applications

Req1	*Represent the component activities of a web transaction, their semantic associations and the properties/ constraints applying to each of them.*
Req2	*Describe the possible workflows of a web transaction.*
Req3	*Define and manage the state of a web transaction.*
Req4	*Specify which activities can be suspended and resumed afterwards during long-lived transactions.*
Req5	*Describe the way two or more types of user involved in a web transaction collaborate in its execution.*
Req6	*Specify the way content navigation and operation execution affect each other and the state of an ongoing web transaction.*
Req7	*Define which contents will be provided to the user in order to support the execution of a particular activity*
Req8	*Define which information objects are affected by the execution of the activity and how.*
Req9	*Describe the way an activity will be customized depending on the state of the ongoing transaction.*
Req10	*Describe the way an activity will be customized depending on the context of execution.*

Let's now suppose that we want the airline web site to be able to store the reservations history of each user. Under this hypothesis the activity of *confirm reservation* needs to update the user reservation history. Let's also suppose that we want to store the user preferences about on-board options (type of meal, seat position, etc.) to propose them as default choice during the *choose on-board options* activity. We may want to update the user preferences according to what he specified when executing the *choose on-board options* activity. Finally, the airways company managers may want the list of available flights presented to a user to be updated when other users book last available seats. The depicted situations are those motivating the requirement Req8 of the list.

As we said above, during the reservation process, the activity of searching for a flight is typically executed more than once in order for the user to find the best flight and fare. While the first time this activity is executed it could propose the usual departure and arrival airport for an authenticated user or empty values for an anonymous user, the following times it is helpful for the user to present the values he specified on the last execution of the activity. The possibility to define how activities should appear depending on the state on the ongoing transaction is what the requirement Req9 in intended for.

The airways company paying for the design and implementation of the reservation system may want the system to be accessible from PC as well as from PDA and published in different languages. The company may also want to have a quicker reservation procedure available for frequently customers. The last requirement of the list, Req10, concern the ability of the web transaction design methodology to cope with these ever more common needs of ubiquity and customization of the web application (and his processes) with the context of execution.

Though not exhaustive, the above list of requirements is representative of the needs of designers when specifying web transactions in a web application. We next show how to use this list to analyze and compare design methodologies.

3. THE ANALYSIS FRAMEWORK

3.1 Three Dimensions of Design

Each of the requirements presented in Section 2 can be classified on three different dimensions of design corresponding to three levels of abstraction as follows:

- *Business requirements (business level of abstraction)*: this category includes requirements derived from the business rules that apply to the application to be designed and implemented. We located here those requirements defining the hierarchical and dynamic organization of application transaction in terms of component activities taking into account the business process structure.
- *User requirements (conceptual level of abstraction)*: this category of requirements makes a design methodology user-centered. A design methodology satisfying the category of requirements enables the designer to take care of the user-experience and the usability of the application to develop.
- *System requirements (implementation and technological level of abstraction)*: we classify here those requirements related to the specification of how the system will manage the state of each transaction guaranteeing the data coherence and the system to work properly. A methodology satisfying only this category of requirements is said to be system-centered.

3.2 Classifying the Requirements with the Three Dimensions of Design

We now classify each of the requirements presented in Section 2 in terms of the three dimensions of design listed above; each of the requirements may belong to more than one level of abstraction and may concern more than one dimension of design.

Req1 is a *business requirement* since it defines the hierarchical structure of the transaction, i.e. the set of activities involved in it and the relations between them. It is a user requirement in that the logical and temporal relations between the activities are designed taking into account the user experience.

Req2 is a *business requirement* in that it defines the transaction from the dynamic point of view, modeling the logical/temporal order with which the user can or must execute the activities of the transaction (the flow of the activities). In addition it aims to improve the usability of the web application, satisfying the user's need to execute a number of activities simultaneously without generating inconsistency in the data, therefore it is also a *user requirement*.

Req3 is on the system level, given that it tackles problems linked to the management aspects of the transaction.

