Hyperbooks

Gilles Falquet[1], Luka Nerima[1], and Jean-Claude Ziswiler[1]

Centre universitaire d'informatique
University of Geneva
7, route de Drize, 1227 Carouge, Switzerland.
falquet, nerima, ziswiler@cui.unige.ch

Summary. The first part of the chapter presents a synthesis of recent works in the domain of hyperbooks and introduces a general hyperbook model. In this model, a hyperbook is made of a knowledge structure, a set of informational fragment, links between the fragments and the knowledge structure, and a user interface specification. This specification is used to generate the actual reading interface which is a hypertext whose nodes and links are derived from the knowledge and fragment structures. The knowledge structure provides a mean to interconnect different hyperbooks in a semantically consistent way, so as to create digital libraries of hyperbooks.

The second part explains in more details the knowledge structure alignment process that is at the heart of the semantic interconnection of hyperbooks. The presentation is based on a real-world example, in the domain of agriculture. It also provides experimental results about the performance, in terms of precision and recall, of this process.

1 Introduction

Hyperbook, or hypertext book, is a term that is commonly used to refer to a hypertext that has some of the characteristics of a printed book. In particular, a hyperbook is generally organized as a set of elements that are grouped together to form larger entities such as chapters or sections. Moreover, the content of a hyperbook should be autonomous and have a clearly identified topic or objective (in this sense, an encyclopedia is not a (hyper)book).

There is presently no consensus about a common hyperbook model. Nevertheless, most of the models proposed in the literature and actually implemented are comprised of a first order hypertext [14], which corresponds to the book content and the logical orgnization of its entities, and a second order structure that represent knowledge about the book's domain. This structure can range from a simple network of concepts, with semantic links, to a formal domain ontology. These two levels already existed in early hypertext systems such as KMS [1] and MacWeb [11]. In addition, some models and systems provide a way to specify the generation of user interface documents. In this case, the hyperbook is a kind of virtual, or potential, document, in the sense of [15] and its user interface is made of actual documents generated from the hyperbook's contents.

In the rest of this introduction we briefly present a list of systems and models that are typical of the hyperbook or virtual document approach.

The Woven Electronic Book Systems (WEBSs) [13] is an electronic book management system for the creation and organiszation of documents (texts, figures, logic-mathematical models, indexes, etc.). The documents, or blocks within documents, can be interconnected through a network of semantic links. It is also possible to create hierarchical structures (similar to table of contents), called browser documents. The WEBSs has a powerful scripting language used to define new user interface components such as browsers or indexes.

The InterBook project [3] presents an e-learning platform that includes two basic models: the domain model and the student model. The domain model is a network of domain concepts. The student model describes the student knowledge as well as the student learning goals, both expressed in terms of the concepts of the domain model. The system uses this model to adaptively generate the content the student can access at a certain point, depending on his or her knowledge and goals. At a hight leve, several books can be integrated in a bookshelf. The interconnection of several books is realized by a shared domain model. In [18], the authors propose a similar 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 understood by the user.

The KBS hyperbook system [12] is also dedicated to e-learning, with a constructivist approach. It makes use of the O-Telos modelling language to create a rich semantic description of the book's domain. An interesting feature of this system is the possibility to create different semantic abstractions over the same information units in order to represent different viewpoints.

Crampes and Ranwez [4] propose two models of virtual documents. Both of them use domain ontologies for indexing informational fragments (the resources). In the first case, a "conceptual backward chaining" strategy can construct reading paths corresponding to the user objectives (described in terms of conceptual graphs). In the second case, a pedagogical ontology defines teaching rules, which guide the assembling of fragments to produce documents with respect to a predefined pedagogical approach. These rules determine the order of appearance of the different types of learning material in the documents. An inference engine generates documents that satisfy these rules.

Garlatti, Iksal and colleagues [9][8] propose a comprehensive and detailed model of virtual documents. It is based on four ontologies for modeling the domain, the metadata, the application and the user. These ontologies allow a fully declarative approach of document composition.

In the e-learning context, we can find similar approaches for instance in [17]. The authors present an ontology-based hyperbook model where the modeling of relations is of particular interest. Links between concepts in the domain ontology are not represented directly in the interface documents, but generated implicitly around concepts of interest. The idea is to avoid users (mostly students) getting confused by seeing too many of the top-level concepts of the ontology. The Learning Object Model (LOM) is fully integrated into the system so that people using other learning systems can reuse their tutoring and student models. They have to adapt only the knowledge base by working out the concepts of the ontology. The authors also provide an example ontology about concepts in computer sciences.

Bocconi [2] describes a hypertext generation system to automatically select and compose scholarly hypermedia. The presented content is generated through a domain ontology containing the concepts and their relations and a discourse ontology containing different roles and narrative units describing different genres. The discourse ontology holds a very detailed and highly formalized description of the points of interest that a user can have about a domain.

In the next section we will present a hyperbook model that synthesizes and generalizes the main concepts introduced in the above-mentioned approaches. Then, in section 3, we will show how to extend this model to create semantic digital libraries of hyperbooks. Section 4 details the automated hyperbooks' alignment process and provides a concrete example together with experimental results.

