ISSN: 2278 – 1323
International Journal of Advanced Research in Computer Engineering & Technology
Volume 1, Issue 3, May 2012

# Analyzing Web Modeling Existing Languages and Approaches to Model Web Application Design

**D.R. Ingle, Dr. B. B. Meshram**

*Abstract*— **Day to day life the establishment of web increased the importance for the rise in social networking sites and the web, and increase in advertisement of scripting languages that is commonly known. Extendibility its functionality allowed to be more in interactive in the application of web. There is need to model these applications. Hypertext languages are focused for structuring web sites for the existing web modeling languages. Technologies that are used in today's web application are introduced in this paper. Web modeling languages are introduced in this paper such as WebML, UWE, W2000 and OOWS, the capability and discussion and the comparision that are been described for the new modelling approaches. The technology that is introduced in this paper for feature modeling that actually consist ontology which provides specification for features modeling. The Semantic Web is based on ontology technology a knowledge representation framework at its core. Ontology technologies are highly useful to organise, personalise, and publish learning content and to discover, generate, and compose learning objects.**

***Index Terms***— **Feature modeling, models interactivity.Ontology, OWL, SWRL, web engineering.**

## I. Introduction

Web modeling that specific issue relates to design and development of large-scale Web applications. In particular, it focuses on the design notations and visual languages that can be used for the realization of robust, well-structured, usable and maintainable Web applications. Designing a data-intensive [1]. In the beginning of Web development, it was normal to accessed Web applications by creating something with no attention to the developmental stage. WebML enables the high-level description of a Web site under distinct orthogonal dimensions: its data content (structural model), the pages that compose it (composition model), the topology of links between pages (navigation model), the layout and graphic requirements for page rendering (presentation model), and the customization features for one-to-one content delivery (personalization model). All the concepts of WebML are associated with a graphic notation and a textual XML syntax. WebML specifications are independent of both the client-side language used for delivering the application to users, and of the server-side platform used to bind data to pages, but they can be effectively used to produce a site implementation in a specific technological setting. WebML guarantees a model-driven approach to Web site development, which is a key factor for defining a novel generation of CASE tools for the construction of complex sites, supporting advanced features like multi-device access, personalization, and evolution management [2]. Feature models (also known as feature diagrams) are treelike structured models used to capture differences and commonalities between software features, enabling the representation of software variability [3]. Technologies like AJAX support thin client applications that can take full advantage of the computer power of the clients. Ajax is a group of interrelated web development techniques used on the client-side to create asynchronous web applications. With Ajax, web applications can send data to, and retrieve data from, a server asynchronously (in the background) without interfering with the display and behaviour of the existing page.

The rest of the paper is organized as follows. Section 2 deals with Modeling concept Section 3 deals Web modelling language. Section 4 design process in WebML, Section 5structural model, Compositional model deals with section 6, section 7 deals with navigational model, features of WebML discuss in section 8, section 9 deals with discussion and section 10 gives the conclusion.

***Manuscript received May 10, 2012***.

***D.R. Ingle****, Department of Computer Engineering, Bharati Vidyapeeth College of Engineering, Navi Mumbai India, 09702777927.*

***Dr. B.B. Meshram,*** *Department of Computer Technology, Veer Jijamata Technical Institute, Mumbai, India, 9969381962.*

## II. Modeling Concepts

### *A. Ontology Framework*

To deal with the challenges mentioned in the introduction, we employ an ontology based approach for the representation of the knowledge contained in feature models. We chose the web ontology language (OWL) [4], to represent our ontology for a number of reasons. First, OWL is the standard (semantic web) language, which will allow making the ontology interoperable among different applications. It will allow easily exchanging different feature models between applications and users. Secondly, OWL has constructs that allows the modeling of both classes and individuals, along with constraints defined on them. This provides a seamless transition from the real-world view of the model to the ontological view of the model. Thirdly, OWL (DL) was designed to support DL reasoning on top of the ontology model. This enables using DL reasoners to infer knowledge and using rules implemented in SWRL [5]. Our goal is not to provide taxonomy for feature models but rather to provide a knowledge representation mechanism for feature models. We do so via providing an OWL representation for the knowledge contained in feature models, and representing all possible feature model semantics. Furthermore, we use SWRL rules [5] to insure the consistency of the feature model and to detect contradicting or conflicting knowledge in the model. This shows the main components of the feature modeling ontology framework. The feature model ontology defines the vocabulary to represent the knowledge within feature models. services based on a formal ontological specification in which the structural knowledge of manufacturing services is formalized using Ontology Web Language (OWL, formerly named DAML+OIL) [6],and the constraint knowledge between structural knowledge is formalized using Semantic Web Rule Language (SWRL) [7]. Heterogeneous manufacturing services are acquired from the Internet, semantically annotated using a manufacturing service ontology, and stored in local manufacturing service hubs.

