Documenting Software Systems with Views IV: Documenting Web Transaction Design with UWAT+

Damiano Distante
Dept. of Innovation Engineering
University of Lecce, Italy
damiano.distante@unile.it

Scott Tilley
Dept. of Computer Sciences
Florida Institute of Technology
stilley@cs.fit.edu

Shihong Huang
Dept. of Computer Science
Florida Atlantic University
shihong@cse.fau.edu

ABSTRACT
This paper describes an approach to documenting the conceptual design of Web Transactions using UWAT+. A Web Transaction is a collection of serial and/or parallel activities that contributes to achieving a user-oriented business objective using a Web-based application. UWAT+ is a meta-model for describing the various aspects of a Web Transaction in a holistic manner. It is an extension of the Transaction Design Model that is part of the Ubiquitous Web Applications (UWA) framework, a comprehensive framework for designing ubiquitous Web applications. A series of (extended) UML diagrams are used to graphically document the UWAT+ meta-model, which greatly facilitates adoption of the approach by practicing software engineers and technical writers. Use of the approach for documenting Web Transaction designs in both forward and reverse engineering processes is described.

Categories and Subject Descriptors
D.2.7 [**Software Engineering**]: Distribution, Maintenance, and Enhancement – *documentation*.

General Terms
Documentation, Standardization

Keywords
Web applications, transaction design, documentation, business processes, UWAT+, software views

1. INTRODUCTION
Business applications have become omnipresent on the Web. Complex systems such as e-commerce, e-government, and e-banking applications, enterprise information systems, and customer relationship management programs have adopted the Web as the meta-platform of choice. Such systems implement business processes by means of Web Transactions, which are collections of serial and/or parallel activities that contribute to achieving a user-oriented business objective.

SIGDOC'04, October 10–13, 2004, Memphis, Tennessee, USA.

The navigational nature of the Web, coupled with the operational and transactional nature of business applications, induces a unique design requirement on these kinds of applications: both hypermedia issues and transaction aspects need to be considered holistically. One the one hand, models and methodologies used in traditional hypermedia application design fall short when addressing transactional aspects. On the other hand, software engineering design practices and models that have proven successful for traditional (non-navigational, non-Web-based) applications fall short when addressing navigational issues.

There are two common practices for modeling Web Transactions that realize business processes. The first is emulating transactions as a sequence of navigational steps (or neglecting them entirely) [3]. The second is considering them as a second-class by-product of the overall site design process [4]. Both practices result in poor quality Web applications due to the presence of erroneous and unpredictable behavior, with undesirable effects on the realization of the business itself (e.g., low customer satisfaction due to low usability of the online service).

From a software engineering point of view, neglecting or improperly addressing Web Transaction design documentation causes numerous problems that can arise during long-term system evolution. These problems include:

- Difficulties in communication and understanding between designer and customer (first), and designer and developer (later), in the application process realization;
- Unpredictability of the development process and resultant application quality;
- Difficulties in verifying the traceability between business process requirements and transaction implementation.

This paper presents an approach for documenting Web Transaction design at the conceptual level that addresses some of these problems. The approach relies on a meta-model, named UWAT+, for describing the numerous aspects that, taken together, characterize a Web transaction. The UWAT+ meta-model [1] is an extension of the Transaction Design Model proposed as part of the Ubiquitous Web Applications (UWA) [14][13] design framework, a comprehensive framework for designing ubiquitous Web applications.

The next section of the paper discusses Web Transactions, various techniques that can be used to document their design, and

requirements for a documentation model working at the conceptual level. Section 3 provides an overview of the UWAT+ meta-model, which can be used to comprehensively document Web Transaction design using standardized graphical representations. Section 4 outlines guidelines for how UWAT+ might be used for documenting Web Transactions both of an existing Web application (reverse modeling) or for a building a new one (forward design). Section 5 summarizes the paper and outlines possible avenues of further work.

2. DOCUMENTING WEB TRANSACTIONS

A Web Transaction is defined as a sequence of Activities with the associated execution flow that enables the user to accomplish a task and/or reach a specific goal by means of a Web application. Thus, a Web Transaction can also be thought as a specialization of the concept of a workflow in a Web application, within the constraints implied by the underlying business process model.

