Conceptual modeling of Web applications featuring e-services and workflows

Piero Fraternali, Marco Brambilla

Joint work with: Stefano Ceri, Aldo Bongio, Sara Comai, Maristella Matera, Ioana Manolescu, Roberto Acerbis, Stefano Butti, Emanuele Molteni

Tutorial Outline

1. Introduction to model-driven Web application development
   - Web Modeling Language
     • Primitives for modeling in the large
     • Primitives for modeling in the small

2. From data-centric to process and service centric applications
   - Requirements
   - Extending WebML with service modeling primitives
   - Extending WebML with workflow modeling primitives

3. Implementation notes

4. Ongoing and future work
1. Introduction to Web application conceptual modeling

Data-intensive Web applications

- Enable users to access and/or manipulate content through the Web
  - Browse & buy books
  - Make hotel (plane, or opera) reservations
- **Data Design:**
  - E/R or UML help describe data requirements visually in a platform independent way
  - Easily mapped to different logical models

- **Hypertext Design**
  - Hypertextual interfaces (content+navigation) should also be modelled at high level in a platform independent way
  - Hypertext diagrams should be (easily) mapped to physical structures (page templates, data extraction components, ..)
Hypertext conceptual modelling with WebML

- Visual Web application modeling language
- WebML specifications consist of:
  - Data model (E-R, UML class diagrams)
  - One or more hypertext models (site views)
- Presentation is dealt with a standard language (XSL rules)

Reference

- Available in Italian from McGraw Hill
- Online reference: www.webml.org
  - White papers, tutorials, slides, teaching materials, 40+ solved exercises
Example: E-R schema for a ski portal web site

Hypertext modeling in the large: SITEVIEW, AREA, PAGE

- **Siteview**: a set of pages and/or areas forming a coherent view of the site. Multiple site views can be defined on the same data model for different users or publication media.

- **Area**: a set of logically homogeneous pages
  - Examples: Sections of a portal: Sport, Music, HiTech, ...
  - Areas can be nested, so that sub-areas can be defined inside areas

- **Page**: a container of one or more pieces of information shown to the user at the same time
Hypertext modeling in the small: CONTENT UNITS

- A content unit is the atomic element for publishing content in a page
  - unitX
  - container

- It is a “view” defined upon a container of objects:
  - All the instances of an entity
  - Instances of an Entity that meet a selection condition called selector
**PREDEFINED CONTENT UNITS**

**DATAUNIT**
- **Purpose:**
  - allowing users to see data
  - to choose items
  - to insert data

**INDEXUNIT**
- **Selector:**
  - conjunction of conditions

**MULTIDATAUNIT**
- **Selector:**
  - conjunction of conditions

**HIERARCHICAL**
- **SCROLLER**
- **MULTICHOICE**
- **ENTRYUNIT**

**Examples**

**ResortData**
- **Resort** [Name="Cortina"]

**ResortIndex**
- **Resort**
- **NEST Slope** [ResortToSlope]
Navigation & context: LINKS

- A link is an oriented connection between two units (source unit and target unit), normally rendered by means of anchors or submit buttons.
- Purpose of links:
  - Allowing the user to move from one place to another
  - Transporting information from one place to another
  - Activating a computation (side effect)

Example of links

Which resort’s hotels?
- Resort
  [OID=p1]

Which hotel?
- Hotel
  [Resort2Hotel(p2)]
- Hotel
  [OID=p3]

Hotel Detail
- Hotel 1
- Hotel 2
- Hotel 3

La Thuile
A marvellous...
photo:
### Modeling business logic: OPERATION UNITS

**Purpose:**
Updating the database or performing arbitrary actions

<table>
<thead>
<tr>
<th>Operation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CREATE</td>
<td>Create Entity</td>
</tr>
<tr>
<td>DELETE</td>
<td>Delete Entity</td>
</tr>
<tr>
<td>MODIFY</td>
<td>Modify Entity</td>
</tr>
<tr>
<td>CONNECT</td>
<td>Connect Relationship</td>
</tr>
<tr>
<td>DISCONNECT</td>
<td>Disconnect Relationship</td>
</tr>
<tr>
<td>SEND MAIL</td>
<td>Email</td>
</tr>
</tbody>
</table>

### Semantics of operation calls

- An operation receives in input one or more links, possibly bringing parameters. One activates, the others only transport.

