



**HAL**  
open science

# Adaptive Web Information Systems: Architecture and Methodology for Reusing Content

Sébastien Iksal, Serge Garlatti

► **To cite this version:**

Sébastien Iksal, Serge Garlatti. Adaptive Web Information Systems: Architecture and Methodology for Reusing Content. AH 2004 workshops, EAW'04: Engineering the Adaptive Web, 23 Août 2004, Eindhoven, Pays, Aug 2004, Eindhoven, Netherlands. pp.36-45. hal-02142943

**HAL Id: hal-02142943**

**<https://hal.science/hal-02142943v1>**

Submitted on 4 Jun 2019

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Adaptive Web Information Systems: Architecture and Methodology for Reusing Content

Sébastien Iksal<sup>1</sup>, Serge Garlatti<sup>2</sup>

<sup>1</sup>LIUM – University of Maine – IUT de Laval  
52 rue des Docteurs Calmette et Guérin  
BP 2045 – 53020 LAVAL Cedex 09, France  
+33 2 43 59 49 26  
sebastien.iksal@univ-lemans.fr

<sup>2</sup>IASC Dept. – GET – ENST Bretagne  
Technopôle de Brest Iroise  
CS 83818, 29238 BREST Cedex, France  
+33 2 29 00 14 53  
serge.garlatti@enst-bretagne.fr

**Abstract.** Nowadays, adaptive web information systems use partially the Web to provide different kinds of content, navigation tools and layouts according to user needs. We focus on AWIS for which users share a common knowledge to work together. For us, AWIS design is an intensive knowledge driven process. We propose the methodology and architecture used in the flexible composition engine called SCARCE. The paper presents the key issues for reusing the content in the methodology: interoperability and W3C standards, consistency of the delivered document and the distinct specification and management of AWIS components. The main benefits of this approach are: i) a generic AWIS architecture which is reusable in different contexts, ii) this architecture is tuned to the explicit knowledge of communities and provide a method for AWIS design.

**Keywords.** Adaptive Web Information Systems, Composition Engine, Semantic Web, Metadata, User Model.

## 1 Introduction

Adaptive web information systems – AWIS - have the ability to provide different kinds of content, navigation tools and layouts according to user needs [1]. Traditional AWIS used to have a domain model, a user model and adaptation rules. The domain model is used to convey resource semantics. The content is typically known to the author of the application and under his control [2]. Thus, the website structure, its content, its adaptation strategies and its presentation are often designed by the same author and are often combined.

Nowadays, AWIS use partially the Web as an information space where resources are distributed. They must have the ability to reuse distributed data repositories [2, 3] and/or web services [4]. One can view AWIS consisting of different components: an organization, content, adaptation strategies and presentation. We focus our study on a methodological approach enhancing the content reuse. Some key issues of that are: interoperability and W3C standards, consistency of the delivered document and the distinct specification and management of AWIS components. The current information-intensive nature of web information systems requires more rigorous design, engineering and development process. As soon as AWIS are computed on the fly from distributed data sources, the consistency and the comprehension of the delivered document is closely related to the content, the organization and the adaptation policies dedicated to users' tasks. So, a methodology is necessary to provide a global approach unifying the different components and fulfilling the user needs.

We are interested in AWIS for which users belong to a kind of community of practices: they share a common knowledge to work together [5]. We claim that the explicit knowledge of communities is the key issue to automate AWIS and thus to ensure consistency and comprehension. The specification and the management of content – **selection**, **organization** and adaptation policies – **filtering** - are separated and based on community knowledge. Thus, the AWIS design is an intensive knowledge-driven process. We have designed an AWIS as a flexible composition engine, called SCARCE - SemantiC and Adaptive Retrieval and Composition Engine based on a semantic web approach in which explicit community knowledge is formalized in ontologies. Our methodological contribution consists of: (i) a semantic organization of resources, (ii) a declarative specification of adaptation and (iii) a knowledge-driven composition engine. The main benefits are: i) a generic architecture which is reusable in different contexts, ii) this architecture is tuned to the explicit knowledge of communities and provide a method for AWIS design. Indeed, new AWIS can be generated as soon as the specification of selection, filtering or organization is modified. Of course, it is limited by the core principles underlying the composition engine. SCARCE is the core of ICCARS<sup>1</sup>, the CANDLE<sup>2</sup> European project and KMP<sup>3</sup>.

