

Revisiting and Versioning in Virtual Special Reports

Sébastien Iksal and Serge Garlatti

Department of Artificial Intelligence and Cognitive Sciences
ENST Bretagne
Technopôle de Brest Iroise
B.P. 832, 29285 Brest Cedex
FRANCE
Tel. : (+33) 2 29 00 14 28
Fax : (+33) 2 29 00 10 30
{sebastien.iksal, serge.garlatti}@enst-bretagne.fr

Abstract. Adaptation/personalization is one of the main issues for web applications and require large repositories. Creating adaptive web applications from these repositories requires to have methods to facilitate web application creation and management and to ensure reuse, sharing and exchange of data through the internet/intranet. Virtual documents deal with these issues. In our framework, we are interested in adaptive virtual documents for author-oriented web applications providing several reading strategies to readers. These applications have the following characteristics: authors have know-how which enables them to choose document contents and to organize them in one or more consistent ways. A reading strategy and the corresponding content are semantically coherent and convey a particular meaning to the readers. Such author's know-how can be represented at knowledge level and then be used for generating web documents dynamically, for ensuring reader comprehension and for sharing and reuse. Then an adaptive virtual document can be computed on the fly by means of a semantic composition engine using: i) an overall document structure – for instance a narrative structure - representing a reading strategy for which node contents are linked at run time, according to user's needs for adaptation, ii) an intelligent search engine and semantic metadata relying on semantic web initiative, and iv) a user model. In this paper, we focus on a semantic composition engine enabling us to compute on the fly adaptive/personalized web documents in the ICCARS project. Its main goal is to assist the journalist in building adaptive special reports. In such a framework, adaptation, personalization and reusability are central issues for delivering adaptive special reports.

Keywords : Semantic Composition Engine, Adaptation/Personalization, Semantic Web, Ontology, Virtual documents, Revisiting, Versioning

Introduction

Numerous applications are available on the Web today and their size and volume are increasing. For instance, portals, e-learning, problem solving systems, decision support systems, digital libraries, on-line information systems, virtual museums, e-business and digital newspapers are current applications. On the World Wide Web, we have large distributed information repositories which convey large amounts of knowledge for internet users as well as for companies which are the owners of this knowledge. Adaptation/personalization is one of the main issues for web applications. Adaptive web applications have the ability to deal with different users' needs for enhancing usability and comprehension and for dealing with large repositories. Indeed, adaptive web applications - also often called Adaptive Hypermedia Systems - can provide different kinds of information, different layouts, different navigation tools according to users' needs [1]. Creating adaptive web applications from these repositories requires the following features: i) methods to facilitate web application creation and management and ii) reuse, sharing and exchange of data through the internet/intranet.

The notion of flexible hypermedia and more particularly of virtual documents can lead to methods facilitating web application design and maintenance. According to Watters, "A virtual document is a document for which no persistent state exists and for which some or all each instance is generated at run time" [2]. Virtual documents have grown out of a need for interactivity and individualization of documents, particularly on the web. Virtual document and adaptive hypermedia are closely related – they can be viewed as the two faces of the same coin. Reuse, sharing and exchange through the internet/intranet require to have a precise search engine. Indeed, it is well known that keyword-based information access presents severe limitations concerning precision and recall. On the contrary, intelligent search engines, relying on semantic web initiative [3] and semantic metadata, overcome these limitations [4, 5].

In our framework, we are interested in adaptive virtual documents for author-oriented web applications providing several reading strategies to readers. These applications have the following characteristics: authors have know-how which enables them to choose document contents and to organize them in one or more consistent ways – author reading strategies. Content and organizations are "semantically" related to ensure reader's comprehension. In this paper, we focus on organizations called narrative structure. The reader has the ability to recognize – sometimes unconsciously – these structures. For instance, scientific papers, courseware, report, special report in journalism, etc., have each of them a distinct narrative structure. At present, the narrative structure is implicit in printed document, but also in digital one. Such author's know-how and skills can be represented at knowledge level and then be shared and reused among authors, used for generating web documents dynamically and for enhancing reader comprehension. A narrative structure provides an overall document structure which is a declarative description of web documents. Then, a web document can be computed on the fly by means of a semantic composition engine using: i) an overall document structure – for instance a narrative structure - representing a reading strategy for which node contents are substituted at run time, according to user's needs for adaptation, ii) an intelligent

search engine, iii) semantic metadata, and iv) a user model. An authoring tool is provided for creating narrative structures, specifying their content and associating metadata.

