Specification and design of workflow-driven hypertexts

(This research is part of the WebSI (Web Service Integration) project, funded by the EC in the Fifth Framework)

Marco Brambilla
Dipartimento di Elettronica e Informazione, Politecnico di Milano
Via Ponzio 34/5, 20133 Milano, Italy
+39 2 23 99 34 08
mbrambil @elet.polimi.it

Stefano Ceri
Dipartimento di Elettronica e Informazione, Politecnico di Milano
Via Ponzio 34/5, 20133 Milano, Italy
+39 2 23 99 35 32
ceri@elet.polimi.it

Sara Comai
Dipartimento di Elettronica e Informazione, Politecnico di Milano
Via Ponzio 34/5, 20133 Milano, Italy
+39 2 23 99 34 74
comai@elet.polimi.it

Piero Fraternali
Dipartimento di Elettronica e Informazione, Politecnico di Milano
Via Ponzio 34/5, 20133 Milano, Italy
+39 2 23 99 73 29
fraterna@elet.polimi.it

Ioana Manolescu
INRIA Futurs
78153 Le Chesnay, France
+33 1 39 63 51 46
mailto:ioana.manolescu@inria.fr

ABSTRACT

While the Web consolidates as the ubiquitous application delivery platform, the features of Web applications evolve to cover new requirements, like the capability of managing complex workflows spanning multiple users and organizations. This scenario challenges the Web engineering methods to address a broader class of applications. This paper introduces Workflow Driven Hypertexts, defined as Web-enabled hypertextual applications serving the workflow of multiple users, and proposes a design method integrating process, data, and hypertext modeling, expressly conceived to support the development of this class of applications.

Keywords

Web site design, web site management, workflow, application integration.

1. INTRODUCTION

Web engineering is about applying the principles, techniques, and best practices of software engineering to Web application development. One of the key ingredients of software engineering is conceptual modeling, that is, the ability of describing the properties of an application at a high-level, for the purpose of exchanging requirements and automatically generating the implementation code. Conceptual modeling has been successfully applied to hypertextual applications for the Web [3, 2] and is gaining momentum also in the market of software development tools.

However, as time passes, the mission of Web applications evolves, from online information browsing to full-fledged collaborative applications, supporting workflows spanning multiple individuals and organizations. In this paper we propose a fusion of data, hypertext, and workflow modeling concepts and methods into a novel approach for modeling lightweight Web-enabled workflows; this approach extends the benefits of high-level conceptual modeling and automatic code generation to a much broader class of

Web applications.

2. FEATURES OF WORKFLOW-DRIVEN HYPERTEXTS

Integrating hypertexts with workflows means delivering Web interfaces that permit the execution of activities and embody constraints that drive the navigation of users. We call these applications *workflow-driven hypertexts* (WFdHT, for short).

Figure 1 pictorially contrasts the characteristics of workflow managements systems, of hypertext applications, and of WFdHTs. "Pure" workflow management systems are characterized by a rich set of constraints, do not allow free navigation, and expose only the data needed for accomplishing the workflow objective. "Pure" hypertexts, at the opposite, do not constrain navigation nor establish the accessible information a priori. WFdHTs lay in the middle of the spectrum. Constraints, the degree of navigation freedom, and the amount of displayed information can be tuned to obtain different application flavors.

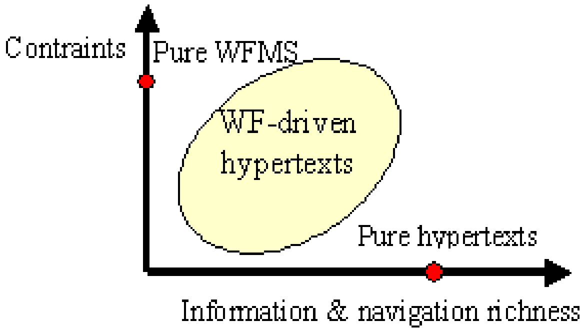

Figure 1 – Hypertexts vs. workflows

We advocate a development process for WFdHTs centered on *conceptual modeling*, blending three ingredients, representing orthogonal perspectives:

- The *process model* represents the process to be executed, in terms of the activities that compose it, their precedence constraints, and the actors in charge of executing each activity.
- The *data model* represents the domain objects published and manipulated in the application, and possibly the meta-data necessary for implementing the application (e.g., data about the users, auditing data, and so on).
- The *hypertext model* specifies the elements of the interface for executing the activities, in terms of the pages delivered to the user, the content published in each page, the operations that can be invoked from the interface, and the links necessary to enable navigation and operation invocation; the boundaries of activities (i.e., where in the application interface an activity can be started and terminated), and the operations necessary to track the execution of the process.