In a Web Transaction, component Activities are sets of simple or more complex operations on the data and the contents the Web application elaborates. Examples of Web Transactions include the process of searching for and booking a flight on an airline's Web site, the process of searching for and bidding on an item using an online auction system, or the process of calling for tender management in a e-procurement system.

Web Transactions are a way for Web applications to support the implementation of business processes and to provide the user with services different than pure content navigation. The user makes use of the service provided by a Web Transaction by surfing the contents of the application and executing the Activities included in the Web Transaction in accordance with the associated execution flow. The execution of a Web Transaction entails both the navigation through the hypermedia contents of the Web application and execution of the related component Activities.

Web Transaction design can be accomplished at several levels of abstraction and several formalisms can be used to represent and document each of them. At least three levels of abstraction can be used: (1) conceptual level, (2) logical level, and (3) implementation level. The conceptual design gives a representation of the system (the Web Transaction in our scope) as perceived by the user and removed from implementation issues. The implementation design focuses on providing the system developers with all the specifications needed for realizing the single components of the system. The logical design is a middle level of abstraction design intended to translate the user-centered specifications of the conceptual design into terms of specifications closer to the implementation issues.

As with all software artifacts, the design of a Web Transaction can rapidly become very complex. The more complex the design, the more challenging it is for other team members to understand its nuances. Documentation is needed to communicate the rational behind the design. The documentation can be textual or graphical. Textual documentation has traditionally been used to record selected low-level implementation details. Since Web Transaction design is at the conceptual level, it is arguably best suited to be represented using graphical documentation.

Some of our own related work in the area has been directly related to the use of "views" as forms of graphical documentation [12][5][10]. Views in this context are visualizations of software systems that support multiple perspectives. For Web Transaction design, views can be effectively captured as a series of Unified Modeling Language (UML) diagrams.

From an adoption perspective, UML diagrams are the most popular form of graphical documentation for software systems. UML diagrams can be used during all phases of the life cycle of a software system. They can also serve several purposes. For example, UML diagrams are commonly used to represent user-level requirements, system-level architecture, and programmer-oriented low-level design details. UML diagrams are traditionally used in a forward engineering setting. However, recent work has also leveraged their value as an output of an automated reverse engineering process [8].

Documenting the conceptual design of a Web Transaction has the following minimal requirements:

1. Specifying the list of Activities in the Web Transaction is conceptually divided.
2. Specifying the execution rules associated with the set of component Activities.
3. Specifying the information needed-by / provided-to the user when executing each Activity and the interplay between Activity execution and hypermedia navigation.

The approach to documenting Web Transactions described below relies on extensions to the UML to graphically represent the conceptual design. The approach meets the requirements stated above by using the UWAT+ meta-model.

3. UWAT+

UWAT+ is a meta-model that can be used to document Web Transaction design. UWAT+ is composed of three sub-models:

1. The Organization Model: A customized version of a UML class diagram that describes the Transaction from a static point of view in terms of the user Activities it includes and the hierarchical and semantic relations among these Activities.

2. The Execution Model: A customized version of a UML activity diagram that models the Transaction from a dynamic point of view. It describes the possible execution flows provided to the user for carrying out the Activities involved in the transaction.