Numerous web sites offer tools to users for personalizing their information space and the presentation. Two projects aims at supplying personalized news on a Web site. Sistemi Telematici Adattativi [6] is a project which aims at filtering and displaying news and advertisement according to users' preferences and characteristics. The system selects a relevant set of news according to the reader's interests via probabilities. Then, some rules are applied for choosing the relevant presentation for a single news (abstract or full text, image or video, etc.). Personalization is done by a filtering process on news and a presentation selection. KMI Planet [7] is a kind of private on-line newspaper where all readers and writers are in a same group – university. It collects news through e-mail, processes them and sends the result to the most interested readers. The tool is able to order articles for filling in gaps, and after to inform the reader when the news is available. It supplies with an advanced interface for searching documents. Each news is annotated via an academic ontology, and then the query interface uses the same ontology for writing queries. The system uses the annotations and all queries given by a user to find out the most relevant news. Personalization mainly consists of a filtering process based on user queries and ontology annotations. In this two projects, personalization is mainly based on a filtering process using user's preferences for selecting the most relevant news.

Our project aims at delivering adapted special reports to news readers. They consist of a set of news – selected by authors - and several semantic structures organizing them. These semantic structures provide different reading strategies to the readers. A reading strategy and the corresponding news collection are semantically coherent and convey a particular meaning to the readers. This meaning can be viewed as a viewpoint on this collection. Some reading strategies are created by authors and others by the system. The reading strategies and their content may be adapted to users' needs. According to C. Watters, revisiting, versioning and reusability are some of the main issues for virtual documents. In this paper, we focus on a semantic composition engine enabling us to compute on the fly adaptive/personalized special reports in the ICCARS project. Its main goal is to assist the journalist in building adaptive special reports. In such a framework, adaptation, personalization and reusability are central issues for delivering adapted/personalized special reports.

Firstly, we present the context of our approach: journalism and reporting on the web via the ICCARS Project. Next, we will give a summary of the architecture of our semantic composition engine. We will present how we will manage revisiting and versioning of dynamic documents according to our context and our composition engine. Then, our adaptation policies is presented. Finally, some directions for the future will be proposed.

ICCARS Project

ICCARS is the acronym for INTEGRATED AND COOPERATIVE COMPUTER ASSISTED REPORTING SYSTEM. It is a joined project between the IASC Laboratory, a SME called Atlantide and a regional daily newspaper called Le Télégramme. It is funded by Brittany Regional Council. The ICCARS prototype will be a computer assisted reporting system. Its main goal is to assist the journalist in creating adaptive special reports. These documents are able to include audio and video material, links, and they are no longer limited in size.

Due to internet features, numerous web sites are offering news. Then, It is not sufficient to filter news. We need to go beyond a news delivery service and to provide new services around information. Special reports seem to be the most representative journalists' task. A special report offers news as well as analysis, debate, synthesis and/or development. It can be viewed as an organized collection of articles offering a viewpoint on events. It is a matter of journalist know-how for creating such type of document.

The digital special report

A special report is a synthesis made by one or more journalists on a particular topic, for instance a yachting race. We consider a special report as a collection of articles with a given narrative structure. In a paper version, there is a single organization which appears through the sequence of pages and the page layout. A digital special report may naturally offer different narrative structures.

New features

Digital special reports can provide new services to the reader. We present some of them: such as various reading strategies for a single special report, enrichment, reusability and adaptation. It is important to notice that as in the printed version, the journalist chooses the set of articles belonging to a special report. That is to say, in a digital special report we can find only the articles that the journalist wants to provide his readers with.

The notion of reading strategy

In a special report, a set of articles can be read in different ways, according to a reader's or author's viewpoints. We call a particular sequence of articles a reading strategy. We distinguish two kinds of reading strategies:

1. An author strategy: This is a narrative structure designed by a writer for presenting a particular angle on a set of articles. One of the main roles of journalists is to analyse events and report them in a consistent and synthetic way. A narrative structure is composed of nodes and semantic relationships. Nodes are spans of texts. Relationships belong to those analyzed by Rhetorical Structure Theory (RST) [8]. RST defines relations between spans of text, each span have a role inside the relation (nucleus and satellite). Each relation is defined by some constraints on the nucleus, the satellite, the

combination of the nucleus and the satellite, and an effect to the reader. Among these relations, we can find are antithesis, restatement, summary, interpretation and so on.

2. A reader strategy: This is an overall document structure computed from a reader's goals. For instance, it can be based on geographic, history or topic criteria – a domain model - organizing the access to articles. The structure delivered is computed on the fly and controlled by the computer according to a generic structure. Nevertheless, journalists are aware of such structures because they associate metadata with articles and special reports for these services.

