

# Modeling Ontology-Driven Personalization of Web Contents

Marco Brambilla, Christina Tziviskou

*Politecnico di Milano, Dipartimento di Elettronica ed Informazione,  
V. Ponzio 34/5, 20133 Milano, Italy  
{mbrambil, tzivisko}@elet.polimi.it*

## Abstract

*Personalization of contents accessed on the web can be improved by proper exploitation of user preferences and navigation behavior. Such profile information may be provided by metadata and should be managed through state-of-the-art web engineering methodologies and notations. In our proposal, we exploit existing solutions in the field of knowledge representation and web applications conceptual specification, and we provide: (i) a set of ontologies for representing both user profile and preferences, together with a good metadata design for describing them; (ii) a set of techniques for registering preferences explicitly declared by the user, as well as user navigation behavior automatically recorded; and (iii) some basic methods for delivering personalized content to the user based on metadata management. We base our method on the well known web modeling language called WebML, extended with a set of new primitives for exploiting personalization metadata.*

## 1. Introduction

In recent years, the Web has become the implementation platform of choice for complex B2C and B2B applications, whose goals comprise navigation of contents, business transactions, value-added services, and so on. Concepts like personalization and context-awareness are crucial for applications to provide services tailored on the users' needs and desires. In this sense, the web represents one of the most efficient means for the enactment of the current trends in marketing and communication strategies, which tend to move from a "one-size fits all" strategy to a "one-to-one" marketing approach.

Several commercial web sites heavily base their strategies on personalization: Amazon.com provides the user with a large amount of personalized contents that are shown besides the requested information. However, this kind of personalization is rather "poor",

in the sense that it can provide personalized contents only in terms of data stored in a database as explicitly connected to the user requests through predefined relations (i.e. books sharing the same author with books previously retrieved by the user). An automatic processing of the context resulting from the user profile, the requested information, and the published data could bring great advantage to the specification and implementation of navigational links built ad-hoc upon the context evaluation for the retrieval of personalized content.

As ontology languages and technologies become more and more available for practical use within web sites and web services, they may be adopted also for embedding machine-processable metadata describing the Web context and enabling its use in the personalization mechanisms. Manipulation of metadata describing the user preferences, his behavior, and the published information, enables the Web applications to integrate the context of Web contents within their decision mechanisms. Such mechanisms extract personalized contents that are more meaningful and effective for the user. However, for correct design and management of the new metadata-based aspects of personalization, state-of-the-art web engineering methodologies and notations must be adopted.

The goal of this paper is to describe an approach for applying software engineering methods, tools, and techniques to the problem of web contents personalization. Our method evaluates the context of Web contents for embedding in the pages ad-hoc navigational links enabling the connection of objects, when it is not known a priori (at design time) the relations connecting them. The context of the page contents is described with ontology schemas. We envision (i) an Entity-Relationship model for representing the metadata describing the Web context expressed as WSML ontologies; (ii) a set of high-level hypertext primitives for Web context management and query; and (iii) web-based content management mechanisms for delivering metadata-driven personalized information. More concretely, this work

consists in defining a conceptual model recording the user preferences in terms of explicitly declared wishes and implicitly recorded behaviors; and describing the publishing of metadata-driven personalized content within the hypertext pages. We base our method on WebML, extended with a set of new primitives for easily exploiting personalization metadata. In the current work, the evaluation of the Web context is achieved thru relational queries upon its relational representation. Future work investigates reasoning mechanisms to take place upon the ontology schema corresponding to the relational implementation of the context.

The paper is organized as follows: Section 2 briefly outlines the background; Section 3 describes the case study presented throughout the paper; Section 4 presents our approach to the metadata-driven personalization of web applications; Section 5 describes the expressive power of the query language for personalization; Section 6 reviews the related work; and finally Section 7 concludes.

## 2. Background

We build our solution upon an ontology of personalization metadata specified using the WSML ontology language [17]. The personalization mechanisms we propose for the hypertext design have been specified using the WebML language ([4], [15]), a high-level notation for data-, service- and process-centric Web applications, and its extensions for modeling ontologies [14]. Next subsections briefly summarize the basic aspects of these topics.

### 2.1. WSML

WSML is a representation language of Semantic Web services. It consists of four elements describing Web services: ontologies, Web services, goals, and mediators. The ontology is the main WSML element and defines the terminology used by all other elements. It defines the meaning of the content exchanged in Web services interactions, as well as the semantics of elements like goals, Web services, and mediators in the architecture implementing Semantic Web services.

*Ontologies* provide a formal and explicit specification of the application domain and of the terminology used by the other components. Optionally, they may be described by non-functional properties and may import existing ontologies. An ontology is composed of: (i) concepts, describing the ontology domain possibly organized in a hierarchy, (ii) relations, representing further connections among concepts, (iii) instances of concepts and relations, setting values to their attributes and parameters respectively, and (iv) axioms for further definition of concepts and relations through logical expressions.

### 2.2. WebML

WebML ([4], [15]) is a conceptual language for specifying dynamic large-scale Web applications developed on top of database content. The WebML specification of a Web application consists of a *data schema*, describing application data, and of one or more *hypertexts*, expressing the Web interface used to publish this data. The WebML data model is the standard Entity-Relationship (E-R) model. Upon the same data model, it is possible to define different hypertexts (e.g., for different types of users or publishing devices), called *site views*. A site view is a graph of *pages*, allowing users from the corresponding

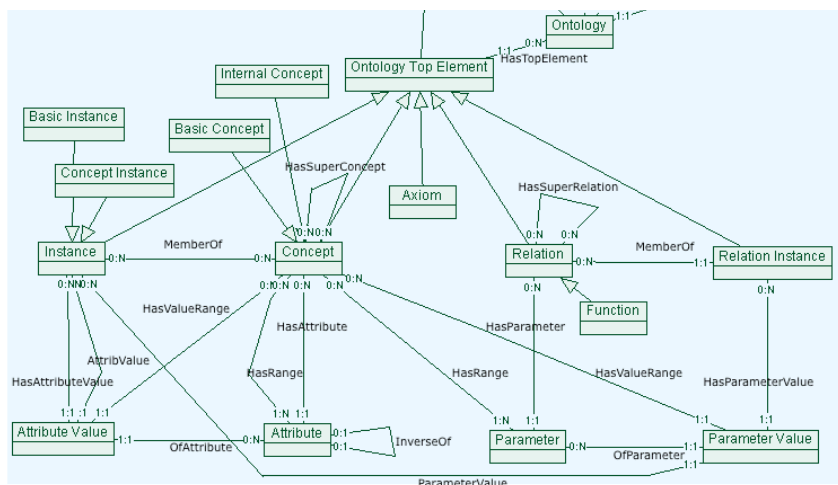
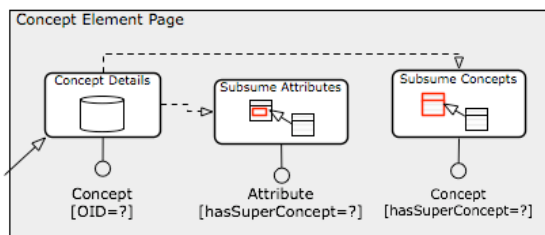


Figure 1. E-R metamodel describing the ontology concepts.

group to perform their specific activities. Pages consist of connected *units*, representing at a conceptual level atomic pieces of homogeneous information to be published: the content that a unit displays is extracted from an entity, and restricted by means of a set of conditions called *selector*.

