

THE DESIGN AND IMPLEMENTATION OF THE KNOWLEDGE POOL IN THE CHIRON PROJECT

Nektarios Moumoutzis

*Technical University of Crete / Laboratory of Distributed Multimedia Information Systems
and Applications (TUC/MUSIC)
University Campus, Kounoupidiana
73100, Chania, Greece
nektar@ced.tuc.gr*

Panagiotis Kontogiannis

*Technical University of Crete / Laboratory of Distributed Multimedia Information Systems
and Applications (TUC/MUSIC)
University Campus, Kounoupidiana
73100, Chania, Greece
pkonto@ced.tuc.gr*

Stavros Christodoulakis

*Technical University of Crete / Laboratory of Distributed Multimedia Information Systems
and Applications (TUC/MUSIC)
University Campus, Kounoupidiana
73100, Chania, Greece
stavros@ced.tuc.gr*

Abstract

In the context of the Leonardo Da Vinci project CHIRON, reference material related to research outcomes is analyzed as well as experiments and best practice solutions for new forms of eLearning, based on integration of broadband web-, digital TV- and mobile technologies for ubiquitous applications in the sector of non-formal and informal life-long learning. This material is organized in a special purpose distributed knowledge pool that provides all the essential management and browsing functionality to allow for the organization and dissemination of the content. In this paper we describe the design of the CHIRON knowledge pool focusing on its extensibility features. We also present its web-based user interfaces.

Keywords

Content management, knowledge pool, user interfaces

1. INTRODUCTION

The CHIRON project is a Leonardo Da Vinci Project that aims to develop reference material presenting and analyzing research outcomes, experiments and best practice solutions for new forms of eLearning, based on integration of broadband web-, digital TV- and mobile technologies for ubiquitous applications in the sector of non-formal and informal life-long learning. One of the most valuable tools for its success is the knowledge pool, a web-based distributed information system that handles this reference material offering functionalities for managing, searching and browsing it.

In particular, the basic elements of the knowledge pool are the following:

- Data base with information of channels for data collection and continuous data sources such as scientific publications and conferences, research projects, thematic web resources, sources of statistical data, sources of technological data, international VET associations, international standardization committees, consultation bodies, task force groups and official documents of the European Commission etc.
- Data base of all the possible users of the reference materials according to the defined project target groups. It will be used to disseminate regular information about the project progress as well the content of the reference materials.

- Reference Material Repository, containing source materials, intermediate and final reference materials.
- Multilingual glossary of keywords in the field of technology supported learning and the applied Information and Communication Technologies.

Clearly, such a system resembles content management systems [Robertson J., 2003]. However, most available content management systems, either open source or not, are highly complicated due to their generality. CHIRON needs are specific and a decision was made to build a special system in order to meet these needs. In other words, a decision was made to follow a solution that is better suited to the specific needs of the project instead of customizing a general purpose content management system to the project.

The rest of this paper is organized as follows: Section 2 presents a rough information analysis and the basic functionality of the system. Section 3 presents the conceptual data model of the knowledge pool using UML class diagrams. Section 4 depicts the current web-based user interface implementation. Conclusions and necessary extensions to meet highly desirable features are presented in section 5.

2. INFORMATION ANALYSIS AND BASIC FUNCTIONALITY

The knowledge pool supports two major user classes:

- **Content providers** that use the CHIRON knowledge pool **Content Manager** to manage information related to reference material.
- **End-users** that use CHIRON knowledge pool **Content Browser** to receive detailed information about the reference material.

In order to define the necessary functionality for each one of the two user classes (i.e. the functionality of the Content Manager and the Content Browser) we must first identify in more detail the types of reference material that should be managed by the system. In order to do so we must first identify the most critical concept of the system: the concept of a *resource*. A resource is an object (such as a publication, presentation, report, software product, etc.) that represents a source of valuable information. Then, a first approach is to divide the information into the following three major sections with corresponding categories:

- *Resources*: In this section belong all the resources that can't contain other resources. These resources must be related with a resource collection. For example a **publication** can be related with the **journal** or the **proceeding** collection it was published in, a **minute** with a **meeting** collection etc. Resources can be related with other resources. For example a **demonstrator** can be related with a set of **testbeds** and technical **reports**, a **publication** with a set of other **publications** (references) etc. Resources are classified into types that include: Deliverables, Demonstrators, Minutes, Presentations, Publications, Reports, Software, and Test datasets.
- *Resource collections*: In this section belong all the resources that may contain other resources creating a part-of hierarchy. For example a **conference** is a collection because it may contain a set of papers, presentations, demonstrations, etc. A resource collection may contain other resource collections. For example the collection **project** contains the collection **work package**; the collection **work package** contains the collections **tasks** and **meetings** etc. Resource collections are classified into types that include **Conference series**, **Conferences**, **Journal issues**, **Journal series**, **Meetings**, **Proceedings**, **Projects**, **Tasks** and **Work packages**.
- *Basic elements*: This section contains some auxiliary resources such as **keywords** forming a multilingual dictionary, **Organizations**, **Groups** of people and **Persons**.

2.1. Basic functionality for content providers

Content providers should have all the necessary facilities to manage (insert/edit/delete) resources, collections and basic elements. Inserting or editing a resource correspond to editing its attributes, establishing its relationships to other resources, collections of resources and basic elements as well as creating relationships with persons that participated in its production. Inserting or editing a resource collection corresponds to editing its attributes, establishing relationships with related resources and basic elements. Editing a basic element means to edit its attributes and establish its relationships with other related objects.

2.2. Basic functionality for end-users

The most important functionality that should be provided for end-users is inspection of resources. In order to provide a powerful browsing mechanism of resources, they should be classified in semantic categories so that the end-users could select the categories that they want and then they could see the related resources.

Apart from the above, end-users should be provided with facilities to insert, access, and give comments on Resources (e.g. scientific publications, technical reports, user manuals, etc.). A search facility will also be available allowing the users to search for information based on specific parameters or using hierarchies of resource categories.

3. CONCEPTUAL DATA MODEL OF THE KNOWLEDGE POOL

The CHIRON knowledge pool resources database follows an expandable philosophy, thus allowing the integration of new resource types with their specific attributes and classification hierarchies. In this section we present its conceptual model using UML class diagrams [Fowler M., 2003].

3.1. Resources and related concepts

The most critical concept of the knowledge pool is the concept of *Resource*. The related classes and their relationships are depicted in Figure 1. A *Resource* is essentially any kind of an object (such as a publication, presentation, report, software product, etc.) that has a unique *id*, a *title*, an *abstract* and a *url* that points to the location of a file that the actual data of the resource resides. The *virtual* property is discussed in section 3.3.

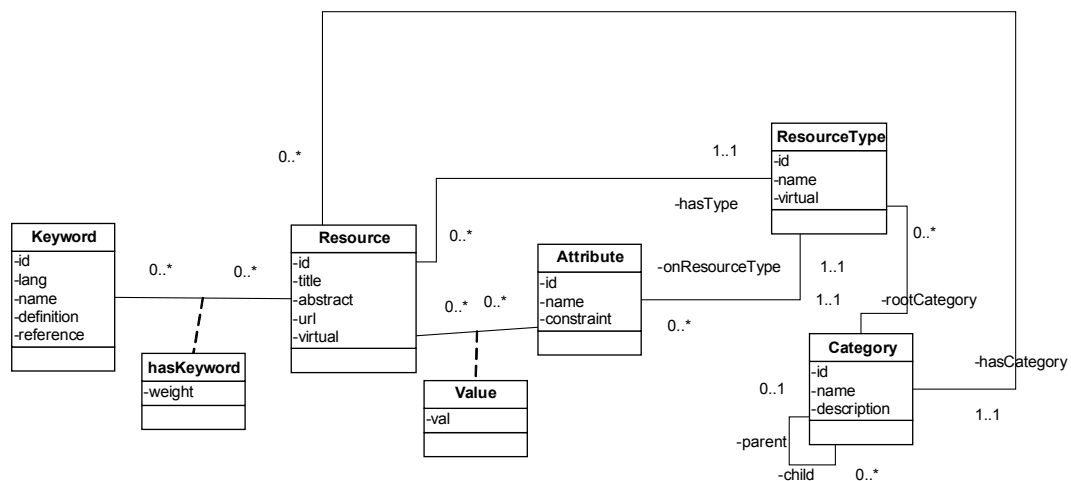


Figure 1: UML class diagram for the description of resources and related classes.

