Chapter XXVII
Semantic Web and Adaptivity:
Towards a New Model

Jorge Marx Gómez
Carl von Ossietzky University of Oldenburg, Germany

Ammar Memari
Carl von Ossietzky University of Oldenburg, Germany

ABSTRACT

The chapter aims at proposing a model that gives an abstraction to the functionalities and data involved in adaptive applications for the Semantic Web. As the quantity of provided information on the Web is getting larger, the need for adaptation in software is getting more and more necessary in order to maximize the productivity of individuals, and more issues are emerging that have to be considered in the new generation of hypertext systems. With the advent of Semantic Web, adaptation can be performed autonomously and in runtime, making the whole process of information adapting transparent to the user.

INTRODUCTION AND MOTIVATION

Recent development within the Semantic Web community suggests that the Internet, or the WWW, as we know it is about to change; the content and the services will be annotated with meta data which can describe and define them.

Traditionally, the Web is seen as a collection of linked nodes, as entailed by the specification of the reference model for hypertext applications, the Dexter Model (Halasz & Schwartz, 1994). This model has long been successful in abstracting the applications that could deliver resources to the Web for human users to use.

However, the supply of information is steadily increasing, and today's Web is the place for expressing ideas, telling stories, blogging, sharing movies photos and sound clips…, anything that anybody ever wanted to say, so the huge amount of knowledge available to any one person is far more than they can possibly absorb (Bailey, 2002) and exposing the human brain to such a big pile of information will cause the "Lost in the Hyperspace" problem, when the useful information remains unfound because it is hidden under a huge amount of useless and irrelevant data.

Fortunately as this supply of information increases, so does also the automatic information processing capabilities. So, there is and will be great potentials to make use of these automation capabilities in order to extract from the overflow of the Web the information and services relevant to the user on an ad-hoc basis, and deliver them over a standardized user interface. The retrieval of data in this adaptive manner gains more and more importance as the mass of provided information grow larger.

We can notice that the portion of the information accessed by any individual on the internet, no matter how large it is, is so small and nearly negligible when compared to the full amount of information available. What if we were able to define this portion of information? Or draw the border of it?

The process of moving from the information domain to the interest domain is similar to moving from the time domain to the frequency domain through Fourier transformation.

Efforts to abstract the functionality and data representation of hypermedia systems in a model, or furthermore a reference model, were conducted. Two of them are mostly used, The Dexter Hypertext Reference model, and the model of the World Wide Web, which is slightly different from Dexter.

These Models can no longer abstract the functionalities and data required for adaptive applications for the Semantic Web. Many efforts were made to come up with a model that extends the Dexter model with adaptive functionalities but most of them carried along some of the limitations of Dexter, and some others concentrated on the static structure of the Semantic Web not on the wider spectrum of both static and dynamic relations between the knowledge and the consumer (see (Memari & Marx-Gomez, 2008)). In fact even the Dexter Model had some aspects of adaptivity, but it had them unintentionally, and they are definitely not enough (Dodd, 2008).

BACKGROUND INFORMATION

Dexter Hypertext Reference Model

"The Dexter model is an attempt to capture, both formally and informally, the important abstractions found in a wide range of existing and future hypertext systems." (Halasz & Schwartz, 1994). The Dexter model is an abstraction of existing hypertext systems -till the day it was presented- and was meant to be an abstraction of future hypertext systems as well. It was created as an answer to the question : what do hypertext systems such as NoteCard, Neptune, KMS, Intermedia, and Augment have in common?. So the model had a goal to collect the common properties and specifications of such systems in order to create a standardized model capable of drawing the border between the hypertext systems and other types of systems , as well as helping to develop interchangability and interoperability standards.

The idea of the model was developed during a series of workshops that took place in the years 1988 and 1989 to develop consensus framework specifications for hypertext systems (Iverson, 2003). Started by the "call to arms" issued by Norm Meyrowitz (Meyrowitz, 1989) and ended up by the formulation and presentation of the Dexter Hypertext Reference Model by Frank G. Halasz and Mayer Schwartz in 1990.

The model divides the hypertext system into three layers: Runtime layer, Storage layer and Within-component layer (Iverson, 2003).

1. **Runtime layer**: The active part of the hypertext structure, that allows for interaction and manipulation of the structure as well as

notification of changes and maintenance of link structures.

2. **Storage layer:** This is the place where the hypertext system stores components and links; no structure within the component is modeled, components are treated as generic containers of data.
3. **Within-component layer:** This layer (undefined in Dexter) manages the internal structure of the hypertext objects. It is assumed that this covers all types of data, including text, graphics, audio, video and all the other aspects of the hypermedia archive.
4. **Presentation specifications:** A critical aspect of the Dexter model. Mechanism for encoding presentation or viewing specifications into the hypertext network. Thus we can infer that the way a component is presented is a function of:
   - o The specific runtime layer and
   - o A property of the component itself which is the presentation specification.
5. **Anchoring:** An extremely critical piece of the Dexter model. The mechanism for addressing (refering to) locations or items within the content of an individual component (Halasz & Schwartz, 1994). This interface provides the clean separation between the storage layer and the within-component layer.

The main focus of the model is on the storage layer and on its adjacent two layers, while the other two layers are not as focused; the within-component layer is purposefully not elaborated within the Dexter model. And concerning the runtime layer the source states: the range of possible tools for accessing, viewing and manipulating hypertext networks is far too broad and too diverse to allow a simple, generic model (Halasz & Schwartz, 1994).

The model defines all of its terms and leaves the important design decisions open, so every hypertext system has the freedom in implementation, and that fits the goal of making a "reference model".

Semantic Web

Semantic Web is about making the Web more understandable by machines (Jeff Heflin & Hendler, 2001). Semantic Web is about building an appropriate infrastructure for intelligent agents to run around the Web performing complex actions for their users (Jeffrey Douglas Heflin, 2001).

Semantic Web is about explicitly declaring the knowledge embedded in many Web-based applications, integrating information in an intelligent way, providing semantic-based access to the internet and extracting information from texts (Gómez-Pérez & Corcho, 2002).

Semantic Web is about how to implement reliable, large-scale interoperation of Web services, to make such services computer-interpretable, create a Web of machine-understandable and interoperable services that intelligent agents can discover, execute and compose automatically (Devedzic, 2006).

Semantic Web is the natural solution for the problems of the current Web, and is the inevitable way to make the huge supply of information that lie within the Web useful and responsive to humans; we can no longer expose the human brain to this massive amount of data and expect it to cope with it and to analyze, extract and compose knowledge out of it. Instead, we must be able to process information automatically and present them to the user, formed and arranged for a particular purpose. No matter how simple the task might seem, it is not that simple; computers have a big advantage over humans concerning the processing capabilities, nonetheless there are many issues to be addressed before we can make any use of this advantage. For example what exists on the current Web is information, and computers are used presently as carriers of this information, so a program does not have a real understanding of the content of them.

