

Implementing Link Services via Semantic Web Services Composition

Sanjay Vivek and David De Roure
Intelligent, Agents, Multimedia Group
Dept of Electronics and Computer Science
University of Southampton, UK
+44 (0) 23 8059 2418
{smv99r,dder}@ecs.soton.ac.uk

ABSTRACT

The Open Hypermedia model is based upon the separation of hypertext links from documents, treating them as separate, first class entities. Distributed link services take this approach a stage further, with separate link services providing and managing the links. In this paper, we describe the development of a system infrastructure for link services using the Web Services architecture. More precisely, we target the development of Web Services composition to enable integration and coordination between link services. A second prototype illustrates the use of Semantic Web technologies to implement a simple open hypermedia system. We discuss the role of Semantic Web technologies in the service-oriented approach to open hypermedia systems.

Keywords

Open Hypermedia, Web Services Composition, Semantic Web

1. INTRODUCTION

One of the advantages of the open hypermedia model, where hypermedia link information is stored and managed separately from the documents it describes, is the ability to deliver the right set of links to the right person (or program) at the right time.

Ongoing developments in networking infrastructure bring new requirements to such link resolution services, particularly with respect to distribution of the link information. We need to provide link services to users with a spectrum of connectivity beyond the intranets that have characterised previous projects. This includes home users, users connecting via service providers and through firewalls, and mobile users with limited bandwidth and intermittent connectivity. In fact these more restrictive modes of connection often apply to the users who stand to benefit most from an effective link service, providing a customised information space to improve the effectiveness of their interaction.

This context-sensitive determination and filtering of links, in a distributed and possibly mobile setting, imposes complex demands on the engineering of the Open Hypermedia system. In this paper we explore a service-oriented approach to this problem: a world of multiple services, including context-sensitive link resolution, brought together by service composition in order to provide the required link resolution processes. We have

adopted the Web Services approach to demonstrate composition, and we also exercise some basic technology of the Semantic Web.

In section 2, we describe open hypermedia and link services including the Fundamental Open Hypermedia Model (FOHM) [22] and the Auld Linky[21] contextual link server. Service composition is described in Section 3, using a tourism scenario to illustrate our adaptive hypermedia requirements. Our implementation, which uses a contextual link server, is presented in Section 4. In Section 5 we describe a preliminary experiment in the use of Semantic Web technologies to express link information. Section 6 describes how link services can be made more open by the use of ontologies and semantic mark-up of service descriptions, and future research directions is also discussed. Related work can be found in Section 7 and our conclusions in Section 8.

2. Open Hypermedia

2.1 Distributed Link Services

The open hypermedia model demands separation of link information from documents, typically achieved through the introduction of link databases (or 'linkbases'). In the early systems we adopted a modular approach to link resolution, using a series of software components called 'filters' in a standalone system ('Microcosm'[16]). Subsequently, we introduced the notion of the third party linkbase, implemented as a network service – this was the essence of the Distributed Link Service (DLS)[1]. A number of distributed architectures for the DLS were investigated[11],[13].

The motivation for the distributed implementation was to deal with multiple link resolution engines, which we must surely do – the full power of open hypermedia remains unrealised when there is only one linkbase. For specific applications it may be appropriate to hard-wire the combination of link services into one overall link resolution process. However, in general we wish to combine services dynamically, to meet the needs of the user in their current context.

Service-oriented architectures support a programming model that allows service components residing on a network to be published, discovered, and invoked by each other. Typically these services components interoperate with each other in a platform- and language- independent manner. The distinctive

characteristics of a distributed Web Services architecture are the size of the network being used and the underlying technologies involved.

Web Services extend the service oriented programming model into a vast networking platform that allows the publication, deployment, and discovery of service applications on Internet scale using Web technologies. These include ‘Simple Object Access Protocol’ (SOAP) for inter-service communication, ‘Web Services Description Language’ (WSDL) for service description and ‘Universal Description, Discovery and Integration’ (UDDI) for service directories. Web services readily enables linkservices to be published, deployed, and invoked by other likeminded services on both a global scale on the Internet, and also in a local-area peer-to-peer and pervasive scale. In this paper, we address issues related to modelling linkbase access as a Web service and implementing it within a Web process (composite Web services).