A *Resource* has one and only one *ResourceType* that signifies its type (i.e. publication, presentation, report, software product, etc). Each *ResourceType* has an *id* that uniquely identifies a type, and a *name*. The *virtual* property will be described in subsection 3.3. Apart from its type, a *Resource* also has one and only one *Category* that refers to a semantic taxonomy of resources (e.g. the ACM 1998 computing classification system [ACM, 1998]). This taxonomy is used to define a tree of categories that can be used for organizing the resources and facilitating content browsing and searching. In order to reflect this hierarchy of categories there is a parent-child relationship defined on the *Category* class. Each instance of the class *Category* has a unique *id*, a *name* and a *description*. Moreover there is a relationship between each instance of the class *ResourceType* and an instance of the class *Category* signifying a *rootCategory* (i.e. a hierarchy tree) that is related to each *ResourceType*. E.g. we may associate, using this relationship, a specific resource type (e.g. reports) with a hierarchy of categories meaning that all reports should be associated with categories belonging to this hierarchy.

The description of resources in an extensible way is handled through the use of a special class called *Attribute*. This class describes attributes of resources (i.e. properties) and may be used to add dynamically new attributes by adding new instances to this class. An instance of the *Attribute* class is described by a unique *id*, a *name* and a *constraint*. The constraint is used to define various data constraints that should hold on the values of attributes. In order to support a schema-based description of attributes, there is a relationship between the *Attribute* class and the *ResourceType* class (named *onResourceType*). Each attribute is connected, through this relationship to one and only one resource type meaning that only resources of this type are able to use this attribute. The relationship between the *Resource* class and the *Attribute* class is a many-to-many relationship called *Value* that has a *val* property used to hold the specific value of each resource with respect to each attribute that is valid for this resource.

Apart from attributes, the resources are also described using structured vocabularies of keywords. The class *Keyword* is used for this purpose and has an *id* that uniquely identifies a keyword, a language code (*lang*), so that keywords could be described in different languages, a *name*, a *definition* and a *reference* to external material. The relation between keywords and resources is captured through the many-to-many relationship *hasKeyword* that has a *weight* property used to record the relative weight of a keyword in a resource. Using this approach it is easy to support similarity-based retrieval techniques based on the controlled vocabularies of keywords.

3.2. Linking resources

The classes used to model the linking between resources are depicted in Figure 2. The model is as simple as possible yet providing all the necessary features to ensure generality and powerful browsing functionality for the end-users.

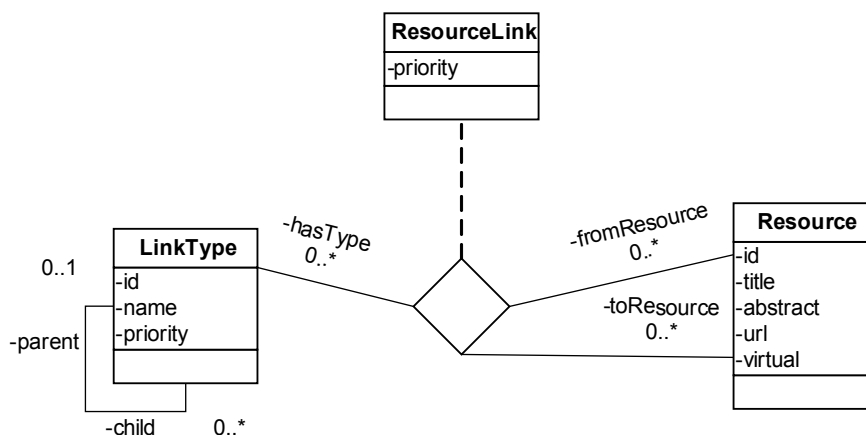


Figure 2: UML class diagram for the description of mechanisms for linking resources.

Links are typed and hierarchically organized. This is captured in class *LinkType*. Each instance of this class has a unique *id*, a *name* and a *priority* that defines the relative importance (in terms of presentation mainly) of each link type in the context of its parent link type. The hierarchical organization of link types is captured through a parent-child relationship defined on the *LinkType* class.

The particular links between resources are not modeled as a separate class but as ternary relationship. In particular a *ResourceLink* is a relationship between a *LinkType* and two *Resources*. The *LinkType* of the relationship specifies the type of the link. The first *Resource* of the link is the source of the link while the second one is the target of the link. The relationship has a *priority* property that specifies the relative priority of a link on a set of links of the same type between two resources in order to give a hint to the presentation software when presenting these links to the user.