2 A Conceptual Model of Hyperbooks

The hyperbook model we present here is comprised of a fragment repository, a domain ontology, and an interface specification (see Figure 1). The fragments and the ontology, together with their interconnecting links, form the structural part of the hyperbook while the interface specification is intended to dynamically generate the actual documents and hyperlinks that form the reading interface. This model has been introduced in [7] and it can be seen as a synthesis of the above mentioned approaches.

2.1 The Hyperbook Structure

The basic informational contents of the hyperbook are made of reusable fragments, which can be texts, images, sounds, mathematical formulae, etc.. Fragments can be connected by structural links, for instance from fragments to sub-fragments, to form compound fragments. These typed links indicate the roles played by the different fragments in the compound fragment. For instance an exercise could be made up of a question fragment, one or more answer fragments, and a discussion. Compound fragments can have different purposes,

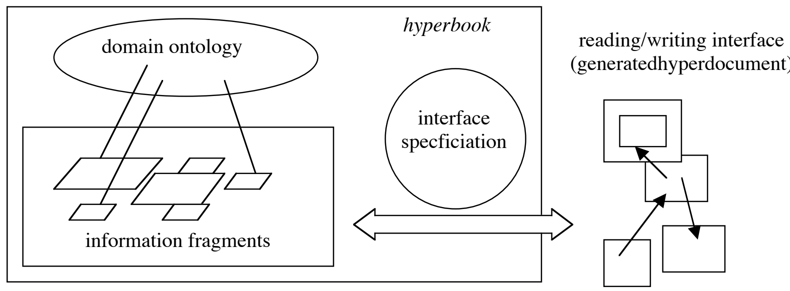

Fig. 1. Components of the virtual hyperbook model

they can represent pedagogical or narrative/rhetoric units (exercise, elaboration, summary, reinforcement, etc.), argumentative units (an issue related to positions, arguments, contradictions, etc.), or even hyperbook management units (group discussions or weblogs). For instance, a discussion structure can be made up of topic and message fragments connected through about and reply-to links. In fact, this structure corresponds to what Rada [14] calls the first order hypertext, it also roughly corresponds to the kind of structure that is supported by markup languages such as XML (considering not only the hierarchic decomposition in elements but also “transverse” links).

The domain ontology of a hyperbook is intended to hold a formal representation of the domain's concepts. Since the hyperbook authors are not supposed to be knowledge engineers or ontologists, the ontology model must be kept simple. For this reason, the ontology is a directed graph whose nodes represent concepts and whose labelled links represent semantic relations. Among these relations, the is a relation (or generalization/specialisation) plays a particular role since it provides a taxonomic structure of concepts. This relation must form a directed acyclic subgraph of the ontology.

The connection between the ontology and the fragments is provided by semantic annotation links. These links not only index each fragment with one or many concepts but they also indicate the role played by a fragment with respect to a concept. 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
- property: The fragment describes a property of the concept
- reference, use: the fragment refers to the concept (it is necessary to know the concept to understand the fragment)

Other link types may exists in specific contexts. For instance, a scientific hypertextbook may have links such as theorem, exercise, algorithm, historical note, etc. Figure 2 presents an example of a hyperbook ontoloy Gray boxes indicate concepts that are connected to fragments.

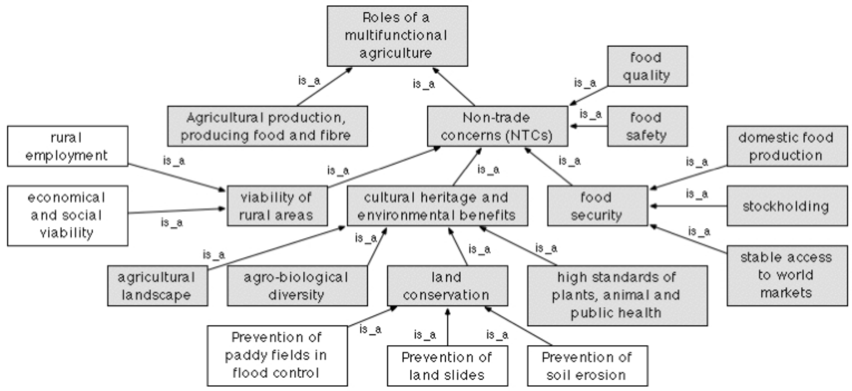

Fig. 2. The domain ontology of the hyperbook about multifunctional agriculture

In the previous section we have seen that the ontological or domain knowledge level may have different purposes such as providing a semantic index of the hyperbook, guiding the production of pedagogical documents, adapting the displayed content to the user, etc. In this section we will emphasize the role of the ontology in the inference of semantically relevant links between fragments and between interface documents; in the following section we will show that hyperbook ontologies play a central role to integrate different hyperbooks into a digital library.

2.2 The Model-based Hyperbook Interface

Following the virtual document approach, we define the user interface of a hyperbook as a navigable hypertext, made of documents and hyperlinks that are derived from the hyperbook structure. The aim is to present the informational content of the hyperbook to the user and to help him or her read, understand, and write the hyperbook's content in different ways and under multiple perspectives.

To reach this aim, the interface documents must be composed by selecting and assembling several fragments that make up semantically coherent units of presentation. Given the richness of the static hyperbook model, it is impossible to design a single "optimal" reading and writing interface. This is why the interface model is designed to support the specification of various views on the hyperbook content, thus enabling the hyperbook designer to adapt the interface to each particular hyperbook [5]. In particular, it is necessary to adapt the hyperbook structure to different pedagogical styles or to specific practice or rules of a domain.