- Two kinds of outgoing links:
  - **OK link**: followed when operation succeeds
  - **KO link**: followed when operation fails

- Operations do not publish content → positioned outside pages
From data-centric to service & process centric applications

- WebML models synchronous user-driven applications
  - Nothing happens until the user clicks
  - Interfaces are displayed to humans
  - Navigation is (quite) unconstrained
- B2B applications and enterprise application integration demand:
  - Application-to-application interfaces
  - Asynchronous interactions
  - Workflow enactment and control (constraints on users’ navigation)
- A “complete” modelling language should support
  - Web Services as a means of interfacing to peer systems
  - Workflow primitives to structure the interaction (human 2 application, application 2 application)

2. Extending WebML with support for Web services
**Service integration layer**

- **Choreography or collaboration layer**
  - Describes the interaction of services within a process
  - Business processes are essential to build complex applications over the message-based paradigm.

- **Example: BPEL4WS**
  - An implementation language: it allows to define new Web services as composition of existing ones
  - The composed service is described by means of WSDL, just as any other Web service
  - The *process specification* describes how the composed service plugs into the global process
Application scenario

- Ski Portal stores touristic info on resorts and slopes plus name/address of hotels
  - Reservations are done through the HMS web services
  - Payment is done through the PS web services
  - The HMS also delivers auxiliary services usable by the Ski Portal (e.g., hotel news, event notifications, user comment management)

Types of interactions (as envisioned in WSDL)

- Central component of a web service: operation = 1 or 2 msg
  - Request-response
  - Solicit-response
  - Notification
  - One-way
Two-message operations may be used

- **Synchronously**: no action is taken between 1st and 2nd msg and the sender waits
- **Asynchronously**: action is performed between the messages, the sender does not wait

**Conversations**

- A Web service conversation is a set of correlated messages involving one or more Web services (and Web applications)
- Conversations entail additional semantics
  - Messages belonging to the same conversation must be recognized and traced
  - Timeouts or error conditions can be specified for the messages of a conversation
  - Exceptions should be handled
Extending WebML to support interactions with web services

- **Goal:** modeling the interaction between the Web application, the human users, and the remote systems
- **New modeling primitives are required:**
  - Data model: specific entities modeling the interaction history with Web services
  - Hypertext model: primitives for handling the various classes of service operations
- **Run-time support for Web services**
  - Ability of exchanging messages
  - Integration of application data and Web services XML messages
  - Support for conversations

**Meta-data model for supporting Web services**
New WebML constructs for supporting Web services

- One WebML operation for each class of Web service operations
  - ReqRsp
  - SolRsp
  - AsReqRsp
  - AsSolRsp
  - OneW
  - Notif

- Marks for operations that start, resp. end conversations
  - ReqRsp
  - SolRsp

WebML service operations are macros

1. Create a new Conversation Instance
2. Create a new Operation Instance
   - Name="getOffer", …
3. Connect the Operation Instance to the Conversation Instance
4. Compose parameters of the incoming links into an XML message
5. Send XML message; block waiting for answer
6. Decompose XML answer
   - may involve populating the entities of the underlying data schema
7. Export selected element from the answer as parameters of the outgoing links
Hypertext of the Ski Portal with service calls (Customer Site View)

Room search
- RoomCrit
  - From
  - To
- Offers
  - RoomOffer
- Book

Reservation
- User data
- [OID=CurrentUser]
- New
  - Reservation
- CreditInfo
  - CCNo
  - Exp.

Confirmation
- Confirmed
  - Reservation
- Pay
  - ReservConv

Hypertext with asynchronous messaging

Added new entity Comment(OID, subject, text, answer)
+ Relationship Comment_User

Send comment
- User data
  - User [OID=currentUser]
- Comment
  - Subject
  - Text

Create
- Comment
  - Comment2User

Connect
- Comment2User
  - User [OID=currentUser]

MyComm
- Comment
  [Comment2User] [answer!=null]
  - Answered
  - Comment

Answers to my comments
- User data
- [OID=currentUser]

Update
- Comment
  <answer:=A>

A: text
C: Comment.OID

Comment
  CommConv
WebML primitives for data transformation

- Web services communicate through XML messages (SOAP)
- Marshalling and transformation between messages and from/into structured data is required
- XML-in & XML-out:
  - marshalling
  - XML documents $\leftrightarrow$ ER instance
- Adapter unit:
  - transformations from XML document to another XML document (through user-defined XSL stylesheet)