First, some requirements for AWIS design are presented. Secondly, we develop the design principles of our architecture and methodology. Thirdly, we briefly present the adaptive composition engine. Finally, we conclude with some perspectives.

## 2 Some Requirements for AWIS Design

In AWIS design, content reuse leads to new requirements and constraints. We briefly analyzed some of them: interoperability and W3C standards, consistency of the delivered document, the separated management and specification of AWIS components.

**Interoperability and standards:** In the semantic web architecture, interoperability depends on the sharing of common architecture and standards. It takes place at three

---

<sup>1</sup> ICCARS : Integrated and Collaborative Computer Assisted Reporting System

<sup>2</sup> CANDLE : Collaborative And Network Distributed Learning Environment

<sup>3</sup> KMP: Knowledge Management Portal, RNRT Project

different levels: i) At address level, the URL/URI referring to resources whatever their location. Indeed, information (content), programs, etc. can be resources; ii) At logical or syntactic level, where resources may be described in XML and have a DTD to explicit their logical structure. Thus, it is possible to reuse such resources with different presentations in different contexts. Nevertheless, XML resources do not have semantics understandable by machines; iii) At semantic level, where resources have metadata defining contexts, conditions and rules for retrieval and reuse. Intelligent search engines provide tools for information retrieval in using ontologies and corresponding semantic metadata associated with resources. At a semantic level, the reuse, sharing and exchange of resources are a metadata and ontology issues.

Moreover, three different views of AWIS may coexist: semantic, logical and layout [6]. Each view has a specific structure organizing it. The semantic structure of a document conveys the organization of the meaning of the content. The logical structure reflects the syntactic organization of a document. The layout view describes how the documents appear on a device. The logical and layout views are combined in an html document, whereas XML standard enable us to manage them separately. The semantic level is often implicit in traditional AWIS. New AWIS architectures, based on the semantic web, could represent explicitly the semantic level. Thus, it is possible to represent the explicit community knowledge and to share/reuse this knowledge. Indeed, such knowledge can be a resource using a standard, like OWL.

To fit the semantic web architecture, the three views can be linked to the three semantic web levels as follows: i) semantic: logic, ontology, RDFS/RDF, ii) logical: syntactic level encoded in XML, iii) layout: XSL/XSLT. New AWIS could have an architecture dealing with these three views to ensure interoperability and web standards. At runtime, the computation of the delivered document to a user will be composed of three sequential processes: semantic, logical and layout.

**Separated specification and management of information system components:**

After having explained how to separate the presentation component in two parts: logical and layout. Now, we analyze content, adaptation and organization. Content can be reused in different contexts and different AWIS. Thus, it is necessary to dissociate organization and content and to design an information retrieval mechanism able to select the relevant content according to user needs from a user query or a content specification. Metadata schema enables designers to associate roles / functions with indexes and to enhance information retrieval. With semantic metadata, indexes have some values associated with ontologies which are formalized vocabularies. In our context, the metadata role is to automate the content reuse, whereas the LOM metadata schema [7] was not designed to automate the learning object reuse. It is difficult to automate this task, because the metadata authors, the AWIS ontologies and the end-users must share explicit and implicit knowledge. The metadata quality is crucial to retrieve the right content. There is an implicit information space corresponding to a metadata schema: all resources linked by its corresponding metadata.

Adaptation rules cannot be included or associated with content. They may not be known and under the control of the content author. As the content selection is made at semantic level through metadata and ontologies, adaptation rules have to match the user model and metadata. They also have to be consistent with the selection mechanism and the metadata schema principles. There are two issues: the granularity level

for retrieval and reuse and the use of indexes for selection and adaptation. Indeed, the retrieved resources can be reused as a whole or partially. For the latter, it is necessary to have rules and/or specific metadata entries to guide the reuse process. For the other issue, the role of indexes must be defined according to selection and adaptation.