An interface specification consists of a set of hypertext node schemas that will be instantiated on demand to produce the actual interface documents [7]. Hence, the interface nodes (the documents the user sees) are instances of node schemas. A node schema is comprised of

- a selection expression: what objects (fragments/concepts) to select
- a content description: how to organize the selected objects within the nodes
- a link description: what kind of links to generate to which nodes

The link model is richer than what exists in hypertext systems such as the Word-Wide Web. In addition to the "jump" links (the usual Web links), it is possible to specify *inclusion* and *expand-in-place* links. Inclusion links are intended to build complex hierarchical nodes by including the content of other (sub)nodes. Expand-in-place nodes enable the user to dynamically create a document by including the content of selected nodes in the current node. Figure 3 shows an interface document of the above-presented hyperbook about multifunctional agriculture. It is an instance of the concept[C] node schema, shown (in abbreviated form) in figure 4, with C set to Cultural heritage and environmental benefits. It includes, among others, an instance of the concept_definitions node schema that displays the contents of the fragments connected to C through a definition link.

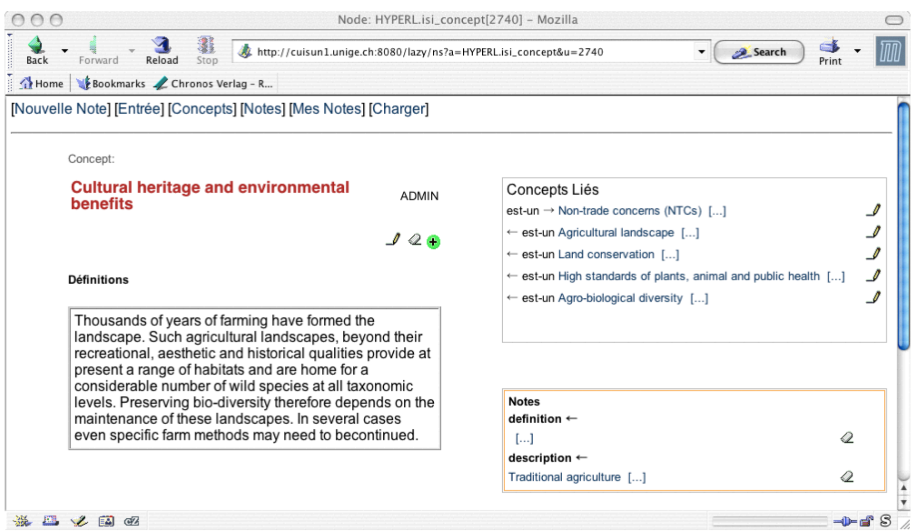

Fig. 3. A user interface node of the hyperbook about multifunctional agriculture

```
node concept[C]
{
  <left-column>(
      C.content ,  /* display the concept name */
      include concept_definitions[C]  /* and all its definitions */
  </left-column>
  <right-colum>
      include related_concepts[C] ,
      include related_notes[C]
   <right-column>
}
from C

node concept_definitions[C]
{
   F.content
}
from  C (definition)> F
/* select all the fragments connected to C
   throug a "definition" link */

... other node schemas ...
```

Fig. 4. Node schemas for producing the node shown in figure 3

2.3 Link Inference

Inferred links correspond to paths starting from a fragment, going through one or more concepts in the ontology, and ending on another fragment. Inferred links are preferred to direct links because authors are generally able to establish correctly typed links from the fragments they write to the relevant concepts, whereas, 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 [6].

Since the hyperbook ontology has a graph structure, an interesting property of the model is that semantically meaningful links can be obtained by simple inference rules that consist of path expressions. If we consider the global labeled graph formed by the domain ontology, the fragment collection, and the concept of fragment links, a path expression is an alternated sequence of nodes and arc specifications. A node specification is composed of a node type (concept or fragment), a category name (for fragments) or a term (for concepts). An arc specification is composed of a link type, a traversal direction. In addition, each node and arc can be associated to a variable. An instance of a path expression is a path in the hyperbook graph that satisfies all the specifications of the path expression. Figure 5 shows an extract of the domain ontology about political sciences, links (1) are instance of the above-described path expression.

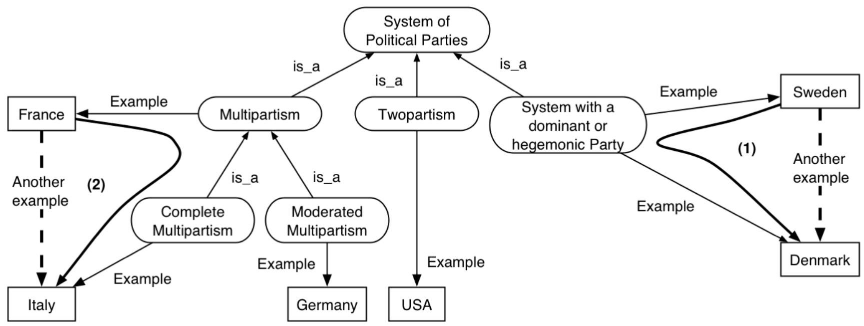

Fig. 5. Link inference through the domain ontology (rectangles are fragments and rounded rectangles are concepts)