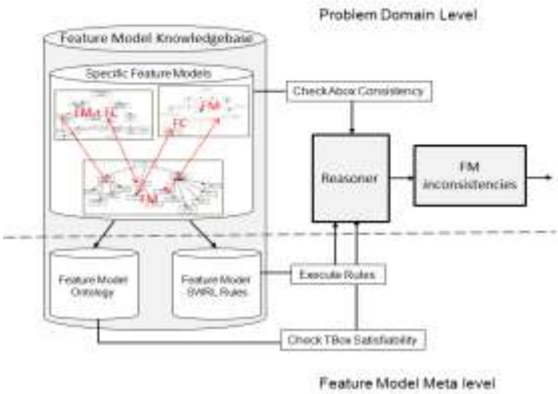

Fig.1. Model of Ontology Framework.

*B. Feature Model Ontology*

Ontology is a conceptualization of a domain. An ontology expresses knowledge of a certain domain of discourse in terms of classes, properties and restrictions. We have chosen the iterative engineering approach described in [8] to model our ontology. We model the feature model constructs as classes. To capture as much as possible all possible semantics represented by feature models, we have conducted a study of the state of the art of feature models to explore the similarities and differences in feature modeling methods. The basic source of our feature model concepts come from FODA representation. We have also added concepts from FORM [9] and FeatuRSEB [10]. We have added semantics for cardinality and feature attributes following there commendations of [11].One of the major benefits of having a feature model ontology is that it allows representing features, as well as both feature relations and features constraints in one model. Furthermore we provide feature constraints that allow integrating multiple distributed or segmented feature models. The integration aims at creating one feature model representing the system while detecting conflicts and contradictions. These constraints represent another form of relation between features, and link together different segments of features. We call these constraints feature to feature constraints (FTFC). It is the case that features interact together and thus influence the selection of other features within a valid composition. The integration of segmented models must insure the consistency and correctness of the overall model and thus the correctness of configuration decisions based on it.

*C. Feature Model Ontology Rules*

The need to introduce rules in our ontology is driven by the fact that we need a mechanism to detect the inconsistencies in a single feature model, as well as in a set of integrated feature models. By inconsistency we refer to the term defined by [12], "A state in which two or more overlapping elements of different software models make assertions about aspects of the system they describe which are not jointly satisfiable". Some contradictions cannot be represented in OWL DL, such as mutual exclusive properties. Therefore, this type of inconsistency cannot be detected by DL reasoners which try to identify sources of inconsistencies by searching for contradictory facts. Therefore, we need to define the rules that represent all these invalid states in order to assure that the underlying feature model is logically consistent. By logical consistency we mean that the facts in the ontology are not contradictory or violate rules governing the real case problem. We refer to this type of consistency as Variability Model Consistency. For the sake of inconsistency detection, we have added tour ontology a class named Inconsistency, which has one property called Problem. Problem has Inconsistency as both a domain and range. The situations that cause inconsistencies are defined by a set of SWRL rules. A rule has a body (antecedent) defining an inconsistent situation and a head (consequent) that marks the individuals causing this inconsistent situation. Marking is done by asserting them to have a problem relation between them. The rules assign inconsistent individuals to the Inconsistency class via the problem property. We capture two types of inconsistency problems. The first type of inconsistency emerges from using two properties that are mutually exclusive for the same features.

## III. WEB MODELING LANGUAGE

WebML (Web Modeling Language) is a conceptual model for Web application design . It is an ingredient of a broader Web development methodology, which is supported by a CASE tool, named Web Ratio. InWebML, the specification of a Web application consists of a data schema specified as an Entity-Relationship (E/R) model or as a UML class diagram, and of a hypertext schema describing the content of the site in terms of *site* views, areas and pages. A site view is a specific hypertext, designed for a particular class of users (Internet customers, administrators, stakeholders and so on) and it may exhibit a hierarchical organization, represented with the concept of *area*, defined as a recursive hypertext sub-module. Site views and areas are composed of *pages* that display elementary piece of contents. A content unit is a component for the publication of information inside a page and it corresponding to a parameterized query over the underlying E/R data model. WebMD offers six predefined units (data, index, multidata, scroller, multichoice index, and hierarchical index), which show one or more database instances. To compute its content, a unit may require the "cooperation" of other units, and the interaction of the user. Making two units interact requires connecting them with a *link*, represented as an oriented arc between a source and a destination unit. The aim of a link is twofold: permitting navigation (intra or inter-page), and enabling the parameter passing from the source to the destination unit. Besides having a visual representation, WebML primitives are also provided with an XML-based textual representation. This facilitates the automatic generation of final applications by means of CASE tools, as well as the automatic analysis of the application quality, as it will be explained in thefollowing sections.**WebML** enables designers to express the core features of a site at a high level, without committing detailed architectural details. WebML concepts are associated with an intuitive graphic representation, which can be easily supported by CASE tools and effectively communicated to the non-technical members of the site development team (e.g., with the graphic designers and the content producers). WebML also supports an XML syntax, which instead can be fed to software generators for automatically producing the implementation of a Web site. The specification of a site in WebML consists of four orthogonal perspective[13].