To investigate composition of services we are developing distributed link service implementations using the Web Services model coupled with Semantic Web technologies. Here we describe two prototypes. In the first, described in sections 3 and 4, composition is achieved using workflow techniques. In the second, described in Section 5, we construct a simple open hypermedia system using web services and Semantic Web technologies.

Effective discovery and composition of services requires effective service description, not just at the WSDL level (which describes how to connect things together) but also at the UDDI level so that the functionality of the services is exposed. For this we propose to make further use of Semantic Web technologies. This is the subject of future work and is discussed in section 6.

In recent years, the Open Hypermedia Systems Working Group (OHSWG) has been working on a series of open hypermedia protocols to achieve interoperability between Open Hypermedia Systems. The original Open Hypermedia Protocol (OHP)[10] effort was followed by the Fundamental Open Hypermedia Model (FOHM)[22], the latter concentrating on the link data model rather than an on-the-wire protocol, and consists of a generic model for expressing hyperstructure by representing associations between data. FOHM is described in greater detail in the following subsection.

2.2 FOHM and Auld Linky

There are three key data objects in FOHM: *data*, *references* and *associations*. *Data* objects serve as wrappers for any piece of information held outside of the model. Typically these data objects are URIs but in some cases can be actual short, anonymous items of data. *Reference* objects are used to point at data objects and associations, and it provides a mechanism to locate a specific region within an object, for example a specific paragraph of a document. *Associations* represent relationships between data objects (or other associations). Bindings are used to attach the references to the association structure. An association has a feature space and each binding specifies the position of the reference in this space. Associations are handcoded and descriptions can be added to provide metadata about the relationships.

Figure 1 shows a basic navigational link that has a source and two destinations. Associations are depicted as long rectangles, which have semi-circular bindings attached to them. Bindings connect to references (circles) that can point to data objects (square) or associations.

In addition to these three first class objects, FOHM also has *context* objects that can be attached to any of the first class objects as shown in Figure 2. Only those structures whose context matches that of the query are returned, while parts of the structures that do not match are culled. This context mechanism provides support for adaptive hypermedia, in which the available links are determined according to contextual information such as the user’s identity, role, task, location and history. Hence it is particularly important in a mobile setting, where enhanced navigation of the information space can be particularly valuable.

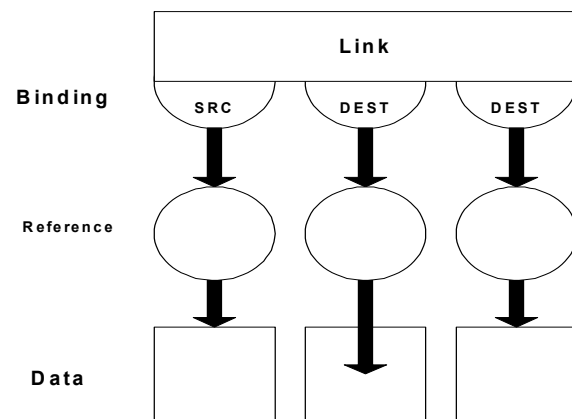


Fig.1. A navigational link in FOHM

For example, a simple hyperlink found on a webpage can be represented in FOHM as an association of type link, with a feature space of direction. It would contain two bindings to two references, one reference bound in the source position and the other in the destination. The references would then each point at the relevant data objects that represent the webpages. In addition the references, where required, would specify where in the data items the link is valid; most commonly this would be an indication of which text in the source document the link is made on. FOHM allows easy representation of far more complex links than those found on webpages, for example bi-directional links and links with multiple sources and destinations.

In addition to example of the simple hyperlink above, FOHM has been used in a number of applications[25], and numerous types of associations were found to be common and relevant to the overall theme of this paper.