Units within a Web site are often related to each other through *links* carrying data from a unit to another, to allow the navigation and computation of the hypertext. WebML allows specifying also update *operations* on the data underlying the site (e.g., the creation/deletion of instances of an entity) or operations performing other actions (e.g. invoking or publishing Web services [11], supporting business processes, and so on). The methodology and the architecture of the language are detailed in [4].

In [14] the WebML data design has been used for defining the data model describing the WSML Ontology elements: *Concept*, *Relation*, *Function*, *Axiom*, *Instance*, and *Relation Instance*. A detailed description of the data structure is available in [13]. Figure 1 shows the ontology metamodel represented by a E-R diagram. The main entity *Concept* contains defined concepts, possibly organized in a hierarchy with properties denoted by the entity *Attribute*. The actual instantiation of a concept and of its attribute values is depicted in the entities *Instance* and *Attribute Value* respectively.



**Figure 2. Concept Page WebML hypertext model**

The WebML language has also been extended with primitives for accessing the information in the relational representation of WSML ontologies. These primitives provide the designer with a simplified high-level access to the ontological content [13]. The new primitives are: *Subsume* and *Plugin units* for retrieving the more generic/specific concepts/relations of a given one respectively; *Exact unit* for retrieving the more specific concept/relation among two or more; *Subsume attributes/parameters unit* for retrieving the inherited attributes of a concept; *Export/Import Ontology units* for elaborating the internal relational representation of an ontology to/from its WSML format. Figure 2 presents the WebML fragment that implements the hypertext for browsing an ontology concept. In the *Concept Element Page*, the user receives the concept identifier, and visualizes the inherited attributes

(*Subsume Attributes* unit) and its super concepts (*Subsume Concepts* component).

WebML is supported by a CASE tool called WebRatio [16] that provides visual design facilities, error checking, and automatic code generation.

### 3. Case study

We exemplify the personalization techniques presented in the article with the scenario of a portal for educational organizations.

The Web site of the portal provides detailed presentation of academic institutes and of their activities. The site structure is composed of three interconnected areas. The first area describes the research carried out in the institutes, and the people involved in it. A second area contains the list of courses, detailed pages about the teaching staff and the educational material, and the last one allows users to browse information about employees (academic and administrative staff) and external collaborators to the institute, and gives access to their personal pages.

We focus on the interactions of the user with the pages relative to a professor of a faculty. The user may reach the professor page through various paths based on the search and select capabilities offered by the portal. Once he reaches the page of interest, he may choose to navigate the projects and the events in which the professor participates, explore the courses he teaches, view his collaborators, and search in his publications. In the above scenario, we present: (i) extraction mechanisms of the user preferences, (ii) querying of user preferences metadata for extracting personalized information, and (iii) model-driven techniques for enriching the pages with such personalized data.

#### 3.1. Academic institution ontology

The metadata describing the domain of the educational portal has been implemented as a WSML ontology built with the OntoWebML editor ([14], [13]), a Web editor for creating, deploying and browsing WSML ontologies, developed using WebML and its extensions for the WSML Ontology management. Our decision of adopting ontologies for representing the metadata that describes Web data is based upon two considerations. First, ontologies are machine-processable enabling the integration of such metadata in the applications logic. Second, ontologies are available on the Web, and hence, they enable easy-to-exchange and easy-to-share user profiles to become understandable by an arbitrary number of Web applications.

The ontology specifies the main concepts and relations (together with their characteristics) that we may find in the portal, hierarchically structured. Part of its WSML syntax is presented in Figure 3, while the full definition is available in Appendix A. The domain data instantiates concepts and relations of the ontology, and gives values to their attributes.

```

namespace
  targetNamespace:
    <<http://dbgroup.polimi.it/WSML/edu#>>
    xsd:<<http://www.w3.org/2001/XMLSchema#>>
ontology
  <<http://dbgroup.polimi.it/WSML/edu.wsml>>
concept Publication
  author impliesType Person
  publicationDate ofType (1) xsd:date
  organization impliesType EducOrganization
  publisher impliesType (1) Organization
  subject ofType xsd:string
concept Thesis subConceptOf Publication
concept Periodical subConceptOf Publication
concept Book subConceptOf Publication
concept Proceedings subConceptOf Publication
relation publicationInResearch
  p ofType (1) Publication
  a ofType Research

```

**Figure 3. Part of the domain ontology**

The above ontology describes one main concept called *Publication*, encapsulating other concepts as it is revealed from the hierarchy: *Scientific Article* (book article, conference paper, journal article, and workshop paper), *Periodical* (journal and magazine), *Book*, *Proceeding*, and *Thesis*. Main attributes are *author*, *publicationDate*, *organization*, and *publisher* that correlate the publication to other concepts. The rest of the ontology, shown in Appendix A, describes the academic institutions and the organization of the personnel: the *Work* concept is the general entity that regroups the more specific activities depicted from the concepts *Course*, *Research*, and *Conference*. Another main concept is the *Education Organization* divided into: *Department*, *Institute*, *Research Group*, *School*, and *University*. Characteristics of the categories like *name* and *address* are incorporated as attributes in the Education Organization concept and are inherited by the subconcepts. New ones are also defined like *topic* for the Research concept, and *specificCourses* for the Course concept. The last main concept is the *Person*. It is the more general concept for the *Student* and the *Employee* concepts. The employee is divided into the *Administrative Staff* the *Faculty Member* concepts. Faculty members may be professors (*Professor* concept divided into assistants, and associate, full and visiting professors), or lecturers (*Lecturer* concept).

Relations express interdependencies among concepts not depicted as attributes. They may be created explicitly for existing objects (e.g. the *advisor* relation relates the student Gianni to the professor Rossi), or may be stated as a consequence to an event

(e.g. the *researchOfInterest* connects persons to the research topics of their publications resulting from their specification as *authors*).

## 4. Personalization

Our approach to personalization consists of:

- (a) a conceptual model for representing the user profile described in terms of ontological terms;
- (b) a conceptual model for recording the explicitly declared preferences of the user in his profile;
- (c) a conceptual model for recording implicitly behaviors and navigation paths followed by the user, described in terms of browsed ontological concepts and/or instances in the user's profile;
- (d) a conceptual model for describing the publishing of metadata-driven personalized content within the hypertext pages, according to the information gathered from the previously specified models.

Therefore, our approach is based on the iterative process of monitoring the user navigation, collecting its requests, storing them as preferences in an user profile ontology, and querying the metadata expressed in the ontologies for delivering personalized content. The use of ontologies for representing both data and its metadata facilitates the tasks of correlating the domain data and the user preferences through management of their metadata. The approach allows to monitor the user interests, and to exploit connections of such interests to the domain data upon evaluation of their metadata for embedding in the Web pages personalized information.