A. **Structural Model:** It expresses the data content of the site, in terms of the relevant entities and relationships. WebML does not propose yet another language for data modeling, but is compatible with classical notations like the E/R model, the ODMG object-oriented model, and UML class diagrams.

B. **Hypertext Model:** It describes one or more hypertexts that can be published in the site. Each different hypertext defines a so-called site view (see Figure 2). Site view descriptions in turn consist of two sub-models.

C. **Composition Model:** It specifies which pages compose the hypertext, and which content units make up a page.

D. **Navigation Model:** It expresses how pages and content units are linked to form the hypertext. Links are either non-contextual, when they connect semantically independent pages (e.g., the page of an artist to the home page of the site), or contextual, when the content of the destination unit of the link depends on the content of the source unit.

E. **Presentation Model:** It expresses the layout and graphic appearance of pages, independently of the output device and of the rendition language, by means of an abstract XML syntax. Presentation specifications are either page-specific or generic.

F. **Personalization Model:** users and user groups are explicitly modelled in the structure schema in the form of predefined entities called User and Group.

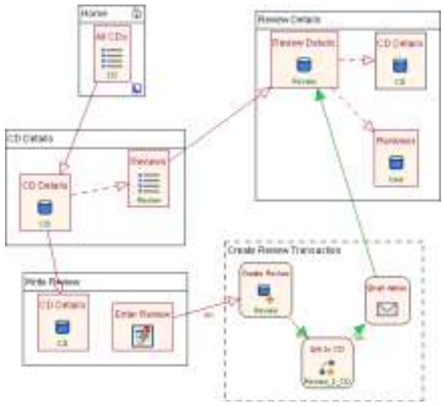

Fig 2: WebML application for a simple Hypertext Model

*A. Events*

Ceri et al. (2002) extends WebML to handle opera-tions and operation chains, which allows WebML to describe server-side events; this can be seen in the transaction area in Figure 1. Bozzon et al. (2006) further extends the model to allow for distinguishing between client and server operations and objects with the C (Client ) symbol, depicted in Figure 3. Events can only execute explicitly from a request, and cannot spawn additional operations in parallel, except through external web services (Manolescu et al. 2005). Exception support is also proposed in Brambilla et al. (2005) through a variety of event models, however this is not yet supported in WebRatio.

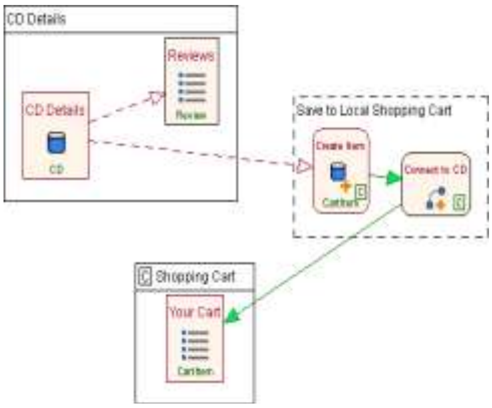


Fig. 3: Client/server extensions to WebML

*B. Browser Control*

A WebML model has little support for controlling the browser, with no support for cookies, scripting or identifying the user agent. It allows for scripted components, but the model cannot describe the scripts themselves.

*C. Lifecycles*
The closest thing we can compare lifecycles to is the use of operation chains and navigation links, but these suffer the same issues as event modelling in WebML. An example of this lifecycle support – to delete the users shopping cart at the end of their session – is presented in Figure 4; note that this operation chain only occurs when the user explicitly logs out14. WebML does not treat lifecycles as separate conceptual objects,and as such, implementation of lifecycles could add a lot of complexity and redundant information to the model.

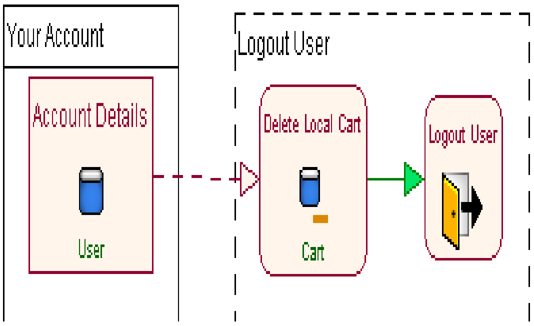


Fig. 4: Emulating lifecycle events in WebML

*D. Users and Security*

WebML has inbuilt support for users and groups, butonly as members of the structural model. Each user belongs to at least one group, and each group can onlyaccess one site view12. The model assumes only one user interacts with the model at any one time, so cannot support modelling the interaction between multiple users. The only permissions represented in the model is the ability for groups to view certain pages; additional security permissions must be checked explicitly through operation chains.[15]

*E .Databases*

The use of an ER model in the structural model of aWebML model abstracts the data from the database system; the hypertext models can abstractly access entities in the model. It has limited support for uploading files or handling multiple databases, but should be extensible.

