Ontology Based Interfaces to Access a Library of Virtual Hyperbooks

Gilles Falquet, Claire-Lise Mottaz-Jiang, and Jean-Claude Ziswiler

University of Geneva, CUI – Department of Information Systems
24, rue Genéral-Dufour, CH-1211 Genve 4, Switzerland
{Gilles.Falquet, Claire-Lise.Mottaz, Jean-Claude.Ziswiler}@cui.unige.ch

Abstract. A virtual hyperbook is a virtual document made of a set of information fragments linked to a domain ontology and equipped with selection and assembly methods or rules. In this paper, we study the problem of accessing and reading in a digital library of virtual hyperbooks. In this case it is necessary to generate hyperdocuments that present information and knowledge originating from several hyperbooks. Moreover, these hyper-documents must fit with the reading objectives or specific point of views of readers. Our approach is based on the integration of domain ontologies and the re-use of interface specifications.

1 Introduction

A virtual document is a set of information fragments associated with filtering, organisation and assembling mechanisms. Depending on a user profile or user intensions, these mechanisms will produce different documents adapted to the user needs. The idea of virtual document has emerged from research on pre-Web hypertext systems, such a MacWeb [20] and, more recently, on adaptive and personalized hypertext systems. Given the rapid development of theoretical and practical tools in this domain, it is reasonable to think that digital libraries will incorporate virtual documents in addition to traditional electronic documents.

It is thus interesting to explore the new accessing and reading possibilities that a library of virtual documents can provide. The main distinction between a traditional digital library and a virtual document library is the disappearance of the monolithic character of a book or an article. The ability to select and assemble informational fragments coming from various virtual documents opens new perspectives on the reading action, but it also raises important questions. In a digital library of virtual documents, a document reading system should be able to compose new documents from all the available informational fragments of the library, according to the readers objectives. For instance, a reader wishing to obtain some information about the concept of recursion should get a document containing a definition of this term, eventually other definitions that represent alternative point of views, examples and exercises drawn from various virtual documents, historical notes, etc. We can also consider that a virtual book, once inserted into a library, will automatically enrich itself by connecting to fragments of other books (new examples and exercises, new comments about several concepts, etc.)

In these two cases it is obviously necessary to check the semantic compatibility of the fragments before re-using them. The objective is to deliver to the reader new documents that are semantically coherent. For this, we propose an approach based on ontology integration and on reusability of virtual document interface specifications.

1.1 Hyperbooks and Virtual Documents

For several years, the concept of virtual document has been studied in different contexts and from different perspectives. Research on hypertexts has tackled several problem areas that are related to our study. Systems like Intermedia [14] or Storyspace [2] were developed essentially for producing hypertext literature, others like KMS [1] or MacWeb [20] have aimed at the management and sharing of knowledge. Concepts like links, anchors, composition of nodes, etc. were studied in detail. This led, among other results, to the definition of the Dexter reference model [16]. Various models and systems have also been proposed for the integration of books and electronic documents into hyperbooks. This concerns the transformation of paper books into hypertext [22] or into electronic books [19], writing directly in electronic form (hypertext) [11] or also integrating existing electronic documents [3]. The hypertext personalization problem led to the definition of models and techniques for adaptation and adaptivity [7], [4]. The capacity of adaptation corresponds to the presentation of different or differently organized contents, depending on a user profile. Adaptivity consists in automatically updating the user profile according to his or her behaviour. A well-known example of the adaptability is the change of colors of the links leading to already visited web pages. In [25], the authors propose a model of adaptive hypertext which includes a domain model, a user model and adaptation rules. The domain model is a semantic network consisting of domain concepts and relations between concepts. This model serves essentially to define adaptation rules, depending, for instance, on the concepts known or appropriated by the user. More recently, a research field has emerged that concentrates on the concept of personalizable virtual documents [6], [13]. Personalizable virtual documents are defined as sets of elements (often called fragments) associated with filtering, organization and assembling mechanisms. According to a user profile or user intensions, these mechanisms will produce different documents adapted to the user needs. For instance, in [5] Crampes and Ranwez define pedagogical virtual documents. Garlatti and Iksal [17] proposed a comprehensive and detailed model of virtual documents based on several ontologies.