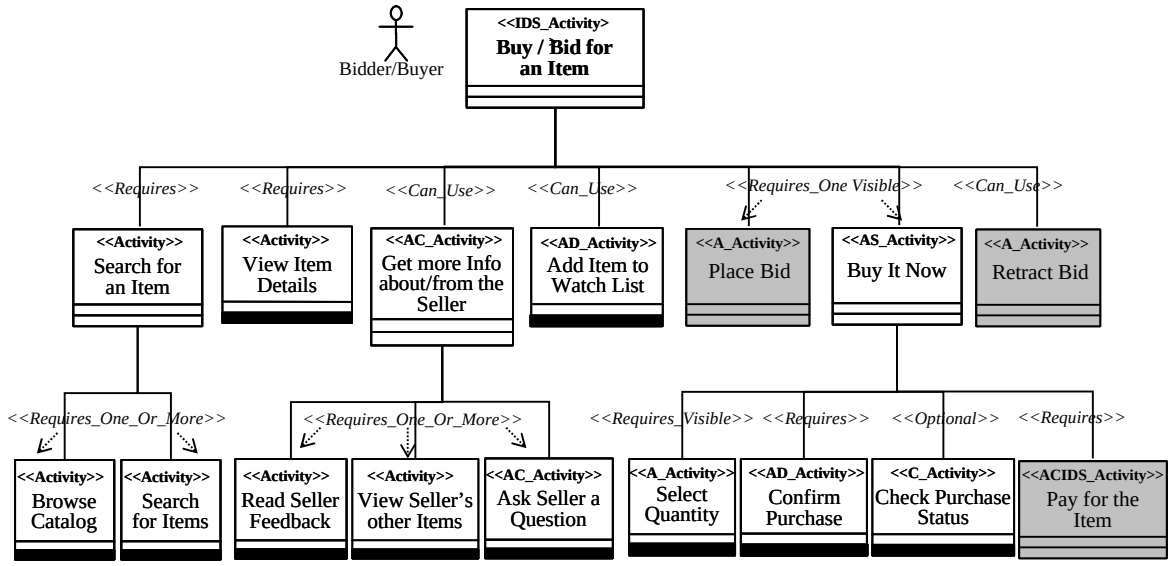


Figure 1: The Organization Model of the Web Transaction for "Buy/Bid for an Item" from the eBay Web site

3. The Navigation Model: A model that specifically describes the informative aspects of each Activity defined by the two previous models and the interplay between informative navigation and execution of the Web Transaction.

UWAT+ is a meta-model born as an extension of the UWA Transaction Design Model [17]. UWAT+ was specifically defined for documenting the conceptual design of Web Transactions by means of three graphical models: the Organization Model, the Execution Model and the Navigation Model. While the Organization Model and the Execution Model are extensions of their original version proposed by UWA, the Navigation model is a new contribution of UWAT+.

Each of the models included in UWAT+ focuses on and satisfies one of the three requirements listed in the previous section. Individually, the three models offer three complementary views of the conceptual design of a Web Transaction. Taken together, they cover all the major issues related with Web Transaction design and reverse modeling.

By focusing on separate aspects of the conceptual design of a Web Transaction, the three models are easy to produce and to read. When drawing the diagrams, the analyst concentrates on one particular characteristic of the Web Transaction design. Similarly, when reading the diagram, the reader is informed about a particular feature of the Web Transaction. The characteristics described and the information provided by the three models graphically documents in a complete and suitable way the conceptual design of a Web Transaction.

The reminder of this section provides an overview of each of the three models in UWAT+, with reference to the models obtained applying UWAT+ for graphically documenting the "Buy/Bid for an Item" Web Transaction of the eBay online auctions Web site [9] as viewed by a Bidder/Buyer of the application.

3.1 The Organization Model

Figure 1 shows the Organization Model recovered by analyzing the Web Transaction of "Buy/Bid for an Item" by direct inspection of the eBay online auction Web application. The model is a customized version of a UML class diagram [7] representing the Web Transaction from a static point of view. It lists the set of Activities involved in the Web Transaction, the relations (hierarchical: Requires, Requires One, and Optional; semantic: Visible, Compensates, and Can Use) among Activities, and the properties (ACIDS: Atomicity, Consistency, Isolation, Durability, Suspendability) associated with each Activity. Activities are represented by means of a class stereotype, arranged to form a tree with its root corresponding with the whole Web Transaction. The name and the set of ACIDS properties of an Activity are specified in the symbol of the stereotype while relations are specified by means of labels associated with the lines connecting them.

The main Activities involved in the "Buy/Bid for an Item" Web Transaction for the Bidder/Buyer type of user are "Search for an Item", "View Item Details", "Place Bid", and "Buy it Now". The first two Activities and one of the latter two are required to complete the Web Transaction. Three variants of the symbol of a class stereotype are used to describe and distinguish between: (a) Composed Activities (Activities composed by simpler Sub-Activities), (b) Elementary Activities (Activities not further decomposed in Sub_Activities), and (c) Composed Activities not detailed in the diagram. For example, "Buy It Now" is an example of Composed Activity, "View Item Details" is an Elementary Activity, and "Place Bid" is a Composed Activity not detailed in the diagram.