In AWIS, organization used to be a directed graph enabling the designer to provide navigation tool and the user to access content. This graph can be explicitly defined or computed on the fly. This kind of structure can be naturally defined at semantic level by means of ontologies, thus closely related to explicit community knowledge. Content, adaptation and organization have to deal with different requirements. Moreover, they also have to face a common goal together: how to ensure consistency?

**Consistency:** The consistency and the comprehension of an AWIS is closely related to the content, the web site structure, presentation and the adaptation policies dedicated to users' tasks. As soon as an AWIS is computed on the fly from distributed data sources and separated views, it is more difficult to ensure the consistency of the delivered document. Thus, it is necessary to have a methodology providing a global approach unifying the different components. We could have a kind of model permitting content, organization and adaptation specifications. This model has to be able to closely link content retrieval, organization and adaptation at runtime and to ensure the global consistency.

### 3 Design Principles

The automation of AWIS and the content reuse can be considered as virtual documents (VD). VD are web documents for which the content, the site structure and the layout are created as needed [8]. In our approach, we consider an AWIS as an adaptive ontology-driven VD, defined as follows: an adaptive VD consists of a set of resources, their corresponding metadata, different ontologies and an adaptive composition engine which is able to **select** and to **filter** the relevant resources, to **organize** and to **assemble** them by adapting various visible aspects of the document delivered. Our adaptive composition engine consists of three composition engines: semantic, logical and layout. The selection, filtering and organization are managed at semantic level and the assembly at logical and layout level according to previous requirements. Content or reusable resources with associated metadata are called fragments. We consider two types of fragments: Atomic fragments are information units and cannot be decomposed; Abstract fragments are composed of other fragments (atomic or abstract) and one or more structures organizing them. We have a document model and an adaptation model to link content, adaptation and organization in a consistent way. These models are instantiated by an author as a generic document which specifies organization, selection and adaptation at knowledge level. The generic document is expanded and instantiated at runtime to compute on the fly a semantic graph. It is a semantic representation of the delivered document adapted to user needs.

In summary, we need a user model and a domain model and on top of that we have also a metadata schema and a document model. These models are formalized in different ontologies. They are as follows: **metadata ontology** at the information level which describes the indexing structure of resources, some index values are taken in

the domain and document ontologies; **domain ontology** representing knowledge in a specific area; **document** ontology consisting of a document model and an adaptation model; **a user ontology** which defines different stereotypes and individual features.

To begin, we present the principles underlying organization, selection and filtering.

- Organization is combined with selection and is based on a generic directed graph, having a single root, in which generic nodes have a content specification and generic arcs that are semantic relationships. The generic graph features are described in the document ontology.
- Selection is an information retrieval process on a set of resources (local or distributed) indexed with a unique metadata schema. Metadata have to be used for information retrieval and filtering which is a selection refinement. A subset of the metadata schema is used for selection specification and another one for filtering specification. The same granularity level (fragment) is used for retrieval and reuse.
- Filtering is based on adaptive navigation methods. The principle is as follows:
  - First, selected resources are evaluated: the evaluation aim is to put each resource in one equivalence class according to class membership rules. These classes are declared in the document ontology according to the community of practices. Generally, a maximum of five equivalence classes are defined. It might be difficult for a user to deal with too many equivalence classes [1, 9].
  - Second, one adaptive technique is chosen for the current user, its preferences and the stereotypes associated with adaptive navigation techniques. Thus, it is applied to delete, annotate, hide, etc. some equivalence classes of resources.

**The Document Ontology:** It is composed of a document model, an adaptation model, subcategories and instances of fragments, semantic relationships and generic documents. The two models belong to the core of SCARCE whereas the others depend on the community of practices. They are parts of the common knowledge shared by community members. The document model is based on the concept of *Abstract Fragment* (fig. 1). In ICCARS and CANDLE, we chose to associate a particular information space with an abstract fragment to enhance AWIS consistency. Nevertheless, this constraint can be released and all fragments indexed by the metadata schema could be used. It depends on the communities of practices.

