WebML models and concepts

Marco Brambilla
Politecnico di Milano and Web Models Srl

The WebML models

- WebML: a conceptual language for high-level design of data-intensive web sites
- Models:
  - **Content**: data organization
  - **Derivation**: redundant data definition
  - **Composition**: definition of site pages as set of subpages and elementary publishing units
  - **Navigation**: definition of links between pages and between units
  - **Presentation**: positioning of the units in the page and definition of graphical appearance
Preview of WebML concepts

- Site = Content + Composition + Navigation + Presentation

Structure Model (1)

- Q: what are the objects published in the site and how they are related?
  - A:
    - Entity: an object type in the application domain
    - Attribute: scalar property of an entity
    - Relationship: A connection between entities
    - IS-A hierarchy: classification and grouping
  - Compatible with Entity-Relationship and UML class diagrams
Structure Model (2)

- Simplified Entity-Relationship model
  - Binary relationships between entities
  - IS-A hierarchies
  - Simple typed attributes in entities
  - Derivation model can be applied for redundant data

![Diagram of entity-relationships]

Relationship cardinality

- Relationship Book_Review
  - book2review minCard: 0 maxCard: N
  - review2book minCard: 1 maxCard: 1

- It reads: a book may have zero or more reviews, a review deals with one and only one book
Derivation Objectives

Derivation is a modelling phase that makes it possible:

- To augment the content of an entity by adding attributes, either imported or calculated from related objects.
- To define the population of entities or of relationships, based on some property of the involved objects.

WebML OCL

Derivation consist of writing expressions called “derivation queries”

Queries can be expressed using WebML OCL

Derivable concepts:
- Entities, relationships, attributes

Derivation queries can be automatically transformed into SQL views installed in the database
Derived entities

- Sub-Entities population in ISA hierarchies can be specified by means of OQL queries
- “A free slot is a slot having 0 reservations”
- WebML OQL: “SuperEntity where count(reservation)=0”

![Diagram showing ISA relationship between slot and freeSlot]

Derived attributes

- Attributes of an entity can be derived by associating to them a query.
- Four types of derived attributes:
  - Constant attribute: affiliation: “Politecnico di Milano”
  - Imported attributes: maritalName: Self.husband.lastname
  - Aggregate attributes: reservation#: count(Self.reservation)
  - Calculated attributes: lastPrice: Self.price*discount
- The Self keyword identifies the current entity in which the attribute is defined
Basic Content Units

Content:
- instances of an entity

Selector:
- set of conditions

Meaning of Content Units

Author
- first name: XXX
- last name: YYY
- photo:

Index of Authors
- S. Ceri
- P. Fraternali
- O. Versand

Books & Authors
- 1. Web Applicat.
- Ceri
- Fraternali
- Versand
- Systems
- Tannenbaum
**Pages**

- A **page** is a container of one or more pieces of information shown to the user at the same time.
- Nesting of pages is allowed: a page can have sub-pages.
- The user navigates a site made of pages.

**Non contextual links**

- A **non contextual link** is a link between pages.
- No context (information) is transported.

The user can browse from a page to another one via an anchor (e.g., `>>Book`).
**Home Page**

- The Home Page is the main page of a site
- It is the first page of the site that the user should see
- Each siteview must contain a page marked as "Home"

**Landmark pages**

- *Landmark pages*: globally visible pages. The user can jump to them from everywhere in the site view
- It is equivalent to a non contextual link implicitly defined from every other page in the site view to the landmark page
Areas

- An area is a set of logically homogeneous pages
- Examples of areas:
  - Sections of a portal: Sport, Music, Technology, ...
  - Elements of a data-management system: Products management, News management, ...
- Areas can be nested, so that sub-areas can be defined inside areas
- Each area should have a DEFAULT PAGE or a DEFAULT SUB-AREA, to give a meaning to landmark areas and non-contextual links pointing to areas

Site Views

- A siteview is 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
- Different site views can be published for different types of users and for different types of output devices
- Site views can be
  - Public: everyone can enter
  - Private: access control with password protection is enforced
Acme site view modeling

- Two site views on the same data model
  - Customer: public, for customers
  - Admin: private, for the administrators and content managers

Acme customer site view

- Purpose: define the customer experience of the site
- Structure:
  - Two main areas: products and combinations
  - Other landmark pages: home, stores
ACME page modeling

Q: what “main” pages do I need in the site?
- Home, Products, Combinations, Stores

Q: what other pages do I need?
- Product detail, Combination detail, ByPrice, BigImages...