A label associated with the line connecting them describes the type of relation existing between two Activities. For example, "View Item Details" is a required Activity to complete the "Buy/Bid for an Item" Web Transaction. The user is also required to execute one of the "Place a Bid" and "Buy it Now" Activities.

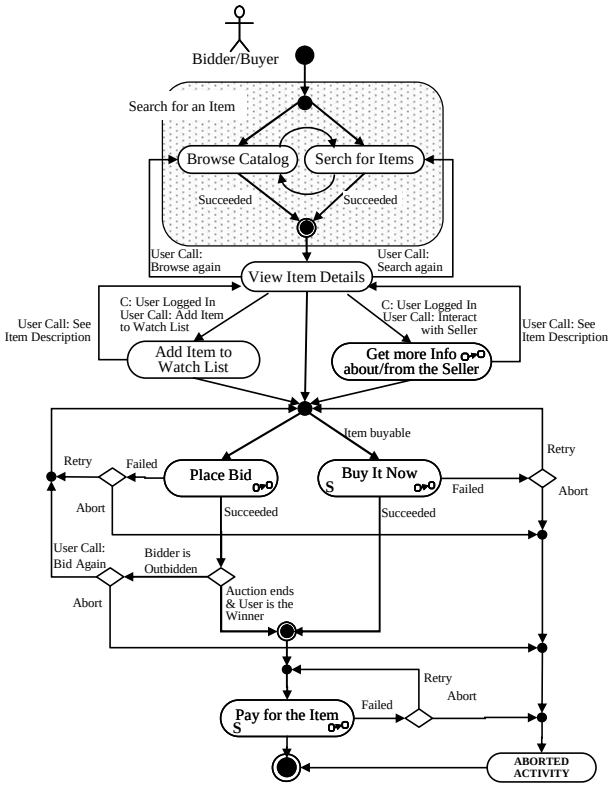


Figure 2: The Execution Model of the Web Transaction for "Buy/Bid for an Item" from the eBay Web site

In the first case, the user would like to bid for an item they are interested in, while in the second case they want to buy the selected item. The user can optionally execute the "Retract Bid" Activity, in case they want to cancel a bid previously made, and the "Get more Info about/from the Seller" Activity, in case they want to get more information before proceeding to bid for the item or buy it. These two Activities are not essential for the "Buy/Bid for an Item" Web Transaction, but are useful for the user when executing it.

The "Search for an Item" Activity is composed of two subActivites: "Browse Catalog" and "Search for Items". The user executes the main Activity by executing one or more of them.

The diagram in Figure 1 also provides other types of information, such as a description of the properties of each Activity. For example, the letters 'A' and 'S' inside the class stereotype for the "Buy it Now" Activity is to let the reader know that the Activity is Atomic and Suspendable. It is Atomic since all its required sub-Activities must be executed with success for it to be completed. It is Suspendable since it can be stopped during the execution of some of its sub-Activities and resumed later by the user.

3.2 The Execution Model

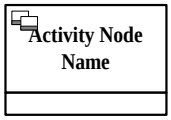
Figure 2 shows the Execution Model recovered for the Web Transaction from a dynamic point of view. It describes the executions flows among the Activities it includes. The Execution Model is a customized version of the UML activity diagram [7] in which Activities and sub-Activities are represented by states (ovals) and the execution flow between them is represented by state transition (arcs). Ovals with a symbol inside indicate Composite Activities whose Execution Model is further detailed elsewhere. Simple ovals correspond to Elementary Activities. A dotted oval represents a composite Activity whose sub-Activities are shown inside it. Thick arcs indicate mandatory execution flows (transitions towards required Activities), while thin arcs indicate optional execution flows (transitions towards optional Activities).

Each possible transition between Activities is explicitly represented with an arc between them. Associated with it is a textual description of the condition(s) required for the transition to happen, or the results of the execution of the source Activity. UML Swimlanes can be used to describe how two or more types of user of the Web application collaborate at the execution of a Web Transaction.