Depending on the link types and fragment categories of the hyperbook, it will be possible to define link inference paths that have a precise and useful meaning for the reader. The following expressions show examples of link inferences that typically occur in a hyperbook.

```
F1 (Another Example)> F2 :-
F1 <(Example) C1 <(is-a*) C2 (Example)> F2.
```

Generates a link, with type Another Example from fragment F1 to fragment F2 if F1 is an example of a concept C1 , and C1 has a sub-concept C2 , which has an example F2 The <(is-a*) notation represents the traversal of zero, one or more is-a taxonomic links in the generic to specific direction.

```
F1 (has property)> F2 :-
F1 (uses)> C (is-a*)> D (property)> F2
```

If fragment F1 refers to concept C , create a link, with type has property to every fragment F2 that describe properties of a concept D that is more generic than C . For instance, If F1 is an exercise, this will link it to all properties of the concepts required by the exercise.

An interesting property of this link inference method is its robustness with respect to the hyperbook's evolution. Since the domain ontology is usually more stable than the hyperbook's fragments, a link to a concept will probably have a longer lifetime than a link to a fragment. Moreover, inferred links are, by definition, always up to date.

3 Semantic Digital Libraries of Hyperbooks

In this section we will explore the construction and development of digital libraries made of hyperbooks instead of traditional electronic documents (such as PDF or HTML files). The main distinction between a traditional digital library and a virtual document library is the disappearance of the monolithic nature of a book or an article.

In a digital library of hyperbooks, 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. We can also consider that a hyperbook, once inserted into a library, will automatically enrich itself by connecting to fragments of other books, For instance, a hyperbook that has fragments related to a concept C can be augmented by finding fragments about an equivalent concept C' in another hyperbook. This implies, of course, that there exists a mean to find a concept that is "considered as equivalent" to C in another hyperbook ontology.

3.1 Establishing Links between Hyperbooks

Once several hyperbooks have been (partly) written, the characteristics of the hyperbook model allow aligning the different hyperbooks by semantic links (Figure 6). The links are realized in a way similar to the semantic relations we have inside a hyperbook ontology. The link can be defined by hand with

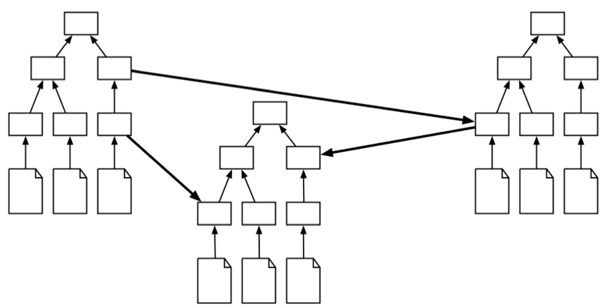

Fig. 6. A semantic digital library made of hyperbooks and semantic links between hyperbook concepts

the advantage that a specific type can be added. As this might become a time consuming task in a bigger environment, we introduce the possibility to establish similarity links in an (semi-) automatic way by applying known ontology alignment methods to our hyperbook model. The technical details are described in the next section that comes also with a sight to experiences we conducted during the last years. The final result might be several connected hyperbooks that are linked by different types of link: Some of them might be set up by hand and can be considered as precise semantic relations between concepts of the hyperbooks' ontologies. Others might be generated just on the fly in the interface level if a user wants to browse through the content of different hyperbooks. These links typically might be generated automatically and don't have a type other than "similar". Finally, we can imagine to establish link verification by social navigation, or that the automatic generation process helps user to detect links that they finally establishes by hand and by giving them a more specific type.

The so created structure is not a fully integrated hyperbook, but a semantic digital library that takes the form of a network of semantically inter-connected hyperbooks. As mainly the user interface plays an important role on how the user will browse in the digital library and how she or he considers the semantic relations between hyperbooks, we particularly consider the re-using of hyperbook interface specification in the following.

3.2 Reusing Interface Specifications and Creation of New Global Books

Another 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 inks. If we consider that a hyperbook represents a point-of-view (semantic or narrative), we will obtain a vision of the whole library according to the point-of-view of this hyperbook. In other words, we extend a hyperbook with the help of the others. The most direct manner to extend a hyperbook consists in using the similar concepts found in the other hyperbooks and links issued from these concepts. For example, the path

```
F<(example)  C -(example)> G
```

becomes

```
F<(example) C -(similar_to)> C' (example)> G
```

A fragment f that contains an example for concept C will be connected to a fragment G if G is an example of a concept C′ that is similar to C (the set of examples of a hyperbook is extended thanks to the example found in other hyperbooks). In the same way, it is also possible to make present other definitions of concept c.

It should be noted that the effectiveness of this approach depends on the quality of ontology integration; This means that we must find the same link types and fragment categories in these ontologies. This last problem, although non-trivial, is nevertheless simpler than the integration of domain ontologies because the number of concerned concepts is quite limited. Another way of re-using an interface specification consists in applying this specification to another hyperbook. In this case, we will see the informational content of one hyperbook with the interface of another. It is the dynamic part of the narration of a book that is applied to another content. This kind of re-use does not require any rewriting of interface node schemas, but it implies that the hyperbook ontologies have been well integrated.

Another case concerns the creation of new "global" books. We suppose that an author wants to create a new book starting from information already existing in the digital 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 interface node schemas, or by re-using schemas of different hyperbooks. As we have already seen, each interface 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.