There is presently no consensus on a common virtual document model. Nevertheless, most of the proposed models are comprised of (at least) a domain ontology and a fragment base. These model generally differ on the user interface part, i.e. how to specify the production of user-readable documents. Existing models use declarative languages, pedagogical or narrative ontologies, inference rules, or other mechanisms.

1.2 Ontology integration

The integration and re-usability of ontologies plays a major role in the domain of virtual books and a fortiori in the domain of virtual libraries. If we suppose that each virtual book has its own domain ontology, we need an integration technique to create a semantically coherent virtual library. It is important to note here that it is not realistic to suppose that all the virtual books will refer to the same (global) ontology, because either such an ontology does currently not exist or, even if it existed, it would contain only stable and well established concepts (thus it would not be convenient for books on new and advanced topics).

The literature about ontology integration is indeed very heterogeneous. As a starting point, we can refer to [21] and [18] for drawing a typology of the principal methods of integration. There are two major approaches to ontology integration, namely, alignment and fusion.

Alignment techniques try to bring two ontologies into mutual agreement by establishing correspondence links between the concepts of the two ontologies [18]. As a subcategory, mapping techniques intend to relate corresponding concepts or relations by an equivalence relation. In both cases, the existing ontologies will persist. This integration process is often chosen if the ontologies cover complementary domains.

The fusion of ontologies consists in creating a new coherent ontology by merging or matching concepts. This process is often quite complex because it may require, among others, the creation of new concepts in order to relate concepts from the two ontologies. It is thus very difficult to automate. Nevertheless, there exists environments and tools, like Chimaera [15], to help merging and diagnosing multiple ontologies.

In our particular case of digital libraries, we should take into account the fact that such a library can be very evolutionary. The arrival of new documents will require a constant process of integration so that the ontology remains adapted to the digital library. Thus, the fusion approach is probably not adequate for integrating hyperbooks in a digital library. Hence, the approach we propose is based on alignment and mapping.

In the rest of this paper, we first propose, in section 2, a simple model for virtual documents (virtual hyperbooks). Next, we describe in section 3 our multi-point of view approach of ontologies and the integration process of hyperbooks into digital libraries. In section 4, we will show how to use the model to define documents for global reading in an integrated library. The conclusion briefly presents the implementation techniques for realizing prototypes of digital libraries.

2 Virtual Hyperbook Model

The hyperbook model we use is comprised of a fragment repository, a domain ontology, and an interface specification, as shown in Fig. 1. The fragments and the ontology, together with their interconnecting links, form the structural part of the hyperbook. The interface specification specifies how to assemble the information fragments, with the help of the domain ontology, to produce a hypertext that constitutes the hyperbooks user interface.

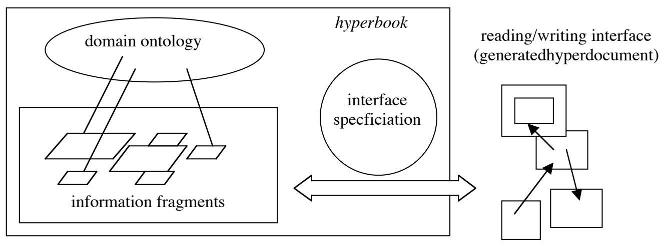

Fig. 1. Components of the virtual hyperbook model and the reading/writing interface