To ease the WebML specification, we define a set of macro units that implement the typical queries needed for implementing the personalization behaviors. For lack of space, the paper cannot include the definition of all the macros. As an example, Figure 4 shows the implementation of a macro unit that allows to extract the user information given his credentials. It is implemented as a double chain of queries retrieving the user with a given username and password. The full list of macros used can be found on Appendix B.

This section addresses the problem of personalization by exemplifying our approach on the use case: we deliver recommended publications and detailed people information related to the research areas of interest to the user. Managing personalized data for users requires the storage and retrieval of user preferences about research areas, and the actions to be taken are: identify the users in the system; collect users' preferences, represented in the system by means of the User Profile ontology; manage metadata for querying the user preferences and the domain data and extract personalized data.

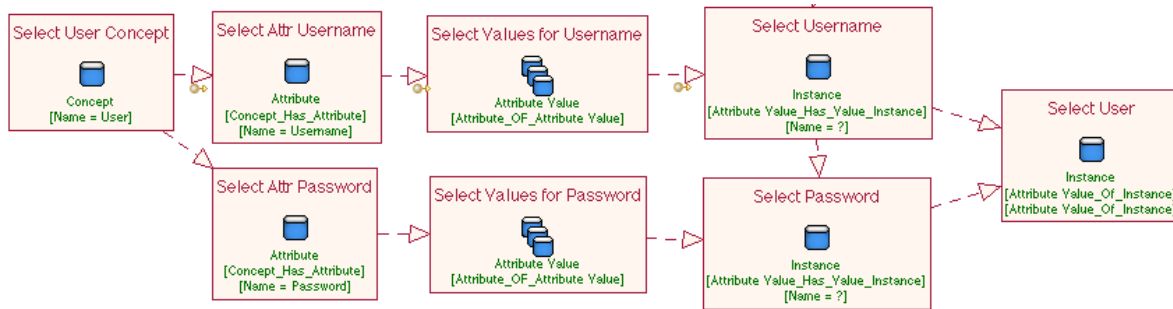


Figure 4. WebML modeling of the Ontological Select User macro

We identify two kinds of users of the portal: registered and unregistered users. In the next subsection we describe how user profiles can be managed within the hypertext.

#### 4.1. User modeling

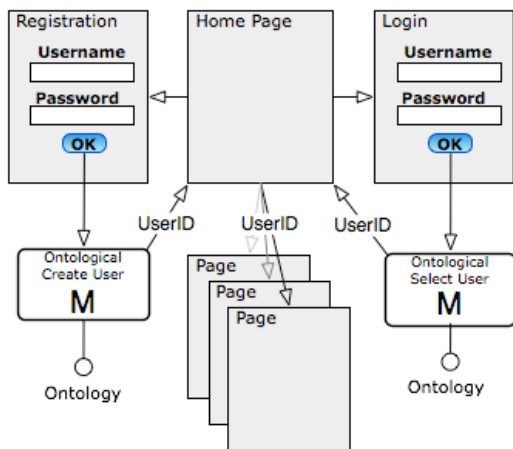


Figure 5. Registered user modeling

Figure 5 represents a piece of the portal modeled with the WebML language: *registered users* enter the Web site through a login page in which they submit their credentials. The *Ontological Select User* macro unit extracts the user ID from the user ontology in the knowledge base, according to the submitted credentials (Figure 4 gives a hint on the implementation of the macro unit). Subsequent requests may exploit the retrieved user ID for (1) generating the navigation context for the user, (2) updating its profile, and (3) presenting personalized data. The registration for this user category takes place through appropriate Web forms. Once the user submits the registration data, the *Ontological Create User* macro unit creates a new instance in the user profile ontology. Further logins of the same user will enrich his profile with more accurate behavior description and preferences.

On the other hand, unregistered users may be dealt with by creating a profile that remains valid only for the current navigation session. Proper WebML primitives can be exploited for building user sessions. As shown in Figure 6, the Get Session primitive in the Home Page retrieves the global parameter Session stored at server side. Navigating the Home Page, a check is performed on that parameter. If the parameter is not set, a new user instance is created in the user profile ontology, the Session parameter is set, and the new user ID is transferred along subsequent requests. Unlike the previous case, the user profile cannot be preserved for several sessions, but it is valid only while the user keeps browsing in the current session.

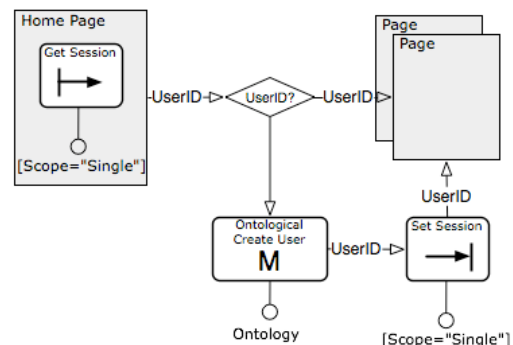


Figure 6. Unregistered user modeling: single session

If multi-session profile storage is needed for non-registered users, a session identifier must be kept at the client side and retrieved whenever the user navigates to the portal Web site. Technically, this can be done with several mechanisms, such as cookies. We have extended the WebML Get and Set primitives of Figure 6, for retrieving and setting session identifiers at the client side by means of the property Scope: if set to the value “Multi”, the Get Session primitive checks for the existence of a session cookie on the client; otherwise, the value is retrieved only from the current session. In Figure 6, by following the link exiting the Home Page, a control is done on that parameter. If the cookie is not found, a new user is created in the knowledge base, the

Session cookie gets set, and the user ID is transferred along subsequent user requests. If the cookie is found, the user session is retrieved allowing the retrieval of the user profile from the knowledge base. This enables the possibility of preserving and updating the user profile according to the Profiling Mechanisms (see Section 4.2), which in turn opens the way to generate and present personalized data.

## 4.2. Profiling mechanisms

User profiling is the process of acquiring data about the user interests that may be used later for generating personalized data. The representation of the gathered data is specified by means of the *User Profile Ontology* (Section 4.2.1), and is correlated with the *Domain Ontology* in order to infer new data for the user. We identify two sources for collecting the user profile information: (i) the knowledge about the user behavior is originated from its navigation on the Web pages; (ii) the user explicit preferences on domain objects is collected by appropriate forms where the user can specify his interests in the concepts, relations and instances of the ontology.