To "Buy/Bid for an Item" on the eBay Web site a Bidder/Buyer type of user first searches the item they is interested in by browsing the eBay catalog ("Browse Catalog" Activity) and/or by searching for a specific item ("Search for Items" Activity). When the user finds an item of interest, they view the item details ("View Item Details") and, optionally (if logged in the eBay system), they can add the item to their watch list ("Add Item to Watch List") or get more information about/from the seller ("Get more Info about/from the seller). For lack in space, the execution flow associated with the sub-Activities of the latter Activity are not detailed in the diagram in Figure 2, as indicated by the symbol , but would be described in another diagram (not shown).

The "Browse Catalog" and "Search for an Item" Activities can be repeated as many times as the user wants. When the user has found an item they are interested in, they can bid for the item ("Place Bid") or (if the chosen item is "buyable") buy the item directly ("Buy It Now"). Both these Activities are Composite Activities and, as indicated by the symbol , their execution flow is supposed to be detailed with another diagram.

If the user executes the "Buy It Now" Activity or if they bid for the item and becomes the winner of the auction (when the auction ends), the user is requested to pay for the won or bought item. The "Pay for the Item" Activity is suspendable, so the user has the option to complete it in a following session, but it is required, as shown by the thick line entering the oval representing it. This is also described by the "Requires" relation that exists between the "Pay for the Item" Activity and the main Activity "Buy/Bid for an Item" of the Organization Model presented in Figure 1. The Succeeded and Failed label associated with the outward arcs of an oval describes a positive and failed execution of the Activity associated with the oval, respectively.



Property	Description
Name	Name assigned to the Activity Node
Source	Name of the Activity associated with the node
Content	List of the information objects and eventually other nodes provided by the node to the user when execution the Activity specified by Source.
Input data	Data requested to the user for executing the Activity
Action elements	Buttons and other interaction elements the user can use to invoke the user calls associated with the Activity in the Execution Model.

Figure 3: The notation and table contents that defines an Activity Node.

3.3 The Navigation Model

The Navigation Model associated with a Web Transaction describes the informative aspects of the execution of the Activities involved in the Web Transaction and the interplay between content navigation and Activities execution. The model is based on two main constructs: the Activity Node and the Activity Cluster.

As specified by the UWA Navigation Model [16], a Navigation Node (or Node for short) defines a unit of information suitable for user "consumption" that is provided to the user as a whole. An Activity Node is defined as an interactive Navigation Node intended to specify:

- The contents provided and the data requested to the user when executing an Activity;
- The action that the user can invoke when visiting the node associated with an Activity and corresponding to the User Call of the Execution Model.

To specify the possible navigation paths among Nodes in a given context, the UWA Navigation Model makes use of the concept of Navigational Cluster (Cluster for short). An Activity Cluster is defined as a new type of Cluster intended to describe:

- The navigation allowed to the user when visiting an Activity Node and how it affects the state of the ongoing Transaction (suspending, completing or aborting it);
- The navigation associated with the execution of an Activity or the invocation of an action by the user when visiting the node associated with an activity.

Typically, an Activity Node and an Activity Cluster are designed for each Elementary Activity of a Web Transaction. An Activity Node is defined by means of a table and represented with a stereotype of a class. Both the contents of the table and the symbol are shown in Figure 3.

An Activity Cluster typically encloses one or more Activity Nodes and Navigation Nodes and defines the navigation links among them. It also specifies the state transitions of the ongoing Web Transaction associated with each navigation link. Figure 4 shows the Activity Cluster obtained for the Activity of "View Item Details" analyzing the eBay Web page in charge of enabling the user to execute this Activity. Figure 5 shows a screenshot of this Web page and highlights the Activity Nodes and navigation links described by the model in Figure 4.

As graphically documented in Figure 5, when browsing the Activity Node "View Item Details" the user can navigate towards the Navigation Node "Seller's eBay Store Homepage" or "Seller Information". Actually, no real navigation step is needed to go from one node to the other, since they are published on the same Web page where the "View Item Details" Activity Node is.