Req4 is a *user requirement* since it meets one of the most important needs of the user: being able to suspend an activity, to resume the execution subsequently in a consistent state. In this sense, it is also a *system requirement*, because this need influences the way of managing and implementing the transaction on the system level. Finally it is a *business requirement* because an activity is or is not suspendable depending on the business logic and rules of the process.

Req5 is a *process requirement* since its objective is to represent the various types of players who participate in the execution of the business process, defining the activities executed by each one, and how they cooperate. It is also a *user requirement*, since it aims to model the web transaction according to the user-experience.

Req6 is a *system requirement* since its objective is to specify which navigation operations causes a transition of state in the transaction (abort, resume, suspend). It is also a *user requirement* because it defines the navigation allowed in the context of the execution of the transaction, taking account of user-experience.

Req7 is a *user requirement* because it takes the user's point of view to define the informational and navigational context of each activity of the transaction.

Req8 is a *system requirement* because it defines what changes can be caused by the execution of an activity in the hypermedia elements.

Req9 is a *user requirement* since it helps to improve the usability of the application; for example in re-executing an insertion of data it might be useful, to show the data inserted in the previous execution, so that the user needs only reinsert some of them. It is a system requirement, because, in order to establish how the activity must be presented to the user, it is necessary to define the state for the transaction and manage it on the system level.

Req10 is a *user requirement* because it makes it possible to model transactions that are usable in any context of use (i.e. any type of user, any device, any geographical area and at any time). It is also a *system requirement*, since designing transactions of this type entails the need to guarantee the continuous consistency of the data when presented in different devices.

Figure 1 provides a graphic representation of the distribution of the requirements on the three dimensions of design.

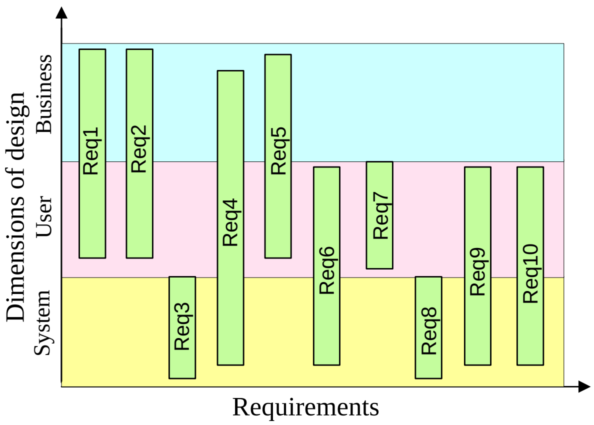

Figure 1. Characterization of the requirements with respect to the three dimensions of design.

3.3 The Analysis Framework

The idea at the base of the framework is to measure the degree to which a methodology meets each of the requirements of the list presented in Section 2 and to use the results of the analysis to position each methodology in a space with three dimensions: Business, User and System. The three dimensions of this space correspond to the three levels of abstraction on which the requirements were classified. In practice, the following procedure is adopted:

1) For each analyzed methodology we evaluate the degree to which it satisfies each of the requirements of the list.

2) The results of the analysis are represented in table form, by reporting for each requirement which of its design dimensions (Process, User, System) are satisfied by each of the methodologies.

3) The graphical representation is finally obtained by summing for each methodology the number of requirements that it satisfies on each design dimension.

Figure 2 shows the three-dimensional space of analysis and a hypothetical methodology, M1. From the collocation of M1 in the three-dimensional space it can be seen that the methodology does not satisfy any of the system requirements, it satisfies one requirement on the user level and two process requirements.

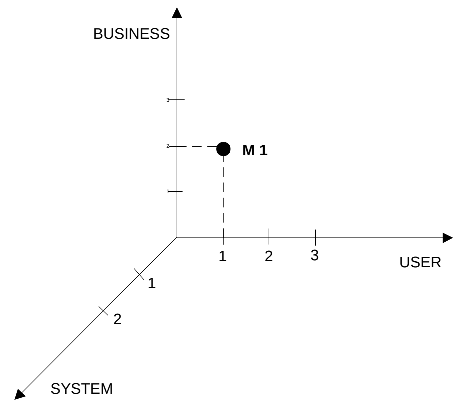

Figure 2. The three-dimensional space of analysis.

