The Next Big Thing: Adaptive Web-Based Systems

Paul De Bra, Lora Aroyo and Vadim Chepegin
Department of Computer Science, Eindhoven University of Technology, Eindhoven, The Netherlands
Email: {debra, larioyo, vchepegi}@win.tue.nl

Abstract

At the ACM Hypertext Conference a panel discussed "The Next Big Thing Inc." in the area of hypertext and hypermedia. The Web has been the "Big Thing" during the past 10 years, but its success has also made it very difficult to find the appropriate information in an ocean of over 3 billion pages. Whereas search engines achieve incredible precision, they suffer from the same "one size fits all" approach that characterizes the Web sites they index. The paper defends the position that personalization, and in particular automatic personalization or adaptation, is the key to reach the goal of offering each individual user (or user group) the information they need. During the panel discussion there was debate about whether the user should always have access and control over the entire (hypertextual) information space. There were different views on whether the "right" to all the information is best guaranteed by offering tools that reduce the information space the user perceives so that the user can actually find and reach the information, or by offering unfiltered access to an ocean of information in which everything is available but in which perhaps nothing can be found. We argue in favor of adaptation but at the same time point out flaws in the way adaptive hypermedia has been used until now. The paper then proposes a new, modular adaptive hypermedia architecture that should lead to adaptive Web-based systems as the "Next Big Thing" indeed. In this architecture, different applications can collaborate in creating and updating a user model. Shared user model servers are not just needed for adaptive Web sites, but are also the key to enabling the development of ambient intelligence. (Many small systems then need to work together and base their actions on common knowledge about their user(s).) Sharing user models can of course cause a "big brother" problem. Legislation is already in place to protect users' privacy by placing legal limits on the kind of user modeling and sharing of user models that is allowed. The paper briefly reviews the legal issues of user modeling and adaptation in order to provide not just a future outlook based on "wild imagination" but based on a realistic vision of what will not only become technically possible but also of what will be "acceptable".

1 Introduction
Nielsen (1990) made two predictions for the future (within three to five years from publication):

- the emergence of a mass market for hypertext;
- the integration of hypertext and other computer facilities.

It has taken a bit longer, but other than that the predictions were spot on. To some extent World Wide Web technology has been used to make both these predictions come true. The Web itself is the best illustration of the mass market come true. With over 3 billion pages at the time of writing this paper - and that is only the part indexed by the largest search engines according to Sullivan (2003) - the Web has indeed reached the masses and is widely used all over the world. For the second prediction we look at help systems. Many software products come with documentation in (html) hypertext form, sometimes included with the product but increasingly available online as well and sometimes even only online. Traditional help systems, like Microsoft Windows help, also have hypertext functionality. Through the help subsystems, hypertext functionality has been effectively integrated with other computer facilities.

Hypertext, and the Web in particular, offers three ways to find information:

- When you know the precise location of the information you want, you type a URL in the browser's location field, and you have immediate access to that information. A bookmark list can help to remember the location of information you found before.
- When you have a good description of what it is you wish to find, the modern search engines are very good at finding the location of that information. While writing (the first draft of) this article the first author was watching a program on hackers on National Geographic and remembered that many years ago he found a security problem in the Solaris "tar" program. Typing the search terms "Paul De Bra", "tar", "vulnerability" and "solaris" into Google, the 16 hits were all copies of the CERT Advisory describing the problem that he reported 10 years ago.
- When you want to explore the Web hoping to find interesting information, for instance hoping to find the sorts of things "Paul De Bra" has been writing or where he has been mentioned on the Web, the real problem starts. You can browse and search all you want, but you will most likely not find all the information that is available about topics you are interested in at that time. When you start browsing the search results for "Paul De Bra", the above-mentioned report does not appear to be among the first 500 hits (out of over 2000). Browsing may provide access to all the information (that is linked) but it is still impossible to find all the information you are interested in.

The core problem in finding the information you want, in all the above cases, is describing what you want. Results from search engines are often disappointing because most search requests are too short and unspecific to yield good results. Once a Web site with interesting information is found, it is often difficult to navigate to interesting pages only, because the site can only be navigated using its predefined link structure, independently of the search request that brought you to that site.