The two most essential questions to be answered are:

- How do we represent the knowledge in such a way that it can be understood and processed automatically by a computer.
- Once we have this information, how can we use it effectively in real applications (Walton, 2006).

As an answer to the first question, we must think of a way to represent the knowledge so that it can be interpreted and reasoned about by both the producers and the consumers of it. We must take into consideration what we are going to represent, what we actually represent is a conceptualization, which is a simplified view of the world (Walton, 2006).

The proposed answer to this question is the concept called ontologies, it is a term borrowed from philosophy, and refers in our case to a specification of a conceptualization (Walton, 2006) ; an ontology may include a description of classes, properties and their instances ("OWL Web Ontology Language Guide", 2004); so ontologies are meant to represent a set of concepts within a domain, and different kinds of relationships between those concepts. In addition to the well-known relationships among classes and objects in the object-oriented languages e.g. generalization, composition, aggregation and association, ontologies represent some more relationships and addresses interrelationships between classes properties and instances (individuals) . Some examples of languages used to describe ontologies are : RDF, RDFS and OWL.

In fact ontologies must be designed for reusability, and they will have big portions of intersection, so an ontology can be a part of other ones, and ontologies can share classes, relationships and individuals.

Ontologies have given us a clue about solving the first problem of representing the knowledge in a way readable by machines; this is required, but it isn't enough. To make a real use of this represented knowledge, we need to design a software that can understand , reason about, collect, summarize adapt to this knowledge, and be able to represent it back as a human-friendly knowledge. The Semantic Web vision promotes the concept of agents as the primary consumers of knowledge (Walton, 2006).

Agents and Adaptivity

Agents are programs that act on behalf of the user, they are not executed or invoked for a specific task, rather they are autonomous and activate themselves, they collect Web contents from different sources, process the information, and contact each other to exchange results, experiences and trustworthiness degrees of the knowledge sources… etc.

we can define an agent with the intention to obtain some useful piece of information, e.g. a list of banks in our area. The agent can then be let loose onto the Web and will attempt to find this information for us by utilizing a variety of knowledge sources, and performing inference on this knowledge. (Walton, 2006, p. 11)

Since agent are going to act on behalf of humans, studies are conducted to give them the human logic, and that was approached in many ways, the most popular of these ways is the Believe-Desire-Intention model (Bratman, 1999).

Agents are sensitive to the changes of their environment, the changes in the environment can have a wide range of variety. According to this event the agent has to decide the way it should act, it will interpret the input considering its intentions. Intentions are the driving force of the BDI model as they determine the actions of the agent (Walton, 2006). In addition to intentions the agent has also its beliefs that correspond to knowledge that the agent has accumulated about the outside environment which can be built by own experience or by others, and can be shared among agents; has its desires which is the state of affairs that the agent would (ideally) like to bring about (Walton, 2006) and has its goals which is a subset of desires that do not conflict between each other.

The vision of the Semantic Web goes beyond the view of agents working in isolation, it goes to considering all the agents as a society of communicative individuals. An agent in such a society can act in a social way by interacting with other agents in order to cooperate, coordinate, negotiate, advise and ask for advice… etc. and this kind of system is called a Multi-Agent System (MAS). If we want to consider the ways of inter-agent communication, we must think beyond the standard ways ; an agent must be able to express its beliefs to another agent (Walton, 2006), so each agent may have its own way of storing and understanding these beliefs, but there must be a common language among agents to express them; and this study is inspired by the philosophical "human dialog" study.

As defined in biology: Adaptation is the change in organisms that allow them to live successfully in an environment (Freeman & Herron, 2003). Adaptation allows the living organism to cope with environmental stress and pressure. The environment around the applications of the Semantic Web is in constant variation, and in order for a system to perform a good and suitable action in such a dynamic environment, it has to adapt to the new circumstances. That has to be done dynamically, autonomously and in runtime; today's accurate decision might be inaccurate tomorrow, the system must be capable of "learning from" and "adapting to" changes in the environment.

The relation between adaptation and agents has two different dimensions:

- The adaptive agent dimension: From this point of view, agents react on changes in the environment, and in other words they adapt to it. We look at the agent itself as an adaptive application, since the features of the way that agents work can be easily mapped into performing tasks that are adaptivity-related; e.g. agents use knowledge bases and rules to govern their actions and that can easily be mapped into the use of rules to personalize information according to a user model. An adaptive agent is an agent with actions that are flexible and may be learnt through interaction (Walton, 2006).
- Using agents for building adaptive applications dimension: Looking at the relation from this point of view leads us to considering a higher level of adaptivity, where a Multi-Agent System (MAS) can compose a layer or a module in the application which will be responsible for giving the application an adaptive property, in a way that this MAS is used for filtering, rating and optimizing the information according to the preferences of a user (or a set of users). Such a use of MAS can be found in a semantic search engine that constitutes a part of a Web portal, or for a semantic Web Service discovery component as the one proposed in this chapter. For another example of a multi-agent framework for personalized information filtering see (Lommatzsch, Mehlitz, & Kunegis, 2007).

In our proposed model, the agents concept will be utilized in the form of a layer, which has standardized interfaces with adjacent layers. This agency layer is a subset of a MAS that will be shared among agency layers of other applications.

WHY DEXTER MODEL

The WWW was a success in itself, but can we build another success on top of it?

Even though we can't consider the model that the WWW was built upon as a separate model, we can't say that the WWW conforms one hundred percent to the Dexter Model.

If we try to make a mapping between the WWW and Dexter model, we can easily consider the Web page as the basic composite component in the WWW, this page is composed of other "Atom" components like text, images, videos, etc. and might also be composed of other "composite" components like the case of framed Web pages or the Ajax-based Web pages.

In addition to that, a Web page contains both "Anchors" and "Links" to other anchors (the <a> and <area> tags), we can here detect a difference between the model and the WWW, since the links and anchors are embedded within the page instead of being separated into independent components. That will blur the position of the Web page in the Dexter model layers because it has also elements from the "Anchoring" layer which cannot be separated. This fact will also make it more difficult to work on the hypermedia structure separately in order to add flexible behavior to the system such as the dynamic adaptive navigation, thus confining the system and preventing the WWW from having an adaptive feature that exist in Dexter (discussed later) and in Open Hypermedia Systems in which "information contents are usually stored apart from hypermedia structures" (Grønbæk, Bouvin, & Sloth, 1997). and makes the WWW "completely lack the editability and link maintenance requirements that Dexter imposes." (Iverson, 2003). Despite the advantages of reduced complexity of embedded links, for the Semantic Web the Open Hypermedia approach seems more realistic. Keeping in mind that annotation of others' work has always been an important feature, and the importance of it gets even bigger in Semantic Web, we can notice that the embedding of links inside the documents is as a big obstacle in front of this feature, because the user who wants to annotate a document must have in this case the write access to it.