3.3. Collections

The concept of a collection captures the fact that resources are organized into sets of interrelated resources based on various criteria. E.g. all papers presented on a conference are considered as a correlated set of resources. The classes used to model the concepts related to collections are depicted in Figure 3.

A *Collection* has a unique *id*, a *name*, a *description* and *url*. *Collections* are organized hierarchically through a parent-child relationship (e.g. a journal may contain journal issues that are also collections). A specific *Resource* is contained to one and only one *Collection* (this is represented through the many-to-one *inCollection* relationship). Each collection has a specific type represented through the many-to-one relationship *hasType* between this class and the *CollectionType* class. This latter class has a unique *id* and a *name*. Examples of *CollectionTypes* include Conferences, Journals, Proceeding etc. In order to represent the fact that for certain *CollectionTypes* only *Resources* of specific *ResourceTypes* are allowed, there is a many-to-many relationship between *CollectionType* and *ResourceType* called *contains*.

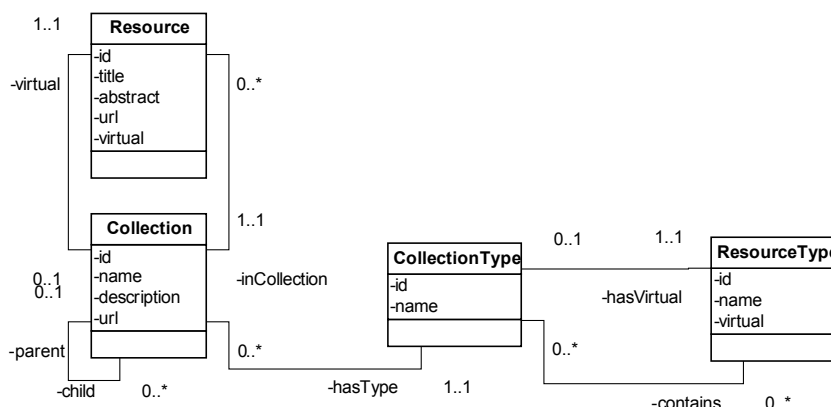


Figure 3: UML class diagram for the description of collections and related classes.

In order to take advantage of the extensibility mechanisms provided for the description of *Resources* and *ResourceTypes* classes (e.g. the facilities to define new *Attributes* or to associate with *Category* trees) in the case of *Collections* and *CollectionTypes* respectively, each instance of the latter two classes is associated to a (virtual) instance of one of the former classes. This way, a *Collection* is associated with a *virtual Resource* through the corresponding many-to-one relationship and by this way it can use all the extensibility features of the corresponding virtual resource. The same holds for a *CollectionType* associated, through the *hasVirtual* relationship to a *ResourceType*. The *virtual* attribute of a *Resource* or a *ResourceType* is a flag that is set when an instance of these classes is used in this mechanism.

3.4. Persons and related concepts

Figure 4 depicts the classes used to describe persons and related concepts. In this class diagram the central class is *Person*. An instance of this class represents an individual that participates in some way in the production of a resource and is characterized by a unique *id*, its *firstname*, *lastname*, *middlename*, *title*, *email*, *url* and *address*.

The relation between persons and resources is captured through the ternary many-to-many-to-many relationship *Credits* that relates a *Resource* with a *Person* that participated in its production along with the particular *Role* that it had. A *Role* has a unique *id*, a *name* and a *description*. The *priority* property of the relationship is used to provide an ordering of the persons that share the same role in the production of a particular resource (e.g. the authors of a paper).

Persons are grouped into *PersonGroups* through the *inGroup* relationship. This relationship allows to record historical information about the time period during which each *Person* is associated to a *PersonGroup* through the *fromDate* and *toDate* properties. A *PersonGroup* is described by a unique *id*, a *name*, a *description* and a *url*. Furthermore, each *PersonGroup* is associated, through the many-to-one *inOrganization* relationship, with an *Organization*. An *Organization* is described by its unique *id*, *name*, *description*, *url*, *country*, and *address*.

Finally, there is a many-to-many relationship between a *PersonGroup* and a *Collection* named *Owner* representing the fact that each collection may belong to some person groups (e.g. a conference is organized by some special interest groups of some scientific organizations). The *ownerrole* property of this relationship may be used to assign different roles to different groups associated to a particular collection.