ACME: HomePage
ACME product page

ACME combination page
Navigation Model: Links

- Semantics of a link:
  1. Moving from one place to another
  2. Transporting information from one place to another (navigation context)
  3. Activating a computation (side effect)

Composition: Pages

A Page is a structured container of units and links

- Possibly structured in and/or sub-pages
- Abstraction of screen, frame, card, deck...
- Permits one to cluster related information for more efficient communication

E.g.:

The index of authors and the selected author are shown together in the same page
Types of links

- Contextual links
  - Between units
  - Context transported

- Non-contextual links
  - Between pages
  - No context transported

Operation Unit

- Models a generic external operation, or a built-in content manipulation operation
- Input from one or more incoming links (at least one is declared as normal link, the others as transport links)
- Two kinds of output links
  - OK link if the operation completes correctly
  - KO link if the operation fails
- The predefined WebML units can be enriched by adding custom external operations (e.g. SendMail, ...
**Built-in Operations**

- WebML predefines a set of frequently used built-in operations to manage a site’s content.
- They are the traditional database operations: create, delete, modify, create relationship, delete relationship.
- Users do not need to define the behaviour and the implementation; they are provided off-the-shelf in the model.

---

**Built-in Operations**

- **CREATE**
  - Create Unit
  - Entity

- **DELETE**
  - Delete Unit
  - Entity

- **MODIFY**
  - Modify Unit
  - Entity

- **CONNECT**
  - Connect Unit
  - relationship

- **DISCONNECT**
  - Disconnect Unit
  - relationship
**In/out flows**

Create Unit

- value1 \(\rightarrow\) attribute1
- value2 \(\rightarrow\) attribute2

Modify Unit

- Value2 \(\rightarrow\) attribute1
- value1 \(\rightarrow\) attribute2

Delete Unit

- OIDs of the object(s) to delete

Example:

- Albums
  - All the way
  - Anthology
  - Born in the USA
  - Imagine
  - Let’s talk...
  - Sing-a-long

- Anthology
- Bridges to Babylon
- Imagine

- All the way
  - 1999
  - 1997

- Sing-a-long
  - 1998

**In/out flow: delete**

- OIDs of the object(s) to delete

Example:

- Albums
  - All the way
  - Anthology
  - Born in the USA
  - Imagine
  - Let’s talk...
  - Sing-a-long

- Anthology
- Bridges to Babylon
- Imagine

- All the way
  - 1999
  - 1997

- Sing-a-long
  - 1998
**In/out flow: connect**

- **Note:** if any of the input parameters is a set of OIDs, the operation is applied to each element of the set
- The disconnect operation is similar

OIDs of source object(s) to connect

OIDs of destination object(s) to connect

Identifiers of the objects to connect

Identifiers of the connected objects

**Example**

One of the most romantic and expressive artists of the nineties.

Select an artist to assign the review:
- **Beatles**
- **Dion**
- **Prince**

First name: **Celine**
Last name: **Dion**

Review:
- One of the most romantic and expressive artists of the nineties.
Siteviews

- 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
Using global information

- It is often necessary to:
  - Set a parameter value (e.g. the country or language preferred by a user)
  - Use this value globally in all the pages the site view, without carrying it explicitly along links

- Solution
  - Use global parameters stored in the session
  - Provide means for setting/getting their value

Context Parameters

- WebML Context Parameters allow to achieve this goal in a simple way
- Designer defines one or more Context parameters.
- A context parameter is defined by:
  - Name
  - ID
  - Duration (User session or Application)
  - Value type: can be either:
    - A Printable value (integer, string, ...)
    - An Entity (thus, the parameter can assume an OID value of that entity)
  - Starting value [optional]
### Set unit
- Unit that allows to SET the value of a parameter
- It is always placed outside a Page
- It has only an incoming link (carrying the value to be assigned to the parameter)
- It has not outgoing links

### Get unit
- Unit that allows to RETRIEVE the value of a parameter
- It is always placed inside a Page
- It has only an outgoing link (carrying the value retrieved from the parameter)
- It has not incoming links
- The retrieved value can be used in any compatible unit
In reserved Web sites commitment wants:
  • One or more public pages readable from anyone
  • A set of private page accessed only after login, which contains personal content and personal services
  • Personalization in terms of delivered pages (which pages user can access) and delivered content (which content user needs/can see)
WebML Development process