The community of user modeling and adaptive hypermedia offers solutions for this problem: using information gathered about the user during the browsing process to change the information content and link structure on-the-fly. User modeling captures the mental state of the user, and thus allows that knowledge to be combined with the explicit queries (or links) in order to determine precisely what the user is looking for. To support the design of this user model-based adaptation, reference models like AHAM (De Bra et al. 1999, Wu 2002) and Munich (Koch and Wirsing 2002), both based on the Dexter Model by Halasz and Schwartz (1994), have been introduced in an attempt to standardize and unify the design of adaptive hypermedia.
applications, used mostly in isolated information spaces such as an online course, an electronic shopping site, an online museum, etc.

To become "The Next Big Thing", adaptive hypermedia systems need to open their architecture to allow collaboration between sites to provide better adaptation than is currently possible. The latest developments within the scope of the Semantic Web are leading adaptive hypermedia towards the "Adaptive Web" of semantically organized information in concept structures, where collaborative Web services allow for the encapsulation of the diverse knowledge, for the modularization of the architecture, and for a more dynamic and sharable framework for automated personalization or adaptation. In this way, current adaptive hypermedia architectures will be extended with powerful reasoning at the level of standardized concept schemes, as opposed to the traditional hand-crafting by authors of the domain concept relations, thus allowing adaptation generation in open information spaces on a higher schema level.

This paper first briefly recalls and summarizes the overall architecture of (Web-based) adaptive hypermedia systems (AHSs) to show how to extend and modularize this architecture to enable different AHSs to work together at different levels: the conceptual structure, the user models, and the adaptation. Last but not least, we point out some legal problems that arise as a consequence of opening the user model to be shared among various applications, which is technically possible but maybe legally unacceptable in the improved adaptive hypermedia architecture we propose.

2 Adaptive hypermedia architecture

On the Web personalization is being realized in two ways:

- Some Web sites require users to register and provide information about their interests. The site takes the interests into account in the presentation. The registered user profile is only updated when the user revisits it to make changes.
- Some Web sites require users to register only so they can be identified. Identification is also possible through cookies, in which case the user may not be aware of the identification taking place. The site monitors the user's browsing behavior (and in particular the pages that are visited) in order to create a user model representing the user's interests and knowledge. The presentation is updated according to the user model and the model is updated during the browsing process.

The former category of systems is called adaptable, the latter adaptive. A lot of research has been performed in the field of adaptive hypermedia. One sub-area has focused on coming up with new ways of adaptation, to offer the user better, more suitable, more acceptable and more visible guidance through the information the Web site has to offer. Another sub-area has focused on finding higher-level ways of describing aspects of the application domain and of the user that together form the basis for performing adaptation. In the AHAM reference model, by De Bra et al. (1999) and Wu (2002), the core of the architecture of an adaptive application is formed by three closely linked components: the domain model (DM), the user model (UM) and the adaptation model (AM). In AHAM and other models there is a close relationship between the three submodels of the storage layer, e.g. the user model is typically an overlay model of the domain model. This means that for every "concept" in the domain model there is a corresponding "concept" in the user model that represents how the user relates to that concept. In a learning application for instance, the user model will keep track of the user's knowledge of each of the concepts in the domain model. The interplay between the submodels is best illustrated by explaining how an AHS processes a user's request, using its adaptation engine:

1. The user requests a page by clicking on a link in a Web page. Every page corresponds to a "concept" in the domain model and the user model
The Next Big Thing: Adaptive Web-Based Systems: De Bra et al.: JoDI

which is an overlay model).

2. The system checks the suitability of the requested page for this user. For this it needs to look at concept relationships from the domain model and at values from the user model. A typical type of concept relationship that plays a role in determining the suitability is the prerequisite relationship. The adaptation rules to check whether the page is suitable are defined in the adaptation model.

3. The system performs updates to the user model through (other) adaptation rules, e.g. the knowledge value of the concept that corresponds to the requested page is increased, and is increased more when the page is considered suitable than when the page is considered not suitable. Also, knowledge often propagates from pages to sections to chapters.

The presentation of the requested page can be adapted through adaptation rules in two ways:

- The information content of the page can be altered, e.g. by conditionally including or hiding fragments. When a prerequisite is not met, an additional explanation can be inserted, or when an interest value is high, additional details can be shown. Another possible adaptation is the choice between media items based on preferences of the user.
- The links that emanate from the page can be manipulated. Links to pages that are considered not suitable can be annotated, e.g. with a red marker, can be hidden, or even removed. The link destination can be changed as well.