4. EVALUATING AND COMPARING OOHDM AND UWA

This section presents the results of comparing two relevant (and archetypical) web application design methodologies, OOHDM and UWA, by means of the framework described in Section 3. Each methodology is briefly described with regard to the approach it proposes for dealing with business processes when designing web applications.

4.1 Designing Business Processes in Web Applications with UWA

The UWA design framework [5] provides a complete methodology and a set of meta-models and tools for designing ubiquitous web applications, i.e., web applications able to be accessed via different devices, by different user types, from different locations, in different languages, etc.. Included in UWA is the UWA Transaction Design phase [15], a design phase specifically intended by UWA to integrate business processes in the designing web application.

A business process (*web transaction* in the UWA jargon) is basically designed by means of two models, the *Organization model* and the *Execution model*.

The Organization model describes the hierarchical relations between the component activities of a web transaction, specifies whether the execution of the an activity is required or optional in order for the user to complete the transaction, or if changes to data resulting from the execution of an activity are visible by other concurrent users. It is a stereotyped version of UML Class diagram [16] in which activities are arranged to form a tree. The activity at the root of the tree represents the whole web transaction, while the intermediate nodes and the leaves of the tree represent its component activities and sub-activities. For each activity a *property-set* and an *operation-set* is specified. The property-set is the subset of the isolation, atomicity, durability and consistency properties (ACID) the activity satisfies [9]. The operation-set represents the set of elementary operations the user can invoke during the execution of the activity.

The Execution model describes the dynamic aspects of a web transaction defining the possible execution flows associated with the transaction component activities included in the Organization model. It is a stereotyped version of the UML Activity diagram in which transaction activities and sub-activities are represented by states (ovals), and the execution flow between them is represented by state transitions (arcs). The conditions for the execution of an activity (e.g., the user inputs) and the results of its execution (e.g., the transaction state) are represented by labels associated with arcs. Swimlanes are used to describe how different user types collaborate in the execution of a transaction.

4.2 Designing Business Processes in Web Applications with OOHDM

OOHDM focuses on three different design concerns: conceptual or application modeling, navigation design and interface design (which is ignored in this paper). OOHDM partitions the *conceptual model* into two types of classes: *entities* and *processes*. Entities model usual business objects; processes represent sets of activities that must be performed to achieve a goal. A *process* is a composite of activities, which encapsulate their own state (active, suspended, etc); control flow is further decoupled from activities and represented in the corresponding process.

In OOHDM, the *navigational model* describes the nodes and links of the hypermedia application. A*ctivity nodes* are the process counterpart of nodes and describe, in an abstract way, the visible attributes, anchors and operations with which the user will interact during process execution. Activity nodes are shown in the context of the corresponding process node to which they belong (a composite in OOHDM). Some links include "process" semantics, namely "suspend", "abort" and "terminate" links. These links complement the usual link navigational semantics by triggering a message to suspend (respectively abort, terminate) the activity that corresponds to the source node. When an activity is left, it is suspended, aborted or terminated and the corresponding process is aware of this change of state. This awareness is achieved as the outgoing links trigger the change of state in the corresponding activity/process. When the process is resumed it can then be started in the corresponding activity; at the same time, as activities are modeled as first class objects, they can store their state and be re-initiated safely.

4.3 OOHDM and UWA by Comparison

In this section we briefly show how we used the analysis performed in the previous sections to compare OOHDM and UWA. This comparison may serve as an example on how to assess different methods and select one of them, depending on their features, by using the proposed analysis framework.

Each of the two methodologies briefly summarized above have been evaluated separately over the requirements on which the analysis framework is based. The results of this evaluation are summarized in Table 2. Each row of the table refers to a particular requirement and indicates whether and on which design dimensions (B=Business, U=User, S=System), each methodology satisfies that requirement. The results are also graphically synthesized in Figure 3, where the two methodologies are located in a three-dimensional space depending on the number of requirements they satisfy on each axis.

