

# Applying Adaptive Hypermedia Techniques to Semantic Web Service Composition

Owen Conlan, David Lewis, Steffen Higel, Declan O'Sullivan, Vincent Wade  
Knowledge and Data Engineering Group (KDEG),  
Department of Computer Science, Trinity College, Dublin 2, Ireland.  
{Owen.Conlan, Dave.Lewis, Steffen.Higel, Declan.OSullivan, Vincent.Wade}@cs.tcd.ie  
<http://kdeg.cs.tcd.ie>

## Abstract

The semantic web has the potential to revolutionize the ways in which users search and manipulate the rapidly expanding amount of information on the web. This expansion, coupled with the capability of this information to be semantically related in a multitude of different ways, leads to increased possibilities for the Web user becoming 'lost in hyperspace'. This danger stems from the fact that technologies which encourage semantic markup do not necessarily encourage appropriate vocabulary usage, resulting in inappropriate or misleading semantic relationships being formed. Users' expectations from information services on the Web are also increasing. This will lead to a proliferation of web services in this Semantic Web. Adaptive techniques, and in particular those used in Adaptive Hypermedia (AH), may be used to alleviate the semantic overload, while at the same time meeting the raised expectations of users. This approach leads to a number of research questions: How can Web Services be (successfully) used in an adaptive environment? and How do adaptive hypermedia and open hypermedia combine? This paper argues for the integration of semantic web, semantic interoperability, web services and AH technologies to meet and exceed the information requirements of users in the emerging Semantic Web.

## 1. Introduction

A Hypermedia system consists of Hyper-documents. Hyper-documents are nodes which have links through which the user can navigate to other nodes [3, 6]. This allows the information in the Hypermedia System to be accessed in a non-linear manner. The user can *jump* along a link from one page to another regardless of the physical location of the linked page in Hyperspace [6]. The content and link structure of a closed hypermedia system should be designed so that any possible path through the links is logically valid for the user. The World Wide Web, however, is an open hypermedia system that has not been explicitly designed and as such there may exist many linked paths that are not obviously valid for the user.

The WWW, as the primary example of an open hypermedia system, is undergoing a fundamental change as node types are developing from human readable media to web services that can be invoked directly by automated agents. Simple web services can be combined to create a more complex and feature rich composite service. Much work in service composition derives from workflow systems where a centralized server controls the flow of control and data between services. However, for service composition to scale successfully to the WWW, the service themselves must be configurable with the conditional links to the services with which they must communicate within a composition [1]. The services, therefore, become hyper-document or hyper-services. Service composition therefore involves the discovery, brokering, intercommunication and monitoring of these hyper-services.

A given service composition will have any number of attributes describing the function it performs, its input and output information, pre-requisites, required effects and other context information, and similarly so for the constituent services (which may themselves be composite services). The end result of a service composition should ensure that some form of input will be given to any service, operated upon and then given to the next service in the chain. At some point, the flow of information will reach a service which is an endpoint in the process, and the result will be passed to a person or device capable of interpreting it.

Some parallels are already visible between this concept and those in hypermedia. There is a navigation (possibly with multiple branches at different nodes) through a structure in service composition, just as there is in hypermedia. There are links between different nodes in service composition which could be statically embedded in the service or dynamically created to suit a differing set of requirements, as there are in hypermedia. This, however, is not enough to apply the more advanced and useful techniques in adaptive hypermedia to enable easier and more automated service composition.

This paper describes the challenges presented by service composition on the WWW and examines how parallels between service composition and adaptive hypermedia (AH) point to areas where advances in AH techniques could be exploited in service composition (and vice versa). The paper examines the potential benefits and challenges from exploiting the semantic mark up of web services in AH-based service composition techniques.

## **2. Web Service Composition**

Service Composition is the orchestration of a number of existing services to provide a richer composite service assembled to meet some user requirements. The current major interest in service composition, however, stems from the emergence of web services and the possibility of composing them to provide value-added services over the WWW. Service composition techniques typically involve expressing elemental services and composite services, the latter being compositions of elemental services and other composite services. The definition of composite services requires the expression of the flow of control and information between the elemental services. Techniques for this draw heavily on business process modeling and languages for enactable work flows.