4 Automatic Hyperbook Integration

This section discusses how different hyperbooks can be aligned to build a digital library. Concretely, our aim is to establish semantic links between concepts of different hyperbooks. We explain in the following how we compute different semantic similarity values between concepts and how we decide which of the calculated similarity values have the right quality to indicate candidates for semantic relations between concepts.

4.1 Determining Semantic Similarity

An approach to determine semantic similarity, to which our problem is close, is the one of Rodrguez and Egenhofer [16]. Their aim is to calculate semantic similarity between concepts of heterogeneous and disconnected ontologies. In their approach, calculating semantic similarity between concepts means calculating semantic similarity between "entity classes" through three different basic components: a set of synonym words (synsets) that denotes the entity class, a set of semantic interrelations among these entity classes, and a set of distinguishing features that characterizes entity classes. In order to determine the similarity between entity classes, they first define similarity functions for synonym sets, semantic neighbourhoods and distinguishing features (parts, functions and attributes). Then, a matching process over the calculated similarities establishes semantic similarity links between the entity classes (concepts).

We are concerned with small-scale domain ontologies where concepts are described by their relationships and by textual fragments, so we can't apply Rodrguez and Egenhofer's algorithm as such. They apply their algorithm to larger top-level ontologies like WordNet or SDTS. In our application, an author will not define several synonyms of a term and all their semantic relationships and features. Determining similarity between different synonym sets is becoming a comparison between two single terms (Word matching WM).

Rodrguez and Egenhofer outline that the more complete and detailed the entity classes' representation, the better the algorithm will work. In the hyperbook structure, we have a rich semantic representation of the concepts, but in a less formal way. Thus, we propose to substitute feature matching that indicates common and different characteristics between concepts with fragment matching (FragM). Instead of comparing concepts' features, we compare their textual fragments.

Only the third similarity measure, which compares concepts and their associated features/fragments in the semantic neighbourhood (NeighM), can be applied as in the original method.

In order to calculate the three mentioned similarities, we have to determine the distance from the concepts to the immediate super-class that subsumes them, or in other words, their common least upper bound. In the case of independent ontologies, this means to connect them by making each of their roots a direct descendant of an imaginary and more general root "anything". The path distance $\alpha(a, b)$ is determined as

$$\alpha(a,b) = \begin{cases} \frac{depth(a)}{depth(a)+depth(b)} & \text{if } depth(a) \leq depth(b) \\ \\ 1 - \frac{depth(a)}{depth(a)+depth(b)} & \text{if } depth(a) > depth(b) \end{cases} \tag{1}$$

where $depth(a)$ and $depth(b)$ stand for the shortest path from concept a (resp. b) to the imaginary root "anything".

The basic similarity function is based on set theory involving both common and different characteristics of concepts:

$$S(a,b) = \frac{|A \cap B|}{|A \cap B| + \alpha(a,b)|A - B| + (1 - \alpha(a,b))|B - A|} \tag{2}$$

where, in the case of WM, A and B stand for the sets of words found in the term designating concept a (resp. b) of the respective ontologies (after stopword filtering). In the case of FragM, A (resp.) B stands for the terms found in all the fragments linked to concept a (resp. b) in the corresponding hyperbooks (after stopword filtering). NeighM applies these two similarity measures recursively to all concepts within a preliminary defined radius.

To compute the final similarity value $S_{final}(a, b)$ between two entity classes (concepts), Rodrguez and Egenhofer weight the three matching values.

$$S_{final}(a,b) = w_1 S_{WM}(a,b) + w_2 S_{FragM}(a,b) + w_3 S_{NeighM}(a,b) \tag{3}$$

with $w_1 + w_2 + w_3 = 1$.

The difficulty is to indicate how to assign the weights w_i for WM, FragM, and NeighM. We will discuss different settings in the next section where we present our experimental results with the matching algorithm.

4.2 An Example from the World of Agriculture

We illustrate the outcomes of the algorithm with the above-presented hyperbook about multifunctional agriculture (HMA). The aim is to establish semantic similarity links towards a second hyperbook about agriculture politics of the World Trade Organization WTO (HWTO). The matching algorithm compares each concept of HMA (36 concepts) with each concept of HWTO (54 concepts). In this way, we established in total 1944 comparisons (36×54). Out of them, we manually identified 79 reference comparisons that imply a real similarity relation between two concepts.

We discuss now the influence of the different matching components by analyzing different settings of the above-presented algorithm. Figure 7 shows the interpolated precision and recall values for the different settings. Interpolation of the precision on recall i is defined as the maximum precision for all recall equal or higher than i. The first setting, we consider is WM only, or in other