4. Optionally the adaptation to content or links (e.g. the inclusion of a fragment) can cause user model updates as well.

Many adaptive hypermedia systems have been developed, targeting various application domains, i.e. educational applications (ELM-ART developed by Weber and Brusilovsky (2001), Interbook developed by Brusilovsky et al. (1998), KBS-Hyperbook by Henze and Nejdl (2001), NetCoach by Weber et al. (2001)), recommender systems, or general-purpose adaptive hypermedia systems, like AHA! by De Bra et al. (2002) and De Bra et al. (2003). A detailed taxonomy of methods and techniques used in AHS is given by Brusilovsky (2001). The content and link adaptation possibilities given above are just a few from a much larger set described in that paper. Most AHSs have been developed with one application or application area in mind. A notable exception is the AHA! system developed as a general-purpose AHS with a centralized architecture shown in Figure 1.
Figure 1 shows that the three submodels (DM, UM, AM) all reside on a central Web server, together with the "local pages" (content fragments). In AHA! the engine handles page requests. (Pages can come from external servers, but the corresponding concepts must exist in the local domain model.) Each page request results in updates to the user model. That user model is used to perform the adaptation.

The centralized architecture of AHA! (and several other systems) has the advantage that the DM, AM and UM can work closely together. A typical optimization is to retrieve the DM and AM when the server starts, and the UM when the user logs on, and to keep all that information cached in memory, periodically saving the updates to UM to disk (or database). For adaptive systems to work together, as suggested in the introduction, a new, decentralized and modular architecture is needed, as will be presented in Figure 2. The issue of optimization is transferred towards the bridging protocols and the information request coordination between the DM, AM and UM in that architecture. We recognize that performance remains very important as a key to the success of adaptive information services in that they should be (nearly) as performant as non-adaptive services.

Apart from the issue of collaboration between adaptive systems, the centralized architecture has some other drawbacks: the semantic information about the application domain is usually stored inside a local domain and adaptation model. Relationships between concepts are "hidden" inside adaptation rules (although some AHSs provide a graphical interface - i.e. Graph Author in AHA!, Figure 1 - for the authors to define the concepts and
relationships and make high-level relationships plainly visible). It is difficult to "export" the semantic information contained in the DM/AM combination to applications that deal with concepts and relationships in another way. It is equally difficult to "import" semantic information, e.g. an ontology, into an AHA! application. Also, the user model cannot be accessed by external applications. It is only shared by applications running on the same AHA! server. Knowledge gained by the user through an online course cannot be used by an online course at another institute or department in order to initialize the user model there.

The next section looks at various research efforts attempting to provide solutions for these problems while facilitating adaptation to various users navigating a diverse and distributed open information space. Methods are proposed to improve the quality, consistency and linking of Web documents while the user is browsing, and for the authors when creating the adaptation. Reflecting on this work we propose a new, modular architecture of an open Web-based adaptive system that overcomes the above difficulties.

3 Modular adaptive hypermedia

While ubiquitous computing was still a far fetched vision when introduced by Weiser at the Computer Science Lab at Xerox PARC in 1988, now it is becoming a reality and a promising part of our everyday lives. After the age of mainframes (one machine - many people), and after the boom of personal computers (one machine - one person), now we are moving towards "the age of calm technology", as described by Weiser and Brown (1996) (one person - many computers and applications), or as Kay refers to it: the "Third Paradigm" of computing.

The essence of ubiquitous computing is to offer to users systems with "invisible" (ambient) intelligence, which will be able to know about the user and offer him/her the desired service, content and presentation, without intrusion and unnecessary human computer interaction. Adaptation across applications, seen as the "Next Big Thing" and realized with the notion of adaptive Web-based systems, defines the goal today: to provide adaptation within software environments, where the users are able to interact simultaneously with various applications. For this we need open and modularized architectures, which are able to interact, exchange data, and share components. A fundamental issue in such architectures relates to the coordination, handling and control of all the components (services) within them. One of the biggest challenges in this context is the sharing, synchronization and interpretation of the user model among the different applications. This way the user behavior within each system will be permanently evaluated and more detailed, and richer user models will be achieved in order to allow for enriched adaptation and personalization of the content. This paper presents our approach for achieving this openness and modularity of the adaptive hypermedia system architecture. To enable different AHSs to work together at different levels (e.g. conceptual, user model, and adaptation) we see the need for four main aspects:

- supporting a strict separation of domain model, application model, adaptation model and user model, to ensure good modularization of the system components;
- maintaining generic sharable (dynamic) models, such as the user model, to serve as a communication point for different AHSs;
- providing semantically rich descriptions of the components' functionality and their internal formats, to allow for interoperability among system components;
- providing mechanisms to describe the management, i.e. coordination and orchestration, of the communication between the system components.