The recent emergence of web services and their expression in the Web Service Description Language (WSDL) has resulted in XML based languages being used to define composition between WSDL service definitions, e.g. BPEL4WS, allowing service composition patterns to be readily exchanged between tools and thus reused. Such service composition specifications necessarily need to be expressed at the level of abstraction used for constituent services. Though techniques for developing business process models from business requirements are well established, they involve skilled manual activities and are not amenable to the dynamic service creation needed to meet the needs of average users lacking in business process modeling skills. A 'semantic gap' therefore exists between the model the user possesses of *what* they want to do, and the service composition models that simply express *how* this may be accomplished. Some work has been performed in reconciling the behavior of a system with a high level representation of its requirements. However, even these high level requirements are complex to express and are typically elicited by skilled requirements engineers. Adaptive hypermedia techniques have been successfully applied in areas such as e-learning, to dynamically map basic user profile information to adapt hypermedia documents to user's knowledge and preferences [4].

### **3. Adaptive Hypermedia Techniques**

Adaptivity is used to tailor a user's view of a subject to their personal requirements. Adaptive Hypermedia (AH) technologies are often used to guide a user through a body of digital material, assisting them in their comprehension of that material. There are several ways in which AH techniques may be employed to assist the user. These techniques may be classified along axes of adaptivity. The two axes from which adaptive techniques are most frequently utilized are the adaptive navigation and the adaptive presentation axes.

#### **3.1 Adaptive Navigation**

Adaptive navigation attempts to guide the user through the system by customising the link structure or format according to a user model. The form of adaptive navigation will determine the level of guidance and freedom granted to the user within the system. Hypermedia experienced users are known to be more likely to navigate in a non-linear way. Similarly users who are familiar with the subject matter are more likely to navigate non-linearly and therefore reap the benefits of Hypermedia systems [7]. The axis of adaptive navigation describes techniques that may be used to aid a user's exploration of a body of material. These techniques range from methods that restrict the user's interactions with the content, such as link hiding, to techniques that aid the user in their understanding of the information space, such as link annotation. Utilizing these techniques the user's path through an information space may be guided by the adaptive technologies. These techniques have generally been applied to closed corpus information spaces, where the content available is well defined. Adaptive Navigation techniques require some information about the nature of the resources in the information space to decide how links to that resource should be adapted for the user's requirements. They also require information about the user and their current objectives to effectively modify the navigation structure of an information space.

#### **3.2 Adaptive Presentation**

Adaptive Presentation is the customisation of content to match characteristics specified by the user model. The granularity may vary from word replacement to the substitution of pages or the application of different media. Content may be customised to contain additional information, pre-requisite information or comparative explanations. This form of adaptivity may be implemented by fragmenting the constituent content components into discrete words, phrases or paragraphs. These components or pagelets may constitute a discrete unit of information about a concept. With this approach different pagelets may be displayed for different users. An example would be a technical term or acronym with which the user is unfamiliar. The system may substitute the unfamiliar content until the user can be introduced to the technical term or acronym.