An abstract fragment can be organized according to one or more *Structures*. A structure is a collection of *Generic Nodes* among which one is its root. A generic node is an abstract object, which only exists inside one structure. It is linked to others through a *Semantic Relation*. Semantic relations are specific to the community of practices (such as RST [10]). The set composed of generic nodes and the corresponding relationships is one organization of the AWIS. An abstract fragment is a directed graph in which the nodes are generic nodes and the vertices are semantic relations between generic nodes. At runtime, a generic node can be viewed as an information retrieval process which uses a description given by the author according to metadata, to send a query to the intelligent information broker. It is able to use the user model to filter the small set of resources. Generic documents can be organized in

a hierarchy in which classes are patterns of generic documents. In ICCARS and CANDLE, organizations are narrative structures which represent the author competences and know-how [11]. In KMP, organization is close to a task model defining practice scenarios.

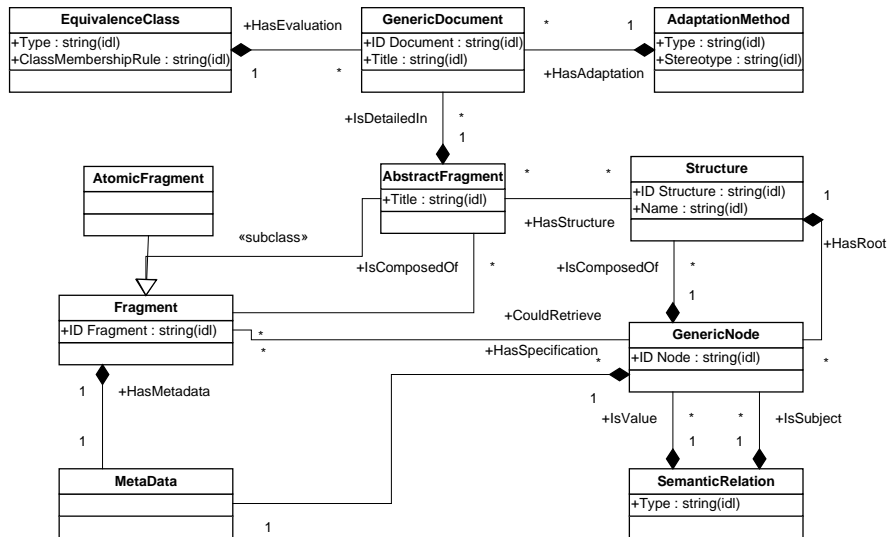


Fig. 1. Document Model

The adaptation model is composed of the equivalence classes, their class membership rules, the set of adaptation methods and their stereotypes. A generic document is also composed of adaptation rules which are instances of the different elements of the adaptation model. A class membership rule is a comparison between user’s characteristics (user ontology), and fragment’s indexes (metadata ontology). An adaptation method is permitted when a user fits the stereotype associated to that method.

**The Metadata Ontology:** The metadata schema (Table 1) provides metadata information about fragments. The semantic composition engine uses the schema for information retrieval. This schema is composed of two kinds of characteristics, typical entries for web resources such as the author, the language and the date of creation, and more specific entries that depends on the community of practices. The typical entries are usually found in numerous metadata schemas, but they are not compulsory, it depends of the application. A gray background shows the specific entries.