The basic idea is that by augmenting encapsulated system modules with rich formal descriptions of their competence, we can further improve and also automate many aspects of the system management. The role of sharable models such as the dynamic user model is crucial for enabling inter-system
interactions, and providing sharability and reusability of user-related modules. Finally, by applying open Web standards we can enable interoperability and wide applicability of the adaptive hypermedia systems.

Currently research in the area of the Semantic Web, originating primarily from the knowledge engineering and artificial intelligence fields, with a special focus on ontologies and Web services, provides a number of standards and accompanying solutions which can be used to achieve the above-mentioned requirements. On the one hand we have the notion of ontology, which plays a role in facilitating the sharing of meaning and semantics of information between different software entities. A number of representational formats have been proposed as W3C standards for ontology and metadata representation. The most current advances with OWL exploit existing Web standards (e.g. XML, RDF and RDFS) and add the primitives of description logic as powerful means for reasoning services. One of the earliest research initiatives to illustrate this is the SHOE metadata annotation of Web content (Heflin et al. 1999). The idea was further elaborated in the CREAM framework (Handschuh et al. 2001), with use of ontology concept instances, considering evolving ontologies and offering annotation in a semi-automated way.

Several other annotation tools are known to the research community, e.g. the Amaya Web editor for RDF-based mark-up of resources while being created. Annotea also points out the advantages of a centralized server, and its unapplicability in "open" environments. It exemplifies a good scenario for collaborating authoring agents within "closed" content spaces, and illustrates its difficult implementation in an "open" information space. Other examples of annotation systems are given by the Enrich project (Mulholland et al. 2000), the CREAM-based Ont-O-Mat/Annotizer (Handschuh et al. 2001), MnM (Vargas-Vera et al. 2002), Letizia (Lieberman 1995), etc.

As the annotation is only part of the content authoring, another rather labor-intensive part is the process of linking the annotated content. The Conceptual Open Hypermedia Services Environment (COHSE) developed by Carr et al. (2001) introduces an ontological reasoning service over domain concepts and their relationships in combination with a Web-based open hypermedia link service: this enables documents to be linked via metadata describing their contents in a conceptual hypermedia system. Another recent project inspired by COHSE and applying ontologies and semantic services for the automation of semantic content annotation is Magpie (Dzbor et al. 2003). This project facilitates various interpretation views on the same content with no prior mark-up, but by means of adding an ontology-derived semantic layer (see Domingue et al. 2004). On top of this semantic layer Magpie deploys semantic services provided to the user as a physically independent layer over a particular Web resource.

Thus, the next step in the process of opening up AHS architectures is applying a Web services perspective on the system components. Web services make use of the above-mentioned semantics and offer means for flexible composition of services (system components) through automatic selection, interoperation of existing services, verification of service properties, and execution monitoring. In an approach such as DAML-S/OWL-S (DAML) for example, the ProcessControl Ontology provides a process definition in terms of its state, initial activation, execution, and completion. The ServiceModel, on the other hand, provides a means for describing the data flow and the control flow in the case of a composite service, and the ServiceGrounding specifies the service access to information by communication protocols, transport mechanisms, etc.

Another relevant approach for describing the role of Web services in system architecture is the Web Service Modeling Framework (WSMF) by Fensel and Bussler (2002). That research shows that Web services appear to be a useful solution for achieving modularization. We can achieve reasonable automation and dynamic realization of the main aspects of Web services (e.g. Web service location, composition and mediation) by extending them with rich formal descriptions of their competence (in standardized languages such as RDF or OWL). In this way we can allow adaptive Web-based
systems to reason about the functionalities provided by different Web services, to locate the best ones for solving a particular problem, and automatically to compose the relevant Web services for dynamic application building.