**4.2.1. User profile ontology.** We express the user profile with the ontology *user* (Figure 7). It is a simplified ontology that serves as the representation format for the user preferences and requests. The concept *User* specifies the characteristics of the Web contents consumer. Registered users have *username* and *password* attributes, whilst unregistered users have *session* context attribute. The values of the former are provided in the registration phase, and of the latter in the session initialization. The *userID* is a common attribute to both user categories. It is initialized by the system during the creation of a new user, and it is available throughout the whole user navigation.

```

namespace
targetNameSpace:
<<http://dbgroup.polimi.it/WSML/user#>>
xsd:<<http://www.w3.org/2001/XMLSchema#>>
edu:<<http://dbgroup.polimi.it/WSML/edu#>>
ontology<<http://dbgroup.polimi.it/WSML/up.wsml>>
concept User
  username ofType (0 1) xsd:string
  password ofType (0 1) xsd:string
  session ofType (0 1) xsd:string
  userID ofType (1) xsd:integer
relation requests
  u ofType (1) User
  w ofType {edu:Person,
edu:Organization,edu:Publication,edu:Work}
  t ofType xsd:datetime
relation prefers
  u ofType (1) User
  w ofType {edu:Person,
edu:Organization,edu:Publication,edu:Work}
  t ofType xsd:datetime

```

Figure 7. User Profile ontology

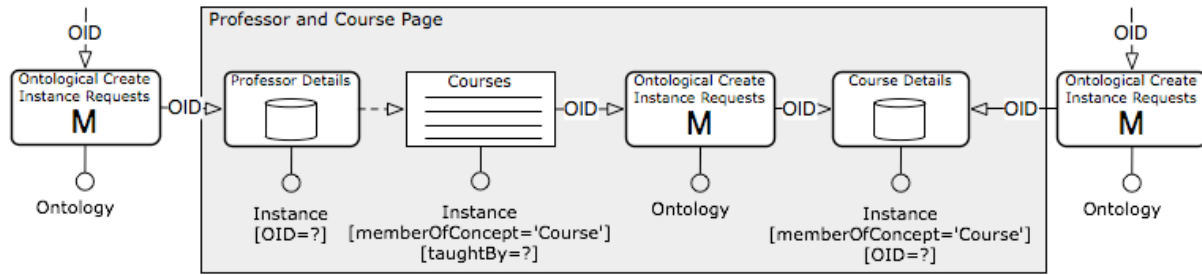
The relation *prefers* expresses the objects of interest for the user in the application domain ontology. The first parameter identifies the user and the second the concept(s) of the preferred objects. The second parameter is defined as the set of concepts that can be selected explicitly by the user. The third parameter contains the recording date-time of the preference. Instances of that relation are created whenever the user makes explicit statements regarding his preferences, according to Section 4.2.2. The relation *requests* expresses the objects of the domain ontology requested by the user. The first parameter identifies the user, and the second one the concept(s) of the requested objects. The second parameter is defined on the concepts we want to monitor during the user navigation. The third parameter marks the date-time of the request. Instances of that relation are created whenever the user requests an URL that is related (or annotated) with an instance of a concept we want to monitor, according to the Implicit Profiling Mechanism in Section 4.2.3.

The two relations express different views on the user profile. The former is built upon statements made by the user, and the latter consists of records of the user navigation. Management of this information allows exploiting the user contextual profile for building personalized data to be delivered to him.

**4.2.2. Implicit profiling.** Monitoring the user navigation is the common source for gathering the user interests. A high number of requests for a specific object indicates a high interest of the user for the object. In order to register into the knowledge base the user requests about objects, we must correlate visited pages with ontological objects.

For every user access to a page, we record in the knowledge base the significant objects contained in the page. For this reason, we enrich the WebML links that interconnect pages or units within pages with an operation chain: an *Ontological Create Instance Requests* macro unit receives the objects presented in the page that we want to monitor and creates a *requests* relation for each of them. In case the page publishes several instances of some concept(s), we may choose to select all the objects that instantiate the concept(s) and create a Requests relation for each of them through an *Ontological Create Concept Requests*.

Figure 8 demonstrates the mechanisms for recording the user requests in WebML. The user may reach the Professor/Course Page by alternative links that transfer the OID of the object to be displayed. If the user requests information about a professor (e.g., Mario Rossi), he visits the page by following the link entering the Professor Details data unit (on the left).



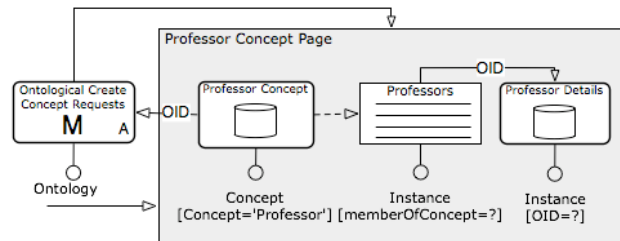
**Figure 8. Implicit profiling thru contextual links**

In this case, we register the request of the user for the specified Professor; the *Ontological Create Instance Requests* macro unit on the link creates a new instance of the Requests relation for the professor object whose OID is transferred on the link. The page displays the detail data for Mario Rossi and the list of all the courses he teaches.

If the user requests information about a course (e.g., the Database course), he visits the page by following the link entering the Course Details data unit (on the right). The *Ontological Create Instance Requests* macro unit on the link creates a new Requests relation instance for the course currently selected by the user. In this case, the page contains only data on the specific course. No context is available for the other units, which therefore do not display any content.

Contextual links between units in the same page act in a similar way. They carry the OID of the object to be displayed, and thus, an *Ontological Create Instance Requests* macro unit can be integrated at wish into them in order to create a new instance of the Requests relation for the object. In Figure 8, when the user selects a specific course from the courses list for the current professor, an *Ontological Create Instance Requests* macro unit registers the new request in the knowledge base.

Notice that the decision about which concepts are of interest for a particular navigation is taken at design time. It is the responsibility of the designer of the application to decide when to register new requests into the knowledge base, depending on the significance of the concepts and of the navigation steps.

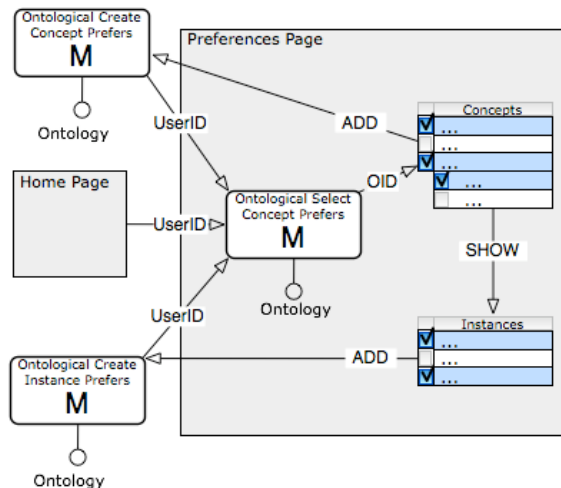


**Figure 9. Implicit profiling on non-contextual links**

The user may also be transferred to a page by activating a non-contextual link, i.e., a link that does not carry context to units and therefore points to a page and not to a unit. In such situations, (an extended version of) automatic links in WebML are used for automatically firing the operation chain that will register the user request in the knowledge base. In Figure 9, the user navigates the link entering the page for the Professor concept. The attributes of the concept are shown in the Concept data unit, and a list of all the professor instances are retrieved through the index unit. By selecting a professor in the index unit, its specific details are displayed in the Professor data unit. When entering the page, the automatic link (marked with "A") is activated and the *Ontological Create Concept Requests* macro unit creates a new instance of the *requests* relation for every instance of the Professor concept.

**4.2.3. Explicit profiling.** Knowledge about the user preferences may be given explicitly by the user through appropriate Web forms that allow him to specify his interests. The user may choose concepts of the domain ontology or specific instances of these concepts. However, a problem arises about the association of preferred concepts to user instances. Due to a limitation in the expressiveness of the WSML language, concepts and instances cannot be related through relations (except for the *Membership* relationship). Therefore, it is not possible to correlate in the knowledge base the current user with a concept. To overcome this issue, we correlate the user with all the instances of the concept he declares as preferred.