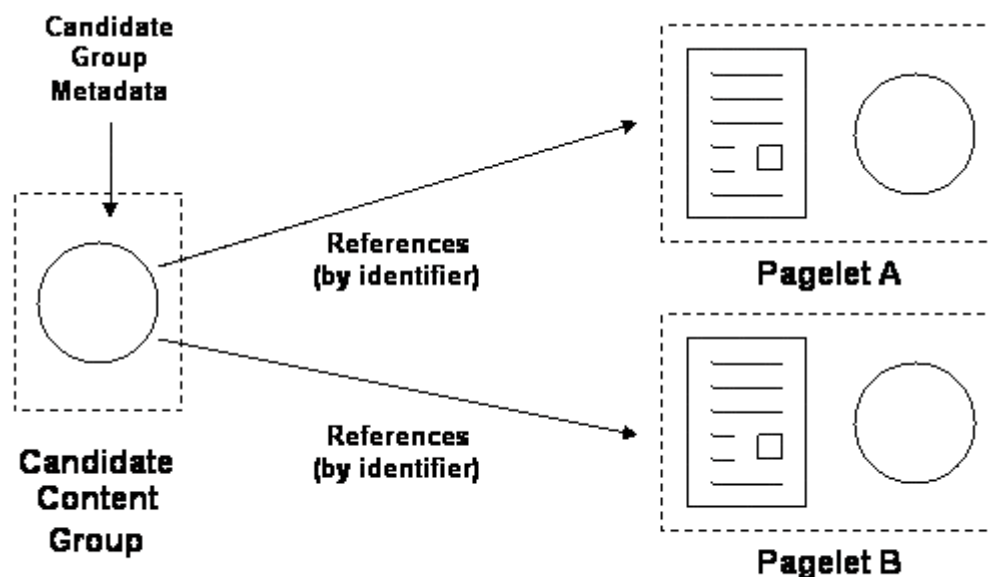
#### **3.3 Abstraction and Candidacy**

In APeLS (Adaptive Personalised eLearning Service), based on the multi-model metadata driven approach [4], the principles of abstraction and candidacy are key to facilitating effective navigation and presentation adaptivity. To facilitate flexibility in the design and implementation of new course offerings the multi-model approach was designed to include an abstraction mechanism. It was envisaged that this mechanism would enable the collaboration of many people on the development of an adaptive course. For example, the abstraction enables the course author (knowledge domain expert) to develop a narrative describing the course sequencing not in terms of the pagelets to be added, but rather using the learning

concepts to be added to realisations of the course. This abstraction allows the course author and instructional designer to design the course without necessarily being concerned with the individual pieces of content that will be used to populate the final course. Similarly the content designer can develop instances of learning content without knowing how or where that content will be used.

This abstraction is facilitated through candidate groups. Candidate groups are used to group together *like* models. For example, a candidate content group concerned with a particular learning concept may contain several pagelets, each covering the learning concept in different ways. In the case of pagelets these differences may be pedagogical or technical – some pagelets may deal with the concept from different perspectives or render the material differently for different devices.

Each candidate group has associated metadata that describes the role of the group and has the identifiers of the constituent models within the group. A simple candidate content group may be visualised as in figure 1.



**Fig. 1.** Candidate Content Group

In the figure above, the candidate content group has two candidates, pagelet A and pagelet B. Pagelets in the same candidate content group are equivalent on some axis, usually the learning concept they cover. Candidate groups can be formed for any set of models in the multi-model system. For example, there can be candidate narrative groups containing narratives that produce equivalent courses according to different instructional approaches.

As candidate groups refer to their candidates by identifier it is possible for any single model in the system to be included in multiple groups. Groups do not have to be homogeneous and may contain models of different types. For example, a candidate group may have some narrative and some pagelet candidates.

In the context of an open corpus environment the capacity for creating candidate groups is limited by the potentially vast number of candidates that may exist. This process, therefore, must be based on a non-manual mechanism.

## 4. HyperService Composition

We propose an architecture for a hypermedia composition system which integrates the composition of services by a domain expert and the adaptive integration service in response to user needs. The whole process should begin with the Service Composition Toolkit (SCT) being fed the general task requirements from a domain expert who has some understanding of how a task can be broken down into smaller tasks which could map onto a composition of different service types likely to be available. The SCT will generate an abstract service composition that can fulfill the proposed user task and store this in a candidate repository made available to the Service Integration Engine (SIE). The SCT also uses input from a repository of User, Service and Context Classifications, which are used to impose preconditions on the use of the abstract composite service to aid the SIE in later matching it to the requirements and context of specific tasks. To improve task to composition mapping over time, the repository of previously developed services can be analyzed to see if a close match already exists.