Special report views

In a digital document, three different views coexist: semantic, logical and layout/presentation [9]. For each view we have a specific structure. The semantic structure of a document conveys the organization of the meaning of the content of a document. This view fits the semantic level of the semantic web architecture. Indeed, it can be represented by ontologies. Ontologies are used to model types of fragment as well as their relationships. For instance, “The fragment A which is an *interview* is the volitional cause of the fragment B which is an *analysis*”, the underlying relationship cannot be represented by a syntactic structure [10]. Interview and analysis are types of fragment. The interview is the satellite and the analysis is the nucleus of this rhetorical relation. This relation is oriented and encode a particular reading guide. In this case, the fragment B will be better understood if the fragment A is read before. It could be interesting to show the type of relation to the reader as explanations or for increasing the comprehension.

The logical structure reflects the syntactic organization of a document. A document (for example books and magazines) can be broken down into components (chapters and articles). These can also be broken down into components (titles, paragraphs, figures and so forth). It turns out that just about every document can be viewed this way. The logical view fits the syntactic level of the semantic web architecture. A logical structure can be encoded in XML [11]. The layout/presentation view describes how the documents appear on a device and a physical structure describes it, (eg. the size and colour of headings, texts, etc). The layout/presentation view may be processed by an XSLT processor [12] for transforming an XML document into an HTML document that can be viewed by any web browser. It can also be processed by a java engine able to compute an XML document for presenting by a web browser.

In a printed document, these three views are intertwined and are not separable. There is no straightforward mapping between the semantic and the logical structure, that is to say, for instance, a paragraph does not correspond to a particular content's meaning. On the other hand, the logical and physical structure are closely related. Indeed, the physical structure encodes the logical structure. For instance, each section element has a particular presentation – font, size, colour, etc. The semantic structure is implicit and so it can be analyzed and/or recognized by a reader. Moreover, it is a key issue for reader comprehension. In a digital document, these three views may be represented and managed.

Reusability

It is very important for journalists to be able to reuse at least an article or a part of a special report in more than one special report and why not in the same report. Indeed, an article and a part of a special report generally concern more than one topic. Watters [2] as well as the Semantic Web Community argues that allowing reusability of fragments – articles, special report parts - leads to associate metadata with fragments. Moreover, narrative structures can also be reused for new special reports.

Enrichment

A special report can be updated. The journalist may add a new article in his base and modify the organization in order to insert this new article. A journalist can organize a subset of articles in order to develop various viewpoints (economic, ecological, for example.). Enrichment is a very important possibility with digital documents, but in order to not disturb the reader, the system must be able to rebuild the same special report, that is to say, the same version of the report. Enrichment leads to the management of versioning in special reports.

Adaptation/Personalization

The digital special report as a particular type of web site is a good candidate for adaptation. As a digital document may be managed at three different levels: semantic, syntactic and layout/presentation, adaptation can take place on each level. For instance, the content and the overall document structure may be adapted to the reader's preferences, knowledge and goals. At a syntactic level, different logical structures may be chosen to fit user needs. At a layout level, the presentation may be adapted to the current device and/or the user stereotype.

Moreover, we have to deal with versioning purposes due to enrichment purposes. Then, personalization involves annotating the documents already read, or those which have been added since the last visit.

Virtual Document for Special Reports

A journalist organizes a set of articles according to one or more reading strategies, but it is necessary to prepare the special report which is relevant to a particular reader and a specific device. First of all, we give a definition of a virtual document in our framework:

- An adaptive/personalized virtual document consists of a set of information fragments, ontologies and a semantic composition engine which is able to select the relevant information fragments, to assemble and to organize them according to an author's strategy or user's goals by adapting various visible aspects of the document delivered to the user.

Fragments can be atomic or abstract information units. The latter are composed of atomic and abstract information units. In ICCARS, fragments are articles – atomic - , special reports and sub-reports – abstract. A special report and corresponding reading strategies are modelled as follows in figure 1.

In order to facilitate comprehension, we prefix all the elements of the special report model with the I of ICCARS. An I-SubReport is composed of a set of articles selected by the author – explicitly associated with it in order to define its relevant information space -, and one or more narrative structures – Reading strategies - between these elements. When a journalist considers that a particular I-SubReport is relevant enough to be a special report, an I-PublishedReport is created and gives a user access to this I-SubReport, that is to say, a document ready to be delivered to readers.

An I-SubReport can be organized according to one or more I-Structures – Reading strategies. An I-Structure is a collection of I-Components among which one is the root of the I-Structure. An I-Component is an abstract object, which exists only inside a particular I-Structure. An I-Component is linked to others through a semantic relation belonging to those of RST. This relationship gives the organization of the I-Structure. That is to say, each I-Component in the structure which is the source of a relationship, is a nucleus in RST and the corresponding destination (an I-Component also) is a satellite. So, we use RST as a basis to build a narrative structure in which nodes are different categories of fragments. If RST is very far from the journalists viewpoint, we will use journalists’ relations in the future. The set composed of I-Components and the relationships is one narrative structure of the special report. An I-Component is a kind of information retrieval service which uses a description given by the author according to metadata, in order to send a query to the intelligent information broker. It is able to use the user model to filter the small set of answers. So the I-Component is able to deliver the most relevant articles or sub-reports. Then, the I-SubReport is a graph where the nodes are I-Components and the vertices are relations between I-Components. Several special reports can be generalized to provide special report templates. So, reusability concerns these templates as well as all the instances of the special report model. For instance, in the case of the wreck of the Erika, the structure can be reused for other wrecks – super tanker oil slick - (Tanyo, Amoco Cadiz, etc.).