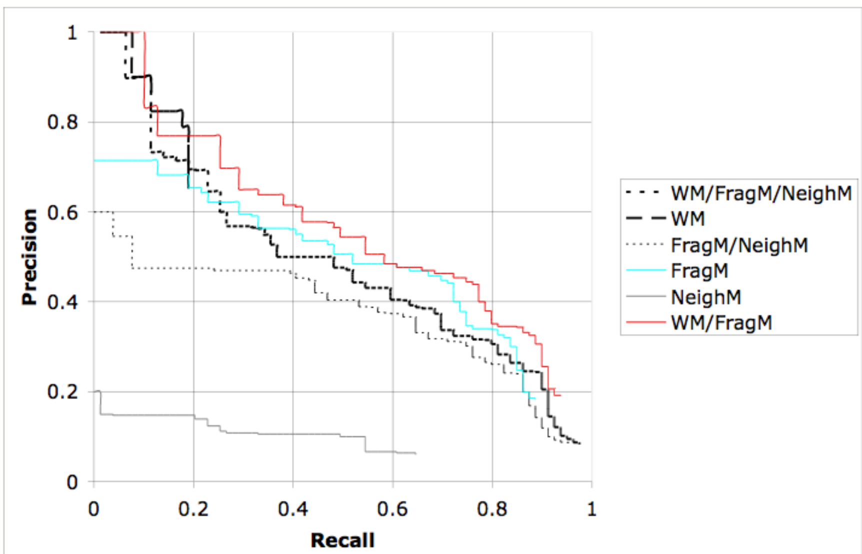

Fig. 7. Interpolated precision and recall values for different settings

words: $1.0 \times WM + 0.0 \times FragM + 0.0 \times NeighM$. Usually, two problems are enumerated if using isolated word matching. The first point concerns polysemy. This happens when a word has different meanings, for instance *bank*, which can denote either a financial institution or a sloping land (a riverbank). In our example, we didn't find a lot of polysemous terms, mostly because we work at the level of domain ontologies and we are not dealing with cross-domain or top-level ontologies where experts might use the same terms to express different concepts. This results in high precision for WM. The second problem concerns synonyms. Synonyms are different words or word sequences that denote the same or very similar concept. Consider for instance *stadium* and *sports arena*. A comparison function based only on the syntactical analysis of words cannot detect similarities well enough, which results in low recall. In our example, WM cannot reach more than 20% recall.

Other problems appear if only adjectives match. We found for instance a high WM value (0.43) between the concepts *Agricultural Training Services* and *Agricultural Landscape*. Stopword filtering is in general known as an effective help to reduce matching between functional words, which do not carry meaning. We apply stopword filtering, but in a domain-specific environment, we are concerned with expressions that are functional and noisy in the range of the domain, but not in general. Using commonly known stopword filters does not resolve this problem. We conclude that even though there is no word ambiguity problem, word matching alone isn't sufficient to evaluate concept similarity.

To provide evidence that our function runs well, we tested WM with different comparison functions. On of the most popular is the Levenshtein distance, which calculates the minimal number of operations needed to transform one string into another, where operation is an insertion, deletion, or substitution of a single characters. This statistical approach is an alternative way to the identification of word stems. In the example, we calculated the distance with equivalent costs to insert, delete or substitute a single character, but found similar results to the applied comparison of words.

Next, we discuss settings that concentrate on FragM or on NeighM especially with the aim to increase recall. We consider FragM only, NeighM only and FragM in conjunction with NeighM ($0.0 \times WM + 0.5 \times FragM + 0.5 \times NeighM$). We preliminary has to mention that not all concepts are illustrated with fragments, only 18 concepts of HMA come with fragments respectively 31 concepts of HWTO. Especially at the leave level of the hyperbook ontology, the hyperbook authors did not detail all concepts through fragments.

As we expected, we can observe in Figure 7 that recall goes up at the price that precision decreases. Further, we detect that the precision of WM is reinforced. If WM gives unexpectedly high values, FragM and NeighM sorts a low value. In the above-mentioned comparison between *Agricultural training service* and *Agricultural landscape*, we found 0.43 for WM while the corresponding values for FragM and NeighM are considerably lower, at 0.044 and 0.036.

We also discovered that FragM enhances results much better than NeighM. This can be explained by the fact that the structure of the hyperbook ontologies is relatively weak and too small to obtain good results with NeighM. NeighM runs better if the ontologies have a certain number of concepts that forms a well-described environment of a concept.

Due to the fact that the structure of the hyperbook ontologies are weak and NeighM results in low values for precision and recall, we finally proceed to the most promising setting we found for this example: WM in conjunction with FragM or $0.5 \times WM + 0.5 \times FragM + 0.0 \times NeighM$. Experiments show that considering all three components is not necessarily the best way to find similarity between two ontology-based hyperbooks. This observation is consistent with the results of the Ontology Alignment Evaluation Campaign (OAEI held in 2006. In [10], the authors of one of the best alignment methods (RIMON) explain that their algorithm first compares a linguistic based and a structure based alignment technique and then decides which one will be dominant in the final alignment. They also mention that in some cases, one of the two methods alone can be better than the combination of both of them.

Table 1 shows the highest scores for WM in conjunction with FragM. Rows in gray indicate relations that we do not consider as real similarities. Table 1.