Table 1. The metadata schema of ICCARS

MD.1	General	General information about the resource	
MD.2	Life cycle	Entries for versioning purposes	
MD.3	Meta Metadata	Information about metadata	
MD.4	Technical	Technical information about the resource	
MD.4.1	Location	Where the resource can be found?	
MD.4.2	Format	Format of the resource	
MD.4.2.1	Type	Type of the resource (ppt, doc, html, ...)	
MD.4.2.2	Size	Size of the resource in Kbytes	

MD.5	Classification	Data about the content and reporting features	
MD.5.1	Domain	Description related to the domain	
MD.5.1.1	Concept	Concept name	Domain ontology
MD.5.1.2	Level	Level of knowledge required	
MD.5.2	Reporting	Reporting classification	
MD.5.2.1	Resource Type	Type of resource (Interview, report ...)	Document ontology
MD.5.2.2	Edition	Edition concerned	Document ontology
MD.5.2.3	City	City concerned	
MD.6	Rights	Use conditions of the resource	

That schema is composed of six sections, typical sections are usually proposed in all metadata schemas (“General”, “Technical”, ...). Some sections are more specific to our architecture, it is the case of “LifeCycle” which is necessary for the management of versioning purposes [12]. “Classification” has to describe the content of fragment according to the domain model and to the community of practices.

**The User Ontology:** It describes the user characteristics for adaptation purposes. It is used by the adaptation model to define the class membership rules and the stereotypes of adaptive methods. The user ontology needs the domain ontology to describe the knowledge of the user – as an overlay model. Generally, the user model consists of the following five parts: personal, preferences, knowledge, history and session. It is possible to add some specific sections according to the community of practices. “Personal” is composed of typical data on the user such as his name. “Preferences” is useful for personalization purposes; it is a section that can be modified easily by the user. “Knowledge” describes the overlay model, and is used by the system for evaluating the relevance of fragments. “History” is used for proposing facilities to the user inside the environment such as bookmarks. “Session” is used only by the system as a storage area for the current session. “History” and “Session” are specific to our architecture because they are used by our composition engines.

**Table 2.** The user model of ICCARS

UM.1	Personal data	Personal data concerning the user	
UM.1.1	Identity	His identity	
UM.1.2	Login	Unique identifying data	
UM.1.3	Classification	Classification data	
UM.1.3.1	Location	Where does he live?	
UM.1.3.2	Professional Activity	What kind of job? (economist, fisherman, student, etc.)	
UM.1.3.3	Role	The role in the application (author, reader ...)	
UM.2	Preferences data	Data about the preference of the user	
UM.2.1	Interest	Topics of interests	
UM.2.1.2	Topic	A list of topics	Domain ontology
UM.2.2	Adaptation	Adaptation preferences	
UM.2.2.1	Element	An element which can be adapted (link ...)	
UM.2.2.2	Rule	A method of adaptation (annotation ...)	Document ontology
UM.3	Knowledge	Data about the knowledge of the user	
UM.3.1	Domain	Knowledge about the domain	
UM.3.1.1	Element	A domain concept	Domain ontology
UM.3.1.2	Level	A level of knowledge	
UM.4	History data	Data about access to generic documents	
UM.5	Session data	Data concerning the current session	

**The Domain Ontology:** It is a typical domain model which is composed of a semantic network for the domain of the application. It is the link between the user knowledge, the fragment description and the generic node content specification.



## 4 Adaptive Composition Engine

The four mechanisms of VD are implemented as follows: selection, filtering and organization are achieved in the semantic composition; assembly is divided into logical and layout compositions. The generation of the document is detailed in [13].