Links in WWW are uni-directional, this fact can decrease interconnectivity of the system, whereas links in Dexter model have a richer and wider meaning. Although Web browsers provide "back" button depending on session history, and this button can be thought of as a second direction of a link, some transactions (using scripts for example) are not recorded in the that history so they are not traceable. To cover the limitation of uni-directional links in the WWW, Google had to implement a part of the PageRank™ algorithm (Brin & Page, 1998) that takes a given page and calculates all the links that refer to it, in a manner that is similar to a reverse-direction link.

Semantic relationships are more complex than simple HTML uni-directional links. Embedded encoding of such information will increase the complexity of authoring Web content and raise maintenance costs. Since the navigation through semantically annotated resources will involve inferring and reasoning, it is more likely for that to be accomplished on the server side rather than implementing such a linking functionality within the browser, especially if we take into consideration that the reasoning process itself might have aspects which belong to the specific domain of the knowledge.

According to Lee Iverson, the Hypertext systems that existed before the Web were far more capable than the Web and far more sophisticated, and yet they seem to have been swept aside for a much less capable and sophisticated option (Iverson, 2003).

So before jumping to the new era of Semantic Web, let's stop and take a look around.

RELATED WORK AND POSITION OF EFFORT CONCENTRATION

Models Extending Dexter

Since the Dexter model was announced in 1990 and until now, a lot of research was conducted to modify the model in order to make it include some emerging techniques, or to make it adapt to some new standards, most of the modifications tried to be conservative i.e. to keep the old systems included under the new model and come up with a solution that includes the new systems in an extension-like manner. The list includes:

- **Amsterdam Model of Hypermedia (Hardman, Bulterman, & Rossum, 1993)** Amsterdam model came to support some aspects of time and synchronization that Dexter model lacks, since Dexter model does not have these concepts (Dodd, 2008) and this can lead to problems for multimedia applications.
- **Adaptive Hypermedia Application Model (AHAM) (De Bra, Houben, & Wu, 1999) (Figure 3 a)** One of the well-known adaptive hypermedia models. It is an extension of the Dexter model which focuses on the storage layer, anchoring and presentation specification as well. AHAM divides the storage layer into three models:
    - The domain model contains a conceptual representation of the application domain. In the Dexter model the storage layer only contained what AHAM calls the domain model.
    - The user model contains a conceptual representation of all the aspects of the user that are relevant for the adaptive hypermedia application. This includes an overlay model of the domain, the user's background, experience, preferences or anything else that contributes towards the adaptation.
    - The adaptation model describes how an event, such as the user following a link, results in a presentation, by combining elements from the domain model and the user model.
- **Munich Reference Model (Koch & Wirsing, 2002)** As an object-oriented version of the AHAM came the Munich Reference Model which is described using the Unified Modeling Language (UML).

Models for Adaptivity

- **Fundamental Open Hypermedia Model (FOHM) (Millard, Moreau, Davis, & Reich, 2000)** FOHM was presented through an approach towards an Agent-based framework to support adaptive hypermedia. It provides the facility to attach context and behavior objects to the model at various locations. A contextual open hypermedia FOHM server was developed called Auld Linky to instantiate and process the model which belongs to the Open Hypermedia family.
- **Goldsmiths Adaptive Hypermedia Model (GAHM) (Ohene-Djan, 2002)** This is an abstract model that consists of three groups of functions: the H-Region functions model non-personalizable hypermedia-based interaction , the P-Region functions model user-initiated tailoring and the A-Region functions model the system initiated tailoring of hypermedia content.
- **Generic Adaptivity Model (GAM) (Vrieze, Bommel, & Weide, 2004)** GAM is not restricted to hypermedia only , but it is a more generic model. This state-machine-based model can be used as the basis for adaptation in all kinds of applications. To use this model for adaptive hypermedia, another model must be built on top of it, and

then GAM can provide functionality that extends the AHAM model's functionality.

Efforts that Cosider the Semantic Web

- **Enhanced Adaptive Hypermedia Application Model (Kravčík & Gašević, 2006) (Figure .3 b)** The model deals with issues related to design and implementation of personalized and adaptive systems, considering the Semantic Web technologies as a means to achieve certain progress in this field. The model aims at the ways that Semantic Web can solve the demands of interoperability between various systems and between formal models. The effort is oriented towards filling the gap in the harmonization of the different standards through a Semantic Web approach.

This model extends the AHAM (which in turn extends Dexter) by adding other divisions to the storage layer.

Towards more efficient Generic Semantic Authoring for Adaptive Hypermedia (Saksena & Cristea, 2006): This effort is a description of steps taken towards creating more efficient generic semantic authoring for adaptive hypermedia, so it focuses on using the semantic methodology in creating content for educational usage that can be reused in many applications, starting from an existing framework, LAOS, an existing system, MOT, and evaluating results thereof. This system is to make use of existing appropriate Semantic Web techniques (Cristea, Stewart, & Sirmakessis, 2006). The effort concentrates on concrete examples (existing systems) but addresses some important general issues.

As illustrated in Figure 1, we can see that our approach is located in the intersection between Semantic Web applications, Adaptive Applications and Hypermedia applications.

EXPLOITABLE ASPECTS OF ADAPTIVITY

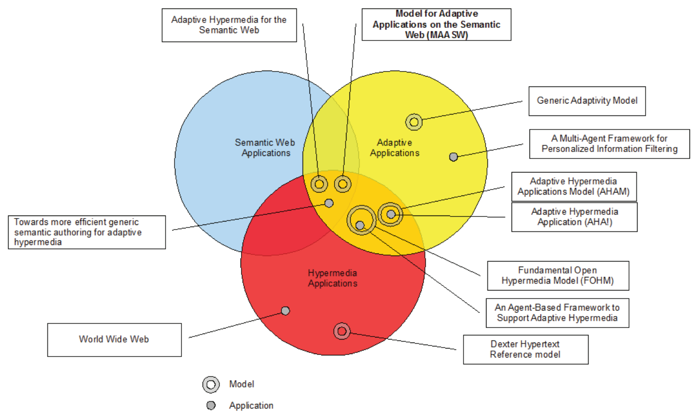
As we can conclude from the motivation, the urgency of the need for adaptation is a current issue. Adaptation is necessary nowadays and in the future systems mainly due to the overflow of available data, thus we can excuse old systems for not concentrating on the adaptation aspects.

Nevertheless, we can still find some handy adaptation-usable features, even in the 20-years old Dexter model:

Dexter model provides the concept of the "Session", which represents a single user interacting with the content and includes a history of user interaction during a session. Although this history was not meant to be used this way, it can be used to help in decisions of selecting among different competing Presentation Specifications, for example if the user was navigating his way through Web pages concerning ancient civilizations and reached our Website which talks about Origami, then there is a better way to present the content of our Website to this specific user, concentrating in the first page on the history of this art. For another user who made his way here through Web pages concerning kids creative games, a different way of presenting our content might sound better to him. All in all, when you see where the man is coming from, you can more easily predict where he is going to.