Table 1. Results with WM and FragM

Source concept	**Target concept**	**WM+FM**
Public stockholding programmes for food security purposes	Food security	0.473
Domestic food aid	Domestic food production	0.453
Domestic food aid	Food safety	0.411
Domestic food aid	Food security	0.374
Public stockholding programmes for food security purposes	Stockholding	0.374
Domestic food aid	Food quality	0.363
Public stockholding programmes for food security purposes	Food quality	0.362
Public stockholding programmes for food security purposes	Food safety	0.344
Public stockholding programmes for food security purposes	High standards of plants, animal and public health	0.308
Domestic support in agriculture	Domestic food production	0.267
Public stockholding programmes for food security purposes	Domestic food production	0.241

The example shows that it is difficult to maintain high precision. We preserve 100% precision only up to 10% of recall. After, we quickly fall down to 75% of precision at 20% of recall, and to 65% of precision for 40% of recall. For instance, 20% of recall means losing 4 of 5 relations. If a concept in hyperbook A has 5 relations to concepts of hyperbook B, we find only one. And if a concept of hyperbook A is similar to only one concept of hyperbook B, the chance to find this relation is only 20%. The situation looks better when we focus on recall. For 75% of recall, we still have 40% of precision. Concretely, this means that a user will find 75% of the similar concepts and almost half of the propositions indicate really a correct similarity relation. Analyzing precision and recall, we see that this setting is a compromise of the above-discussed situations. Word matching increases precision, while FragM increases the recall of the comparison

4.3 Using the similarity links in an interactive interface

The right boundary between precision and recall depends on the application and on the user. If we consider for instance users who use the hyperbook much in the sense of a glossary or a lexicon, they would probably prefer a high recall or in other words as much generated links as possible. Initially considering one concept and exploring others in its surrounding thanks to generated similarity links is an argument for higher recall. On the other side, if we consider users who want to explore a whole book, we must move towards higher precision to avoid cognitive overload. A clearly structured content with selected similarity links might be much better than a proliferation of links that lead to less relevant information..

One way to solve this issue consists in letting the user set the similarity threshold (for instance with a graphical slider in the user interface). When the user considers the number of generated similarity links as too high (or too low) or the quality as too low, he or she can raise the threshold and interactively see the effect on the hyperbook interface.

Another argument for a user-determined setting of precision and recall is the problem of the graphical representation of the similarity relations in a hyperbook. We compare precision and recall settings in hyperbooks with settings in Information Retrieval (IR). Search engines graphically present the results in lists, ordered by the similarity value and often presented in pages with ten or twenty hits. The first page contains the hits with the highest score, or in other words, the results with the highest precision. The more the user browses through the following pages, the more recall will rise and precision decrease. Considering the graphical user interface of a hyperbook, it is obvious that we can't reproduce a similar list, even less if we think that such a list might spread over several pages.

Figure 8 presents a user interface that includes generated similarity links (on the right). When clicked, these links expand to a list of related fragments of the corresponding hyperbook (figure 9, top right). The fragments contents are shown (bottom right) when their name is clicked.

5 Conclusion

In this chapter we have presented a semantic hyperbook model that integrates concepts developed over these last years in the domain of hyperbooks and virtual documents. We have particularly emphasized the user-interface aspect of the model to show that a specification-based approach, relying on node schemas and link inference, is well suited to easily generate reading interfaces that are well adapted to the author and reader aims. In addition, this approach

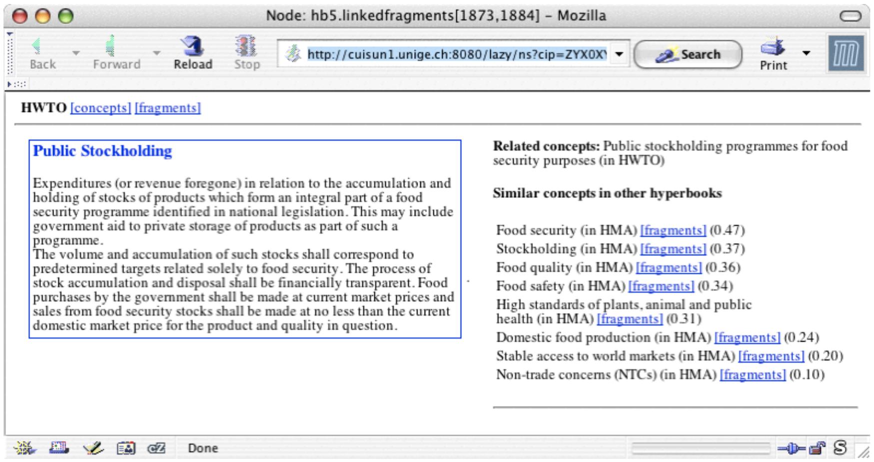

Fig. 8. A graphical interface with generated similarity links

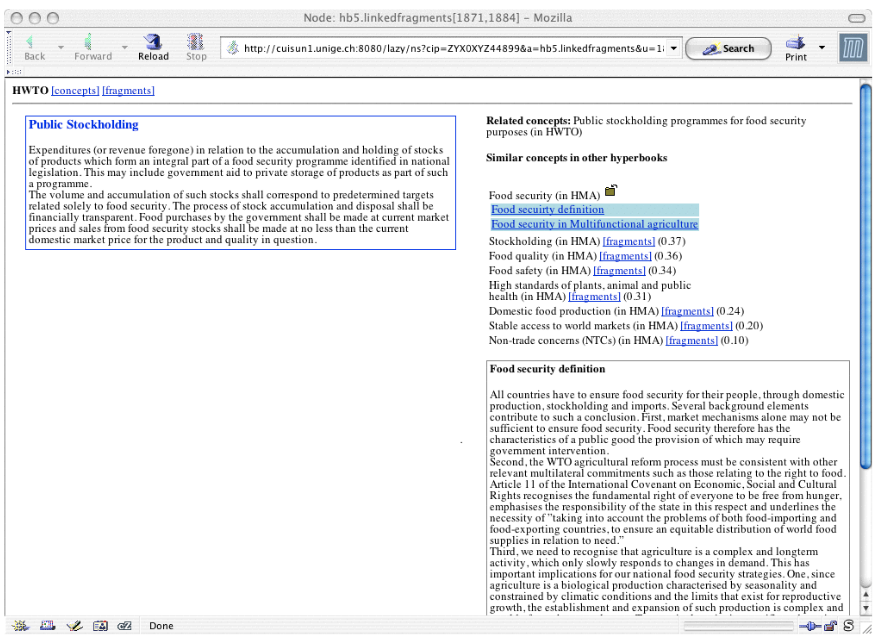

Fig. 9. An opened link with the fragments of the related concept *Food security*

readily handles the generation of "global books" within a digital library of integrated hyperbooks.