An interesting approach that could serve as the basis for a successful application of the Web service perspective on AHS architecture is given by an existing Web service framework like the Internet Reasoning Service (IRS-II) introduced by Motta et al. (2003). They show how we can support the publication, location, composition and execution of heterogeneous semantic-rich Web services. The service uses UPML (Unified Problem-solving Method description Language) to specify reusability in knowledge-based systems by defining how we can build elementary components and how these components can be integrated into one whole system, as described by Fensel et al. (1999). The IRS-II approach supports capability-driven service invocation (e.g. find a service that can solve problem X) because of the explicit separation of task specifications (the problems which need to be solved), method specifications (the ways to solve problems), and domain models (the context in which these problems need to be solved). This separation of system components actually fits quite nicely with the requirements for exploiting the Web service paradigm in the context of AHSs, as shown by Figure 2.
Figure 2. Architecture for adaptive Web-based systems (adapted from Motta et al. 2003)
Figure 2 illustrates our vision of the modular architecture for adaptive Web-based systems (Chepegin et al., 2003). One of the first characteristic aspects we observe is that the different system components are all equipped with facilities to communicate with the (other) components in terms of service invocations. In this architecture, bridges are used in accordance with the UPML framework connector defined by Fensel et al. (1999), in order to specify mappings between the different model services within the architecture. Ontologies also play an important role to define and unify the system's terminology and properties to describe the knowledge of each system service. Each service can be specified by means of a corresponding ontology, providing common ground for knowledge sharing, exploitation and interoperability among the services. This leads to a highly modularized architecture which offers a high degree of flexibility.

In the case of adaptive systems access to the user model via a Web service is a good example of this flexibility. It means that designers can design systems that will interact with or react to the user intelligently without knowing anything in advance about that user, but simply using the knowledge collected by other applications and interpreting it in the context of the current application. Crucial for achieving such a flexible architecture is the need for a standardized protocol for encoding information about users. As mentioned above, open standards (e.g. XML, RDF, OWL) allow for the specification of ontologies to standardize and formalize meaning and to enable reuse and interoperability. Another key aspect is to facilitate mobile user models that follow the user across applications. Results in the field of software agents provide implementation views supporting mobility and autonomous behavior. A final but important requirement is for the user modeling system to be able to reuse system and knowledge components, and thus benefit from other applications.

A second characteristic seen in Figure 2 is the separation of the different components. In the traditional AHS approach, exemplified by AHAM, three submodels are distinguished (section 2). When transforming these components into services the need for a fourth component, the application model, emerges. The main reason for this lies in the fact that in the traditional AHS approach the adaptation model unites the actual process of how to adapt with the decisions why to adapt. In most applications, e.g. those realized in AHA!, the designer's knowledge about why the user is served in a certain way is more or less left implicit: the adaptation model implements the way in which the information is adapted to the user and does so based on the designer's decisions, which are not made explicit. In the situation where we want to share and exchange the different functionalities between systems, it becomes relevant to separate the "how" and "why" in the adaptation. While capturing the designer's intentions about the roles, goals and tasks in the application (related to domain model concepts and user model values), the adaptation model restricts itself to the actual realization of the adaptation to follow the directions given by the application model. In fact this aligns well with the proposal from the IRS-II approach, mentioned above. The application model service contains a generic description of the user tasks in the context of a Role-Goals-Tasks model.

It is clear that this gives the application model service a crucial role in the system architecture. It divides the adaptation process into two parts, where the actual technical adaptation is performed by the adaptation model service, while the management of the service process is coordinated from the application model service. The entire architecture as displayed in Figure 2 emphasizes the fact that the core knowledge about the application processes and the user activities (tasks, roles and goals) lies in the application model service. In the interaction with the application the user is represented by a particular role (e.g. guest, a system expert, administrator, student). This role defines for him/her a corresponding behavior in terms of goals to achieve. To accomplish the user's goals appropriate tools (applications) are used, which realize one or several corresponding methods. The adaptation model service receives the direct user input and interacts with the application model service in order to define the context for the user input for its most precise adaptation. Further the adaptation model service queries the domain model service in order to select the relevant content to be presented to the user. The domain model service is responsible for the explicit storage and description of the domain knowledge in terms of concepts of a domain ontology. Finally, it updates the user model with new values. For instance, when users work with a selected application, every action they perform on the user interface is communicated to the user model service, which is responsible for updating the user model with the new values. The user information is
stored there and a reasoning engine infers new knowledge from it and makes predictions concerning future user behavior. In this way the user model service allows for sharability of the user model between applications by following the user (inside and outside the system) in order to collect and further analyze data about the user's activities.