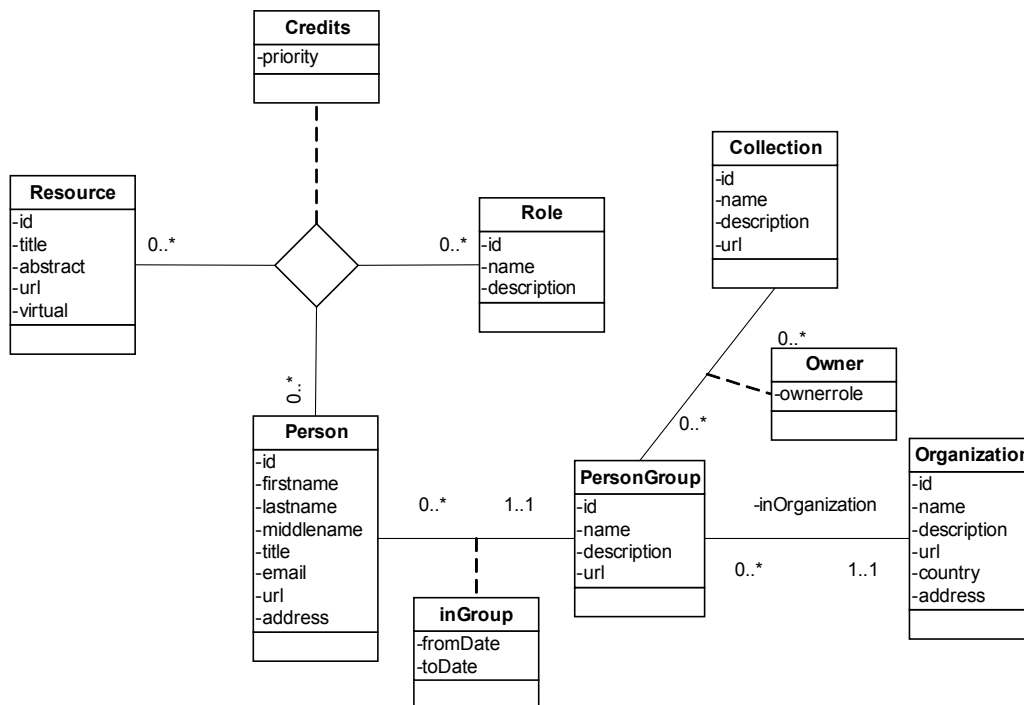


Figure 4: UML class diagram for the description of persons and related concepts.

3.5. Users and related concepts

Figure 5 depicts the classes used to describe users and related concepts. In this class diagram the central class is *User*. An instance of this class represents an individual that has some permission to use some of the functionality of the knowledge base and some of his actions are recorded by the system. Such an individual is characterized by a unique *id*, its *firstname*, *lastname*, *email*, *username*, *password*, *profession* and *country*.

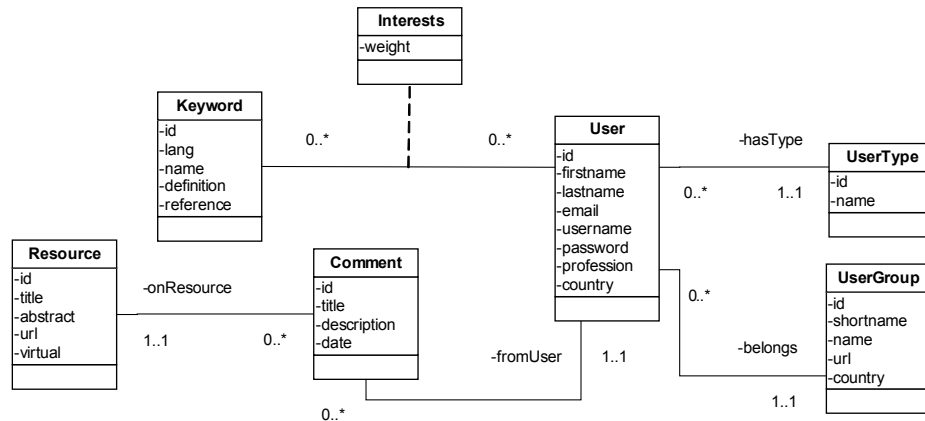


Figure 5: UML class diagram for the description of users and related concepts.