For the explicit preference selection, in order to handle realistic scenarios in which we may have hundreds of concepts in the domain ontology, we add the non-functional property *Preferable* that states if the concept may be selected as an explicit preference by the user. Correlating a concept to the above property implies that the concept will appear in the Web forms for collecting the user preferences.



**Figure 10. Explicit profiling of user preferences**

Figure 10 shows a WebML model that demonstrates the mechanism for gathering the user preferences. The user from the Home Page may reach his Preferences page (the user ID is transferred along the request). For reasons of simplicity, we do not repeat the user modeling hypertext in the figures that follow. The page contains two multi-choice index units allowing the selection of several items in the lists by means of checkbox mechanisms. The first one shows in a tree structure the ontology concepts marked by the non-functional property *Preferable*.

The *Ontological Select Concept Prefers* macro retrieves the previously defined preferences about concepts for the current user, and allows the appropriate concepts in the Concepts multi-choice unit to be shown as checked when the page is displayed (therefore, the user can identify his previous selections). Selecting concepts from that list and activating the ADD link causes the firing of the macro *Ontological Create Concept Prefers*. The macro first retrieves all the instances of the selected concepts, and then creates a *Prefers* relation instance connecting every retrieved object to the current user. Finally, the user is redirected back to the Preferences Page, in which his preferences are retrieved and displayed again in the multi-choice unit, and he may submit other preferences.

The second multi-choice index unit allows the user to specify his preferences about specific instances. The user interaction allows selecting a concept (through the SHOW link) and display all the objects that instantiate the selected concept in the Instances multi-choice. Once again, by checking instances in the list and submitting a new ADD request, the user triggers the macro *Ontological Create Instance Prefers*. The macro creates a *prefers* relation instance connecting every selected instance to the current user. When the user

enters the page again, the system retrieves the newly specified user preferences.

### 4.3. Personalized data delivery

In this section, we present how to deliver personalized Web content according to the knowledge gathered about the user. The extracted personalized information may be published into Web pages and may be browsed by the user as any other Web content; user navigation upon it can contribute to build the user profile as any other content, according to the mechanisms presented in Section 4.2. In the following, we exemplify the personalization extraction mechanisms with WebML macro units that formulate complex queries on the user profile.

Generally speaking, any metadata transformation can be applied to the profile data to extract the best matching personalized data. For sake of simplicity, we present a basic level of metadata management that can be extended at wish. Our approach provides that in a page showing an object (i.e., a concept instance), the delivered personalized contents can include:

- the objects that are related (through any semantic relation or attribute of the WSML language) to the current object;
- the objects that are registered as preferred by the current user. For this purpose, we will consider both explicit preferences and implicit preferences (records of page requests related to the object). For implicit preferences, the number of requests is considered too. Then, priority ranking of personalized content is computed by weighting the different contributions from the explicit preferences and the implicit user requests. The weights are defined according to empirical evaluations over real experiments and to the expected results of the users.

**4.3.1. Instances-based personalization.** Figure 11 exemplifies on a concrete case the abovementioned approach. The user enters the Course page that shows a selected instance of course. Let's suppose that (i) the user is looking now at the Database course identified in the knowledge base with the value 187, (ii) he has visited the page of the professor Mario Rossi four times in the past, and (iii) he has explicitly declared a preference about the Information Systems Course. The *Ontological Select Instances* macro unit performs a query extracting all the instances related to the current Web content. The actual result of the query is extracted by means of a two-phase process.



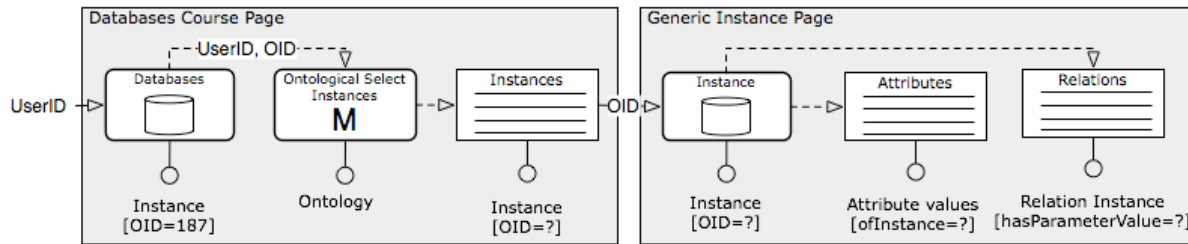


Figure 11. Personalization of instances

In the first phase, we retrieve all the instances related to the Database Course through any attributes and relations. If we consider a sample set of instances as represented in Figure 12, we extract:

- the professors Mario Rossi and Mario Bianchi related to the Database Course with the *teacherOf* relation instance;
- the courses Information Systems and Small Databases appearing as values for the attribute *specificCourses* for the course Database;
- the research DBResearch correlated to the Database Course with the *researchOfCourse* relation instance.

In the second phase, we discard the elements that do not appear in the user profile. In the example, we discard the instances Mario Bianchi, Small Databases and DBResearch from the previous results because they do not appear in the user profile. Indeed, they are not marked as preferences for the user, and they have never been browsed in the past.

The final results are ordered based on a weighting algorithm. We give a weight of 1 to every requested instance from the user in the list, and a weight of 3 to every explicitly preferred instance in the user profile. The instance Mario Rossi is presented first because of a total weight of 4 (because of the four user requests in the past), followed by the instance Information Systems with a weight of 3 (because of the explicit preference of the user). Optionally, we may display the kind of correlation (*teacherOf*, *specificCourses*, *researchOfCourse*) that associates the extracted objects to the content of the page. Selecting an instance from the results index unit transfers the user to the Generic Instance Page; within this page, details, attributes values, and relations of the selected instance are presented, thanks to the very flexible structure of the ontological contents.

```
instance DEI memberOf Department
  name hasValue "DEI"^^xsd:string
  address hasValue "Via Ponzio 34/5, Milano, Italy"^^xsd:string
instance MarioRossi memberOf Professor
  name hasValue "Mario Rossi"^^xsd:string
  email hasValues {"mario.rossi@polimi.it"^^xsd:string}
```

```
affiliation hasValues {DEI}
instance MarioBianchi memberOf Professor
  name hasValue "Mario Bianchi"^^xsd:string
  email hasValues {"mario.bianchi@polimi.it"^^xsd:string}
  affiliation hasValues {DEI}
instance Database memberOf Course
  specificCourses hasValues {InformationSystems, SmallDatabases}
instance InformationSystems memberOf Course
instance SmallDatabases memberOf Course
instance DBResearch memberOf Research
relationInstance teacherOf(MarioRossi, {Database})
relationInstance teacherOf(MarioBianchi, {Database})
relationInstance researchOfCourse (Database, {DBResearch})
```

Figure 12. Example instances for the ontology

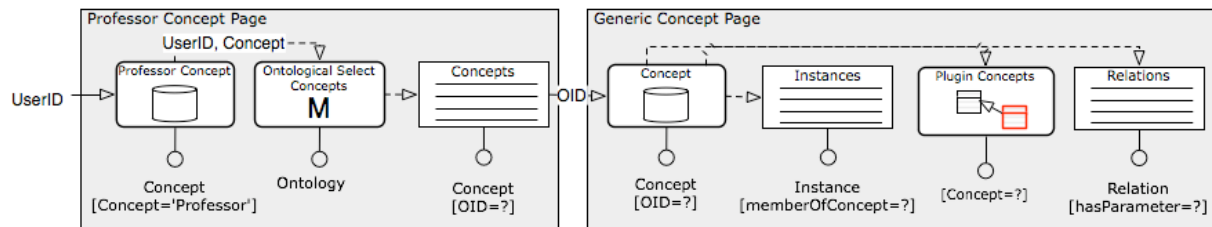
**4.3.2. Concept-based personalization.** Also in the generic concept page we may apply the profiling mechanisms and the personalized data delivery discussed so far. Another kind of personalization is made on presented concepts in this case. This is the case in which a page is describing a concept instead of an instance.