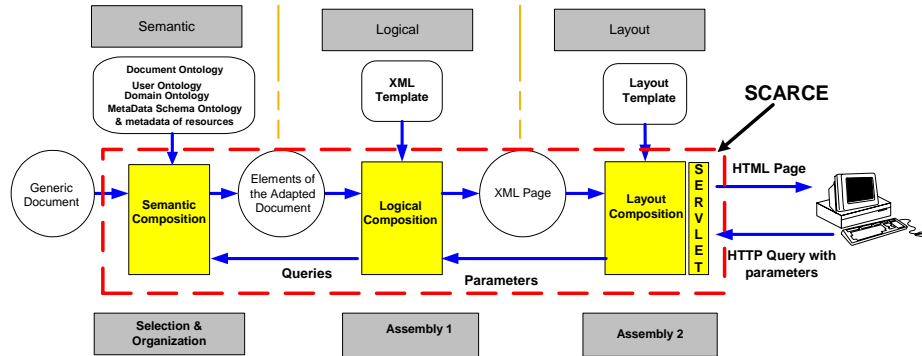
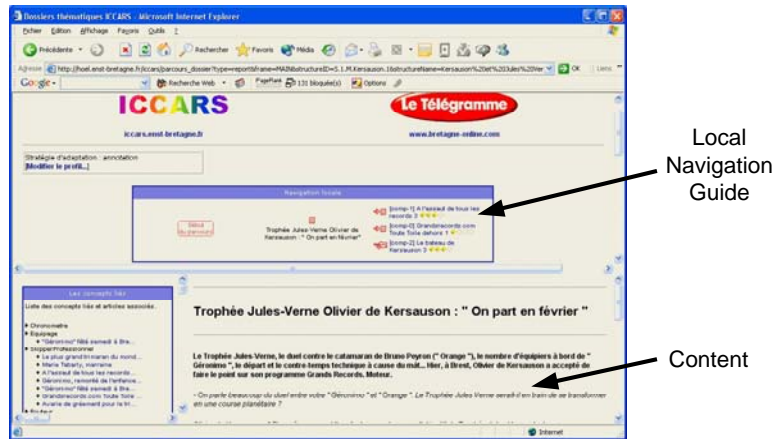


Fig. 2. The Composition Engine Architecture

The semantic composition engine computes on the fly a **semantic graph** from a **generic document**. This semantic graph represents a narrative structure adapted to the current user and linked to its corresponding content. The aim of the logical and layout composition engines is to convert the semantic graph into a real document. This process depends on the nature of the real document. The navigation and the resource access are based on the semantic graph. The logical composition engine requests the semantic composition engine to build and to browse the semantic graph according to user interactions and computes for each node an XML web page with its content and navigation tools. The layout composition engine generates an HTML page from the XML web page by applying the layout rules.

Now, we present briefly the request-response cycle of the composition engine. The real document is computed on the fly stage by stage. A stage consists of the computation of the next web page according to user interaction. An XML template describes a generic web page which is composed of several dynamic components (navigation guides, etc). For instance, we assume that we have local and global navigation guides and content inside a template. In our example, we focus on the local navigation guide which is paradigmatic (fig. 3.). In this figure, the local navigation guide consists of three types of links: previous, current and next. It applies the annotation technique: one or several stars are associated with a link. To be able to choose the right number of stars, it is necessary to evaluate the corresponding resources and to classify them in an equivalence class. All the resources are associated with a node of the semantic graph and then with the corresponding node of the generic directed graph. The “next” links enable the user to select the direct neighbors of the current web page. They consist of all the next nodes in the semantic graph. Therefore, we have to compute in advance the direct neighbors in the semantic graph before displaying this component.



**Fig. 3.** Web Page layout

From a user interaction, the servlet engine receives an HTTP query with parameters: the component of the query and the names of the XML template, the generic document, the next node and the content of the next web page. The layout composition engine sends these parameters to the logical composition engine. It accesses the XML template with several XML components. For each XML component, a java component requests the semantic composition engine to instantiate its XML component and generates an XML structure. For instance, the local navigation guide component asks the semantic composition engine to compute the previous, current and next web pages of the new one and their equivalence classes. Then, it is able to get an XML structure having these types of links with their parameters and equivalence classes. From these computations, the logical composition engine is able to instantiate the entire XML template and to obtain an XML web page. Then, the layout composition engine generates the web page from the corresponding layout template.