Users are grouped into *UserGroups* (through the many-to-one relationship *belongs*). Each *UserGroup* is described by a unique *id*, a *shortname*, a *name*, a *url* and *country*. Moreover, *Users* are associated with *UserTypes* (through the many-to-one relationship *hasType*). Each *UserType* is described by a unique *id*, and *name*. Examples of *UserTypes* are end-users, content-providers, and administrators.

The specification of user interests in order to be able to represent simple forms of user profiles is handled through the association of users and keywords. In particular, there is a many-to-many relationship between the *User* and *Keyword* classes that has a *weight* property.

Finally, there is a simple mechanism for supporting user comments on resources. There is a *Comment* class described by a unique *id*, a *title*, *description* and *date* of issue. Each *Comment* is associated to one *Resource* (through the many-to-one relationship *onResource*) and to one *User* (through the many-to-one relationship *fromUser*).

4. USER INTERFACES

In this section we give a brief description of the current implementation of the CHIRON knowledge pool interfaces exporting the functionality of its content manager and the content browser components. All interfaces have been implemented as web pages communicating with the knowledge pool relational databases.

The screenshot shows the HIRON Content Manager interface. The header includes the HIRON logo and the text "REFERRING UBIQUITOUS LEARNING". Below the header, there are navigation links: "home | search | contact us | sitemap". The main content area is titled "Resources: publications" and contains a table of publications. The table has three columns: "Title", "Persons", and "Category". Each row in the table has a checkbox in the first column. Below the table are buttons for "new", "edit", and "delete".

	Title	Persons	Category
<input type="checkbox"/>	3D content-based retrieval with spin images	Del Bimbo A., D'Amico G., Assfalg J., Pala P.	Information Storage and Retrieval
<input type="checkbox"/>	A Graph Edit Distance Based on Node Merging	Del Bimbo A., Berretti S., Pala P.	Information Storage and Retrieval
<input type="checkbox"/>	A Hybrid User Model for News Classification	Billsus D., Pazzani M.	Information Systems
<input type="checkbox"/>	A Natural Language Model and a System for TV-Anytime Information for Mobile Services	Christodoulakis S., Karanastasi N., Kazasis F.	Information Systems Applications
<input type="checkbox"/>	A personal news agent that talks, learns and explains	Billsus D., Pazzani M.	Data
<input type="checkbox"/>	An Ontology-based Natural Language Interface Generator for Multimedia Repositories	Christodoulakis S., Karanastasi N.	Information Systems
<input type="checkbox"/>	Casimir: Information Retrieval Based on Collaborative User Profiling	Berney B., Ferneley E.	Information Storage and Retrieval
<input type="checkbox"/>	CiteSeer: An Autonomous Web Agent for Automatic Retrieval and Identification of Interesting Publications	Bollacker D., Lawrence S., Giles L.	Data
<input type="checkbox"/>	Combining Content-Based and Collaborative Filters in an Online Newspaper	Claypool M., Gokhale M., Miranda T., Murnikov P., Netes D., Sartin M.	Information Systems
<input type="checkbox"/>	GraphOnto: A Component and a User Interface for the Definition and Use of Ontologies in Multimedia Information Systems	Christodoulakis S., Tsinaraki C., Polydoros P.	Information Systems

Pages: 1 2

© MUSIC-TUC, Chania Crete, 2005

Figure 6: The page of the Content Manager presenting all publications.

4.1. Content Manager Interface

The Content Manager allows content providers to insert/edit information about reference material into the knowledge pool. In particular, based on the division of the material presented in section 2, the Content Manager allows for the insertion and editing of information for resources, collections of resources and auxiliary (basic) elements (shown in a menu at the left hand side of the page in Figure 6). The user is able to select one of the offered types of collections, types of resources or the auxiliary elements. After that, the main area of the page shows a table with a list of objects of this specific type. The user is able to select an object and then he may edit or delete it. He is also able to create a new object of this type by pressing the corresponding button at the bottom of the table. In Figure 6 the publications resource type has been selected and the corresponding resources are presented.