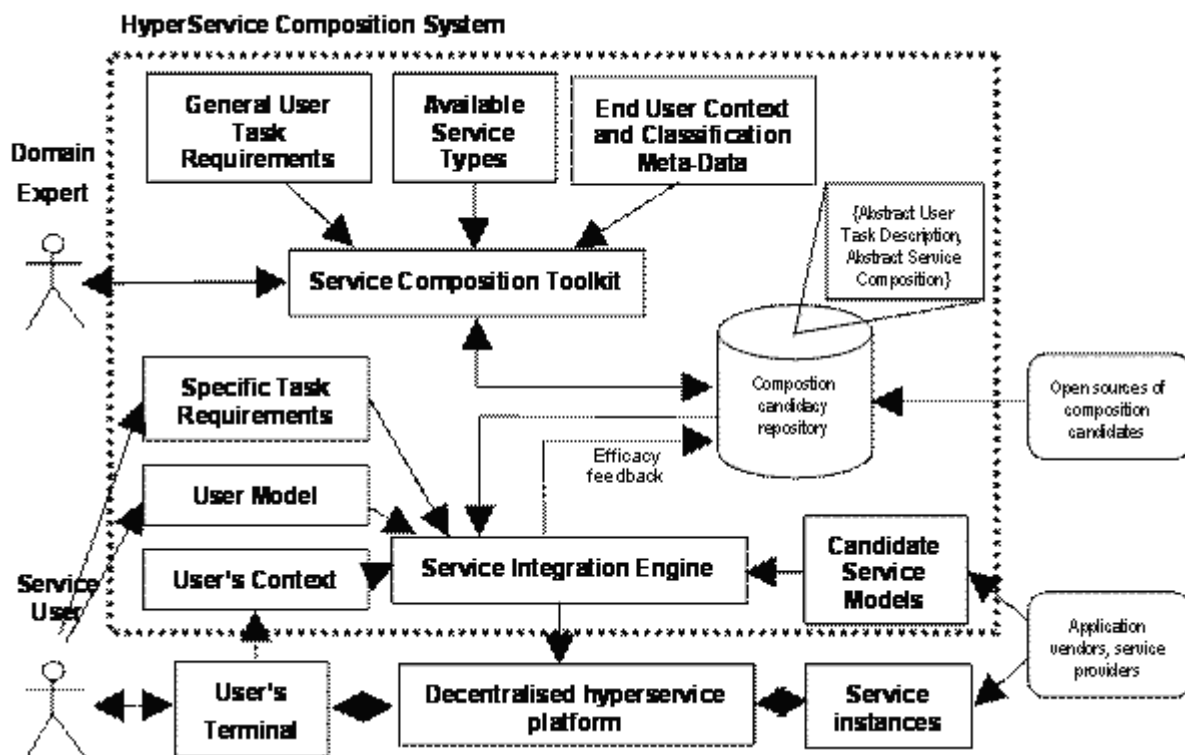


Fig. 2. HyperService Composition System

For any given task required by a user the SIE attempts to match the task requirements to general requirements and corresponding abstract service compositions contained in the candidate repository. Once a composition is selected the SIE attempts to resolve the constituent services onto available running services. The SIE also uses the User Model and User Context to personalize the composite service to the user. Failures to match the abstract composite service to concrete services or failure of user model and user context information matching the preconditions of the abstract service may result in an alternative abstract service being selected for adaptation. Successful uses of an abstract service composition are recorded in the candidate repository to aid in future matching by both the SCT and the SIE.

To illuminate this architecture further a mapping of the above concepts to existing concepts in adaptive e-learning engines is described below -

The list of sub-tasks required to complete the overall task could be mapped onto a list of learning concepts in a course (typically called a narrative). When transformed from e-learning to service composition, a sub-task like “pay for the transaction using a credit card before shipping it” could be mapped onto a learning requirement like “teach the student about regular expressions before teaching lexical analysis”.

The user model and context model could be combined to map onto the learner model of an adaptive e-learning engine. Attributes like “the user is walking down Grafton St.” and “the user would prefer to have an Irish company handle the payment for a transaction” could map onto attributes in e-learning like “the learner likes content to be explained with examples rather than theoretical descriptions” or “the learner responds well to visual explanations”.

Finally, there is the conceptual mapping of different, available services onto different, available pieces of content. In the APeLS system [conlan02], the adaptive engine is presented a number of pieces of candidate content, each of which would teach the same content but in a different manner. If the adaptive engine is presented with a candidate service, described in a vocabulary that can be mapped onto that which describes a candidate piece of content, it should be possible for the engine to select an appropriate service based on the user's preferences, as it would select a piece of content based on the same criteria.