In Figure 13, the user enters the page of the concept Professor. Let's assume again that he has already visited the page of the Professor Mario Rossi four times in the past, and he has declared his preference about the Information Systems Course. The *Ontological Select Concepts* macro unit performs the two-phase query for extracting the personalized contents. In the first phase of the query, it retrieves all the concepts related to the Professor concept through any semantic attribute and relation. Therefore, from instances in Figure 12 it extracts:

- the concept Course related to Professor through the *teacherOf* relation;
- the subconcepts Assistant Professor, Visiting Professor, Full Professor, and Associate Professor;
- the concept Publication related to Professor by the *author* attribute.

The kind of association involved (*teacherOf*, *subConceptsOf*, *author*) can be shown as well.

In the second phase, we discard the concepts from the previous results for which no instances appear in the user profile. The final results are ordered based on



**Figure 13. Personalization of concepts**

the weighting algorithm. We give to every concept a weight of 1 for every requested instance, and a weight of 2 for every explicitly preferred instance of it in the user profile. The different weight with respect to the case of instance-based personalization is motivated by the fact that if a user marked a preference on a concept, all its instances were stored as preferred, thus it's much more likely that the global amount of preferred objects of a concept grows quickly, therefore it must be balanced by a lower weight. In the example, the concept Professor is presented first with a global weight of 4 (since Mario Rossi is a requested instance of this concept in the user profile), followed by the concept Course with a weight of 3 (since the Information Systems is a preferred instance of the concept in the user profile). In case the number of the returned results is less than a predefined limit (for example 3), we may decide to present in the list even the sub-concepts of the results. Therefore, the resulting list of the example contains also the concepts Full Professor, Assistant Professor, Visiting Professor, and Associate Professor.

Selecting a concept from the results index unit transfers the user to the Generic Concept Page; within this page, details, attributes, relations, and instances of the selected concept are presented. Even in this case, we may apply in the page the profiling mechanisms and the personalized data delivery discussed so far.

## 5. Query language expressive power

The capabilities of the query language used in WebML for expressing the above queries are based on the combination of both ontology data querying and ontology schema querying. Since ontologies (i.e., data instances and metadata representing the ontology schema) are both represented in an E-R model, we formulate queries as WebML primitives that interact with the backend database repository. In order to express some ontological correlations (hierarchy), we extend WebML with new units that encapsulate the semantics of such relations (inheritance, overriding, and so on). Other correlations are inferred from existing WebML units acting on top of the relational

schema. We identify the topology of the permitted queries in: *schema queries*, *data queries*, and *schema and data queries*, depending on the type of the resources retrieved.

Basic *schema queries* are defined upon the metadata structure represented with ontological terms like concepts, relations, concept attributes, and relation parameters, and return the metadata fulfilling the defined conditions. Queries that return the sub-concepts (sub-relations) of a concept (relation), and analogously the super-concepts (super-relations) of a concept (relation), are defined within the new WebML primitives *Subsume Concepts* (*Subsume Relations*) and *Plugin Concepts* (*Plugin Relations*) in [14]. Also, inheritance of attributes (parameters) is defined in the same work within the *Subsume Attributes* (*Subsume Parameters*) unit.

*Data queries* are formulated upon the metadata represented as ontology schema, and retrieve data (ontological instances and relation instances) that correspond to the defined metadata. The selection is based on the type of concepts the instances instantiate, on attribute values, and on participation in relation instances as parameters.

*Schema and Data queries* are used when we want to extract the concepts, attributes or the relations as well as their instances.

## 6. Related work

A first category of related work deals with non-ontological specification of personalized contents, and hence they are not based upon contents semantic markup through metadata. These approaches are effective for extracting simple personalized information, where the relationships between objects are codified and repetitive. However, semantic interpretation and reasoning add up a lot of expressive power and meaning to the personalization.

Previous works have addressed the personalization on the Web through semantic markup as a goal for meeting user requirements without requiring him to explicitly make statements about its preferences. We make a first classification of related work based on the

kind of data being used to generate recommendations for the current user. Approaches like [8], [9] and [18] base their selection methods on similarities among domain data; instead user-centric systems observe user behavior, and inference recommendations for him based on his interests upon the application data. In [13], the authors have followed an intermediate approach. First, they create the user context from direct sources including physical sensors, triggered application events, and external sources, enriched with inferences obtained by applying semantic rules on it. At last, the recommendation procedure it is based on similarities among both the application data and the user profile.

We focus on the second category, and more specifically we focus on the REWERSE project ([1], [2]). They provide a framework for the design and implementation of Personal Readers ([1]). Similarly to our approach, the application knowledge is ontology-based. Moreover, the domain knowledge is enriched with dynamic content extracted from the Web, and ontological rules are applied to it to construct personalized views of it. In [2], the user model for the Personal Readers is presented. It is represented by an ontology and describes characteristics of the user that may be used for generating both personalized data and adaptive contents. The user behavior triggers events resulting to the registration of its requests and to the update of its profile.

In [12], the user profile is built for a single user, but in case the user is new to the system or its profile is relatively poor, an external ontology is used as a resource for generating recommendations for him. The approach focuses on the ontological user modeling on top of the recommender systems Quickstep and Foxtrot. In both cases, the user rating is used to update its profile. Contextual information relative to user requests, Web content in the delivered pages, and time spent on a specific page do not influence the decision procedure. The mechanisms of building personalized data are based on rather simplified relations (membership of an instance to a concept) among data and do not take full advantage of ontological relations like classifications.

Some works already exist that exploit conceptual models and Web engineering techniques for designing semantic web applications ([3]), but they don't address explicitly the conceptual modeling of the semantic markup of user profiles for personalization.

Although most of the presented works study the user behavior either by means of requests on the Web (in terms of requested pages or submitted search criteria for contents), or as explicit statements about its preferences, not many of them take into account the user behavior within a specific request as an indicator

of further knowledge about its interests. In [6], the time the user spends on a Web page indicates an implicit preference for the presented object, and in [5] an extended list of indicators expressing implicit interests for the user on the Web is presented. These aspects can be easily incorporated into our approach by extending the weighting criteria in the definition of the weights of the personalized contents.