Dexter model allows to leave the selection of which subset of components to show and which to hide to be made at runtime, which allows for a comprehensive collection of components. Selection can be based on Presentation Specifications, Session history, and component attributes (name/value pairs that help describe the component). Leaving such decision to the runtime gives adaptivity-enabled applications the possibility to make a resource adapt to some needs of the user. Keeping in mind that links according to Dexter are also components and they have their own Presentation Specifications, we can make it possible

Figure 1. Location of different models within the space of the Semantic Web, Adaptive



for example to decide during runtime where to navigate the user when he clicks a specific link. (Note that it is harder to do that in the case of embedded links).

PAVING THE WAY FOR A NEW MODEL

The effect of incorporating the Semantic Web as the platform upon which the new generation of applications will be built is vast, and it is not limited to the static structure of the Web, rather it will also affect the ways of information storing, retrieving, interpreting and presenting.

When we want to think about the next generation of hypermedia systems that resides on top of the Semantic Web, we have to consider the following issues (Ossenbruggen, Hardman, & Rutledge, 2001) and (Halasz, 1988):

1. The environment of the Semantic Web is highly decentralized, distributed and heterogeneous, and when we import technologies like ontologies and knowledge bases from the field of Artificial Intelligence, we have to keep in mind that these technologies focused originally on representing centralized, localized, consistent and trustworthy knowledge. The concept of adaptivity is strongly connected to filtering, which will remove the redundant, irrelevant and inappropriate information from the available heap, providing transparently a stable foundation of information to the upper layers.
2. As mentioned before concerning the multi-directional links, we also have to keep that in mind concerning the URIs inside the RDF annotation. Given an RDF annotation, we can easily resolve the resources using URIs, given a resource, it is hard to resolve all the RDF notations that refer to it without using crawlers (as in the aforementioned Google approach). Learning from the lessons of Open Hypermedia, we should consider having link services in order to separate the structure of the hypermedia from the content. but that will compromise the acentricity of

the system by making it depending on the providers of such services.

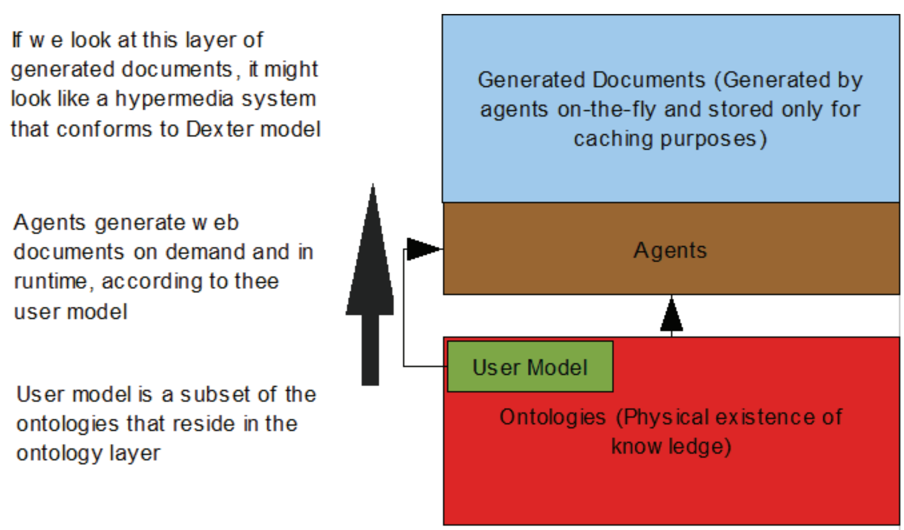
3. Considering binary-encoded data formats in terms of indexing, annotating, recognizing and management. Such formats include images, video files and audio files. None of the already existing systems (which are metadata-management based) is able to define a strong image or A/V semantics derived from the image or A/V structure only. In order to give the knowledge management systems the ability to manage these data types like other contents (text files ...), new languages need to be developed to be able to point into these time-variant and compressed data; enabling for example the search by audio, video, location, image, on-screen text, face, pattern and concept. These formats are common in the multimedia domain. This domain with advances in computer graphics is suggested to play a key role in the evolution of the World Wide Web composing the (other) way leading to Web 3.0.
4. Computer Supported Cooperative Work (CSCW): We are witnessing a steady rise in B2B transactions, enabled by the inter-company exchange of data over a mutually agreed format and recently, the story even goes further, when a few companies make use of the internet platform to move forward towards a across-company integration of their entire information system so as to achieve strategic alliances. CSCW should be considered in the next generation of hypertext systems, since the entire information infrastructure of the company is moving towards the Web platform, offering a unified and standardized access for customers, suppliers and employees to the information and services the companies offer. Making use of applications offered over the employees portal, for instance groupware, staff members can meet, cooperate, and coordinate their work over the internet. Even with the infrastructure of the Semantic Web, many of the features which are main aspects of the cooperative work such as authentication, access control, concurrency control and version control are not yet fully integrated in the Web's infrastructure. We can solve the technical part of this problem by providing extensions which realize these features, but the social and dynamic aspects of the collaboration problem will remain unsolved. It is only because of the Web's initial focus on "read-only'' browsing that these features hardly received any attention. A notable exception is the joint IETF/W3C work on WebDAV (Whitehead Jr. & Wiggins, 1998). While WebDAV predates RDF, it has a similar property-based model for Web resources. In addition to that, it is important to provide means to enable users to annotate the work of others as mentioned earlier in "Why Dexter Model" section, it is an important feature that has been realized in previous efforts namely Stanford ComMentor (Röscheisen, Mogensen, & Winograd, 1995) and others, but disappeared over time due to lack of interoperability, so the annotations could not be shared across applications in the same way as other Web resources.
5. Conceptual Hypertext introduces two layered approach, the bottom layer consists of interconnected "concepts" whereas the upper one contains documents that are presented to the users, separating knowledge representation from document representation as shown in Figure 2. Using the full power of ontologies to improve hypertext linking based on the semantic relations among the associated concepts is a promising methodology that found its implementation in (Carr, Wendy Hall, Bechhofer, & Goble, 2001). Using agents to mediate between the two layers is an approach that adds much

flexibility to the manner of generation by making the processes of adding, deleting and modifying the methodologies of generation as simple as adding, deleting or modifying agents, which can be done in runtime and on the fly.

6. Is the basic model of node, link and anchor (Dexter) still the best way to present information to humans ? (Rouet, 1992) has conducted a study concerning cognitive processing of hyper documents, it includes different experiments that were done to compare linear presentation formats (normal text) to hypertext and has got by then some results, and discussed some ideas. Online definition of of unfamiliar words can improve comprehension only if the defined words are important to the meaning of the text. computerized assistance has to do more than providing additional information, this information must be of correct type and trigger effective comprehension process. Moreover it should be clearly signaled, immediately accessible, and unambiguous. Thus the hypertext in itself should not be thought of as the final answer for information representation, more semantic layers should be involved in the authoring phase. The semantic structure of the domain should also be reflected in the content map of the hypertext in order to facilitate the process of reviewing the document, according to (Rouet, 1992).