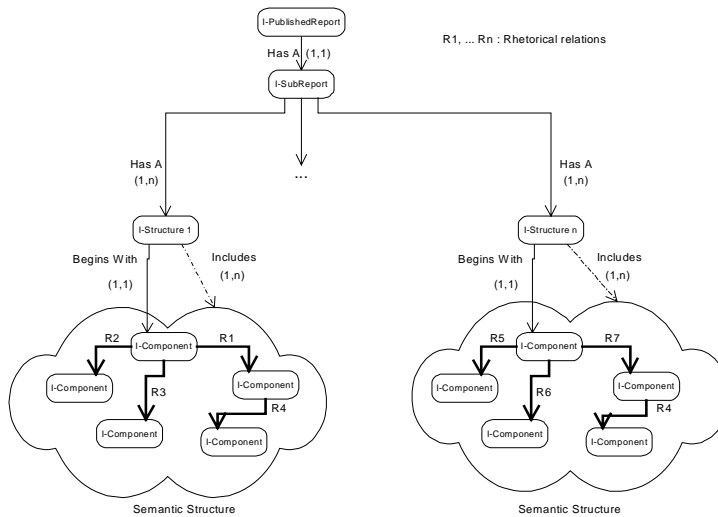


Fig. 1. Model for a special report

This special report model is an input for the semantic composition engine which computes an adaptive/personalized special report for a given reader.

Semantic Composition Engine Architecture

Our semantic composition engine relies on OntoBroker for ontology management and intelligent search engine. OntoBroker is a knowledge management engine which is useful for filtering and information retrieval in a large amount of data as well as in the model specification – ontologies [5, 13, 14]. OntoBroker contains four ontologies and facts closely related to them. These ontologies are: a domain ontology for representing contents, a metadata ontology at the information level which describes the indexing structure of fragments, a user ontology which may define different stereotypes and individual features and a special report ontology which represents the author’s competences and know-how for creating special reports. The domain ontology defines a shared vocabulary used in the metadata schema for the content description of data. It will also be used by the semantic composition engine as an overall document structure, by the user as an information retrieval tool because the user often has difficulty in defining his/her interests, and it is easier for him/her to recognize required information in a domain model than to specify it.

According to the three views of a document, our semantic composition engine architecture is described below (cf. fig. 2).

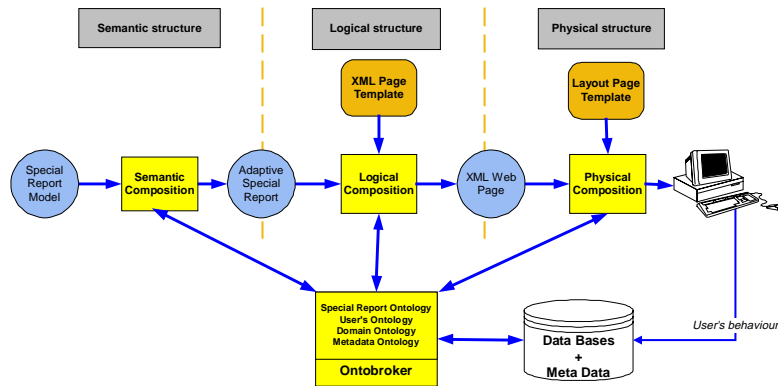


Fig. 2. The Semantic Composition Engine Architecture

One of the main ideas behind the notion of semantic composition engine is to declare as much as possible all the user’s tasks and interactions. The semantic composition engine is composed of three different stages: a semantic composition which manages the semantic structure of a special report model for defining a user adapted special report and selects its contents, a logical composition which computes an XML web page from the user adapted special report and a physical composition

which computes the current web page layout from the XML structure¹. This architecture is based on two different studies: ICCARS Project and CANDLE Project (Collaborative and Network Distributed Learning Environment) which is an european project.

Semantic Composition: for an author's reading strategy, the main role of the semantic composition is to define the special report content and to adapt the chosen I-Structure to user needs. Indeed, each I-Component has only a specification of the content. From this specification, one or more fragments may be selected from the relevant set of articles associated to the considered special report. Indeed, only a subset of metadata entries are used for content specification by the authoring tool. The others are used for defining variants of fragments - according to adaptation policies. The special report structure may also be adapted: an I-Fragment or an I-SubReport may be hidden according to reader's needs, that is to say topics, its professional activity and/or expertise level. Articles may be dedicated to specific professional activities and/or to some levels of knowledge. The semantic composition produces a user adaptive special report – user document - in which all I-components have contents according to their specification and to the user model.