Some existing methods investigate the use of ontologies for the design of semantic Web applications. In [10] the authors make use of RDF(S) to specify the design of a Web information system, but they do not explicitly address the personalization problem, while our solution is tailored to the specific needs of more meaningful information personalization. The survey in [7] presents current technologies for the implementation of adaptation using ontologies. Our approach is distinguished from the included methods as it results also to the conceptual modeling of the provided solution. Such produced models may become more easily reusable patterns in the field of the semantic Web engineering.

## 7. Conclusions and future work

This paper presented a model-driven approach to the design of web applications that exploits user preferences and behavior metadata for delivering targeted contents. We specified a simple user profile ontology and a set of modeling patterns for gathering the user profile knowledge (in terms of explicitly declared preferences and of recorded requests of objects) and for exploiting the metadata describing such profile in order to retrieve new connections between the application data and the user preferences, and thus, for delivering personalized content.

Our main contribution consists in a first attempt of using well-established conceptual modeling languages, namely WebML, for defining metadata-driven personalization. The advantages of this approach are related to (i) the relational implementation of metadata expressed as ontology and describing user profile and domain data, (ii) the easy definition of management and exploitation patterns upon such metadata for the delivery of objects in Web pages that are connected to the user interests thru relations unknown at design time, and (iii) the low cost of evolution of the existing models.

Examining the details of our approach, first we have selected WSMML, a Web-based ontology language to represent both data and its metadata. The advantage of such decision is the easy-to-exchange user profile among applications on the Web. This facilitates the user during the creation of his profile, since he can

reuse his profile within different Web applications. Thanks to the selected ontology language, emerging semantic standards that enable reasoning mechanisms can be applied in the near future. This is indeed one of the future works we plan to address, for the purpose of allowing semantic inference upon the data. In the present work, the manipulation and evaluation of the metadata describing the user profile and the domain data is made thru relational queries upon its relational implementation. Future works will consist in the extension of the approach towards reasoning rules onto the user profile and application data ontologies, at the purpose of extracting more significant personalized information for the user. On the other hand, the choice of using WSML as an ontology language facilitates the realization of our method in a Web services framework. Since WSML is a representation language for Semantic Web services, future works will include the integration of personalization mechanisms in Web services communication.

Second, we have extended WebML with new components for the management of the ontology schema, and therefore, for creating and exploring the metadata describing the user profile and the application data. Such components hide from the developer the implementation details of the WSML language, although he is still required to understand the semantics of the ontology language constructs. Since current implementation of the ontology is on a relational system, some aspects such as relational properties (like transitivity and asymmetry) or multiple hierarchies, have been ignored since they increase the metadata complexity. However, it will be simple to incorporate them in a future implementation within an ontological repository.

Finally, we have used the new WebML components to create modeling patterns that may be used for exploring the metadata describing the user profile and the application data. The whole personalization mechanism may be controlled by assigning weights to the retrieved connections. These weights play the role of filters allowing the developer to increase or decrease the importance of some personalization aspects with respect to others.

**Acknowledgement.** This work was partially supported by the MIUR PRIN project “Basi di dati crittografate” (2006099978).

## 8. References

[1] G. Antoniou, M. Baldoni, C. Baroglio, C. Patti, R. Baumgartner, T. Eiter, M. Herzog, R. Schindlauer, H. Tompits, F. Bry, S. Schaffert, N. Henze, and W. May, “Reasoning Methods for Personalization on the

Semantic Web”, *Annals of Mathematics, Computing & Teleinformatics (AMCT) Vol. 2(1)*, 2004, pp. 1-24.

[2] R. Baumgartner, N. Henze, and M. Herzog, “The Personal Publication Reader: Illustrating Web Data Extraction, Personalization and Reasoning for the Semantic Web”, *European Semantic Web Conference 2005*, pp. 515-530.

[3] M. Brambilla, I. Celino, S. Ceri, D. Cerizza, E. Della Valle, and F. Facca, “A Software Engineering Approach to Design and Development of Semantic Web Service Applications”, *International Semantic Web Conference (ISWC)*, Athens, GA, USA, 2006, pp. 172-186.

[4] Ceri, S., P. Fraternali, A. Bongio, M. Brambilla, S. Comai, and M. Matera, *Designing Data-Intensive Web Applications*, Morgan-Kaufmann, USA, 2002.

[5] M. Claypool, P. Le, M. Waseda, and D. Brown, “Implicit interest indicators”, *Proceedings of the 6th International Conference on Intelligent User Interfaces*, Santa Fe, USA, 2001, pp. 33-40.

[6] J.A. Constan, B.N. Miller, D. Maltz, J.L. Herlocker, L.R. Gordon, and J. Riedl, “GroupLens: Applying collaborative filtering to usenet news”, *ACM Communications Vol. 40(3)*, 1997, pp. 77-87.

[7] P., Dolog, and W., Nejd, “Semantic Web technologies for the Adaptive Web”, *The Adaptive Web: Methods and Strategies of Web Personalization*, Springer Verlag, LNCS 4321, 2007, pp. 697-719.

[8] M. Eirinaki, C. Lampos, S. Paulakis, and M. Vazirgiannis, “Web personalization integrating content semantics and navigational patterns”, *WIDM '04: Proceedings of the 6th annual ACM international workshop on Web information and data management*, Washington DC, USA, 2004, pp. 72-79.

[9] R. Ghani, and A. Fano, “Building recommender systems using a knowledge base of products semantics”, *Workshop on Recommendation and Personalization in E-Commerce*, at the 2nd Int'l Conf. on Adaptive Hypermedia and Adaptive Web Based Systems, Malaga, Spain, 2002.

[10] G.-J., Houbert, P., Barna, F., Frasinca and R., Vdovjak, “Hera: Development of Semantic Web Information Systems”, *International Conference on Web Engineering*, Spain, July 14-18, 2003, pp. 529-538.

[11] I. Manolescu, M. Brambilla, S. Ceri, S. Comai, and P. Fraternali, “Model-Driven Design and Deployment of Service-Enabled Web Applications”, *ACM TOIT Vol. 5(3)*, 2005, pp. 439-479.

[12] S.E. Middleton, N.R. Shadbolt, and D.C. De Roure, “Ontological User Profiling in Recommender Systems”, *ACM Transactions on Information Systems Vol. 22(1)*, 2004, pp. 54-88.

[13] X. Pan, Z. Wang, and X. Gu, “Context-Based Adaptive Personalized Web Search for Improving Information Retrieval Effectiveness”, *International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM)*, 2007, pp. 5422-5425.

[14] C. Tziviskou, and M. Brambilla, “OntoWebML: A Knowledge Base Management System for WSML Ontologies”, *2nd International Workshop on Semantic Web Enabled Software Engineering*, USA, 2006.

[15] WebML Web site <http://www.webml.org>

- [16] WebRatio site <http://www.webratio.com>
- [17] Web Service Modeling Language (WSML) site <http://www.wsmo.org/wsm>
- [18] V. Wolowski, N. Ishikawa, and H. Sumino, "Semantic Web Approach to Content Personalization", *International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies (UBICOMM)*, Papeete, France, 2007, pp. 109-117.