The "View Item Detail" Node is the Activity Node associate with the homonymous Elementary Activity of the Web Transaction. The "Seller's eBay Store Homepage" Node is the Navigation Node corresponding to the homepage of the eBay seller's store for the selected item. The "Seller Information" Node is a Navigation Node that provides the user with basic information on the seller of the item.

The two symbols  and  are used to distinguish between Activity Nodes and Navigation Nodes. The other main navigation links the user can follow when viewing an item details are the ones towards the Activity Clusters associated with the following Elementary Activity: "Place Bid", "Buy It Now", "Get more Ingo about/from the Seller", "Search for Items" and "Browse Catalog".

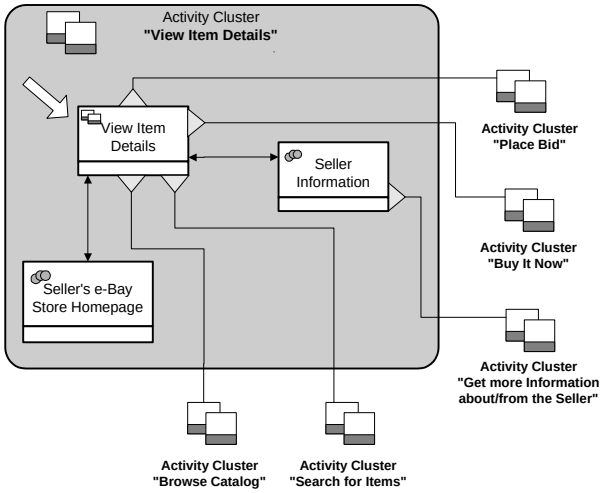


Figure 4: The Activity Cluster for "View Item Details".