To illustrate how this system may work in practice, an example of a course coordinator wanting to build a peer review system for her students will be used. Already developed and available services within the college might include:

- Document viewer Service - given a list of capabilities of a device and the file to be viewed, this service will convert a given document to some form which can conveniently be viewed on said device. It understands that PDF is computationally intensive, so conversion to a series of JPEGs of a given size and detail can be generated for easier viewing on a low powered device. For low bandwidth devices, a text only version could be displayed.
- Access Control List (ACL) Service - Specifies that a certain entity has the right to access a certain resource from a given service. This could be built on top of existing security frameworks like Kerberos [8].
- Document Repository Service - Allows retrieval and storage of files. It should be possible to specify rules which control what can be downloaded by who. It can verify such operations either from another service or from a local set of rules.
- Candidate Selector - Very generic service; it is supplied with a group of candidates (this can be anything from users to documents) and a metric by which to select one or many of them (e.g. which document has been viewed the least times or which user has done the least amount of work).
- Grading Service - A service that accepts a user name, the name of the resource that is being graded, the scale(s) on which they should be graded and what to do with the results (typically, they should be stored in a database).
- Generic Database service - Allows simple access to a persistent relational database. The service does not perform the storage itself... it simply provides the interface to any number of databases, including MSSQL, ORACLE, MySQL, Postgresql, Access, Xindice, etc. Its input will consist of a list of operations and a set of arguments for each. Output will consist of a return code, and any data that was requested.

- Results Generator - This will be specifically developed for the task of peer review. It understands the data stored in the database, how to manipulate it and how to present results in a manner which is meaningful to both students and course administrators. The input will consist of a user name, password, operation name and output will be text displayable in a browser.

The course admin will contact the service development toolkit and she will supply it with a task description with loose keywords, along the lines of "Peer review: Project Submission, grading, limited access, document viewing" The SCT will examine a history of service compositions, seeing if there is a close match for the given criteria. Assuming there isn't, it will have to request the input of some person who (or software that) understands the required outputs.

The resulting abstract service composition will contain a "flow-chart", outlining the different sub-tasks, their interaction with one another and various pre-conditions that must be met to allow this interaction to take place. An example could be forbidding a student to review another student's work until they themselves have submitted their project.

The list of sub-tasks for this scenario would probably include all of the afore mentioned services.

The service developer would submit this flow-chart to the SIE, which, using information based on the course administrator's circumstances and preferences, would fill in the sub-tasks with actual references to given services. A given lecturer might be very strict on plagiarism, or very concerned about security, so appropriate services could be invoked in the chain to ensure that these concerns are dealt with.

This description would be sent back to the service developer for review, to ensure that it functions correctly. If this is not the case, the flow-chart is refined and sent back to the service integrator. This process is repeated until a satisfactory result is produced.

Once a successful composition has been completed, the composition will be stored on a server somewhere, for students to access. Upon contacting the peer review compound service, the student will be asked to confirm their identity. Any running services under this instance of the composition will be marked as belonging to this user.

Firstly, the service should ask the top level results service should check to see if this user has submitted their project. If this is the case, the candidate selector should be provided with information on where to get a list of candidates (it should use the ACL service to get this information from the database service) and a metric through which it can choose which document to return.

The user may be using a PDA to view the document on the bus, so the document viewer service could convert it into an appropriate file format. Once the document has been read, the user will be given the option of submitting a review of this document. The grading service should validate that the inputs are within the correct bounds, and use the database service to store the results.

The course coordinator should be able to view submitted reviews, interpreted in a useful manner (to display various averages) with the possibility of saving the results to a spreadsheet file.