## 5 Conclusion

In this paper, we have presented a methodology and an architecture for the content reuse in the design of AWIS. It ensures interoperability in separating the specification and the management of resources, organization, adaptation and layout. Content, organization and adaptation are specified by an author according to its explicit community knowledge. Our methodological approach relies on: (i) a semantic organization of resources by means of the document model and the community know-how, (ii) a declarative specification of the generic document at knowledge level, (iii) a knowledge-driven engines for the composition of AWIS. We propose a generic architecture for AWIS that have to be instantiated for specific communities of practices. Our approach is limited by the core principles of the architecture. To enhance the consistency of the delivered document, we chose to associate an information space to an abstract fragment. The main idea is to ensure that we have the right content at the

right place for the right user. It will be interesting to release this constraint according to the user goals and the services we want to propose. In the case of a typical information retrieval system, it is necessary to deliver the maximum of relevant fragment, but in the case of a pedagogical application, it is necessary to ensure the coherence between all fragments retrieved. We don't have a solution at the moment; it is an interesting problem that has to be studied in the future. SCARCE engines are currently implemented in Java and linked to the servlet engine. In ICCARS and CANDLE, the ontology-based inference engine used is Ontobroker [14]. In KMP, we are developing a new interface in the semantic composition engine for Corese [15].

## References

1. Brusilovsky, P., *Methods and techniques of adaptive hypermedia*. User Modeling and User-Adapted Interaction, 1996. 6(2-3): p. 87-129.
2. Aroyo, L., P. De Bra, and G.-J. Houben. *Embedding Information Retrieval in Adaptive Hypermedia: IR meets AHA!* in *AH 2003: Workshop on Adaptive Hypermedia And Adaptive Web-Based Systems*. 2003: Technische Universiteit Eindhoven.
3. Brusilovsky, P. and H. Su. *Adaptive Visualization Component of a distributed Web-based Adaptive Educational System*. in *ITS 2002*. 2002. Biarritz: Springer Verlag.
4. Conlan, O., et al. *Applying Adaptive Hypermedia Techniques to Semantic Web Service Composition*. in *Workshop on Adaptive Hypermedia and Adaptive Web-Based Systems in The Twelfth International World Wide Web Conference*. 2003. Budapest, Hungary.
5. Wenger, E., *Communities of Practice - Learning, Meaning and Identity*, in *Learning in Doing: Social, Cognitive and Computational Perspectives*. 1998, Cambridge University.
6. Christophides, V., *Electronic Document Management Systems*. 1998, UCH/FORTH: <http://www.ics.forth.gr/~christop/>.
7. LTSC, *Learning Object Metadata Working draft document 3*. 1999, Learning Technology Standard Committee: <http://ltsc.ieee.org/doc/wg12/LOM-WD3.htm>.
8. Milosavljevic, M., F. Vitali, and C. Watters. *Introduction of the virtual document workshop*. in *Workshop on Virtual Document, Hypertext Functionality and the Web*. 1999. Toronto.
9. De Bra, P. and L. Calvi, *AHA! An open Adaptive Hypermedia Architecture*. The New Review of Hypermedia and Multimedia, 1998. 4: p. 115-139.
10. Mann, W.C. and S.A. Thompson, *Rhetorical Structure Theory: Toward a functional theory of text organization*. Text, 1988. 8(3): p. 243-281.
11. Iksal, S., et al., *Semantic composition of special reports on the Web: A cognitive approach*, in *H2PTM'01*, P. Balpe, et al., Editors. 2001, Hermès. p. 363-378.
12. Iksal, S. and S. Garlatti, *Revisiting and Versioning in Virtual Special Reports*, in *Hypermedia: Openness, Structural Awareness and Adaptivity, Third Workshop on Adaptive Hypertext and Hypermedia*, S. Reich, M.M. Tzagarakis, and P.D. Bra, Editors. 2001, Springer Verlag: Århus, Denmark. p. 264-279.
13. Garlatti, S., S. Iksal, and P. Tanguy, *SCARCE: an Adaptive Hypermedia Environment Based on Virtual Documents and the Semantic Web*, in *Adaptable and Adaptive Hypermedia Systems*, S.Y. Chen and G.D. Magoulas, Editors. 2004, Idea Group. p. To be appear.
14. Fensel, D., et al. *On2broker in a Nutshell*. in *the 8th World Wide Web Conference*. 1999.
15. Corby, O. and C. Faron-Zucker. *Corese: A Corporate Semantic Web Engine*. in *Workshop on Real World RDF and Semantic Web Applications 11th International World Wide Web Conference*. 2002. Hawaii.