## APPENDIX A – Academic Institution WSML Ontology

```

...
concept EducationOrganization
  name ofType (1) xsd:string
  address ofType (1) xsd:string
concept Department subConceptOf
  EducationOrganization
concept Institute subConceptOf EducationOrganization
concept Program subConceptOf EducationOrganization
concept ResearchGroup subConceptOf
  EducationOrganization
concept School subConceptOf EducationOrganization
concept University subConceptOf
  EducationOrganization
concept Work
concept Course subConceptOf Work
concept Research subConceptOf Course
concept Conference subConceptOf Course
concept Person
  name ofType (1) xsd:string
  email ofType xsd:string
concept Employee subConceptOf Person
  affiliation impliesType
  EducationOrganization
concept FacultyMember subConceptOf Employee
concept Professor subConceptOf FacultyMember
  doctoralDegree impliesType University

```

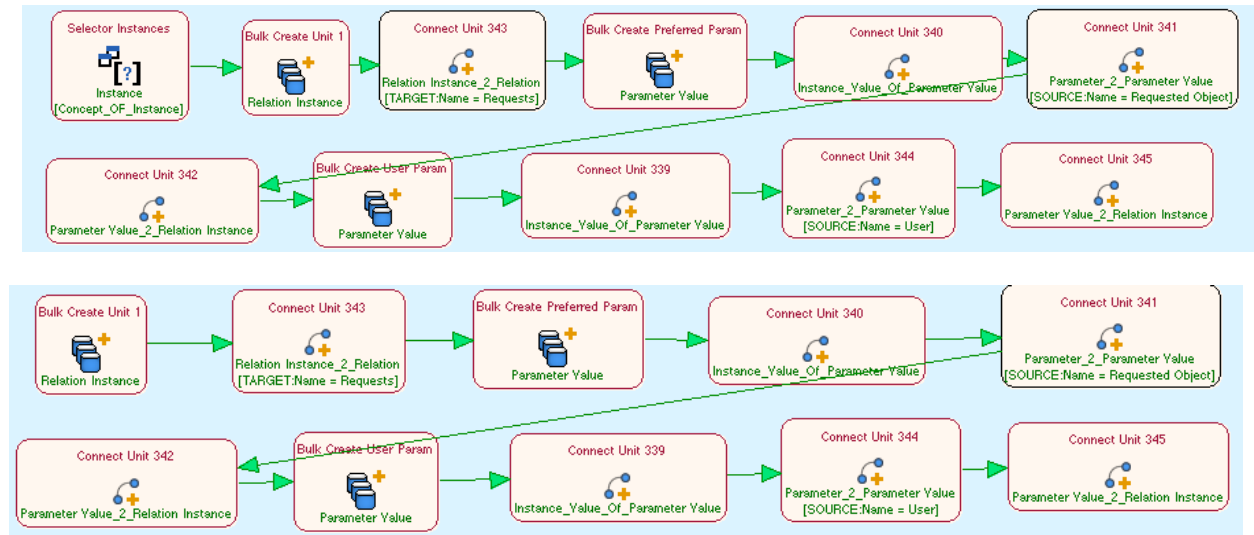
```

concept AssistantProfessor subConceptOf Professor
concept AssociateProfessor subConceptOf Professor
concept FullProfessor subConceptOf Professor
concept VisitingProfessor subConceptOf Professor
concept Lecturer subConceptOf FacultyMember
concept AdministrativeStaff subConceptOf Employee
concept Director subConceptOf AdministrativeStaff
concept ClericalStaff subConceptOf
  AdministrativeStaff
concept SystemsStaff subConceptOf
  AdministrativeStaff
concept Student subConceptOf Person
concept UndergraduateStudent subConceptOf Student
concept GraduateStudent subConceptOf Student
  degree impliesType University
relation advisor
  s ofType Student
  p ofType Professor
relation citations
  p ofType (1) Publication
  c ofType Publication
relation courses
  u ofType (1) University
  c ofType Course
relation researchOfInterest
  p ofType (1) Person
  a ofType Research
relation researchOfCourse
  c ofType (1) Course
  a ofType Research
relation takesCourse
  s ofType (1) Student
  c ofType Course
relation teacherOf
  t ofType (1) FacultyMember
  c ofType Course
relation teachingAssistantOf
  t ofType (1) TeachingAssistant
  c ofType Course

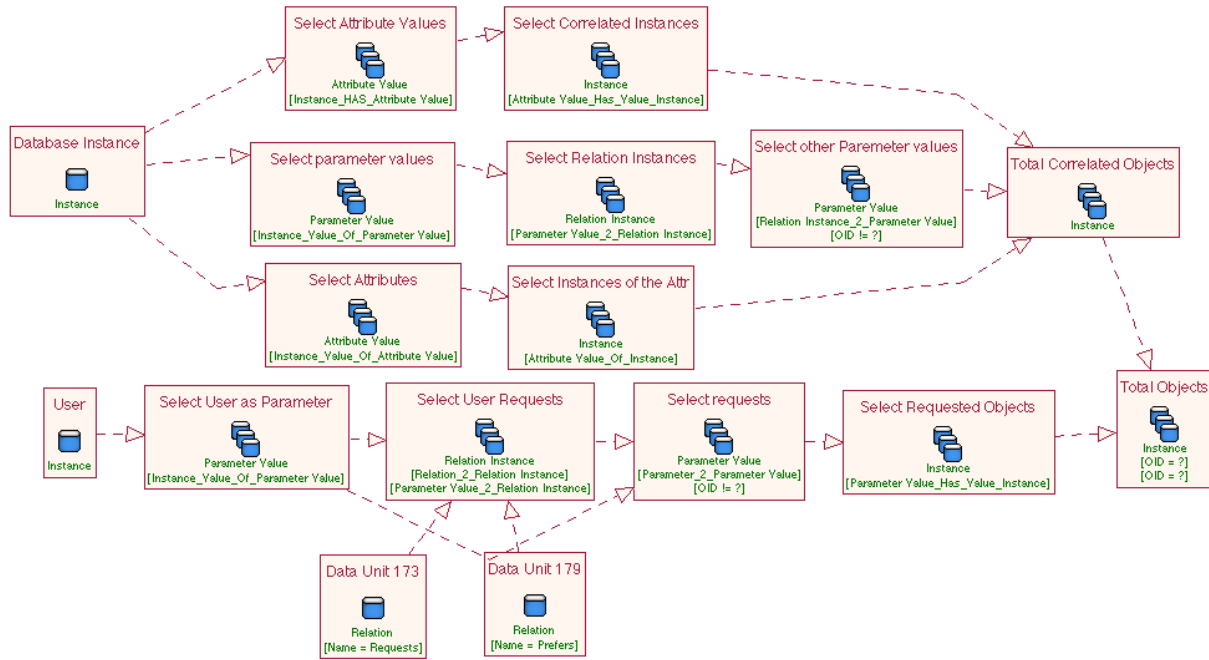
```

## APPENDIX B – Macro implementation in WebRatio

**Macro for User Implicit Profiling.** The first operation chain presents how implicit user requests of concepts are registered in a WebML application. The second chain demonstrates how implicit user requests of instances are registered.



**Macro for Instance-based Personalized Data Delivery.** The operation chain depicts the selection of all the instances in the user profile that are correlated to the input instance.



**Macro for Concept-based Personalized Data Delivery.** The operation chain depicts the selection of all the instances in the user profile that are correlated to the instances of the input concept.