2.1 Structural Part of a Hyperbook

The hyperbook structure is shown in Fig. 2 as a set of classes and associations (expressed in UML). The structural part of a particular hyperbook is a set of objects that are instance of these classes. Classes *OF_Link* and *FF_Link* are associative classes that represent the links between the domain ontology and the fragments and between fragments. Links between fragments can have different natures, such as: structural links (from fragments to sub-fragments); argumentative links (arguments, positions, contradictions, ...); narrative or rhetoric links (elaboration, summary, reinforcement, ...). The domain ontology is a set of concepts connected through semantic relations that have a type and possibly restrictions (such as number restrictions). Every concept is connected to one or more terms

The domain ontology plays two roles. On one side it describes the concepts of the domain. On the other side, it serves as a reference to describe the information content of the fragments. By establishing typed links from fragments to concepts, one can qualify not only what the fragment is about but also what relationship it has with the domain concepts. Typical link types are:

- instance , example , illustration: the fragment describes a particular instance of the referred concept
- definition: the fragment contains a textual (or audio, or graphical) definition of the concept

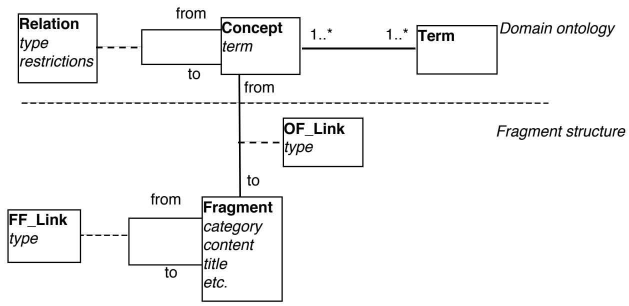

Fig. 2. Classes of the hyperbook structure

- property: the fragment describes a property of the concept
- reference, use, required: the fragment refers to the concept (it is necessary to know the concept to understand the fragment)

2.2 Interface Specification

The interface of a virtual hyperbook is a (real) hypertext, made of nodes and links, derived from the informational structure according to an interface specification. An interface specification is a set of node schemas, the instantiation of which produces the real nodes (XML documents) and links of the interface. A node schema is an expression of the form

`node` *name* [*parameters*]
    content and_link_specification
`from` *selection_expression*

A selection expression is a path expression with attribute conditions. A path expression is a sequence $E_1, \ldots, E_n$ where each E_i is a path element is of the form *class_name variable* [condition] or of the form `-(`*association_name variable* `[`condition`])->`. The evaluation of such an expression yields an n-tuple of interconnected objects that belong to the classes of the hyperbook structure (fragments, concepts, ontology-fragment links, etc.) and that satisfy conditions on their attributes. For instance, the expression

```
Concept c -(OF_Link k [type="example"])-> Fragment f
```

specifies the set of triples (c, k, f) such that c is a *Concept*, k is an *OF_Link* with k.type = *example*, f is a *Fragment*, and k.from = c and k.to = f. In other words, it selects all the concepts and fragments that are connected through an ontology-fragment link of type *example*. The path expressions can be abbreviated by omitting the associative class names when there is no ambiguity. Moreover, when the condition has the form type = *type_name*, we will simply write *type_name*. Hence, the above path expression will be written as

```
Concept c -"example"-> Fragment f
```

The content specification of a node is a list of XML elements that may contain string constants, object attributes, or expressions with string or arithmetic operators.

Example 1. The following node schema selects all the fragments that are linked to a given concept C. The content of a node instance is comprised of

- a title element that contains the main term associated to concept C
- for each fragment F connected to C through a link L: the link type and the fragment's title and content

```
node connected_fragments[C]
   <title> Fragments connected to, C.term </title> ,
   { <subtitle> L.type, : , F.title </subtitle>
     <text> F.content </text>
   }
from Concept C -(L)-> Fragment F
```

(The content specification between { and } is repeated for each selected n-tuple of objects)

The actual presentation of a node instance of this schema will be determined by XSLT or CSS style sheets.