Logical Composition: the logical composition aims at computing an XML page with a content and navigation tools for accessing the different fragments of a special report, by means of a template. A web page, represented as an XML structure [11], is generated from a particular template. A template describes the logical structure of a web page but without any content or navigation tools. It has queries for computing navigation tools and for loading the content via OntoBroker. The content is given by the current node in the narrative structure. For accessing the other nodes, the logical composition engine has to browse the narrative structure. The logical composition has also to associate properties to hyperlinks – Xlinks - for managing annotation, hiding, sorting and direct guidance. All interactions between a user and the web page are also represented in this template in order to respond to users' requests. The database will provide several templates which will be indexed with user's tasks, type of fragments, user's category, user's level of expertise, etc. Then, several templates can be available for a given type of content. By means of these templates, adaptation may take place at this stage.

Physical Composition: finally, the physical composition has to map some presentation rules on the web page. The final process of this architecture is concerned by the design of the web pages of a special report. The final layout of each page may be tailored to the user's preferences: print size, color, and so on and/or use standardized styles from corporate, SMEs or institution style sheets. The physical composition has also to manage the adaptive navigation. From author specification or user stereotypes or user preferences, he has to hide, to annotate, etc; the different types of hyperlinks in a web page. There is a style sheet for each template. It is one way for mapping a presentation on a web page. Indeed, a java process can be applied

¹ A logical structure is associated with one web page and not with the entire document – it cannot be relevant in such a framework.

to an XML structure to provide a web page.

The above description has presented the semantic composition engine architecture and the process for delivering a special report to a user. Now, we present the management of revisiting and versioning in a virtual special report.

Revisiting and Versioning

Computing a document on the fly is very interesting because it is cheaper to produce a single virtual document which can be the source of numerous real documents according to a user's requests than to prepare all these documents in advance. Nevertheless, a real document is ephemeral. Readers have an expectation that documents found once will be available on a subsequent search. In fact, they expect to retrieve the same document in the same state. So the system needs enough information to recreate the document as it was. It is called revisiting purposes which may lead to bookmark an I-Component. In fact, there is no URL to store because of the dynamic generation of the document, so we need to recreate the document and to display the same I-Component.

Thanks to the enrichment capabilities of virtual special reports, it is necessary to manage versioning [15]. Version control is a central issue for special report management, readers need to retrieve a former version of the document, may be because they have already read it or because they want to be aware of the life cycle of this report. They also need to go forward and backward in time through changes in order to develop their own analysis of the situation.

Revisiting and Versioning are very closed together but these are two different issues. Even if we often intertwine them. The main difference between them concerns the domain, providing revisiting features requires the storage in a user model of all the data enabling the semantic composition engine to compute the same real special report again. On the other hand, the versioning process is related to the metadata schema and the special report ontology, that is to say it works directly on documents, structures and so on. According to versioning purposes, revisiting requires to have in the user model the version number of the last visited special report.

Versioning management

Versioning management is related to enrichment features and to the metadata schema. The author will be able to add or remove all the elements of the special report model – I-SubReport, I-Structure, I-Component - and also the relationship between I-Components. We provide, by default, the latest version of the element except if the user asks for a particular version.

The metadata schema (Table 1) provides metadata information about fragments. The semantic composition engine uses the schema for information retrieval. It

matches content specification against metadata. The current version of the metadata schema consists of six parts:

MD.1	General	General information about the resource	Unique Instance
MD.1.1	Identifier	Unique Identifier	Single Value
MD.1.2	Title	Title given by the author	Single Value
MD.1.3	Authors	One or more authors for the resource	Unsorted List
MD.1.4	Description	Short description of the resource	Single Value
MD.1.5	Language	Language of the resource	Single Value
MD.2	Life cycle	Entries for versioning purposes	Unique Instance
MD.2.1	Version	Version number of the resource	Single Value
MD.2.2	Status	Status of the resource (draft, final ...)	Single Value
MD.2.3	Authors	Authors of the version	Unsorted List
MD.2.4	Date	Date of the version	Single Value
MD.3	Meta Metadata	Information about metadata	Unique Instance
MD.3.1	Creator	Author of metadata	Single Value
MD.3.2	Validator	Reviewer of metadata	Single Value
MD.3.3	Language	Language of metadata	Single Value
MD.3.4	Date	Date of metadata	Single Value
MD.4	Technical	Technical information about the resource	Unique Instance
MD.4.1	Location	Where the resource can be found?	Single Value
MD.4.2	Format	Format of the resource	Single Instance
MD.4.2.1	Type	Type of the resource (ppt, doc, html, ...)	Single Value
MD.4.2.2	Size	Size of the resource in Kbytes	Single Value
MD.4.3	Requirements	What is required to read the resource?	Single Instance
MD.4.3.1	Software	Software names and version	Unsorted List
MD.4.3.2	Hardware	Hardware description	Unsorted List
MD.5	Classification	Data about the content and reporting features	Unique Instance
MD.5.1	Domain	Description related to the domain	Unsorted List
MD.5.1.1	Concept	Concept name	Single Value
MD.5.1.2	Level	Level of knowledge required	Single Value
MD.5.2	Reporting	Reporting classification	Single Instance
MD.5.2.1	Resource Type	Type of resource (Interview, report ...)	Single Value
MD.5.2.2	Edition	Edition concerned	Single Value
MD.5.2.3	City	City concerned	Single Value
MD.6	Rights	Use conditions of the resource	Unique Instance
MD.6.1	Copyrights	Copyrights or licenses	Single Value
MD.6.2	Access	Access restrictions	Single Value
MD.6.3	Cost	Cost needed	Single Value
MD.6.4	Publisher	Publisher identity	Single Value
MD.6.5	Remarks	Usage remarks	Single Value

Table 1. The metadata schema

Now, according to the special report model and the metadata schema, we manage versioning features as follows :

- The I-PublishedReport does not have versioning purposes, because if the author changes the main I-SubReport, we create a new I-PublishedReport.
- An I-Component: This exists only inside an I-Structure and it works as a small information retrieval service. We do not have versioning for this kind of element, because the internal information retrieval features are not updated. If needed, a new I-Component is created. But relationships between I-Components can change, so we add a validity period (cf. fig. 3,

R_4 ($m-n$) to relations. This property corresponds to the list of the I-Structure's version in which this relation is valid.

- The I-Structure is composed of a set of I-Components and a link to the first I-Component which is the root of the I-Structure. Two changes can be applied to an I-Structure. First, the root can be replaced, and next the set of I-Components can be updated. An I-Component can only be added. Indeed, an I-Component may be not relevant on a new version, but has to be present for a previous one. For each new version we have to increase the set of I-Components and to give the validity period of the root (cf. fig. 3 ($k-n$)).
- The I-SubReport is composed of a set of I-Structures and a data collection (in our case, this is a list of articles selected by the author). Between two versions of I-SubReport, these two sets can be updated (inserts). For each new version we store the new sets.

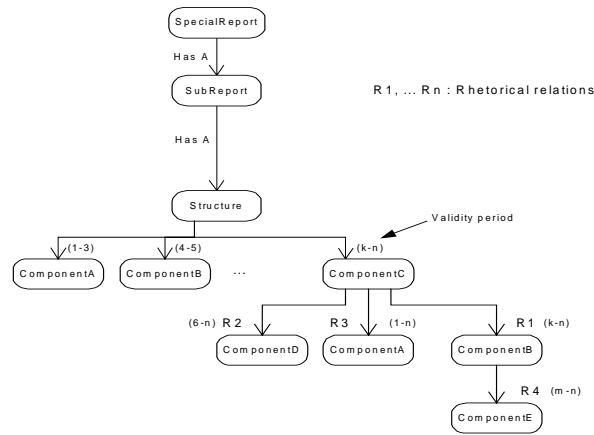


Fig. 3. Versioning application on an instance of the special report model

For each new version, we instantiate a new element of the special report model. We associate the metadata information related to the “Lifecycle part”. This will enable us to use information retrieval methods to find the relevant version. Instead of managing versions via numbers like many versioning systems, we manage version via the date. In Press Institutions, especially daily local newspapers, several editions are printed every day according to different geographical areas. It is therefore more relevant for a reader to ask for the special report of a particular date, than the third version of the special report. According to this versioning management, retrieving the correct version is in fact a filtering process.

History and revisiting

As well as the metadata schema useful for versioning features, the user model is necessary for history and revisiting issues. The current version of the user model (Table 2) is an individual model because it deals with individual features, unlike

stereotype models which are about user class features. The user model consists of five parts: personal, preferences, knowledge, history and session.