- Business Requirements
  - REQUIREMENTS SPECIFICATION
  - DATA DESIGN
  - HYPertext Design
  - Architecture Design
  - Testing & Evaluation
  - IMPLEMENTATION
  - Maintenance & Evolution

Web services architecture

<table>
<thead>
<tr>
<th>Service</th>
<th>Provider</th>
</tr>
</thead>
<tbody>
<tr>
<td>BPEL4WS</td>
<td>IBM, Microsoft, &amp; BEA</td>
</tr>
<tr>
<td>WCSI</td>
<td>Sun</td>
</tr>
<tr>
<td>BPML</td>
<td>BPMI.org</td>
</tr>
<tr>
<td>WSFL</td>
<td>IBM</td>
</tr>
<tr>
<td>XLANG</td>
<td>Microsoft</td>
</tr>
<tr>
<td>UDDI</td>
<td>UDDI.org</td>
</tr>
<tr>
<td>WSDL</td>
<td>W3C</td>
</tr>
<tr>
<td>SOAP</td>
<td>W3C</td>
</tr>
<tr>
<td>XML</td>
<td>W3C</td>
</tr>
<tr>
<td>HTTP</td>
<td>W3C</td>
</tr>
<tr>
<td>Service Integration</td>
<td></td>
</tr>
<tr>
<td>Service Discovery</td>
<td></td>
</tr>
<tr>
<td>Service Publication</td>
<td></td>
</tr>
<tr>
<td>Service Description</td>
<td></td>
</tr>
<tr>
<td>XML-based Messaging</td>
<td></td>
</tr>
<tr>
<td>Transport</td>
<td></td>
</tr>
</tbody>
</table>
**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 from WSDL)**

- Central component of a web service: operation = 1 or 2 msg
  - Request-response
  - Solicit-response
  - Notification
  - One-way

---

**Synchronous vs. asynchronous**

- Two-message operations may be used
  - **Synchronously**: no action is taken between 1\(^{st}\) and 2\(^{nd}\) 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.

WebML support for 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 Web services

WebML for supporting Web services

- One WebML operation for each class of Web service operations

- Marks for operations that start, resp. end conversations
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 with service calls
**Hypertext with async messaging**

Added new entity \(\text{Comment(OID,subject, text, answer)}\)

+ Relationship \(\text{Comment>User}\)

**WebML 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

BPMN* → WebML transformation

(5) Orchestration view: a controller component invokes the activities
The NEXT unit

- The Next unit encapsulates the process control logic
- It exploits the information stored in the Process Metadata
- It calculates the current process status and the enabled state transitions
- It needs the following input parameters:
  - caseID (the currently executed process instance ID)
  - activityInstanceID (the current activity instance ID)
  - conditionParameters (the values to evaluate the conditions)
The NEXT unit (cont.d)

4 execution modes:

- **Process start.** Instantiates and starts a case for the given ProcessID.
- **Process end.** Given the activityInstanceId of the last activity, sets the Case to “Ended”.
- **Calculate and execute.** Given the actInstID of the last concluded activity, enables the execution of the subsequent activities.
- **Execute.** Given an actInstID, sets its status to “Active” and launches its execution.

![Diagram of NEXT unit execution modes](image)
Next Unit

Design process
WebRatio 6 BPM

- WebRatio 6 BPM permits you to:
  - Model and Document your processes
  - Execute and Validate your processes by instantly prototyping
  - Developing, Improving and Refining your processes and create a complete Web application

WebRatio functionalities

- Content model design
- Business processes model design
- Data derivation (Derivation Wizard)
- Hypertext model design (siteviews)
- Consistency checks (warnings)
- Mapping onto a datasource
- Units positioning in the pages
- Automatic web site generation
- Integration of advanced features (BPM, Web services, AJAX)
Model transformations

1. WFEEditor
   - Process Modeling
   - Workflow design
   BPMN, Ajax, ...

2. Transformation Functions
   - Data (Database Filling)
   - Application (Project Creation)
   - Java, SQL, XML, ...

3. Next Unit
   - Process management
   - Activities Orchestration
   WebML, Java, Hibernate, ...

marco.brambilla@polimi.it
http://home.dei.polimi.it/mbrambil