Examples

- Book
  - New
  - ReservConv
  - Reservation
- Book
  - Adapt
  - Store
  - XML
  - New
  - ReservConv
  - Reservation
- Pay
  - Confirm
  - ReservConv
- Pay
  - Adapt
  - Confirm
  - ReservConv
Web services references

- [SOAP], [WSCL], [WSDL], [XQuery], [XSchema]
  [http://www.w3.org/TR](http://www.w3.org/TR)
- [UDDI] [http://www.uddi.org](http://www.uddi.org)
- [WSCI] [http://wwws.sun.com/software/xml/developers/wsci](http://wwws.sun.com/software/xml/developers/wsci)
- [XLang] [http://www.gotdotnet.com/team/xml_wsspecs/xlang-c](http://www.gotdotnet.com/team/xml_wsspecs/xlang-c)

3. Extending WebML with workflow primitives
Basic workflow concepts

- Workflow Management Coalition definition
  - WF: Complex interaction involving several users
- **Process:** type of the interaction
- **Case:** instance of the interaction
- Process structure:
  - Activity: performed by 1 user
  - Activity composition primitives:
    - Sequence
    - And-split, and-join, or-split, or-join
    - Pre-conditions, post-conditions etc.

Workflow-driven hypertexts

- An hypertext with “some” contraints on the use of the hypertextual interfaces
- Constraints are other than those due to data dependencies to be preserved by navigation, they stem from the process to be followed
- Constraints span the site views of multiple actors (e.g., “you cannot do this until I do that”)
The WF-hypertext continuum

PF-driven hypertexts

Pure WFMS

Constraints

Pure hypertexts

Information & navigation richness

Meta-data model for supporting workflows

Workflow Data Model

Group

ActivityType

Process

User

Activity Instance

Case

Application data

0:N

Assigned To

1:1

Type

0:N

StartTimeStamp

Execution By

1:1

Name

0:N

EndTimeStamp

0:N

Name

1:N

Status

0:N

PartOf

StartTimeStamp

1:1

Type

EndTimeStamp

Assigned To
Workflow-Driven Hypertext
Design Principles

• Each user role maps into a site view
  – The site views offers the interfaces for performing all the activities associated with the user’s role

• Each activity maps into one area or set of correlated pages
  – Entering the pages entails starting the activity
  – Leaving the pages entails ending the activity, thus enabling the subsequent ones according to the WF schema
  – Page content and navigation may depend on activity status and assigned work load

WebML units for activity & case management

• Managing activities

  Start Activity
  ActivityName

  End Activity
  ActivityName

  Meaning: they create and update the metadata describing the workflow progress

• Managing cases

  Start Activity
  ActivityName
  StartCase

  End Activity
  ActivityName
  EndCase
Units for conditional navigation

- IF unit

- Switch unit

Units for work assignment and progress checking

- Assign unit
  Assigns an object to be processed in an activity

- Get Activity Status
  Retrieves the status of an activity

- Workflow-aware content unit
  Shortcut: extracts relevant data for a specific activity instance or user
**Process model**

- A simple loan requests management system
- Workflow-driven application
- Double approval
- Check of applicant’s financial and job status

**Data model**

- Workflow data model
- User model
- Application specific data model
• The customer submits his personal information to apply for a loan

• A new Application is created and assigned to next activity (Manager’s preliminary Validation)

• Personal information submission (with “cancel activity” option)
Manager sv:
Application Preliminary Validation

- Validation vs. rejection of the loan request
- Assignment to (two parallel) next activities, if validated

Manager sv:
Application Validation
Workflow references

- Workflow Management Coalition: www.wfmc.org

Implementation status

- WebML is implemented in a commercial CASE tool called Webratio (www.webratio.com)
- WebRatio has a plugin architecture allowing developers to invent new content/operation units (Java2EE & Microsoft .NET)
- Both Web Service and workflow primitives are implemented as plugins to the core of WebML
- Online demo (use of real Web Services): http://www.webratio.com/WebServicesDemo/page1.do
Ongoing and future work

- Web Services:
  - Semantics of conversations
  - Exception declaration and handling
  - Full support to asynchronicity
- Workflow-Driven Hypertext
  - Automatic derivation of hypertext skeletons from WFCM schemas
  - Checking of correspondence between WF and hypertext models