*F. Messaging*

Messaging support is limited to the contextual information passed between hypertext model units; units can also access global parameters. Ceri et al. (2007) adds units to allow the model to access query parameters directly. WebML has excellent support for describing a wide range of web services (Manolescuet al. 2005), and also has good support for sending multi-part e-mails with attachments using its Power Mail unit.

*G.UI Modelling*

There is no formal model in WebML for modelling the user interface; the presentation model takes generated XML pages and transforms them using XSL stylesheets into HTML pages. The CASE tool WebRatio provides a presentation editor which uses a grid-based layout; other presentation attributes have to be composed manually, possibly assisted with external stylesheet design software[16]

*H. Standards*

WebML is platform-independent by design. The generatedpages can be in any language, limited only by the implementation of the software tool; WebRatio currently outputs to JSP pages. TheWebML model is generally proprietary, except for the use of ER diagrams for its structural model. It has no support for meta-modelling tools or standards, and the basic WebML model has only recently been implemented as a UML 2.0 profile (Moreno et al.2006). It lacks comprehensive support for MDA concepts.

*I. Verification*

WebML models have no formal verification process.The WebRatio tool provides a means to search the model for structural and content warnings, but the model itself cannot currently be integrated with any existing model checkers (such as Alloy: Jackson 2006). The model generation process itself has previously been tested with a custom internal prototype in Baresi, Fraternali, Tisi & Morasca (2005).

*J. Software Support*

WebML is actively supported by the development of the CASE tool WebRatio, which provides support for most of the features discussed in this paper. Some extensions, such as exception support, are not yet implemented. The newest version 5.0 extends the popular Eclipse framework (Eclipse Foundation 2007b) and is currently in beta, but does not appear to utilise the EMF framework (Gerber & Raymond 2003).

## IV. DESIGN PROCESS IN WEBML

Web application development is a multi-facet activity involving different players with different skills and goals. Therefore, separation of concerns is a key requirement for any Web modeling language. WebML addresses this issue and assumes a development process where different kinds of specialists play distinct roles: 1) the data expert designs the structural model; 2) the application architect designs pages and the navigation between them; 3) the style architect designs the presentation styles of pages; 4) the site administrator designs users and personalization options, including business rules.

A typical design process using WebML proceeds by iterating the following steps for each design cycle:

- Requirements Collection. Application requirements are gathered, which include the main objectives of the site, its target audience, examples of content, style guidelines, required personalization and constraints due to legacy data.

- Data Design. The data expert designs the structural model, possibly by reverse-engineering the exisiting logical schemas of legacy data sources.

- Hypertext Design ``in the large". The Web application architect defines the structure ``in the large" of the hypertext, by identifying pages and units, linking them, and mapping units to the main entities and relationships of the structure schema. In this way, he develops a "skeleton" site view, and then iteratively improves it. To support this phase, WebML-based tools must enable the production of fast prototypes to get immediate feedback on all design decisions.

- Hypertext Design ``in the small". The Web application architect concentrates next in the design ``in the small" of the hypertext, by considering each page and unit individually. At this stage, he may add non-contextual links between pages, consolidate the attributes that should be included within a unit, and introduce novel pages or units for special requirements (e.g., alternative index pages to locate objects, filters to search the desired information, and so on). During page design in the small, the Web application architect may discover that a page requires additional information, present in another concept semantically related to the one of the page currently being designed. Then, he may use the derivation language, to add ad hoc redundant data to the structure schema and include it in the proper units.

- Presentation Design. Once all pages are sufficiently stable, the Web style architect adds to each page a presentation style.

- User and Group Design. The Web administrator defines the features of user profiles, based on personalization requirements. Potential users and user groups are mapped to WebML users and groups, and possibly a different site view is created for each group. The design cycle is next iterated for each of the identified site views. ``Copy-and-paste" of already designed site view pages and links may greatly speed up the generation of other site views.

- Customization Design. The Web administrator identifies profile-driven data derivations and business rules.

Some of the above stages can be skipped in the case of development of a simple WEB application. In particular, defaults help at all stages the production of simplified solutions. At one extreme, it is possible to develop a default initial site view directly from the structural schema, skipping all of the above stages except the first one. A systematic approach and a specification of the Web application to be built in the form of visual models are recommended if we need to develop complex Web applications.(17).