2.3 Ontology and Link Inference

The links between the ontology and the fragments play a crucial role to establish relevant links between fragments and to generate interface documents. The idea is to replace direct linking between fragments (often called horizontal linking) by inferred links that correspond to paths starting from a fragment, going through one or more ontology concepts, and ending on another fragment. Inferred links are preferred to direct links because users (authors) are generally able to establish correctly typed links from the fragments they write to the relevant concepts. But when they are asked to link their fragments directly to other fragments they have difficulties finding relevant fragments to link to and deciding on what type of links to establish. Since the ontology has a graph structure, semantically meaningful links can be obtained by simple inference rules that consist in path expressions. For instance, Fig. 3 shows two derived links (1) and (2) obtained by going up into the domain ontology and then down to another fragment. In the next section we will present the path expression language.

Example 2. The following node schema has a selection expression that corresponds to the above-mentioned link inference path (1). An instance of this schemas will display the content of the fragment *Ex* and a list of hypertext links (`href`) to nodes (`showFragment`) that display other examples of the same concept *C*. The titles of the other examples are used as anchor text for the hyperlinks (`OtherEx.title`).

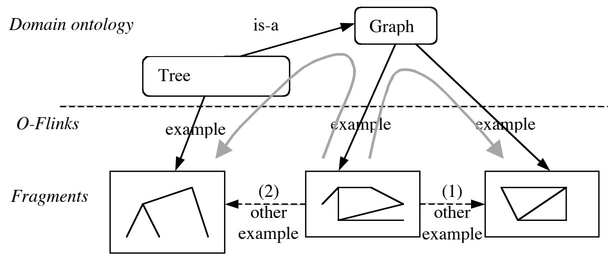

Fig. 3. Classes of the hyperbook structure

```
node exampleAndOthers[Ex]
   Ex.content ,  ,
   "Other examples of concept ", C.term ,
   {
      href showFragment[OtherEx] ( OtherEx.title )
   }
from Fragment Ex <-"example"- Concept C
     -"example"-> Fragment otherEx
```

This same link inference mechanism will be used to generate links accross hyperbooks. In addition to the standard hyperlinks shown on the previous example, the interface specification language also provides inclusion links. Inclusion links enable the interface designer to create complex contents that show several fragment contents together in a single hypertext node. Had we used inclusion in this example, we would have obtained a single node (document) showing an example together with all the other examples.

3 A Multi-Point of View Approach to Hyperbook Integration

With the objective of creating interface documents for reading not only virtual documents, but also for accessing a whole library, we could choose a very direct approach that consists of integrating all hyperbooks into one large hyperbook. In this case, we must create a global ontology from the available hyperbook ontologies. However, this approach is very limited because:

– It forces to strongly unify the concepts that does not conduct to a problem for the well established terminology of a domain, but which might be problematic when concepts have vague environments or when there remains divergent and contradictory interpretations.
– It does not reflect the fact that each book represents the point of view of an author on a subject. This diversity of point of views would be lost.

– It loses the diversity of narrative styles (reflected in the interface documents) adopted by the different authors. This is why we propose an approach based on multi-point of view ontologies [8, 9].

3.1 Concept Conflicts and Point of views

Gaines and Shaw [12] propose a methodology to compare conceptual systems of several domain experts. This method is based on analyzing domain entities (concepts), terms used to design them, and their attributes. The aim is to highlight the divergences between experts in order to facilitate the discussion to obtain a consensus. This analysis can lead to four different situations:

Consensus The experts use the same term for describe the same concept.
Correspondence Different terms are used for the same concept.
Conflict The same term is used for different concepts.
Contrast The experts identified different concepts and use different terms to name them.