## 5. Further Work

Ultimately, accurate mapping of user level requirements to service composition requires modeling of the real world which forms the context in which the user naturally expresses their usage needs. Ontologies provide a way of capturing and exchanging models of the real world and making them available to automated agents. The DARPA Agent Mark-up Language initiative is defining XML-based standards for supporting the development of the Semantic Web [2], which aims to make the content of the web more amenable to processing by automated agents. As part of this initiative DAML-S [5] uses the DAML+OIL ontology language to provide semantic mark-up of web services. This includes modeling the links between a service and the outside world, by using ontologies for expressing the inputs, outputs, preconditions and effects of a service and the resources which provides it, in a way that can be mapped to semantic descriptions of the real world.

DAML-S aims to support the automation of web service discovery, invocation, composition, interoperation and execution monitoring. The principle element of the language is a Service. A Service has one or more Service Profiles which describes what the service does and includes input, output, preconditions and effects. A Service is described by a Service Model, which captures a Process Model of how the Service works. A Process can be atomic, in which case it is a single step execution, or composite, in which case it allow simple processes to be invoked through bindings to control constructs, similar to workflow control constructs, e.g. if-then-else, sequence, repeat-until, split and join etc. Both atomic and composite processes can be expressed in an abstract form called a simple process. This allows a more abstract version of a process to be defined, perhaps hiding specific preconditions or effects not relevant to a particular application or hiding the control constructs linking sub-processes of a composite service. A Service may have one or more Groundings which describe how a service may be accessed. This part of the language is integrated with WSDL, to exploit the latter's existing mapping onto communication protocols such as SOAP. WSDL is extended by using DAML+OIL for expressing the information types used in input and output parameters. Finally, the language includes the description of Resources, which provide the Service. These have types related to allocation and capacity of the resource in question, and can be composed in various forms providing guidance on the sharing of resources between services.

In terms of the proposed application of adaptive hypermedia techniques to service composition, the formal basis of DAML+OIL in description logic opens the door to the use of a range of automated reasoning techniques in the Service Development toolkit [9]. This could range from the use of rule-engines reasoning on simple conditions on service input and output to perform limited automated service composition [13] to the use of situation calculus to reason about semantic information in DAML-S specs in tools to aid developers in the simulation, verification and automated composition of web services [11]. The abstraction of existing services as DAML-S Simple Processes could be used to populate the repository of Service Types. Simple Process expressions could also be used to capture the users required inputs, output, preconditions and effect in terms that they can relate to that may be taken from the growing population of ontologies about the real world, e.g. the UNSPSC ontology of commercial product and service codes ([www.unspsc.org](http://www.unspsc.org)). Candidate service composition can therefore be expressed a DAML-S composite processes made up from Simple Processes from the service type repository. By using an open language such as DAML-S for these service composition candidates, exchange or purchase of successful candidates from other organizations could be undertaken, e.g. universities could exchange successful candidate compositions combining eLearning services and specific administration services, such as coursework assessment or fee collection.



Our work on the integration of adaptive hypermedia techniques with service composition is centering on eLearning in a ubiquitous computing environment, or 'uLearning'. Ubiquitous computing is seen as a ripe application area for semantic web technologies, since similar problems in terms of the number of potential services and huge range of potential compositions exist. Therefore we are actively developing a base ontology for describing uLearning resources and basic services, including mining existing ontologies on academic learning. This will provide the semantic definitions of terms use to express user's uLearning requirements and atomic services available for composition in this context.

One potential problem raised by defining service composition candidates at an abstract level is that when that composition is mapped by the Service Integration Engine onto available concrete services there is some mismatch between semantics implemented by the concrete services. In some cases these mismatches will mean the candidate composition is not viable for the user task at hand and another alternative must be sought. However, we envisage a large number of cases where the semantics of services that must interoperate within a composition are close enough that tailored interoperability services may be automatically generated. We have conducted some research into such semantic interoperability based on the automatic generation of XSLT scripts for adapting communications between services based on a Topic Map expression of their semantics [12]. A close analogy between the expression of composition in hypermedia authoring languages and Architectural Description Languages (ADLs) is identified in [10], with hypermedia nodes mapping to components in ADLs, which implement services, and links mapping to the concept of connectors which represent communication between components. However, ADLs elevate connectors between components to first class objects, whereas the corresponding links between nodes in hypermedia models tend not to be so. Connectors are the ideal abstraction for representing the semantic adaptation functions we describe above. We therefore propose to elevate links in hypermedia model to the same status as connectors, both to provide support for representing semantic adaptors with a HyperService composition and to allow such semantic connectors to be stored as candidates for use on other service compositions. Connectors could also be defined at a syntactic level, where interlinked services in a composition have different WSDL protocol and data format bindings in their Service Groundings and thus require a suitable syntactic connector. Connectors can also represent conditional interactions and n-array interactions between services, and so could be used as reusable sub-components within composition candidates.

