WebML: Model-driven design of Web Applications

Massimo Tisi

Stefano Ceri, Piero Fraternali, Aldo Bongio, Marco Brambilla, Alessandro Bozzon
• **WebML: Web Modeling Language**
  - Structured approach for the design of Data-intensive Web applications
  - Patent in Europe and U.S.
• **WebRatio**
  ▪ A commercial tool based on WebML
  ▪ A university spin-off: WebModels

• Example: Acer-EMEA [www.acer-euro.com](http://www.acer-euro.com)
  ▪ 37 countries
  ▪ 22 multi-lingual applications
  ▪ >600 page templates
  ▪ >3500 queries
  ▪ from design to deployment in 8 weeks
Motivations

• Data-intensive Web site design and implementation are complex processes
  ▪ based on methodologies borrowed from different sectors
  ▪ involving different actors (DB, software eng., designers...)
• Complexity of modern Web applications (e.g., multi-device output)
• Always evolving
  ▪ standards
  ▪ best practices
  ▪ architectures

SOA  SW
RIA
...
WebML Purpose

• WebML provides a **structured approach** to the design of Data-intensive Web sites
  - navigational interface
  - browsing and management of data
• A set of **domain specific models** helps designers in high-quality Web sites production
• **Separation of concerns** is enforced
  - database design
  - application design
  - business logic development
  - presentation and style design
A rigorous modeling approach:

- Can reduce development efforts (cost and time)
- Allows a more structured development process
- Produces more usable and coherent final results
- Design models are self-documenting and always up-to-date projects

Immediate prototyping can be achieved
Requirements for Web modeling

• Expressiveness
  ▪ Real-life cases should be expressible
  ▪ Frequently used design patterns should be captured

• Ease of use
  ▪ Intuitive notation
  ▪ Clear semantics
  ▪ Consistency checks

• Implementability
  ▪ Efficient mapping to physical data structures
  ▪ Flexible code generation from behavioral specifications
Data Intensive Web Applications

- entities, relationships
- structure + derivation
- units, pages, links
- hypertext
- site views
- presentation
- styles

The WebML models
Simplified Entity-Relationship model

- Binary relationships between entities
- IS-A hierarchies
- Simple typed attributes in entities
- Derivation model can be applied for redundant data
• Redundant data can be easily specified using a WebML-OQL (Object Query Language).
• E.g.:
  ▪ Author.BooksNumber = count(self.Author2Book)
  ▪ BestSeller := Book where Book.Sales > 50,000
Goals

- Modelling at a high level:
  - the front-end of a dynamic Web application
  - the interactions with the back end business logic and data
- Using a simple visual notation
- Enabling automatic generation of dynamic page templates and business logic java classes
A **WebML unit** is the atomic information publishing element

It is a “view” defined upon a **container** of objects.

E.g.:
- All the instances of an **entity**
- Instances of an entity that meet a selection condition called **selector**
Hypertext: examples of Content Units

**DATAUNIT**

**DATAUNIT**

Data Unit

To publish information about a SINGLE object (e.g. Forum Message)

**INDEXUNIT**

**INDEXUNIT**

Index Unit

To publish a list of objects (e.g. Forum Messages)

**Forum Message**

Sender: XXX
Text: YYY
Timestamp:

**Index of Messages**

1. WebML?
2. WebRatio?
3. RIA?
### Meaning of Content Units

#### DATAUNIT

<table>
<thead>
<tr>
<th>Author</th>
</tr>
</thead>
<tbody>
<tr>
<td>first name: XXX</td>
</tr>
<tr>
<td>last name: YYY</td>
</tr>
<tr>
<td>photo:</td>
</tr>
</tbody>
</table>

#### INDEXUNIT

<table>
<thead>
<tr>
<th>Index of Authors</th>
</tr>
</thead>
<tbody>
<tr>
<td>• I. Asimov</td>
</tr>
<tr>
<td>• M. Twain</td>
</tr>
<tr>
<td>• C. Dickens</td>
</tr>
</tbody>
</table>

#### MULTIDATAUNIT

<table>
<thead>
<tr>
<th>All Authors</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

#### ENTRYUNIT

<table>
<thead>
<tr>
<th>Insert Your Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Fname</td>
</tr>
<tr>
<td>• Lname</td>
</tr>
</tbody>
</table>

#### SCROLLERUNIT

<table>
<thead>
<tr>
<th>Browse Authors</th>
</tr>
</thead>
<tbody>
<tr>
<td>5/12: go</td>
</tr>
<tr>
<td>1/12</td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

#### MULTICHOICE

<table>
<thead>
<tr>
<th>Choose Authors</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Asimov</td>
</tr>
<tr>
<td>• Twain</td>
</tr>
<tr>
<td>• Dickens</td>
</tr>
</tbody>
</table>

#### HIERARCHICAL

<table>
<thead>
<tr>
<th>Books &amp; Authors</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. I., robot</td>
</tr>
<tr>
<td>Asimov</td>
</tr>
<tr>
<td>2. Hard times</td>
</tr>
<tr>
<td>Dickens</td>
</tr>
</tbody>
</table>
Content units computation