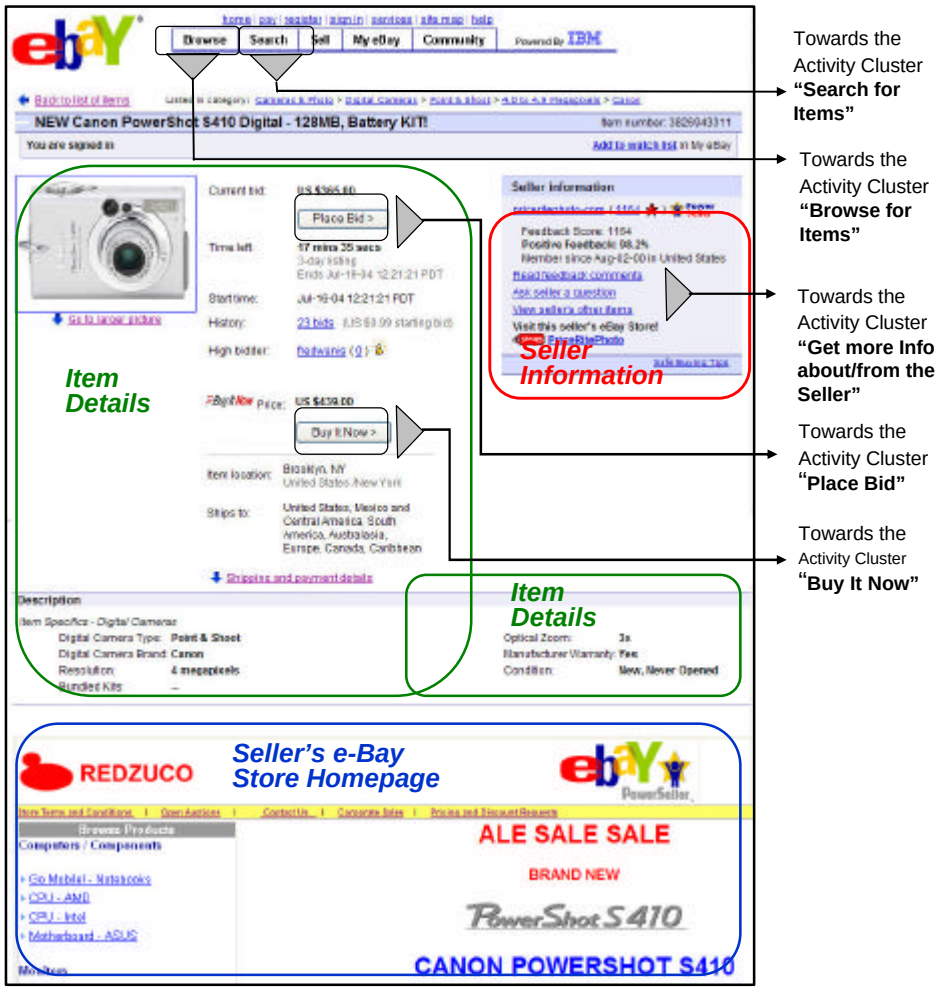


Figure 5: A screenshot of the Web page for the activity of "View Item Details" in the e-Bay Web site.

These navigation links causes the associated Activity to be started. The clusters are shown with icons outside the "View Item Details" boundaries.

For the examined Activity, none of the possible navigation links causes the ongoing Web Transaction to be suspended nor aborted. In a different case, a label upon the link describes the state transition associated with a navigation link.

4. DOCUMENTING VIEWS USING UWAT+

The UWAT+ meta-model described in Section 3 can be used both for documenting the Web Transaction design of an existing Web application, and to produce the conceptual user-centered design of a Web Transactions for a new Web-based application.

A Web Transaction reverse engineering technique based on a preliminary version of the UWAT+ meta-model is detailed in [2] and [11]. The reengineering technique has been found to be useful both for Web site evolution purposes and program understanding objectives. The reengineering technique consists of recovering the Execution Model first and then the Organization Model for the Web Transaction of interest. The model so recovered enables the analyst to evaluate the Web Transaction design from the user perspective and to plan possible changes to improve the user's experience.

When using UWAT+ for designing the Web Transactions of a new Web application, the design process follows basically the UWA design methodology [13]. The design starts with the Requirement Elicitation design activity [15] that is intended to find out the stakeholders of the application, their goals, and to refine these goals to define the requirements for the Web-based application.

The design of the Web Transactions begins with "procedural" goals to flesh out the Requirement Elicitation activity, which will result in an initial Web Transaction design. "Procedural" goals are those goals the user can fulfill with carrying out a set of Activities by means of the Web application.

The design process of a Web Transaction is structured to produce the Organization Model first, then the Execution Model, and finally the Navigation Model. To produce the Navigation Model,

the contents of the Web application and the access structures to them need to be designed first. This can be accomplished by means of the UWA Information Design activity [16]. Once the Information design is created, the information objects defined in it can be used to design the Activity Nodes and the Activity Clusters of the UWAT+ Navigation Model for each of the Web Transactions to design.

Irrespective of the lifecycle direction in which the UWAT+ approach is used (forward or reverse), the outcome is a set of extended UML diagrams that document the Web Transaction design. Assuming the reader is already familiar with the UML, the extensions are sufficiently minor so that little cognitive dissonance should arise. An advantage of using graphical documentation (in the form of extended UML diagrams) is that they succinctly depict salient aspects of the design in a standardized and uniform manner. The textual descriptions of Figures 1, 2 and 4 are much longer, and in some ways less clear, than their graphical counterparts. The graphical view provides a more complete picture of the Transaction design, which in turn helps the software engineer and/or technical writer gain a better understanding of the conceptual model.

5. SUMMARY

Documentation has long played an important role in aiding system understanding. Graphical forms of documentation that rely on software visualization techniques to make complicated information easier to understand are commonplace during reverse engineering. Graphical forms of documentation that exploit the familiarity of standardized representations, such as the UML, are commonplace during forward design.

This paper presented an approach to graphically documenting the conceptual design of Web Transactions using UWAT+, which is a meta-model for describing the various aspects of a Web Transaction in a holistic manner. The graphical documentation produced using the UWAT+ approach consists of a collection of views that are represented as (extended) UML diagrams. Since the UML is the de facto standard for modeling modern software applications, the use of UML in this context can greatly facilitate adoption of the approach.