## V. THE STRUCTURAL MODEL

The fundamental elements of WebML structure model are entities, which are containers of data elements, and relationships, which enable the semantic connection of entities. Entities have named attributes, with an associated type; properties with multiple occurrences can be organized by means of multi-valued components, which corresponds to the classical part-of relationship. Entities can be organized in generalization hierarchies. Relationships may be given cardinality constraints and role names. As an example, the following XML code represents the WebML specification of the structural schema illustrated in figure 1:

```
<DOMAIN id="SupportType" values="CD Tape Vinyl">;

<ENTITY id="Album">
  <ATTRIBUTE id="title" type="String"/>
  <ATTRIBUTE id="cover" type="Image"/>
  <ATTRIBUTE id="year" type="Integer"/>
      <COMPONENT id="Support" minCard="1"
maxCard="N">
    <ATTRIBUTE id="type" userType="SupportType"/>
    <ATTRIBUTE id="listPrice" type="Float"/>
.        <ATTRIBUTE id="discountPercentage"
type="Integer"/>
    <ATTRIBUTE id="currentPrice" type="Float"
          value="Self.listPrice *
              (1 - (Self.discountPercentage / 100))"/>
  </COMPONENT>
     <RELATIONSHIP id="Album2Artist" to="Artist"
inverse="ArtistToAlbum"
          minCard="1" maxCard="1"/>
     <RELATIONSHIP id="Album2Track to="Track"
inverse="Track2Album"
          minCard="1" maxCard="N"/>
</ENTITY>

<ENTITY id="Artist">
  <ATTRIBUTE id="firstName" type="String"/>
  <ATTRIBUTE id="lastName" type="String"/>
  <ATTRIBUTE id="birthDate" type="Date"/>
  <ATTRIBUTE id="birthPlace" type="String"/>
  <ATTRIBUTE id="photo" type="Image"/>
  <ATTRIBUTE id="biographicInfo" type="Text"/>
    <RELATIONSHIP id="Artist2Album" to="Album"
inverse="Album2Artist"
          minCard="1" maxCard="N"/>
    <RELATIONSHIP id="Artist2Review" to="Review"
inverse="Review2Artist"
          minCard="0" maxCard="N"/>
 </ENTITY>

<ENTITY id="Track">
   <ATTRIBUTE id="number" type="Integer"/>
   <ATTRIBUTE id="title" type="String"/>
   <ATTRIBUTE id="mpeg" type="URL"/>
   <ATTRIBUTE id="hqMpeg" type="URL"/>
     <RELATIONSHIP id="Track2Album" to="Album"
inverse="Album2Track"
          minCard="1" maxCard="1"/>
</ENTITY>

<ENTITY id="Review">
   <ATTRIBUTE id="text" type="Text"/>
   <ATTRIBUTE id="autho" type="String/>
     <RELATIONSHIP id="Review2Artist" to="Artist"
inverse="Artist2Review"
          minCard="1" maxCard="1"/>
</ENTITY>
```

The structural schema consists of four entities (Artist, Album, Review and Track) and three relationships (Artist2Album, Artist2Review, Album2track). Entity Album has a multi-valued property represented by the Support component, which specifies the various issues of the album on vinyl, CD, and tape. Note that each issue has a discounted price, whose value is computed by applying a discount percentage to the list price, by means of a derivation query. Derivation is briefly discussed.

## VI. COMPOSITION MODEL

The purpose of composition modeling is to define which nodes make up the hypertext contained in the Web site. More precisely, composition modeling specifies content units (units for short), i.e., the atomic information elements that may appear in the Web site, and pages, i.e., containers by means of which information is actually clustered for delivery to the user. In a concrete setting, e.g., an HTML or WML implementation of a WebML site, pages and units are mapped to suitable constructs in the delivery language, e.g., units may map to HTML files and pages to HTML frames organizing such files on the screen.

webML supports six types of unit to compose an hypertext:

- Data units: they show information about a single object, e.g., an instance of an entity or of a component.
- Multidata units: they show information about a set of objects, e.g., all the instances of an entity or all the sub-components of a composite object.
- Index units: they show a list of objects (entity or component instances), without presenting the detailed information of each object.
- Scroller units: they show commands for accessing the elements of an ordered set of objects (the first, last, previous, next, i-th).
- Filter units: they show edit fields for inputting values used for searching within a set of object(s) those ones that meet a condition.
- Direct units: they do not display information, but are used to denote the connection to a single object that is semantically related to another object.

Data and multidata units present the actual content of the objects they refer to, whereas the remaining types of units permit one to locate objects. Data units refer to a single object. Multidata, index, filter, and scroller refer to a set of objects. Therefore, they are collectively called container units.