As reported in Table 2, both OOHDM and UWA satisfy requirements Req1 and Req2 on the business level: both methodologies model the hierarchical structure of the transaction and its execution flows. Unlike UWA, OOHDM also satisfies Req2 on the conceptual level, being able to define which activities can be performed simultaneously. Both methodologies satisfy Req3, defining a state for each activity that allows the management of long-lived transactions.

Table 2. UWA and OOHDM evaluated and compared against the list of requirements

Requirement	Requirement classification	Level of satisfaction for UWA	Level of satisfaction for OOHDM
Req1	B - U	B	B
Req2	B - U	B	B - U
Req3	S	S	S
Req4	B - U - S	B – U - S	B – U - S
Req5	B - U	None	None
Req6	U - S	None	U - S
Req7	U	None	U
Req8	S	None	None
Req9	U - S	S	U - S
Req10	U - S	U - S	U - S

Both UWA and OOHDM satisfy Req4, while Req6 is satisfied only by OOHDM, which is able to show the interaction between the execution of the transaction and informational navigation including both "information objects" and "operational objects" in the Conceptual Schema and the Navigational Schema. Regarding Req9, OOHDM was found to fully satisfy this requirement since it defines a navigational context for customizing activities, while UWA satisfies Req9 only on the system level; it distinguishes the various times that the same activity is carried out but does not explicitly provide a way to customize the activity depending on the context of execution. OOHDM fails to satisfy only two of the requirements on the list, both of which are classified on the conceptual level. This may be seen in the graph in Figure 3 which shows that OOHDM does not completely cover the conceptual (User) dimension. Indeed OOHDM does not satisfy Req5, since it plans the control flow of the web transaction considering a single user, neglecting the common situation of having a number of players involved. Furthermore OOHDM does not satisfy Req8, given that it does not define any model representing which elementary operations (creation, cancellation, modification etc.) on the data and on the information objects involved are at the basis of each of the activities.

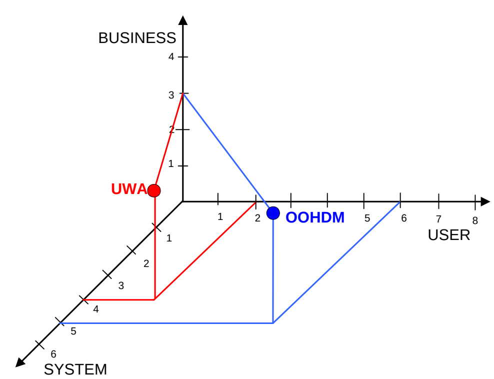

Figure 3. UWA and OOHDM located in the three-dimensional space.

The graph in Figure 3 indicates that, on the whole, OOHDM lends itself better than UWA to the design of business processes in web applications. OOHDM develops along all three of the design dimensions (Business, System, User) according to which the requirements for an ideal methodology are classified, thus considering the user-experience, the constraints of the business process, and the need for a correct functioning of the system.

The results provided by the analysis framework were used to define UWAT+ [10][17][18][19], an extended version of the UWA Transaction Design model that tries to overcome the identified shortcomings by adopting some of the solutions proposed by OOHDM and other methodologies.

5. CONCLUSIONS AND FUTURE WORK

The addition of business processes to modern web applications entails new challenges and risks to be faced when designing and implementing this kind of application and thus generates the need for the adoption of suitable design methodologies. Most of the design methodologies originally conceived for designing informative web applications have acknowledged this need by proposing extensions to include design activities and modeling concepts specifically tailored to cope with business process design issues.

In this scenario we have developed an approach for analyzing, evaluating and comparing currently available web application design methodologies with regard to their ability to design business processes in web applications. The approach is based on an extendible set of requirements we identified and classified; they arose from a number of case studies of the design and reverse engineering of business web applications and from the analysis of existing methodologies. The approach is useful for highlighting which design dimensions each methodology covers best (Business, User and System), and suggests possible directions in which a methodology should be extended in order to satisfy all the requirements. Hypotheses concerning the improvement of a given methodology in order to satisfy a particular requirement can be borrowed or derived from the solutions adopted by other methodologies that satisfy that requirement. This reasoning is behind the extension of the UWA transaction design modeling proposed in [10] and [19].