In a situation of conflict the domain experts must work together to reach a consensus, i.e. to define a single concept that correspond to the term in question. In a multi-point of view approach, the resolution of conflicts is carried out differently, by considering that there can exist several concept definitions associated to the same term, provided they belong to different point of views on the domain. When integrating hyperbook ontologies we will consider that each hyperbook represents a point of view on its domain. Since the hyperbooks may belong to completely different domains, we will find the following three situations:

– Two concepts designated by the same term do not belong to the same semantic domain. For example, the concept table of the furniture ontology and the concept table of a ontology about databases.
– Two concepts effectively belong to the same domain, but they have different definitions. The two definitions represent different point of views of this concept.
– The definitions of the two concepts are considered to be equivalent. The point of views of the two hyperbooks coincide for this concept.

3.2 Ontology Integration for Virtual Documents

Since the objective of the integration process is to lead to a multi-point of view ontology and not just to a "monolithic" ontology, the most appropriated integration techniques are those which establish links between concepts (mapping of ontologies).

For this, we propose to use an extension of the technique of Rodríguez and Egenhofer [23]. There, the similarity between two concepts is the weighted sum of three measurements: similarity of the terms (set of synonyms), similarity of the attributes (set of values) and similarity of the semantic neighbourhood (set of the concepts close to the semantic links in the graph). Moreover, the similarity function takes into account the difference of depth of the concepts (relative to their respective ontologies).

In the case of virtual documents, we make use of additional information to evaluate the similarity between concepts thanks to the fragments related to each concept. If two concepts A and B are bound by links of the same type t to sets of fragments $t(A)$ and $t(B)$ respectively, the similarity between $t(A)$ and $t(B)$ can be taken into account in to compute the similarity between A and B. The similarity between $t(A)$ and $t(B)$ can be obtained with well-known document similarity measures (for instance, the cosine between the *tf-idf* vectors representing the documents in the space of terms [24] or the Kolmogorov distance). Then, we define the similarity between $t(A)$ and $t(B)$ based on the similarities between documents (for example by taking the maximum similarity found between all the fragments of $t(A)$ and $t(B)$). The similarities obtained for all types of links will then be added up to the similarity measure computed at the conceptual level. It is important to remark that link typing is crucial here. Indeed, the comparison makes sense only if the compared fragments play the same role with respect to a concepts. If, for instance, fragment a is an example of concept A whereas b is a counter-example of B, a strong similarity between a and b does not imply a strong similarity between A and B, on the contrary.

4 Generation of Interface Documents for Libraries

An interesting characteristic of the virtual hyperbook model and of the integration model is the possibility of re-using specifications of virtual interface documents to create global reading interfaces.

A first technique for building a global interface consists in re-using the specification of a hyperbook interface, but to apply it to the whole information space of the library, i.e. to the fragments and ontologies of all the hyperbooks and their interconnections through similarity links. A new, extended, version of each node schema of the interface is derived by extending its selection expression as follows:

Each element of the form

```
Concept c
```

is replaced by

```
Concept c [hybook = L] - ("sim") -> Concept c'
```

Thus every path through c can now "jump out" to another hyperbook as shown in Fig. 4.

The initial book is thus enriched with other point of views of the subject. The following node schema is the extension of the schema shown in Example 2.

```
node Extended_exampleAndOthers[Ex, threshold]
   Ex.content ,  ,
   "Other examples of concept ", C.term ,
```

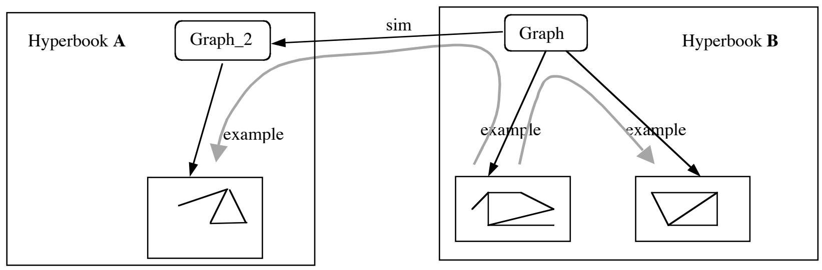

Fig. 4. Selection path to another hyperbook

```
   { href Fragment[OtherEx, R.value] ( OtherEx.title }
from Fragment Ex <-"example"- Concept C
     -(R [type="sim" and value>threshold])-> Concept C2
     -"example"-> Fragment otherEx
```