The UWAT+ meta-model can be used both in reverse and forward engineering processes. It is able to capture and describe several aspects of the design of Web Transactions. These design aspects include the set of activities (unit of work) in which the Transaction is decomposed, the rules that govern the execution of the Activities, the Activities' properties such as their suspendability, the information to provide to the user when executing each Activity, the contents of the application affected by the execution of an Activity, the navigation paths the user is allowed to follow when executing a given Activity and their influence on the state of the Transaction, and so on.

There are a number of avenues of future work in this area. When using the approach in a reverse modeling manner, there is currently no tool support available to support the process. Having an automated aid would aid completeness, correction, and adoption.

There is a clear need for evidence-based studies to assess the efficacy of the UML diagrams produced by using the UWAT+ approach. Stylistic issues, spatial layout considerations, and selective content elusion are all areas that warrant further attention. Fortunately, there is already some initial work in this area, including workshops at SIGDOC such as GDOC 4 [6].

REFERENCES

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

[2] Distante, D., Parveen, T., and Tilley, S.: "Towards a Technique for Reverse Engineering Web Transactions from a User's Perspective." *Proceedings of the 12th IEEE International Workshop on Program Comprehension* (IWPC 2004: June 24-26, 2004; Bari, Italy), pp. 142-150. Los Alamitos, CA: IEEE CS Press, 2004.

[3] G. Rossi, H. Schmid, F. Lyardet. "Engineering Business Processes in Web Applications: Modeling and Navigation Issues." *Proceedings of the 3rd International Workshop on Web-Oriented Software Technology* (IWWOST 2003: July 15, 2003; Oviedo, Spain).

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

[5] Hartmann, J.; Huang, S.; and Tilley, S. "Documenting Software Systems with Views II: An Integrated Approach Based on XML." *Proceedings of the 19th Annual International Conference on Systems Documentation* (SIGDOC 2001: October 21-24, 2001; Santa Fe, NM), pp. 237-246. ACM Press: New York, NY, 2001.

[6] Murphy, S.; Tilley, S.; and Huang, S. "The 4th Workshop on Graphical Documentation: UML Style Guidelines." To be held as part of *The 22nd Annual International Conference on Design of Communication* (SIGDOC 2004: October 10-13, 2004; Memphis, TN).

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

[8] Pierce, R. and Tilley, S. "Automatically Connecting Documentation to Code with Rose." *Proceedings of the 20th Annual International Conference on Systems Documentation* (SIGDOC 2002: October 20-23, 2002; Toronto, Canada), pp. 157-163. ACM Press: New York, NY, 2002.

[9] The eBay online auctions Web site. Online at www.ebay.com.

[10] Tilley, S. and Huang, S. "Documenting Software Systems with Views III: Towards a Task-Oriented Classification of Program Visualization Techniques". *Proceedings of the 20th Annual International Conference on Systems Documentation* (SIGDOC 2002: October 20-23, 2002; Toronto, Canada), pp. 226-233. ACM Press: New York, NY, 2002.

[11] Tilley, S., Distante, D., Huang, S.: "Web Site Evolution via Transaction Reengineering." *Proceedings of the 6th IEEE International Workshop on Web Site Evolution* (WSE 2004:

September 11, 2004; Chicago, IL), pp. 31-40. Los Alamitos, CA: IEEE Computer Society Press, 2004.

[12] Tilley, S.; Müller, H.; and Orgun, M. "Documenting Software Systems with Views." *Proceedings of the 10th Annual International Conference on Systems Documentation* (SIGDOC '92: October 13-16, 1992; Ottawa, Canada), pp. 211-219. ACM Press: New York, NY, 1992.

[13] UWA Consortium, Deliverable D2: General Definition of the UWA Framework. Online at www.uwaproject.org (2001).

[14] UWA Consortium, Ubiquitous Web Applications, *Proceedings of e2002 eBusiness and eWork Conference* (October 2002; Prague, Czech Republic).

[15] UWA Consortium. Deliverable D6: Requirements Elicitation: Model, Notation and Tool Architecture. Online at www.uwaproject.org, 2001.

[16] UWA Consortium. Deliverable D7: Hypermedia and Operation design: model and tool architecture. Online at www.uwaproject.org, 2001.

[17] UWA Consortium. Deliverable D8: Transaction design, Online at www.uwaproject.org, 2001.