## VII. NAVIGATION MODEL

Units and pages do not exist in isolation, but must be connected to form a hypertext structure. The purpose of navigation modeling is to specify the way in which the units and pages are linked to form a hypertext. To this purpose, WebML provides the notion of link. There are two variants of links:

A. Contextual links: they connect units in a way coherent to the semantics expressed by the structure schema of the application. A contextual link carries some information (called context) from the source unit to the destination unit. Context is used to determine the actual object or set of objects to be shown in the destination unit.

B. Non-contextual links: they connect pages in a totally free way, i.e., independently of the units they contain and of the semantic relations between the structural concepts included in those units. Syntactically, contextual and non-contextual links are denoted by element INFOLINK and

HYPERLINK, respectively nested within units and pages.

The following example demonstrates the use of contextual links, by showing a piece of hypertext composed of three linked units: a data unit showing an artist's data, an index unit showing albums, and a data unit showing album's data.

```
<DATAUNIT id="ArtistUnit" entity="Artist">
  <INCLUDEALL/>
  <INFOLINK id="link1" to="AlbumIndex"/>
</DATAUNIT>

<INDEXUNIT id="AlbumIndex" relation="Artist2Album>
  <DESCRIPTION key="title"/>
  <INFOLINK id="link2" to="AlbumUnit"/>
</INDEXUNIT>

<DATAUNIT id="AlbumUnit" entity="Album">
  <INCLUDEALL/>
</DATAUNIT>
```

The Artist Unit data unit, based on entity Artist, is linked via an INFOLINK to the index unit, which is based on the relationship ArtistToAlbum. Such index unit in turn is linked by a second INFOLINK to the AlbumUnit data unit, based on entity Album. The semantics of the above contextual links is that:

- Due to the first link (link1) a navigation anchor is added inside the artist's data unit by means of which the user can navigate to the index unit listing all the albums of a specific artist.
- Due to the second link (link2), a set of navigation anchors (one per each entry in the index) is added to the index unit by means of which the user can navigate to one of the listed albums.

Context information flows along both links. The identifier of the artist whose albums are to be listed in the index unit flows from the source to the destination of the former link (link1). The identifier of the selected album flows from the source to the destination of the second link (link2), to determine the object shown in the data unit.

Figure 5 shows the WebML graphic notation for representing the above contextual links and a possible rendition of such piece of hypertext in an HTML-based implementation. In this example, each unit is placed in a separate page, therefore three distinct HTML pages are generated. Grouping units within pages and establishing contextual links are two orthogonal design primitives, as demonstrated. Where units ArtistUnit and AlbumIndex are kept on the same page. By linking data units and container units it is possible to obtain a variety of navigation modes, as shown in figure where the index and album data unit are replaced by a multidata unit showing all albums of an artist together, both on the same page of the artist (case c), and in a separate page (case d)

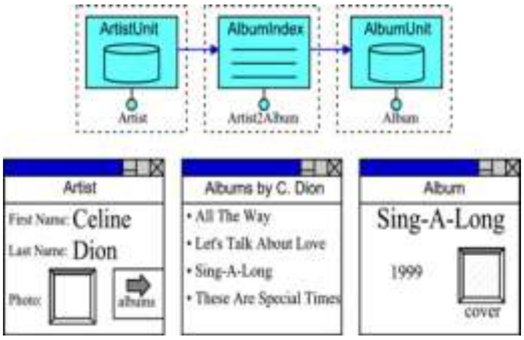


Figure 5 - Index-based navigation (index in a separate page), and a possible rendition in HTML

## VIII. FEATURES OF WEBML

We next present briefly the other features of WebML; for greater detail, we refer the reader to [7] and to the W3I3 project's documentation available at the site http://www.toriisoft.com.

### *A. Derivation*

Derivation is the process of adding redundant information to the structure schema, in order to augment its expressiveness. Derivation in WebML is expressed in a formal query language, which is a restricted version of OQL [5]; it is possible to derive entities, attributes, components, and relationships. An example of derivation is in the definition of the attribute currentPrice of the component Support shown in Section 2.

### *B. User Modelling*

In order to support personalization, WebML includes an explicit notion of group and user. Groups describe sets of users with common characteristics, whereas users denote individuals. Users always belong to at least one group, possibly the default one (called everyone). Users may belong to more than one group, but in this case they are required to provide a default group or to indicate one preferred group when accessing the site. Each user or group is described by means of specific properties (collectively called profile), modeled as special type of entities in the structure schema. As the normal entities, user and group profiles may be internally sub-structured into attributes and components, classified by means of inheritance, semantically related to other entities, and used for writing derivation queries. Typically, profiles include user- or group-specific data (e.g., the most frequently or recently visited objects, the list of the last purchases), whose content is progressively modified during the Web site evolution as result of user's actions.

### *C. Declarative and Procedural Personalization*

Personalization is the definition of content or presentation style based on user profile data. In WebML, units, pages, their presentation styles, and site views can be defined so to take user- or group-specific data into account. This can be done in two complementary ways:

- Declarative personalization: the designer defines derived concepts (e.g., entities, attributes, multi-valued components) whose definition depends on user-specific data. In this way, customization is specified declaratively; the system fills in the information relative to each user when computing the content of units.