   Rouet also came to the conclusion that some of the users who were the subject of the experiments were unable to take advantage of the ability to self-organize the content (provided by the hypertext), and they had better results with the linear pre-organized texts, because they didn't build meaningful navigational strategies and they lack the "cognitive monitoring". Since that time and until now, the familiarity of the hypertext to the people has increased significantly, but we still can't say that all the people are able to organize their own learning activity. Furthermore when we consider that the user has to find his way through this huge pile of available information, we can say that all the users are lacking these organizational skills to some extent. Nevertheless, with the progress of personalization and adaptation

Figure 2. Conceptual Hypertext Layers (simplified)



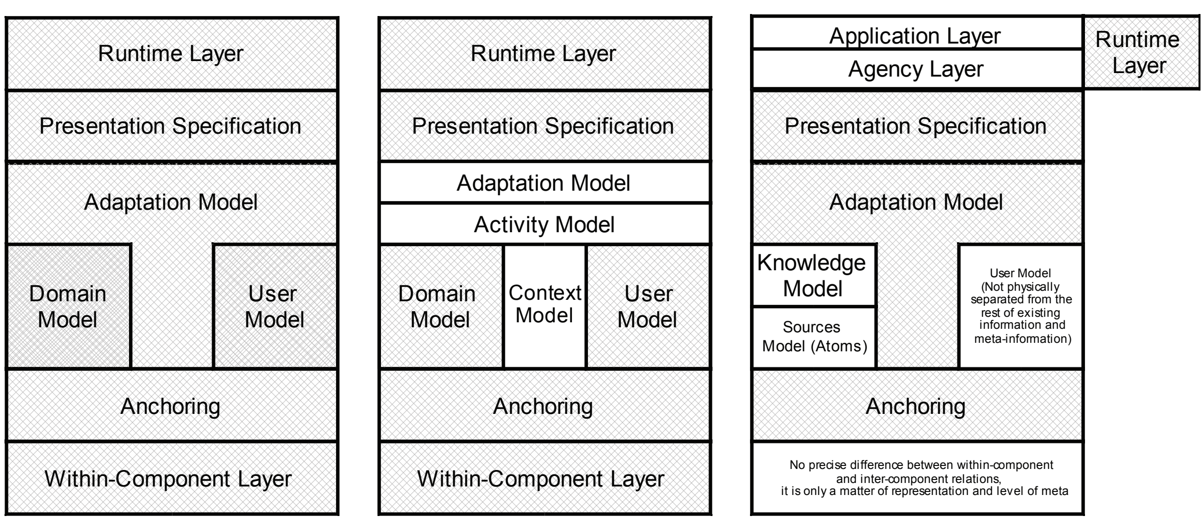
techniques and technologies, we can safely assume that the Semantic Web can compensate this lack of reasoning skills by building user models which can by continuous learning be able to "know" the user more than he knows himself, and thus agents using this model can play the role of the tutor.

7. Different approaches for linking across documents exist on the current Web, they can range from the simple embedded links found in HTML to the conceptual, more advanced linking as what we can find in the RDF family. We are also able to conclude something from examining that the last version of the XML Linking Language (Xlink) (DeRose, Maler, & Orchad, 2001) was released in 2001 and no further development took place, a fact that can support the idea of (Ossenbruggen et al., 2001) that taxonomic hypertext systems might benefit more from ontology-oriented languages than from languages oriented towards navigational hyperlinks such as Xlink. Runtime computation of links and anchors is an appealing idea, using that in addition to statically defined links and anchors that are defined at authoring time can add much flexibility and self-stabilizing ability to hypertexts systems and enhance their infrastructure required to become adaptive systems. They can enjoy the benefits of ontology driven linking that allows documents to be linked via metadata describing their contents and hence improving the consistency and breadth of linking at retrieval time and authoring time (Carr et al., 2001). In addition to the agent-based navigation assistance (El-Beltagy, DeRoure, & W. Hall, 1999).
8. RDF-enabled search engines have the potential to provide a significant improvement over the current keyword-based engines, especially when it comes to metadata and structure-based searching. An example of such a system, albeit not using RDF for encoding its semantic annotation, is the Ontobroker system (Decker, Erdmann, Fensel, & Studer, 1998) which has matured and went commercial, available under Ontoprise (http://www.ontoprise.de/).
9. Beyond the classic user interfaces: what was really special and original, is what Nintendo did by releasing its fifth home video game console known as Wii (Wii: The Total Story, n.d.), Wii didn't introduce the tremendous cutting-edge graphics, nor the market-braking High Definition audio and video, but it introduced a revolutionary controller nicknamed "Wiimote", this wireless pointing and movement detecting device had simply provided a new way of interaction between humans and games, opening new dimensions and exploring new areas that were ignored before in the development of home video games console. The domain of human machine interaction was also ignored in the various existing hypertext models, we can safely say that about Dexter model since it had abstracted away from the user interface details, also Open Hypermedia research considered the user interface as part of the application's functionality and it was more or less ignored. However in adaptive hypermedia the presentation and interactive behavior of hypermedia structures is more complex than the typical button-like behavior of navigational links, and is often tightly intertwined with the underlying semantics of these structures (Ossenbruggen et al., 2001). The ability of the Semantic Web to model the semantics of hypermedia structures explicitly provides new opportunities to improve the hypermedia user interface.

When we take these points in consideration and try to reflect them on a piece of clay in order to get a model that can contain all these aspects, we will find that it is not as easy as it sounds. Nevertheless we still can define the broad strokes that form the clay into the desired adaptive hypermedia model (Figure 3c) and shake the thrones of existing models. Basing on the Enhanced AHAM (Kravčík & Gašević, 2006) which in turn is based on the Dexter-based AHAM (De Bra et al., 1999) the main effects on the different layers of the model are:

- **Within-component layer:** Most of what we used to call a component will be composed in runtime, by the accessing agents for instance. Composite components (e.g. a Web page) will not exist as physical stand-alone components anymore (except for caching purposes) and the pages that the user will see are composed by the system (agents) according to his preferences (Figure 2). Even the so-called atom component are not an exception, e.g. the system can autonomously compose a video out of a collection of images, video clips and audio clips, the same can be easily said about texts, images, audio content that used to be called atom components in Dexter. Of course there will be atom components in the system, components that cannot be divided anymore, but these ones will have more granularity, and they don't have to comply with the definition of atom components that comes from the human point of view. A different way for representing knowledge will be used especially when we consider the conceptual hypertext, and by using ontologies there can be no clear borders between components since ontologies can intersect, one can be part of another and some sources can belong to more than one ontology. Thus no precise differentiation between within-component and inter-component relations, it is only a matter of presentation and level of abstraction.
- **Domain model:** The Enhanced AHAM (Kravčík & Gašević, 2006) divided this model into two interconnected networks of objects, complying with the conceptual hypertext approach mentioned earlier. With different kinds of semantic relations between the concepts in the knowledge space in addition to the relationship between the

Figure 3. Three models in comparison



a - Adaptive Hypertext Applications Model(AHAM)

b- Enhanced Adaptive Hypertext Applications Model

c - The proposed model

hyperdocuments in the hyperspace which have different standards, such can be found for example in the Dublin Core Metadata Initiative ("dublincore.org"). The third kind of relationships is the one that interconnects the two networks, this kind can be thought of as the relationship between sources that "glues" a Web page together, it isn't the semantic relationship that we can find in the knowledge space, rather it is presentation specific, user specific, and is the core of the adaptiveness. So considering that the third kind of relationships belongs only to the domain model can be mistakable, these relations are the ones which will be constructed by the agents in order to "weave" the user-adapted Web.