By adjusting the threshold value, the user can define the type of extension he or she desires. A very high threshold corresponds to an extension with very close point of views while a lower threshold accepts dissimilar point of views. A second way to re-use an interface specification consists of applying the interface specification of a hyperbook to another one. In this case, we will see the informational content of one hyperbook with the interface of another. If we consider that the interface of a hyperbook represents its narrative style, we obtain a vision of the content of a hyperbook in the style of another one. This kind of re-use does not require any rewriting of node schemas, but it implies that the hyperbook ontologies use the same types of relations.

It is also possible to define a completely new reading interface on the whole library. In this case, we suppose that an author wants to create a new book starting from information already existing in the library. This is a second level author, who will not create information, but invent new narrations and presentations. This task can be achieved either by creating new node schemas, or by re-using schemas of different hyperbooks. As we have already seen, each node schema can be applied to any hyperbook. As a consequence, a second level author can create new schemas that include or refer to existing schemas, without having to modify the latter.

5 Conclusion and Future Work

In this paper, we have presented a virtual hyperbook model and an ontology integration approach adapted to the reading of hyperbooks in a digital library environment. Each hyperbook has its own domain ontology and hypertext interface specification. Our approach to globally reading in a library of virtual hyperbooks is based on the idea that each hyperbook corresponds to a point of view on a domain. By applying integration techniques to the hyerbooks ontologies, we can create a multi-point of view ontology that describes a set of hyperbooks. A hypertext interface specification language can use this ontology to infer semantic relations between informational fragments and to construct new semantically and narratively coherent documents that are based on the content of several hyperbooks. Thus, a user will read something that is more than each individual book in the library. We have created various implementations of hyperbooks using techniques of hypertext views on databases. The domain ontology, the hyperbook ontology and fragments are stored in a relational database (the class diagram of Fig. 3 can be readily translated into a relational database schema). The node schemas of the interface are specified in the Lazy language, which is a declarative language to specify hypertext views on relational databases [10]. This language corresponds to the node schema specification language presented in this paper. With this technology, we have developed a hyperbook management system for courses. Every course has its own hyperbook. The reading interface provides the user with different views to help him or her grasp the meaning of concepts and see the direct or indirect interconnections between the courses concepts. We are currently using the integration techniques described here to define global reading interfaces for these hyperbooks. We plan to implement different similarity measures and compare them on hyperbooks in different domains.

References

1. Akscyn R., McCracken D., Yoder E.: KMS: a distributed hypermedia system for managing knowledge in organizations. Communications of the ACM, vol. 31, July 1988, 820–835.
2. Bernstein M.: Storyspace. Proceedings of the thirteenth Conference on Hypertext and Hypermedia, 2002, College Park, Maryland, 2002, 172–181.
3. Brusilovsky P., Rizzo R.: Map-Based Horizontal Navigation in Educational Hypertext. Proceedings of the ACM Hypertext 2002 Conference, College Park, Maryland, 2002.
4. Brusilovsky, P.: Methods and techniques of Adaptive Hypermedia. User Modelling and User-Adapted Interaction. Adaptive Hypertext and Hypermedia, Kluwer, 1998, 1–43.
5. Crampes, M., Ranwez, S.: Ontology-Supported and Ontology-Driven Conceptual Navigation on the World-Wide Web". Proceedings of the ACM Hypertext 2000 Conference, San Antonio, USA, 2000.
6. Crampes M.: User Controlled Adaptivity versus System Controlled Adaptivity in Intelligent Tutoring Systems. Proceedings of the Artificial Intelligence in Education Conference, Le Mans, France, July 1999, 173–180.
7. De Bra P., Calvi L.: Creating Adaptive Hyperdocuments for and on the Web. Proceedings of the AACE Web-Net Conference, Toronto, 1997, 149–155.
8. Falquet G., Mottaz Jiang C.-L.: A Model for the Collaborative Design of Multi-Point of View Terminological Knowledge Bases. In R. Dieng R., Matta N. (Eds) Knowledge Management and Organizational Memories, Kluwer, 2002.
9. Falquet G, Ziswiler, J.-C.: A Virtual Hyperbook Model to Support Collaborative Learning. Supplemental Proceedings of the Artificial Intelligence in Education Conference, Sydney, Australia, July 2003.