## **6. Conclusion**

This paper outlines how adaptive hypermedia techniques may be applicable to the analogous field of service composition of web services in an approach we call Hyper-Service composition. An architecture is proposed that builds on an existing adaptive hypermedia engine by introducing a toolkit for developing service compositions to match user task requirements based on abstract models of existing services and known compositions. The adaptive hypermedia engine is then used to deploy concrete service composition on demand for a particular user task, based on a combination of the abstract service composition, the availability of concrete services and the user's current context. We then describe how this approach can be made more open by the use of ontologies and semantic mark-up of service descriptions, and describe research that directs us towards the use of semantic and syntactic connectors in HyperService compositions.

## References

1. Benatallah, B., Dumas, M., Sheng, Q.Z., Ngu, A.H.H. (2002), 'Declarative Composition and Peer-to-Peer Provisioning of Dynamic Web Services', Proceedings of the 18th International Conference on Data Engineering (ICDE'02), ISBN: 0-7695-1531-2, March 2002, pp 297 – 308
2. Berners-Lee, T., Hendler, K., Lassila, O. (2001), 'The Semantic Web', Scientific American, pp 35-43, Issue 284 (3), 17th May 2001
3. Brusilovsky, P., "Methods and Techniques of Adaptive Hypermedia" in User Modeling and User Adapted Interaction, 1996, v6, n2-3.
4. Conlan, O., Wade, V., Bruen, C., Gargan, M. (2002), 'Multi-Model, Metadata Driven Approach to Adaptive Hypermedia Services for Personalized eLearning', Proceedings of Second Conference on Adaptive Hypermedia and Adaptive Web-Based Systems, Eds. De Bra P., Brusilovsky, P., Conejo, R., Springer, LNCS 2347, May 2002, pp 100-111
5. 'DAML-S: Semantic Markup for Web Services', The DAML Service Coalition, <http://www.daml.org/services/>, October 2002.
6. De Bra, P., "Definition of hypertext and hypermedia", <http://www.wis.win.tue.nl/ah/>, 1998.
7. Eklund, J., Brusilovsky, P., "The Value of Adaptivity in Hypermedia Learning Environments: A Short Review of Empirical Evidence", <http://www.wis.win.tue.nl/ah/>, 1998.
8. Jochheck W., 'KERBEROS: Secure Network Authentication, Myth or Reality', Published online
9. McIlraith, S.A., Son, T.C., Honglei Zeng, H. (2001), 'Semantic Web Services', IEEE Intelligent Systems, 16(2), March/April 2001.
10. Muchaluat-Saade, D.C., Soares, L.F.G. (2001), 'Towards the Convergence between Hypermedia Authoring Languages and Architecture Description Languages', Proceeding of the ACM Symposium on Document Engineering, Atlanta, Georgia, USA, ACM Press, pp 48-57.
11. Narayanan, S., McIlraith, S.A., (2002) "Simulation, Verification and Automated Composition of Web Services", Proceedings of 11<sup>th</sup> World Wide Web Conference, May 7-11, 2002, Honolulu, Hawaii, pp 77-88.
12. Conlan, O. (2002), 'Enabling "Smart Space" Service Mobility Negotiation by Ordinary Users', Proceedings of Eurescom Summit 2002 Heidleberg, Germany.
13. Ponnekanti, S.R., Fox, A. (2002), "SWORD: A Developer Toolkit for Web Service Composition", To appear in The Eleventh World Wide Web Conference (Web Engineering Track), Honolulu, Hawaii, May 7-11, 2002 (<http://swig.stanford.edu/public/publications>)