- A unit may need some “context” to be computed
- Each unit exposes input and output parameters

```
IN  unitX  OUT
```

**entity**

\[ \text{[selector (par 1, .., parN)]} \]

- A content unit is not computed until its needed context is available
  - Parameters pre-defined for the unit +
  - Other parameters required by the selector of the unit
- A content unit is recomputed (and eventually redrawn) each time its context changes
- Output can be used to compute other unit(s)
• A Page is a structured container of units
  ▪ Possibly structured in and/or sub-pages
  ▪ Permits one to cluster related information for more efficient communication

  ▪ E.g.:

  ![Diagram](image.png)

  The index of messages and the selected message are shown together in the same page
• Models a generic operation

• Built-in operation units:
  ▪ Data manipulation
  ▪ Session context management
  ▪ User authentication
  ▪ ...

• The predefined WebML units can be enriched by adding custom external operations (e.g. SendMail, …)
Data Management Operations

CREATE

DELETE

MODIFY

CONNECT

DISCONNECT
An operation unit is computed each time an incoming link is activated
- Exception for transport links
- Some operation units can be activated also by other events
(Normal) links

Source unit → Target unit

Semantics of links:
- Allowing the user to move from one place to another
  - Rendering by means of anchors or submit buttons
- Transporting information from one place to another
  - Context propagation by parameters coupling
- Activating an operation

Outgoing links from operation units two labels:
- **OK** link if the operation completes correctly
- **KO** link if the operation fails
A transport link has a context that the source unit makes available to the target unit immediately after its computation, *without user intervention*.

The user *cannot* change the context and therefore the link is not rendered with an anchor.
An automatic link has both the behaviors of a normal link and a transport link:
- makes a context immediately available to the target unit
- Is rendered and can be selected by the users for subsequent activations
• E.g.: creation of a message
A Siteview is a set of pages that the user can experience as a whole Web site.

Different site views can be defined for different devices and different groups of users.

Thus, access control and multi-devices delivery is achieved.
Areas
Transactions
Master Pages
Alternatives
Global Parameters
Functionalities:

- Structure model design
- Data derivation (Derivation Wizard)
- Hypertext model design
- Consistency checks (warnings)
- Structure Mapping onto a datasource
- Units positioning in the pages
  - Grid for main content
  - Unlimited named locations
- Automatic web site generation with presentation styles
  - HTML + Custom Tags + CSS
- Compatibility with best selling tools for presentation editing
  - E.g., WebML extensions for Dreamweaver
Web Ratio: a short demo

WebRatio 4.3
Working in Offline mode
Copyright (C) 2001-2006 Web Models s.r.l. - All Rights Reserved
Protected by United States patent: 6,591,271
Patent pending in Europe
Ongoing work: Event Modeling

RIAs enforce decoupling of user interaction and browser requests

• Allow asynchronous polling (persistent connection technologies), non-interruptive application interaction

• Enable server-to-client communication (server PUSH)

• Make **event-driven** Web applications reality, e.g.:
  ▪ instant messaging, shared calendars, online auctions
Data Model Extensions

- application-specific event types are represented by adding new entities to the data model
- all event types extend the predefined *Event* entity
- Specific event types can have relationships with application domain entities

Composition/Navigation Model Extensions

- We extended the WebML hypertext model to support event notifications by means of two WebML operations:
  - *send event*: send an event notification to a (set of) recipient(s);
  - *receive event*: receive the notification and trigger a reaction;
Extension of Data Design to capture processes

• Based on the WfMC concepts
Extension of Hypertext Design to capture processes

Activity delimiters

Start Activity

End Activity

ActivityName

ActivityName

Process delimiters

Start Activity

End Activity

ActivityName

ActivityName

WF indexes

Assignements

Conditions

Current User

Requests

Assign

If

RefundingRequest
[ActivityType="Approval"]
[User=CurrentUser]

ReqID

Amount

Amount>1000$

RefundingRequest
[Activity="Approval"]
[User=CurrentUser]
[Case=CurrentCase]
Semantic Web Services

- Semantic Web Services (SWS) have a great potential
  - easy web service discovery
  - automatic web service integration
  - easy interoperability
  - ...

- Till now SWS are rarely used in practice
  - annotations are an extra cost

- Software Engineering (SE) tools and methodologies can push the use of SWS
  - model driven development techniques can be improved to include annotation and generate Semantic Web Services
The most complete solution presented at phase-II of SWS/Challenge.
• Integration with other modelling languages / models, tools (esp. UML, WFMC, BPEL, MDWEnet)
  ▪ Production of a (part of) WebML model from other diagrams
  ▪ Reverse transformation
  ▪ Manage correspondences and check consistency

• Current approach:
  ▪ Ecore metamodel
  ▪ ATL transformations
  ▪ ATL HOTs
• Code generation testing:
  ▪ Transformation for test set generation
  ▪ Coverage metrics

• Debugging
  ▪ Adding traceability to the code-generation transformation
  ▪ Adding a debugging environment to WebRatio

• Metrics
  ▪ Transformation for functional size metrics