UM.1	Personal data	Personal data concerning the user	Unique Instance
UM.1.1	Identity	His identity	Single Instance
UM.1.1.1	Last name	His last name	Single Value
UM.1.1.2	First name	His first name	Single Value
UM.1.1.3	Age	His age	Single Value
UM.1.2	Login	Unique identifying data	Single Instance
UM.1.2.1	Login name	The login name	Single Value
UM.1.2.2	Password	The corresponding password	Single Value
UM.1.3	Classification	Classification data	Single Instance
UM.1.3.1	Location	Where does he live?	Single Value
UM.1.3.2	Professional Activity	What kind of job? (economist, fisherman, student, etc.)	Single Value
UM.1.3.3	Role	The role in the application (author, reader ...)	Single Value
UM.2	Preferences data	Data about the preference of the user	Unique Instance
UM.2.1	Interest	Topics of interests	Single Instance
UM.2.1.2	Topic	A list of topics	Unsorted List
UM.2.2	Adaptation	Adaptation preferences	Unsorted List
UM.2.2.1	Element	An element which can be adapted (link ...)	Single Value
UM.2.2.2	Rule	A method of adaptation (annotation ...)	Single Value
UM.3	Knowledge	Data about the knowledge of the user	Unique Instance
UM.3.1	Domain	Knowledge about the domain	Unsorted List
UM.3.1.1	Element	A domain concept	Single Value
UM.3.1.2	Level	A level of knowledge	Single Value
UM.4	History data	Data about access to special reports	Unique Instance
UM.4.1	Access	Access log	Unsorted List
UM.4.1.1	Report ID	Which special report have been accessed	Single Value
UM.4.1.2	Structure ID	Structures used	Single Value
UM.4.1.3	Component ID	Component read	Single Value
UM.4.1.4	Date	Date of access	Single Value
UM.4.2	Bookmark	Bookmark data	Unsorted List
UM.4.2.1	Report ID	Identifier of the special report	Single Value
UM.4.2.2	Structure ID	Identifier of the structure in the special report	Single Value
UM.4.2.3	Component ID	Identifier of the component in the structure	Single Value
UM.4.2.4	Date	Date of the storage, for retrieving the relevant version	Single Value
UM.5	Session data	Data concerning the current session	Unique Instance
UM.5.1	Choice	Stores the first choices of the user	Single Instance
UM.5.1.1	Special Report ID	The Special Report chosen	Single Value
UM.5.1.2	Structure ID	The structure chosen	Single Value
UM.5.2	Current	Stores current data about the reading path	Single Instance
UM.5.2.1	Special Report ID	The current Special Report	Single Value
UM.5.2.2	Structure ID	The current structure	Single Value
UM.5.2.3	Fragment ID	The current fragment	Single Value
UM.5.3	Device	Description of the current device	Single Instance
UM.5.3.1	Software	Software names and version	Unsorted List
UM.5.3.2	Hardware	Hardware description	Unsorted List

Table 2. The user model

Comments: Personal: The geographical area is relevant for a local daily newspaper, and the professional activity (economist, fisherman, student, etc.) will be used to

provide some stereotypes of needs in the future; **Knowledge:** The estimated level of knowledge about some domain concepts. At present, it is given by the reader; **History:** It stores all features required to retrieve the last special report(s). Then, the semantic composition engine will be able to deal with revisiting and versioning. We plan to add two new entries about readers' Behavior and Background.

We have to store all data needed to recreate a web page. According to our special report model, these data are: the I-PublishedReport ID, the I-Structure ID, the I-Component ID and the date for retrieving the relevant version. The I-SubReport ID is not necessary because we can obtain it via the date and the I-PublishedReport ID. Concerning the versioning issue, the I-PublishedReport ID and the date chosen by the user are sufficient to recreate the same special report. Then we will store the user's bookmark in the appropriate section of his individual model. Concerning history purposes, these are necessary for two reasons. First, the system has to propose navigational guides to the reader. In this case, it has to show the path covered by the reader. Next, it is interesting to be able to show, on another visit, the element already visited or elements newly added since the last version. In the first case, we have to store the path exactly, that is to say, an ordered list of elements where something can be found more than one time. In the second case, we just need a list of elements without duplicates, but with the date of the last visit. This date is very useful in order to re-open a document in the same configuration the user left it. We will use in the user model, the history section, and the current session.

Adaptation

Our semantic composition architecture is composed of three engines. Each one is able to offer different types of adaptation to readers.

1. **Semantic Composition:** at this level, the engine adapts the special report structure and the content to the user. By means of the authoring tool, the author is able to select the I-Component content. The authoring tool chooses some metadata entries for specifying this content, at least the classification section (5.1.1 concept name, 5.2.1 resource type) implicitly. Moreover, it has to ensure the consistency of the special report. In other words, an article may be referred in several I-Components according to the author, but not elsewhere. An I-Component may have several contents which are variants due to knowledge level, technical requirements or versioning. As soon as a new article is included in a special report, the authoring tool has to check if another specification matches this article. In this case, more metadata entries has to be added in the specification. We shall have to learn what are the relevant implicit metadata entries in order to help the authors. The composition engine can match the following user model entries: topics of interests and knowledge against the classification entries in metadata. If there are not compatible with, the I-Component is deleted from the I-Structure. The resulting structure with the corresponding content becomes the user adapted special report.
2. **Logical Composition:** the logical engine may select the relevant template according to the resource type, the user task – for instance reading -, the device, etc. for computing the next XML web page. The templates will have metadata

entries which will be used to select the most relevant one. They will be indexed with user's tasks, type of fragments, user's category, user's level of expertise, etc. The composition engine determines the XML page content and the corresponding navigation tools. Hyperlink annotation will be defined at this level by means of properties associated to hyperlinks - Xlink.