The Chiron Project: Referring Ubiquitous Learning - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Add a new publication

Step 1 of 3: Insert the attributes, the category and the keywords of the resource

Main Attributes

Name:

Abstract:

Url:

Extra Attributes

filename:

pages:

status:

Category

Select one category from the hierarchy:

Data

Collection

Select the collection (project, journal, conference etc.) the resource belongs to:

conferences - 116 Audio Engineering Society (AES 2004)

Keywords

Select keywords from the list below:

accessibility
Digital libraries
Personalization

Selected Keywords:

or

Figure 7: The first page for the insertion of a new publication in the Content Manager.

Figure 7 presents the page that is used to insert a new publication. The pages used to insert a new resource of some other type (e.g. a new deliverable, presentation etc.) are similar. The insertion process is split in three separate steps (pages) for convenience. Figure 7 presents the first step of the process. The first section in this page is used to insert the standard attributes that are present in any resource. The next section is used to insert custom attributes (i.e. the ones used only for this particular resource type). The third section is used to insert the resource's category. The next section is used to insert the collection that the resource belongs to. The last section is used to select the keywords that the resource has.

Two more pages follow to complete the insertion of a publication. The first one is used to insert information about persons that participated in the production of the resource and their roles (see section 3.4) the next one is used to create links with related resources based on link types (see section 3.2). The web pages used to edit collections and auxiliary elements follow a similar philosophy and design.

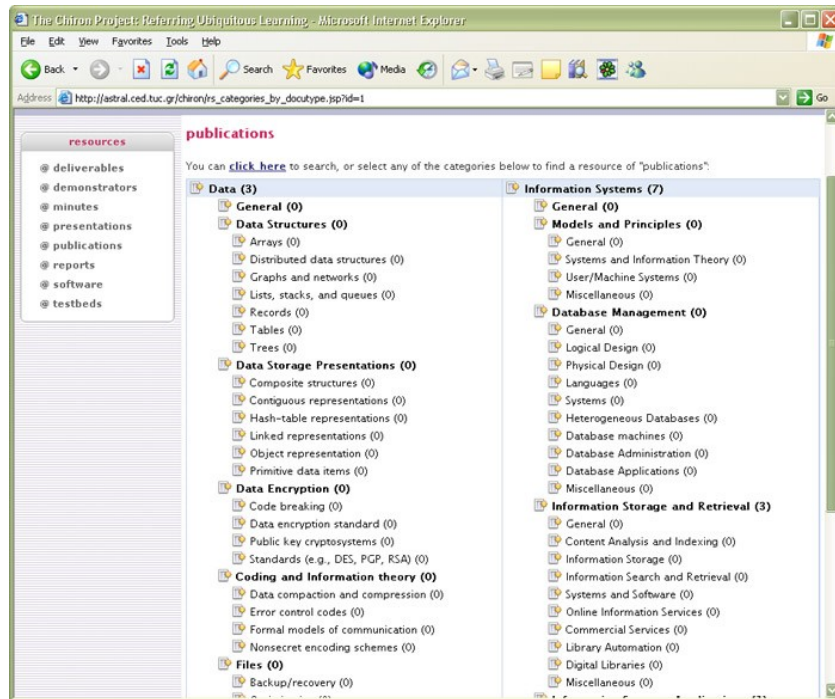


Figure 8: The page of the Content Browser publications by category.

4.2. Content Browser Interface

The Content Browser allows end-users to browser information about reference material stored in the knowledge pool. In particular, based on the classification of the material presented in section 2, the Content Browser allows for the presentation of information about resources and the detailed presentation of their properties and relationships.

The end-user is given first a list of available resource types (see left-hand side menu of Figure 8). In this particular case the publications resource type has been selected. After a resource type has been selected, all the category hierarchy associated with it is presented in the main area of the page (recall the role of the *rootCategory* relationship between *ResourceType* and *Category* classes from section 3.1). Then the end-user is able to see the number of resources associated with each category in the tree and he can select one category to see a list of the resources associated with it. This kind of browsing mechanisms is particularly suited for situations when a big number of resources are available and the end-user wants to be able to restrict the number of resources that he receives in an effective and user-friendly manner. The category hierarchy is a very nice mechanism to address this need.

Figure 9 shows the page presented after the end-user has selected the “Information Systems” category of publications. A similar picture will be shown when the end-user selects any other category on any other resource type. The main area of the page presents the list of resources (i.e. publications in this case) that have been registered under the specific category in the knowledge pool. Each resource is presented in a concise manner giving only the most important information about it. The end-user is able to select one of these resources in order to see more detailed information.