The following section discusses some possible disadvantages and legal problems raised by this open modularized architecture with respect to the user model exchange and access by various collaborating parties.

4 Legal issues in adaptive hypermedia

We do not claim to have any meaningful expertise in legal matters. However, before people start envisioning a great future for user modeling services that enable companies to perform personalization and adaptation for new and existing customers based on a "big brother" type of server that sells them a comprehensive overview of everything a user has done online, it is only fair to summarize a few existing and emerging efforts to make offering such services illegal (and in our opinion rightfully so). Kobsa (2002) points out that whereas privacy laws vary between countries, there are common principles, and some countries decide that their law applies to everyone who performs actions that have an effect in these countries. As a result it is wise to abide to the strictest laws when creating adaptive applications. Some noteworthy issues are:

- Usage logs must be deleted after each session. This implies that adaptive systems may store the effect of user actions in a user model, but must not keep a log of "raw" actions. AHA! version 1.0 for instance kept a permanent log of user IDs and timestamps of every access to a page and of the time a user left the page. In AHA! version 2.0 the logging functionality has been removed, and in a future version we will introduce a log that is kept for the duration of a session only. It would be legal to keep a complete log when the user agrees, but only if the system would offer the same functionality when the user disagrees and when the system does not keep a log.

- Usage logs of different services may not be combined. It is still unclear what the impact of such laws is on the proposed modular architecture in which the user model becomes a service that can be used by different and distributed applications. We would of course like to create user models that combine knowledge about the user, gathered from different sites, but this may or may not be allowed depending on the interpretation of the word "service". We would argue that an online course program is one service, thus justifying the sharing of user model information between applications or courses within that program. Likewise, different departments of an electronic shopping mall that interact with customers independently could be considered one service, and thus share a user profile for each user. However, it remains to be seen whether this interpretation will be followed by the courts.

- Users should have a real, voluntary choice between an "anonymous" login (or login using a pseudonym) and a normal login using their real identity. As of AHA! version 2.0 anonymous logins are possible, yielding the same adaptation as non-anonymous logins. It helps to provide users with a comprehensive and intelligible advance notice about the data that is to be collected, and to indicate the purposes for which this is being done.

- User modeling should be "transparent", or "scrutable": users should be able to inspect their user model and understand what it means. They should also be allowed to change or erase data, especially the assumptions the system inferred about them.

- The user models should be protected using adequate security measures. When user modeling becomes a service, accessible by multiple applications, it becomes non-trivial to shield user models from intruders and eavesdroppers.

This short list is far from complete, but it already shows that for adaptive Web-based systems to become a "big thing" there are a number of legal issues
that need to be clarified and then implemented into the systems.

5 Conclusions and future work

Adaptive Web-based systems are ready to make the jump from single applications to modular distributed frameworks in which multiple applications can share user models and adaptation rules. The challenge for the future is to get research groups to work together to develop standards for exchanging information at the user model and adaptation model level, so that different systems can indeed start to share user modeling and adaptation information. While new technology is being developed we also need clarity in the legal issues involved in sharing user modeling information. We are opposed to a "big brother" style of user modeling, but we are hopeful that with voluntary user consent some sharing will be allowed under controlled circumstances. With many applications of adaptive Web-based systems, with small clusters of collaborating, distributed, modular systems, the idea of adaptive systems can still become a "next big thing".

Acknowledgements

We would like to thank the anonymous reviewers for their helpful comments, and also Geert-Jan Houben for fruitful discussions leading to an improved description of the modular article.

References


http://lieber.www.media.mit.edu/people/lieber/Lieberary/Letizia/Letizia-AAA/Letizia.html

http://www.cs.unibo.it/~gaspari/www/iswc03.pdf


http://searchenginewatch.com/reports/article.php/2156481


http://www.sis.pitt.edu/~peterb/papers/JAIEDFinal.pdf


**Links**

Amaya Web Editor  
http://www.w3.org/Amaya/
Annotea http://www.w3.org/2001/Annotea/


DAML Services http://www.daml.org/services/