The application of the analysis framework to OOHDM and UWA included in this paper shows that it ought to be valid for the evaluation and comparison of other design methodologies. We are currently completing a more thorough comparison of web design methodologies, including, WSDM, UWE, OO-H and WebML.

REFERENCES

[1] Rossi, G., Schmid, H., Lyardet, F. Engineering Business Processes in Web Applications: Modeling and Navigation Issues. In *Proceedings of the 3rd International Workshop on Web Oriented Software Technology (IWWOST03)* Oviedo, Spain, 2003.

[2] Schmid, H., Rossi, G.: Modeling and Designing Processes in E-Commerce Applications. *IEEE Internet Computing*, January/February 2004.

[3] De Troyer, O., Casteleyn, S. Modeling Complex Process for web Application with WSDM. In *Proceedings of the 3rd International Workshop on Web Oriented Software Technology (IWWOST03)*, Oviedo, Spain, 2003.

[4] Koch, N., Kraus, A., Cachero, C., Meliá, S. Modeling Web Business Processes with OO-H and UWE. *3rd International Workshop on Web Oriented Software Technology (IWWOST03)*, Oviedo, Spain, 2003.

[5] Brambilla, M., Ceri, S., Fraternali, P., Manolescu, I. Process Modeling in Web Applications. *ACM Transactions on Software Engineering and Methodology (TOSEM)*, 2006, in print.

[6] UWA Consortium: Ubiquitous Web Applications. Proceedings of e2002 eBusiness and eWork Conference, Prague (Czech Republich), October 2002.

[7] Kappel, G., Pröll, B., Retschitzegger, W., Scwinger, W., Hofer, T. Modeling Ubiquitous Web Application: A Comparison of Approaches. *3rd International Conference on Information Integration and web-based Applications & Services*. Linz Austria, September 10-12, 2001.

[8] Alitalia.it. Online at www.alitalia.com.

[9] Chrysanthis, P., Ramamritham, K. Synthesis of Extended Transaction Models Using ACTA. *ACM Transactions on Database Systems* Vol. 19, No. 3 (Sept. 1994), pp. 450-491.

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

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

[12] Distante, D., Tilley, S., Huang, S. Web Site Evolution via Process Restructuring from a User's Perspective. In *Proceedings of the 6thIEEE International Workshop on web Site Evolution* (WSE 2004: Sept. 11-17, 2004; Chicago, IL). IEEE CS Press, 2004.

[13] Facecchia, A., C., Distante, D., Bochicchio, M. An Approach for Evaluating and Comparing Web Transaction Design Methodologies. Master thesis in Computer Science Engineering, University of Lecce, Italy, December 2004.

[14] Fischetto, G., Bochicchio, M. A., Distante, D. Designing Web Transactions: Theoretical and Practical Aspects. Master thesis in Computer Science Engineering, University of Lecce, Italy, February 2004.

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

[16] Object Management Group (OMG). Unified Language Modeling Specification (Version 2.0). www.omg.org. 2004.

[17] Distante, D., Tilley, S., Huang, S. Documenting Software Systems with Views IV: Documenting web Transaction Design with UWAT+. In *Proceedings of the 22nd International Conference on Design of Communication* (SIGDOC 2004: October 10-13, 2004; Memphis, TN). ACM Press, 2004.

[18] Distante, D., Tilley, S. Conceptual Modeling of web Application Transactions: Towards a Revised and Extended Version of the UWA Transaction Design Model. In *Proceedings of the 11th International Multi-Media Modelling Conference* (MMM 2005: Jan. 12-14, 2005; Melbourne, Australia). IEEE CS Press, 2005.

[19] Distante, D., Rossi, G., Canfora, G., and Tilley, S. A Comprehensive Design Model for Integrating Business Processes in web Applications. To appear in *International Journal of web Engineering and Technology (IJWET)*. Inderscience Publishers, Vol. 3, No. 1, 2007.