10. Falquet G., Mottaz Jiang C.-L.: A Framework to Specify Hypertext Interfaces for Ontology Engineering. Proceedings of the Knowledge Management and Organizational Memories workshop on the 18th IJCAI Conference, Acapulco, Mexico, 2003.
11. Fröhlich P., Nejdl W.: A Database-Oriented Approach to the Design of Educational Hyperbooks. Proceedings of the Workshop on Intelligent Educational Systems on World Wide Web, 8th World Conference of the AIED Society, Kobe, Japan, 1997. 18–22.
12. Gaines, M., Shaw, B.: Comparing Conceptual Structures: Consensus, Conflict, Correspondence and Contrast. Knowledge Science Institute, University of Calgary, 1989.
13. Garlatti, S., Iksal, S.: Revisiting and Versioning in Virtual Special Reports. Proceedings of the Hypertext 2001 conference, Arhus, Denmark, 2001.
14. Garret L. N., Smith K. E., Meyrowitz N.: Intermedia: Issues Strategies and Tactics in the Design of a Hypermedia Document System Proceedings of the Computer Supported Collaborative Work Conference, Austin, Texas, 1986.
15. McGuinness D. L., Fikes R. Rice J., Wilder S.: An Environment for Merging and Testing Large Ontologies. Proceedings of the Seventh International Conference on Principles of Knowledge Representation and Reasoning (KR2000), Breckenridge, Colorado, April 2000.
16. Halasz F., Schwartz M.: The Dexter hypertext reference model. Communications of the ACM, vol. 37, no 2, 1994, 30–39.
17. Iksal, S., Garlatti, S., Tanguy P., Ganier F.: Semantic Composition of Special Report on the Web: A Cognitive Approach. Proceedings of the H2PTM01 conference, Valenciennes, France, 2001.
18. Klein M.: Combining and relating ontologies: an analysis of problems and solutions. Proccedings of the IJCAI2001 workshop on Ontologies and Information Sharing, Seattle, 2001.
19. Landoni M., Crestani F., Melucci M.: The Visual Book and the Hyper-Textbook: Two Electronic Books One Lesson? Proceedings RIAO Conference, 2000, 247–265.
20. Nanard J., Nanard M.: Should Anchors be Typed Too? An Experiment with MacWeb. Proceedings of ACM Hypertext 93 Conference, 1993, 51–62.
21. Pinto, H. S., Gmez-Prez, A., Martins, J. P.: Some Issues on Ontology Integration. Proceedings of the IJCAI-99 workshop on Ontologies and Problem-Solving Methods (KRR5), Stockholm, Sweden, 1999.
22. Rada R.: Hypertext writing and document reuse: the role of a semantic net. Electronic Publishing, vol. 3, no. 3, 1990, 125–140.
23. Rodríguez M. A., Egenhofer.M. J.: Determining Semantic Similarity among Entity Class from Different Ontologies. IEEE Transactions on Knowledge and Data Engineering, Vol. 15, No. 2, March/April 2003.
24. Salton, G.: *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer*. Addison-Wesley, 1989.
25. Wu H., de Kort E., De Bra P.: Design Issues for General-Propose Adaptive Hypermedia Systems. Proceedings of the ACM Hypertext 2001 Conference, Aarhus, 2001, 141–150.