- **User model:** Is the model that will define what the application should adapt to. It will exist as an ontology and thus it will not be physically separated from the rest of the existing information and meta information only loosely defined. This model can also contain fuzzy information in addition to the normal discrete ones. Different approaches to combine the user model and the domain model also exist such as Probabilistic Latent Semantic Models (PLSA) which is such a concept that integrates information about the domain with user data. A framework based on that concept is (Jin, Mobasher, & Zhou, 2004).
- **Adaptation model:** This model used to represent pairs or multi-end relations between the subjects of adaptation (Domain Model) and the specifications that the subjects must adapt to (User Model), these relations will still exist but not as a separate package, and not limited to any type or group of relations.
- **Presentation specifications:** This layer will remain and will have a similar role to its previous role, will also be connected to both the sources (components) and the agents accessing them, the presentation specifications will be a part of the meta data that is describing the information, and will not constitute a certain group as the group of presentation specifications of a component. The way the information is presented to the user is now a function of much more parameters, which may include a bigger portion of the properties of the information to be presented, and the preferences of the user (both in the storage layer) in addition to the specifications of the accessing agent (in the Agency layer).
- **Runtime layer:** This layer will be divided into two layers:
  - *Application layer:* Has the same role as what used to be called "Runtime Layer" and it has open-ended possibilities that it is hard and unworthy trying to have a model that abstracts all of them. Going beyond the classic user interface can be implemented in lower abstraction levels.
  - *Agency layer:* This is the layer that will contain the activities of different kinds of software agents, the features of agents are needed, The features of the way that agents work can be easily mapped into performing tasks that are adaptivity-related (Memari & Marx-Gomez, 2008). The results that are generated for humans may follow the Dexter model. but the way that these "pages" are generated is the task of agents and it is the part that will hold the adaptiveness. So they will take into consideration:
    - Individual user preferences (user model),
    - Knowledge about sources and other users that can be collected by acting as a Multi- Agent System (MAS).

The layer can have different kinds of agents each with different tasks and algorithm implementations e.g. rating agents, meta data agents, resolver agents (which will perform the "resolve" function described in Dexter), accessor agents, search agents and every other functionality imaginable can be assigned to different kinds of agents. This kind of task assignment can give the system a great amount of flexibility, responsiveness and self-* features; adding, removing and modifying functionalities can be a straight forward process, enhancing a specific feature can be thought of as increasing the number of agents assigned to it, all that can be done in runtime without having to stop the system or recompile from source. That can greatly help breaking down complex and heavy-data-requiring problems. This layer will have a standardized interface with the application layer, application layer will have access to the user model only through the agency layer, this will facilitate the content adaptation but may make the presentation adaptation a little harder.

AN APPLICATION ACTIVITY

In order to get a clearer picture about the role of agents in the process of adaptivity, we can consider the following scenario: let's assume that we have an SOA-Enabled Application, based on a Service Oriented Architecture (SOA), see (Newcomer & Lomow, 2004). That means that the user who will be using this system, will be dealing with it as a service, and this system in turn will be composed of and using other services. This architecture is based on the component-based architecture which has the benefit of simplifying the design of the software, and increasing the reusability of sections of code (components) by making each component appear as a black box with inputs and outputs, so the software designer doesn't have to worry about what's inside or how tasks inside are done. Services in SOA have taken the place of normal software components. The advantage of such an approach is to further simplify the application design and to maximize reusability through letting each component run on a different machine, so it is a total black box to the designer, in a way that he doesn't have to care even about the requirements that a component needs to run properly (because that will be the task of the service provider).

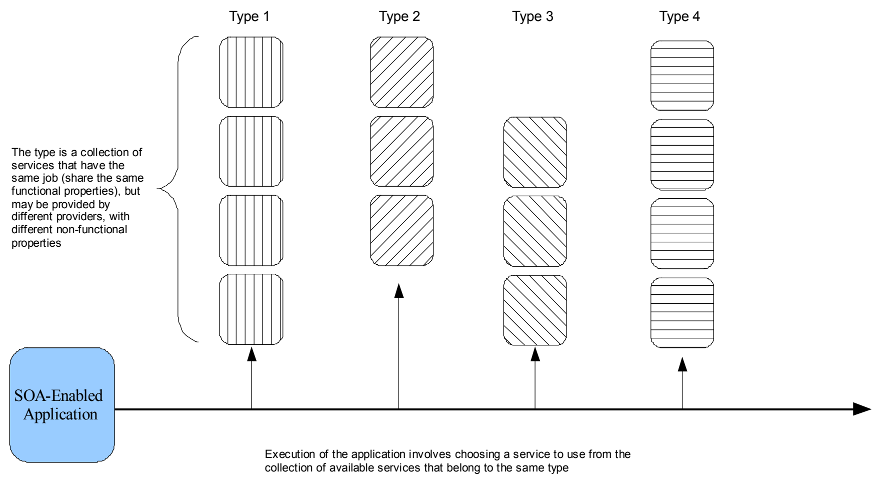
As illustrated in Figure 4 we can see that there are different possibilities for using the available services. Throughout the execution of the application, it will utilize different types of services along the way.

For each type of these services, there is a variety of implementations provided by different vendors, and running on different hardware, that will make them vary in price of use, delay of execution, accuracy of the results etc. which are machine-measurable attributes, additionally they will vary in some attributes that are unmeasurable by machines, like the way a picture looks, or the way a diagram is shown...

Here we can detect two visible problems, the first is that with the huge amount of available services, we are facing the problem of categorizing these services into the appropriate types, since the point of view of the user is always different from the point of view of the programmer, and the essence of reusability is in using the component (service) in different ways, the programmer can only partially categorize his service, and the main burden of this is on the user.

The second problem comes after the categorization, because in each category, there will be a big amount of available services that vary in lots of machine-measurable and machine-unmeasurable attributes as mentioned above, on top of that the sea of provided services is restless, they may change in number, availability, price, quality etc. every minute. That will make the process of taking a

Figure 4. Execution process of an SOA-enabled application



decision about which service of each type to use, too complex for any human mind to solve.