We have defined these features as an extension of the WebML language [1], a high-level notation for specifying Web applications. In WebML the database content is described using the Entity-Relationship model, and hypertexts are represented as graphs of *pages* to be presented on the Web. Pages enclose *content units*, representing atomic pieces of information to be published (e.g., indexes listing item lists, details of a single object, entry forms, and so on); content units have a *selector*, which is a predicate identifying the entity instances to be extracted from the underlying database and displayed by the unit. Pages and units can be connected with *links* to express a variety of navigation effects. Besides content publishing, WebML allows specifying *operations*, like creating, modifying and deleting entities and relationships, sending e-mail, login and logout, e-payment, and so on. Further details about WebML can be found in [1].

The extensions required for supporting WFdHT include the addition of some standard entities to the data model (for recording activities, activities instances and process cases) and of few primitives to the hypertext concepts (for defining activities, for assigning objects to activities, and for expressing process constraints). The new hypertext primitives are graphically represented in Figure 2 . *Start and end activity* units delimit the portion of hypertext associated with the execution of an activity. They can be tagged respectively as the start of the case, when the activity to start is the first one of the entire process, or, dually, as the end of the case, when the activity is the last one of the entire process. *Assign* operations are conceived for connecting an application object (or a set of objects) to an activity instance, typically to specify the future job to be performed on the object. Objects connected to the instances of a particular activity can be retrieved by means of *workflow-aware content units* (e.g. indexes of such objects). Finally, *if* and *switch* units can be used to express navigation conditions. Together with assign/workflow-aware content units, they allow to represent any process constraint described by the process model.

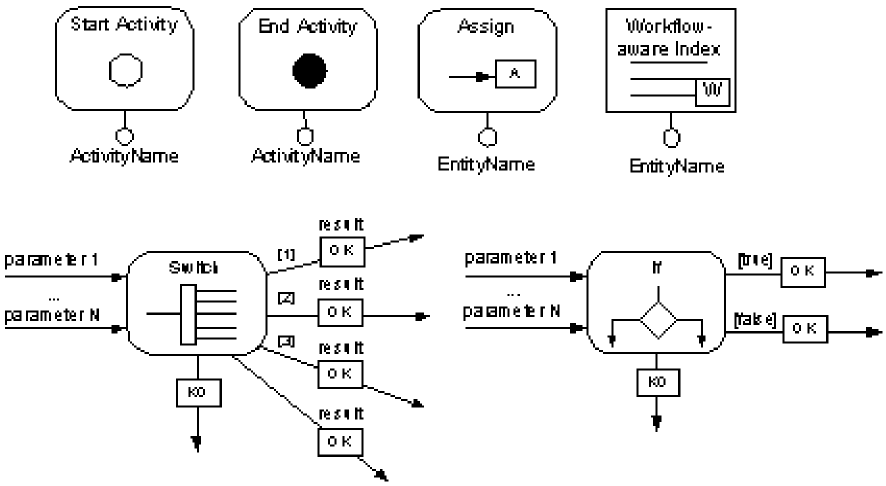

Figure 2 – Workflow primitives

Example. Figure 3 presents a WFdHT implementing a simple process consisting of two activities performed by two different users: the creation of a document and its translation. Two hypertexts (Creator and Translator sites) represent the operations required for performing the two activities. In the first hypertext the first kind of user starts the process and the creation activity by following a link exiting the home page: the activity consists in filling a form; when submitted, a new document is created and assigned to the translation activity (by means of the assign operation). The activity is ended and the user is led back to the home page. In the second hypertext the translator first selects a document to be translated from an index showing the documents assigned to the translation activity (through a workflow-aware index unit), then the translation activity starts for the selected document. The user fills in a form with the translation, and a new document is created and associated with the original one. This activity ends the whole process.