- Procedural personalization: WebML includes an XML syntax for writing business rules that compute and store user-specific information. A business rule is a triple event-condition-action, which specifies the event to be monitored, the precondition to be checked when the event occurs, and the action to be taken when the condition is found true. Typical tasks performed by business rules are the assignment of users to user groups based on dynamically collected information (e.g., the purchase history, or the access device), the notification of messages to users upon the update of the information base (push technology), the logging of user actions into user-specific data structures (user tracking), and so on.

As an example of declarative personalization, the computation of an album's discounted price could be based on personalized discounts, associated with users or user groups. As an example of procedural personalization, a business rule could assign a customer to the "best buyer" group, based on his/her purchase history. WebML declarative and procedural personalization are discussed in greater details in .

### *D. Presentation model*

Presentation modeling is concerned with the actual look and feel of the pages identified by composition modeling. WebML pages are rendered according to a style sheet. A style sheet dictates the layout of pages and the content elements to be inserted into such layout, and is independent of the actual language used for page rendition. For better reusability, two categories of style sheets are provided: untyped style sheets (also called models) describe the page layout independently of its content, and thus can be applied regardless of the mapping of the page to a given concept; typed style sheets are specified at a finer granularity and thus apply only to pages describing specific concepts.

### *E. Updates and operations*

WebML is currently being extended to support a new type of pages for performing operations and updating the site content. Write access modeling is achieved by extending the current set of WebML concepts with the introduction of operation units, a novel type of unit whereby users can invoke operations on the site. An operation unit specifies the operation to be performed, and is linked to other units, which may show the alternative results of performing the operation (e.g., in case of success or failure). A set of generic update operations (such as: insert, delete, modify for entities, and drop, add for relationships) are predefined and need not be declared. Parameters needed to perform the operation either come from the context flowing to the operation unit via an incoming link, or are supplied by the user via forms.

Operation units generalize the notion of filter units, which can be regarded as a particular operation unit associated to a search operation over a set of objects. Thanks to the orthogonal nature of WebML, operation units can be freely combined with the other types of units to compose complex pages, mixing information to be presented to the user and interfaces to perform operations on the site. Operation units are currently being designed by looking at trolleys and online purchase procedures used in the most sophisticated e-commerce sites

TABLE 1. RESULT AND DISCUSSION OF WEBML

| Feature | WebML | UWE | W2000 | OOWS |
|---|---|---|---|---|
| Events | Ok | Poor | - | - |
| Browser control | Poor | - | - | - |
| Lifecycle | Poor | Good | Poor | - |
| Users | Good | Poor | Poor | Poor |
| Security | Ok | Ok | Poor | - |
| Data Support | Good | Ok | Poor | Poor |
| Platform Independeny | Excellent | Excellent | Good | Excellent |
| Standard support | Poor | Excellent | Excellent | Ok |
| Use of meta model | Poor | Excellent | Excellent | Poor |

## IX. DISCUSSION AND FUTURE WORK

In this paper we have presented a framework for representing, integrating and validating feature models by using OWL and SWRL. We have also applied it to a small example that demonstrates its use. Although OWL was initially proposed for the semantic web, its expressive power and formal semantics made it usable in many other different domains. We provided an ontology for feature models that captures a large category of the semantics in existing feature modeling techniques. We have also added feature-based integration semantics to our ontology enabling integration of distributed feature models. Furthermore we have formulated a list of SWRL rules that define conflicts or inconsistencies in the model as well as rules that infer information regarding variability. For our future work towards a complete framework to model and manage feature models, we aim to enrich the ontology by applying more examples.

From a usability point of view, it is not only important to detect inconsistencies but it is also very important to be able to explain to users the reason(s) for the inconsistency. Currently, Pellet provides some support for debugging possibilities, but in a very non-user-friendly format. Feature modeling is an accumulative process with many changes; therefore we plan to explore more the possibilities of reasoning with changing ontologies. In our case changes only happens in the Abox which should not complicate the process. As a second stage, an easy to use user interface to query the features model ontology is required to allow users to query about features and their relations within the ontology.Our final goal is to provide a tool that will allow collaborative analysis and modeling of feature models while pinpointing conflicts and inconsistencies on the spot allowing for accurate decisions and error free configurations. Furthermore our tool should allow for evolutionary development and act as a repository for feature models.