Now let's assume that our application complies to the aforementioned model, that means that the underlying platform is the Semantic Web (or a semantic represented knowledge), also means that we have an agency layer in our application which will be responsible of adaptivity.

In the light of that, let's have a spotlight on the two aforementioned problems, which are summerizable to: a representation problem and an effective usage problem, see (Walton, 2006, p. 3).

The problem of categorization will be solved by representing the service semantically using ontologies as denoted in the vision of the Semantic Web, which means declaring the "What exactly this service does" in a way that is understandable by machines, the process might sound like flipping the service inside out, or penetrating the black box but that's not the case, because we are not declaring how this service works but only what. By using ontologies to describe that, we can guarantee that through the generalization relations within the ontologies, these services are easily categorizable.

The second problem is solved by the agency layer since agents have more processing power than human brains, and they are also in a restless condition and in a constant motion towards better adaptation with the environment, choosing autonomously and in runtime the best possible path among all the available paths, according to certain desires and criteria of the user, in order to accomplish the goals (functionality) of the application.

Now let's have a more concrete example, a simple application that calculates distance between cities, and shows a map to display the shortest path.

A quick look at this application will suggest that we use three types of services: a service that calculates the shortest path between two given cities, another service that calculates the length of a given path, and a third one that displays a map with a given path on it. Assuming that the services are semantically described, the agents will try to find and categorize the relevant services, let's suppose an agent finds a service that does "cartography" and draws "tracks", the agent can reason about that service using relations in the describing ontology that connect "cartography" to "map" and "track" to "path" and conclude that it is relevant and we need it to fill the third position. In the same way the agent can determine that a service that calculates the shortest path between metropolises is relevant because there is an "is a" relation that connects a city to a metropolis.

After getting the results and categorizing them the agent has to choose an appropriate service from each category, according to the preferences of the user.

The preferences of the user are described in the user model which can be modified manually or by learning as mentioned before. The manual modification interface can look like an audio mixer, with different qualifications displayed as trackbars, and the mission of the agent is to find the possible mixture, that best satisfies the given criteria. These criteria in addition to the precise values can have also fuzzy values, so the "price of use" attribute can have values like "0.5", "< 0.2", "0.3 - 0.6" as well as values like "reasonable", "expensive", "cheap"...

After deciding about the best combination of services (which is the first stage of adaptation), it's time to get feedback from the user about the final result. The user might dislike some measurable aspects of the results, like the price or speed or resolution of the displayed map or the accuracy of the resulting distances... in this case, the agent can easily detect the responsible service and try to find other alternative for it.

But sometimes the user might give unmeasurable impressions like: he doesn't like the result, or the result is ambiguous or the result is not worth the price paid for it, in this case a combination of services as a whole is responsible rather than one service, so the agent has to find a better combination to better suit the user. That calls to mind the genetic algorithms, that might be one good solution to use in such case.

The most important issue at this stage, which will draw the border between the agent and the Decision Support Systems (DSS) is that instead of suggesting a possible solution to the user, our agent will act by itself trying to reform the combination of services in order to have a result that conforms more to the needs and preferences specified. In order to do this, the agent must have a margin of freedom, which is be specified by the user through the "audio mixer" interface, the width of this margin reflects the trust that the user gives to the agent.

CONCLUSION

We have presented here the reasons behind the emergency of adaptiveness and personalization in current environments. The example above goes about a simple problem from the usefulness point of view, but real world business problems are more complex and complicated, most of them cannot even have a solution on current hardware within a reasonable time span using conventional methods. The physical limits in front of increasing the hardware resources are another reason for us to think about alternative ways of getting problems solved, and among them: adaptation.

Adaptation has been thought about a long time ago, however it has not been as necessary and urgent as it is today. Models and reference models were conducted in order to capture the abstractions of existing and future systems; we have briefly presented some of them and spotted the light on a few points to be considered for the upcoming hypermedia research.

Finally we tried to look at the Enhanced AHAM model through the eyes of these nine points, taking the Semantic Web as the platform, giving a rough sketch of a new model that tries to abstract the functionality and structure of emerging adaptive hypermedia systems.

No matter whether we use SOAP & WSDL or the simpler alternative REST, whether we use RDF & OWL or the user-friendly XHTML &

Microformats, the Web is definitely going smarter, and that's the future of it: the Semantic Web.

REFERENCES

Bailey, C. P. (2002, December 1). *An agent-based framework to support adaptive hypermedia*. Retrieved June 6, 2008, from http://eprints.ecs.soton.ac.uk/7248/

Bratman, M. E. (1999). *Intention, plans, and practical reason* (p. 208). Center for the Study of Language and Inf.

Brin, S., & Page, L. (1998). The anatomy of a large-scale hypertextual Web search engine. *Comput. Netw. ISDN Syst., 30*(1-7), 107-117.

Carr, L., Hall, W., Bechhofer, S., & Goble, C. (2001). Conceptual linking: Ontology-based open hypermedia. In *Proceedings of the 10th international conference on World Wide Web* (pp. 334-342). Hong Kong, Hong Kong: ACM. doi: 10.1145/371920.372084

Cristea, A., Stewart, C., & Sirmakessis, S. (2006). Adaptivity, personalization, and the Semantic Web. In *Proceedings of the joint international workshop on Adaptivity, personalization & the Semantic Web* (pp. 1-2). Odense, Denmark: ACM. doi: 10.1145/1149933.1149934

De Bra, P., Houben, G., & Wu, H. (1999). AHAM: A dexter-based reference model for adaptive hypermedia. In *Proceedings of the tenth ACM Conference on Hypertext and hypermedia: Returning to our diverse roots* (pp. 147-156). Darmstadt, Germany: ACM. doi: 10.1145/294469.294508

Decker, S., Erdmann, M., Fensel, D., & Studer, R. (1998). Ontobroker: Ontology based access to distributed and semi-structured information. In *Proceedings of the IFIP TC2/WG2.6 Eighth Working Conference on Database Semantics-Semantic Issues in Multimedia Systems* (pp. 351-369). Deventer, The Netherlands: Kluwer, B.V. Retrieved June 25, 2008, from http://portal.acm.org/citation.cfm?id=647498.728000&jmp=cit&coll=GUIDE&dl=GUIDE

DeRose, S. J., Maler, E., & Orchad, D. (2001, June 27). *XML linking language* (XLink). Retrieved from http://www.w3.org/TR/xlink/

Devedzic, V. (2006). *Semantic Web and education* (1st ed., p. 353). Springer.

Dodd, R. (2008). 20 years on: The dexter model of hypertext and its impact on Web accessibility. *SIGACCESS Access. Comput.,* (90), 3-15.

El-Beltagy, S., DeRoure, D., & Hall, W. (1999). A multiagent system for navigation assistance and information finding. In *Proceedings of the Fourth International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology* (pp. 281-295). Retrieved June 25, 2008, from http://citeseer.ist.psu.edu/el-beltagy-99multiagent.html

Freeman, S., & Herron, J. (2003). *Evolutionary analysis* (3rd ed., p. 816). Benjamin Cummings.