In order to create semantic digital libraries of hyperbooks we propose to take advantage of the knowledge structure (the domain ontology) of each hyperbook to integrate them in a semantically consistent way. Thus, a digital library of hyperbooks is not a mere collection of hyperbooks, it provides a semantic interconnection structure among the hyperbooks.

This approach is realistic because, as we have shown, the integration process can be automated, even when the hyperbook ontologies are not fully formalized. This process is based on known ontology alignment techniques that we have extended to the specific case of hyperbooks and tested on a real-world example.

References

1. R. Akscyn, D. McCracken, and E. Yoder. Kms: a distributed hypermedia system for managing knowledge in organizations. *Communications of the ACM*, 31:820–835, 1988.
2. S. Bocconi. Automatic presentation generation for scholarly hypermedia. In *Proc. of the 1st International Workshop on Scholarly Hypertext at the Hypertext 2003 conference*, Nottingham, U.K., 2003.
3. P. Brusilovsky, J. Eklund, and E. Schwarz. Web-based education for all: A tool for developing adaptive courseware. *Computer Networks and ISDN Systems*, 30:291–300, 1998.
4. M. Crampes and S. Ranwez. Ontology-supported and ontology-driven conceptual navigation on the world-wide web. In *Proc. ACM Hypertext 2000 Conference, San Antonio, USA*, 2000.
5. G. Falquet, L. Nerima, and J. Guyot. Languages and tools to specify hypertext views on databases. In P. Atzeni, A. Mendelzon, and G. Mecca, editors, *Porc. of The World Wide Web and Databases*, volume 1590 of *LNCS*, Berlin Heidelberg NY, 1999. Springer.
6. G. Falquet, L. Nerima, and J.-C. Ziswiler. Utilisation des ontologies dans les environnements d'écriture collaborative d'hyperlivres, expériences et perspectives. *Revue STICEF*, 11:333–350, 2004.
7. G. Falquet and J.-C. Ziswiler. A virtual hyperbooks model to support collaborative learning. *AACE International Journal on E-Learning (IJEL)*, 4:39–56, 2005.
8. S. Iksal and S. Garlatti. Spécification declarative pour les documents virtuels personalisables. In Serge Garlatti and Michel Crampes, editors, *Actes du congrès Documents virtuels personalisables (DVP2002)*, 2002.
9. S. Iksal, P. Tanguy, and F. Ganier. Semantic composition of special report on the web: A cognitive approach. In *Proc. H2PTM'01 Conf.*, Valenciennes, France, 2001.
10. Y. Li, J. Li, D. Zhang, and J. Tang. Result of ontology alignment with rimom at oaei'06. In *Proc. of the International Workshop on Ontology Matching (OM-2006), Nov. 5, 2006, Athens, Georgia, USA*, 1998.

11. M. Nanard and J. Nanard. Should anchors be typed too? an experiment with macweb. In *Proc. of the ACM Hypertext '93 Conference*, 1993.
12. W Nejdl and M. Wolpers. Kbs hyperbook - a data-driven information system on the web. In *WWW8 Conference, Toronto*, 1999.
13. J. Pasquier-Boltuck and G. Collaud. An object-oriented approach to conceptualizing and programming an interactive system for the creation and consultation of electronic book. In J. Pasquier-Boltuck, G. Collaud, J. Monnard, R. Ingold, and Armangil D., editors, *Electronic Books and their Tools. The WEBS Prototype and the OSCAR Project*, University of Fribourg Series in Computer Science, pages 23–39, Bern, 1992. Peter Lang European Academic Publishers.
14. R. Rada. Hypertext writing and document reuse: the role of a semantic net. *Elec- tronic Publishing*, 3(125–140), 1990.
15. S. Ranwez and M. Crampes. Conceptual documents and hypertext documents are two different forms of virtual document. In *Proc. Virtual Documents Hypertext Functionality and the Web workshop of the 8th Intl. World Wide Web Conference*, Toronto, Canada, 1999.
16. M. A. Rodrguez and M. J. Egenhofer. Determining semantic similarity among entity class from different ontologies. *IEEE Transactions on Knowledge and Data Engineering*, 15, 2003.
17. M. Ronchett and S. Paramjeet. Knowledge management in an e-learning system. In *Proc. of the 4th IEEE International Conference on Advanced Learning Technologies (ICALT2004)*, Joensuu, Finland, 2004.
18. H. Wu, E. de Kort, and P. De Bra. Design issues for general-purpose adaptive hypermedia systems. In *Proceedings of the ACM Hypertext 2001 Conference*, pages 141–150, Århus, Denmark, 2001.