Figure 10 shows the page that follows a selection of a particular publication to be presented in detail (a similar page is presented for any type of resource). The page presents the information about the resource in a tabular manner so that it is possible to extend this table to accommodate extra attributes and relationships with persons and links with other resources. The comments added by users are presented at the bottom of the page. The current user can also add his comments by selecting the ‘*add your comments here*’ link.

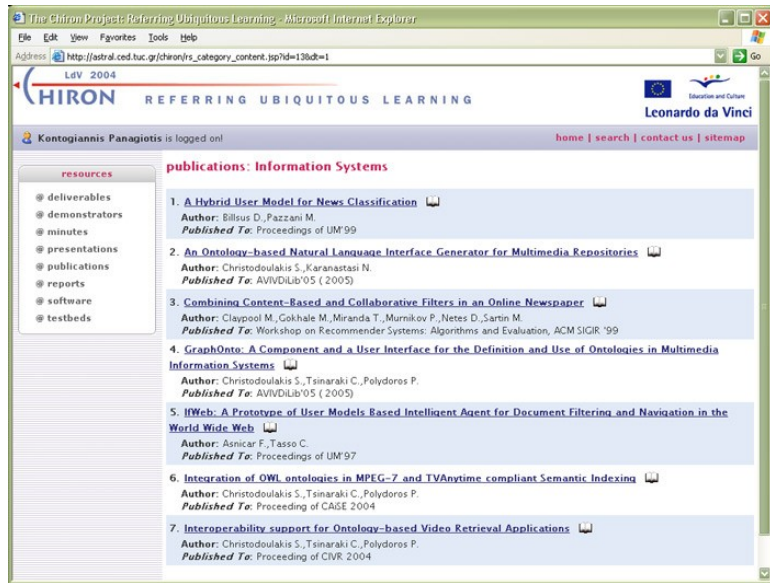


Figure 9: The page of the Content Browser presenting the list of publications under a specific category



Figure 10: The page of the Content Browser presenting a publication.

5. CONCLUSIONS AND FUTURE WORK

In this paper we have presented the design and the current implementation of the CHIRON project knowledge pool, a distributed information system that allows for the management of reference material collected and created in the context of the CHIRON project and its dissemination inside and outside the CHIRON consortium.

The knowledge pool distinguishes between the roles of content providers that provide information about reference material and end-users that browse the knowledge pool to receive information about the material. Its design follows an extensible philosophy in order to accommodate new types of material easily.

A number of extensions in the design and implementation of the CHIRON knowledge pool are necessary to support more sophisticated functionality that is necessary for the needs of the users of the system. One of the most important ones is the support of user permissions so that only authenticated users, groups of users or types of users are able to change specific resources, resource types, collections or collection types. The corresponding mechanisms should be based on the existing modeling mechanism for resources, collections and users and build on top of them a robust model for user permissions that will allow for a flexible specification of permission and deployment of corresponding strategies from the system.

Another direction of extensions refers to the support of sophisticated search filters and user profiles. At the current version of the knowledge pool, user profiles are essentially vectors of keywords that allow the end-users to specify their interests in terms of keywords that describe the content they like. However the resources are also described through attributes, categories, collections that they belong to, persons that they are associated to, and other resources that they are linked to. A profile should take into account all this information and moreover it should be able to use connectives like Boolean operators (not, and, or) in order to build complex filters. Taking into account that most of the end-users may find it difficult to use such an involved model, appropriate user interfaces should be developed that will hide the complexity of the model and will present an intuitive interface to the users exploiting well known user interface paradigms.

ACKNOWLEDGEMENT

This work presented in this paper has been funded in the scope of the Leonardo Da Vinci CHIRON project [PROJECT FR/04/C/F/RF-81103 (2004-2237)].

REFERENCES

- Robertson J. (2003). *So, what is a content management system?* Available at http://www.steptwo.com.au/papers/kmc_what/index.html .
- Fowler M. (2003). *UML Distilled: A Brief Guide to the Standard Object Modeling Language*, Addison-Wesley Professional, Third Edition, (September 19, 2003) ISBN: 0321193687
- ACM (1998). *The ACM 1998 computing classification system* available at <http://www.acm.org/class/1998/ccs98.html>