## X. CONCLUSION

The paper is basically the comparison of web modelling languages which describes the future and scope for the web application software system. In this paper we have presented a framework for representing, integrating and validating feature models by using OWL. Although OWL was initially proposed semantic web, its expressive power and formal semantics made it usable in many other different domains. We provided an ontology for feature models that captures a large category of the semantics in existing feature modeling techniques. We have also added feature-based integration semantics to our ontology enabling integration of distributed feature model. For our future work towards a complete framework to model and manage feature models, we aim to enrich the ontology by applying examples. Feature modeling is an accumulative process with many changes; therefore we plan to explore more the possibilities of reasoning with changing ontology's. Our final goal is to provide a tool that will allow collaborative analysis and modeling of feature models while pinpointing conflicts and inconsistencies on the spot allowing for accurate decisions and error free configurations. Furthermore our tool should allow for evolutionary development and act as a repository for feature models.

## REFERENCES

[1] Web Modeling, available on http://en.wikipedia.org/wiki.

[2] Stefano Ceri, Piero Fraternali, Aldo Bongio : Web Modeling Language (WebML) a modeling language for designing Web sites

[3] Bosch, J.: Design & Use of Software Architectures: Adopting and Evolving a Product-Line Approach, Addison-Wesley, 2000.

[4] OWL, available [online] on http://www.w3.org/TR/owl-features/

[5] Horrocks, I., Patel-Schneider, P. F., Boley, H., Tabet, S.,Grosof, B., Dean, M.: SWRL: A Semantic Web Rule Language Combining OWL and RuleML, available [online]

[6] D. L. McGuinness, R. Fikes, J. Hendler, and L. A. Stein, "DAML +OIL:An ontology language for the Semantic Web," IEEE Intell. Syst., vol. 17, no. 5, pp. 72–80, Sep./Oct. 2002.

[7] I. Horrocks, P. F. Patel-Schneider, S. Bechhofer, and D. Tsarkov, "OWL rules: A proposal and prototype implementation," Web Semant., vol. 3, no. 1, pp. 23–40, Jul. 2005.

[8] Noy, N. F., McGuinness, D. L.: Ontology Development 101:A Guide to Creating Your First Ontology. StanfordKnowledge Systems Laboratory Technical Report KSL-01-05 and Stanford Medical Informatics, Technical Report SMI-2001-0880 (2001)

[9] Kang, K., Kim, S., Lee, J., Kim, K., Shin, E., Huh, M.:FORM: A Feature-Oriented Reuse Method with Domain-Specific Reference Architectures. In: J. Annals of SoftwareEngineering. vol. 5, pp. 143-168 (1998)

[10] Griss, M., Favaro, J., d'Alessandro, M.: Integrating Feature Modeling with the RSEB. In: Fifth International Conference on Software Reuse, pages 76–85 (1998)

[11] Czarnecki, K., Kim, C. H. P.: Cardinality-Based Feature Modeling and Constraints: A Progress Report. In: OOPSLA'05 International Workshop on Software Factories(2005)

[12] Spanoudakis, G., Zisman, A.: Inconsistency management in software engineering: Survey and open research issues. In: Handbook of Software Engineering and KnowledgeEngineering, 1, pp. 329-380, 2001.
.
[13] WebML, available [online] on http://en.wikipedia.org/wiki/WebML

[14] Brambilla, M., Ceri, S., Fraternali, P. & Manolescu,I. (2006), 'Process Modeling in Web Applications', ACM Trans. Softw. Eng. Methodol. 15(4), 360–409.

[15] CARE Technologies (2007), 'OlivaNova: Products Overview'.URL: http://www.care-t.com/products/index.asp.

[16] Ceri, S., Daniel, F., Matera, M. & Facca, F. M. (2007), 'Model-driven development of contextawareWeb applications', ACM Trans. Inter. Tech.

[17] Gerti Kappel, Wieland Schwinger, Nora Koch" Modeling Web Applications," V2.March 2006.

**First Author:**

**Mr. D.R. Ingle** (ISTE LM'2004) is Associate Professor of Computer Engineerig Department at Bharati Vidyapeeth College of Engineerig, NaviMumbai, Maharastra state, India. He received bachelor degree, and Master degree in computer engineering. He has participated in more than 10 refresher courses to meet the needs of current technology. He has contributed more than 20 research papers at national, International Journals. He is life member of Indian Society of Technical Education.His area of interest are in Databases, intelligent Systems, and Web Engineering.

**Second Author:**

**Dr. B.B.Meshram** (CSI LM'95, IE '95) is Professor and head of Computer Technology Department at VJTI, Matunga, Mumbai, Maharastra state, India. He received bachelor degree, Master degree and doctoral degree in computer engineering. He has participated in more than 30 refresher courses to meet the needs of current technology. He has chair more than 15 AICTE STTP Programs and conferences. He has received the appreciation for lecture at Manchester and Cardip University, UK. He has contributed more than 200 research papers at national, International Journals. He is life member of computer society of India and Institute of Engineers. His current research interests are in Databases, data warehousing, data mining, intelligent Systems, Web Engineering and network security.