3. **Physical Composition:** the layout engine selects a style sheet, at least, according to the template. This style sheet has to manage adaptive navigation according to user's preferences (annotation, hiding, direct guidance).

Perspectives

In this paper, we have presented our framework which consists in delivering adaptive special report according to an author oriented viewpoint. Authors have know-how which enables them to choose document contents and to organize them in one or more consistent ways by means of narrative structures. Authors can share and reuse these narrative structures. A particular knowledge elicitation method is used to formalize this knowledge because they are unable to explicit this knowledge. This method relies on theories and methods stemming from cognitive psychology and psycholinguistics. The reusability of this knowledge leads to Knowledge management. It aims to exploit an organization's intellectual assets for greater productivity, new value, and increased competitiveness².

We have proposed a semantic composition engine which delivers a user adapted special report by means of a user model and metadata. This composition engine is also studied for another European project called CANDLE which concerns with distance learning. In this paper, we have considered the management of revisiting and versioning because of the non persistent state of virtual documents. Our system has to ensure that a dynamic document can be recreated every times in the same state for a particular reader.

We plan to offer a kind of free browsing mode which will use a narrative structure as a guide. In other words, the intelligent search engine will not be limited to the information space dedicated to the special report. Indeed, the content specification of each I-Component will be applied to the entire database. A reader will be able to access all articles fitting the different content specifications and then to get articles closely related the current I-Component.

In the future, readers' strategies will be managed by adding enough metadata to compute their structures on the fly according to user's goals. For instance, a clustering process can be applied to geographical (or temporal) criteria present in the metadata and according to a domain ontology. For a local newspaper, which has several issues organized by editions (geographical areas), readers use this criteria for information retrieval from its website.

² <http://www.aifb.uni-karlsruhe.de/WBS/ontoknowledge/>

References

1. Brusilovsky, P., Methods and techniques of adaptive hypermedia. *User Modeling and User-Adapted Interaction*, 1996. 6(2-3): p. 87-129.
2. Watters, C. and M. Shepherd. Research issues for virtual documents. in *Workshop on Virtual Document, Hypertext Functionality and the Web*. 1999. Toronto.
3. Berners-lee, T., *Weaving the Web*. 1999, San Francisco: Harper.
4. Decker, S., et al., Knowledge Representation on the Web. 2000, On to Knowledge Project: <http://www.ontoknowledge.org/oil/papers.shtml>.
5. Decker, S., et al. Ontobroker: Ontology Based Access to Distributed and Semi-Structured Information. in *Conference on Database Semantics*. 1999. Rotorua, New Zealand: Kluwer Academic Publishers.
6. Ardissono, L., L. Console, and I. Torre. Exploiting user models for personalizing news presentations. in *2nd Workshop on Adaptive Systems and User Modeling on the WWW, AH99 and UM99*. 1999: Eindhoven University of Technology, Computer Science Reports.
7. Domingue, J. and E. Motta. A Knowledge-Based News Server Supporting Ontology-Driven Story Enrichment and Knowledge Retrieval. in *11th European Workshop on Knowledge Acquisition, Modelling, and Management (EKAW '99)*. 1999.
8. Mann, W.C. and S.A. Thompson. Rhetorical Structure Theory: Toward a functional theory of text organization. *Text*, 1988. 8(3): p. 243-281.
9. Christophides, V., *Electronic Document Management Systems*. 1998, UCH/FORTH: <http://www.ics.forth.gr/~christop/>.
10. Decker, S., et al., The Semantic Web - on the respective Roles of XML and RDF. 1999. <http://www.ontoknowledge.org/oil>.
11. Bray, T., et al., Extensible Markup language (XML) 1.0, (Second Edition). 1998, W3C: <http://www.w3.org/TR/2000/REC-xml-20001006>.
12. Adler, S., et al., Extensible Stylesheet Language (XSL) Version 1.0. 2000, W3C: <http://www.w3.org/TR/xsl/>.
13. Fensel, D., et al. On2broker in a Nutshell. in the *8th World Wide Web Conference*. 1999. Toronto.
14. Fensel, D., et al. On2broker: Semantic-Based Access to Information Sources at the WWW. in *World Conference on the WWW and Internet, WebNet 99*. 1999. Honolulu, Hawaii, USA.
15. Vitali, F. Versioning hypermedia. in *ACM Computing Surveys*. 1999.