Gómez-Pérez, A., & Corcho, O. (2002). Ontology specification languages for the Semantic Web. *IEEE Intelligent Systems, 17*(1), 54-60.

Grønbæk, K., Bouvin, N. O., & Sloth, L. (1997). Designing dexter-based hypermedia services for the World Wide Web. In *Proceedings of the eighth ACM conference on Hypertext* (pp. 146-156). Southampton, UK: ACM. doi: 10.1145/267437.267453

Halasz, F. (1988). Reflections on NoteCards: Seven issues for the next generation of hypermedia systems. *Commun. ACM, 31*(7), 836-852. doi: 10.1145/48511.48514

Halasz, F., & Schwartz, M. (1994). The dexter hypertext reference model. *Commun. ACM, 37*(2), 30-39. doi: 10.1145/175235.175237

Hardman, L., Bulterman, D. C., & Rossum, G. (1993). *The Amsterdam hypermedia model: Ex-*

*tending hypertext to support *real* multimedia.* CWI (Centre for Mathematics and Computer Science). Retrieved June 16, 2008, from http://portal.acm.org/citation.cfm?id=869550&dl=GUIDE&coll=GUIDE

Heflin, J., & Hendler, J. (2001). A portrait of the Semantic Web in action. *IEEE Intelligent Systems, 16*(2), 54-59.

Heflin, J. D. (2001). *Towards the Semantic Web: Knowledge representation in a dynamic, distributed environment.* University of Maryland at College Park. Retrieved June 19, 2008, from http://portal.acm.org/citation.cfm?id=933917&dl=ACM&coll=portal

Jin, X., Mobasher, B., & Zhou, Y. (2004). A unified approach to personalization based on probabilistic latent semantic models of Web usage and content. In *Proceedings of the AAAI 2004 Workshop.* San Jose, CA. Retrieved July 26, 2008, from http://citeseer.ist.psu.edu/715309.html

Koch, N., & Wirsing, M. (2002). The Munich reference model for adaptive hypermedia applications. In *Adaptive hypermedia and adaptive Web-based systems* (pp. 213-222). Retrieved June 6, 2008, from http://dx.doi.org/10.1007/3-540-47952-X_23

Kravčík, M., & Gašević, D. (2006). Adaptive hypermedia for the Semantic Web. In *Proceedings of the joint international workshop on Adaptivity, personalization & the Semantic Web* (pp. 3-10). Odense, Denmark: ACM. doi: 10.1145/1149933.1149935

Iverson, L. (2003). *Lee Iverson: The dexter hypertext reference model.* Retrieved June 6, 2008, from http://www.ece.ubc.ca/~leei/Weblog/2003/07/the_dexter_hypertext_reference.html

Lommatzsch, A., Mehlitz, M., & Kunegis, J. (2007). *A multi-agent framework for personalized information filtering.* Retrieved June 6, 2008, from http://edoc.mpg.de/316556

Memari, A., & Marx-Gomez, J. (2008). A model for adaptive applications on the Semantic Web. In *Proceedings of the 3rd International Convference on Information and Communication Technologies: From Theory to Applications*, 2008 (pp. 1-6), Damascus.

Meyrowitz, N. (1989). The missing link: Why we're all doing hypertext wrong. In *The society of text: Hypertext, hypermedia, and the social construction of information* (pp. 107-114). Cambridge MA: MIT Press. Retrieved June 19, 2008, from http://portal.acm.org/citation.cfm?id=68452.68459

Millard, D. E., Moreau, L., Davis, H. C., & Reich, S. (2000). FOHM: A fundamental open hypertext model for investigating interoperability between hypertext domains. In U*K Conference on Hypertext* (pp. 93-102). Retrieved June 16, 2008, from http://citeseer.ist.psu.edu/millard00fohm.html

Newcomer, E., & Lomow, G. (2004). *Understanding SOA with Web services* (p. 480). Reading, MA: Addison-Wesley Professional.

Ohene-Djan, J. F. (2002). *A formal approach to personalisable, adaptive hyperlink-based systems.* Unpublished doctoral dissertation, Department of Mathematical and Computing Sciences Goldsmiths College University of London.

Ossenbruggen, J. R., Hardman, L., & Rutledge, L. W. (2001). *Hypermedia and the Semantic Web: A research agenda.* CWI (Centre for Mathematics and Computer Science). Retrieved June 20, 2008, from http://portal.acm.org/citation.cfm?id=869006&coll=GUIDE&dl=GUIDE

Röscheisen, M., Mogensen, C., & Winograd, T. (1995, February 21). *Shared Web annotations as a platform for third-party value-added information providers: Architecture, protocols, and usage examples.* Stanford, CA: Stanford University.

Rouet, J. F. (1992). Cognitive processing of hyperdocuments: When does nonlinearity help? In *Proceedings of the ACM conference on Hy-*

pertext (pp. 131-140). Milan, Italy: ACM. doi: 10.1145/168466.168508

Saksena, M., & Cristea, A. (2006). Towards more efficient generic semantic authoring for adaptive hypermedia. In *Proceedings of the joint international workshop on Adaptivity, personalization & the Semantic Web* (pp. 11-20). Odense, Denmark: ACM. doi: 10.1145/1149933.1149936

Vrieze, P. D., Bommel, P. V., & Weide, T. V. D. (2004). *A generic adaptivity model in adaptive hypermedia.* The Netherlands: Eindhoven University of Technology. Retrieved June 6, 2008, from http://citeseer.ist.psu.edu/632739.html

Walton, C. (2006). *Agency and the Semantic Web* (1st ed., p. 272). New York: Oxford University Press.

Whitehead, E. J., Jr., & Wiggins, M. (1998). WebDAV: IETF standard for collaborative authoring on the Web. *IEEE Internet Computing, 2,* 34-40.

Wii: The total story. (n.d.). Retrieved November 19, 2006, from http://wii.ign.com/launchguide/hardware1.html

KEY TERMS AND DEFINITIONS

Adaptation: A process that includes in general the selection of relevant items from a set of items in a way suitable for given requirements or environment conditions.

Agency Layer: Is a layer in our proposed model which has standardized interfaces with the upper and lower layers and contains the activities of a wide variety of software agents.

Hypermedia: Is a term used for hypertext which is not constrained to be text, it can include graphics, video and sound.

Hypertext: Is a concept that is basically a text which contains links to other text.

Multi-Agent System (MAS): Is a system composed of multiple interacting intelligent agents, such a system can be used to solve problems that are impossible for any agent solely to solve.

Reference Model: An abstract representation of the entities and relations within a problem space; it forms the conceptual basis to derive more concrete models from which an implementation can be developed.

Semantic Web: An extension of the World Wide Web in which the semantics of the offered informational and transactional resources are provided and represented in a machine-understandable manner.