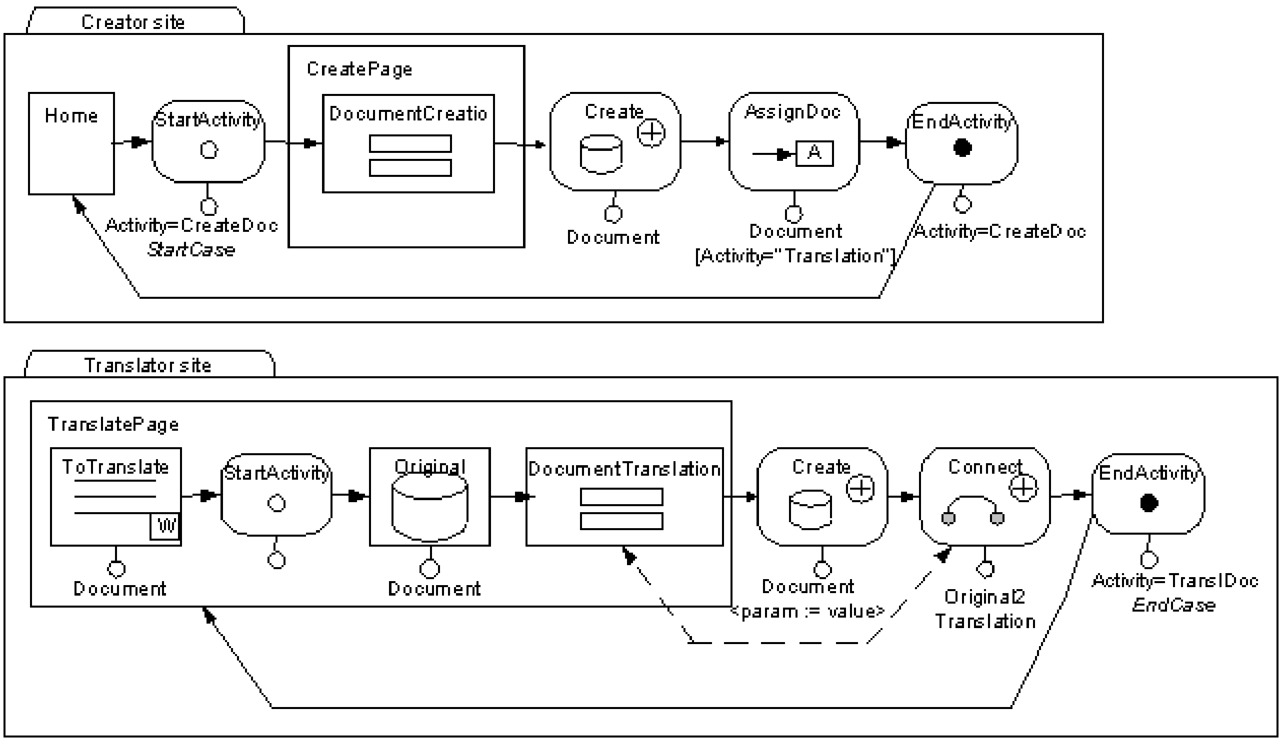

Figure 3 – Hypertext implementing a simple workflow

3. IMPLEMENTATION AND EXPERIENCE

The modelling approach briefly introduced in this paper has been experimented in the context of a real-world CASE tool, called WebRatio [4], which supports the model-driven development of "traditional" Web applications. Our ideas have been benchmarked in the development of a real workflow-driven Web application (www.aceradvantage.com). The application deals with the process of requesting, checking, and activating a warranty extension for laptops and personal computers, which requires the execution of a workflow spanning several individuals and business units of the service provider. The application is a perfect representative of the concept of WFdHT, because it requires the use of hypertextual interfaces, to which the customer is accustomed, the delivery of data (e.g., the warranty contracts) and the enforcement of process constraints. The application was successfully delivered in a very short time (five weeks), and has been generated automatically from a conceptual model described with WebML enhanced with workflow primitives.

4. REFERENCES

1. S. Ceri, P. Fraternali, A. Bongio, M. Brambilla, S. Comai, M. Matera: Designing Data-Intensive Web Applications, Morgan-Kaufmann, December 2002.
2. J. Conallen: Building Web Applications with UML. Addison Wesley (Object Technology Series), 2000.
3. WebML Project Homepage: http://www.webml.com
4. WebRatio Homepage: http://www.webratio.com