FOHM is just a data model and we require a service to provide concrete structures. For this we use Auld Linky[21], a contextual open hypermedia link server. The role of Auld Linky is to act as a repository of FOHM structures that can be queried. Since FOHM is a highly structured data model, queries to the server are made using a pattern matching approach. In Auld Linky, the data model is extended by allowing objects to be marked as a variable. A client constructs a query as a FOHM structure,

possibly with parts marked as variable, and passes the query to Auld Linky. The query is then matched against the structures held in the server by comparing objects on an attribute-by-attribute basis, as well as any further sub-objects. The development of Auld Linky focused on creating a simple link server that would be lightweight, easily installable, and only serve links without any access to external services. To this end, Auld Linky has a footprint of less than 70 KB and consists of a single executable with a simple API, and is ideal for pervasive environments.

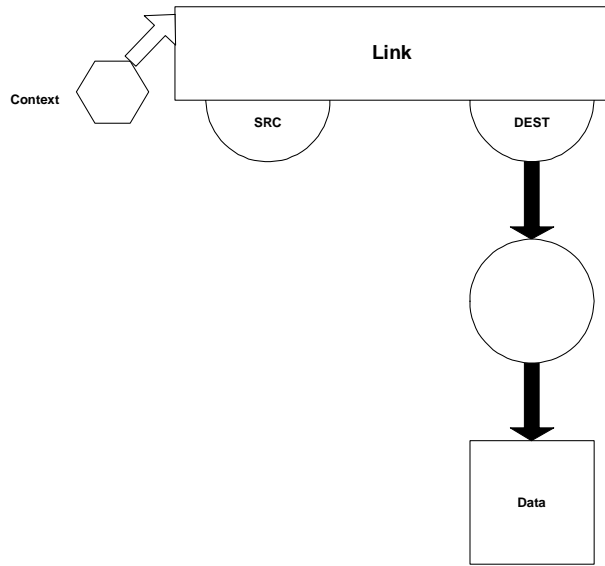


Fig.2. Context placement on FOHM structures

3. Web Services Composition

Web service composition is the ability to take existing services and combine them to form new services. Web service composition can either be static or dynamic composition. In static composition, the services are predetermined during the design of a Web process, for example, if the process to be composed is of a fixed nature wherein the business partners/alliances and their service components rarely change. In a dynamic composition, the Web service that is to be deployed is decided at run-time. Dynamic composition is more suitable if the process has to adapt dynamically to unpredictable changes in the environment. Dynamic composition may involve run-time searching of registries to find services.

A 'Web process', such as our link resolution process, needs to be described in a similar manner to the description of a Web service's interface. IBM's Web Services Flow Language (WSFL) [19] and Microsoft's XLANG[28] were two of the earlier languages to define standards for Web service composition. 'Business Process Execution Language for Web Services' (BPEL4WS)[8] is a recently proposed specification, building on both WSFL and XLANG. BPEL4WS is an XML-based workflow definition language that allow businesses to describe complex business processes that can both consume and provide Web Services. We chose BPEL4WS to represent composed processes because it combines graph oriented process representation of WSFL and the structural construct based processes of XLANG,

and it is the new unified standard for Web services composition. Moreover, BPEL4WS is an executable language, which means it can be used not only to write business processes but also to execute them. It will allow developers to publish their business processes so other applications can call and execute those processes as part of their own workflow.

Web service composition can be represented in a variety of ways. For example, digraphs[5],[27] and Petri Nets[1] have been widely used to represent business processes. We chose to represent Web Services composition as a digraph since it is more easily implemented with BPEL4WS. This representation describes a Web process in terms of activity nodes, control links and data links. The activity nodes in the digraph represent the tasks (services) in the Web process. Control links specify the control flow between the various tasks. It also defines standard constructs like XOR splits, AND splits, XOR joins, and AND joins that are used to define the execution logic in the process. The data links in a digraph represent the data flow between services and it also describes the mapping required between the output of one service to the input of another service.

Orchestrating Web Services by means of BPEL4WS process flows can be used to analyse inter-organisational cooperation, and to create and implement both public and private parts of a Web process. BPEL4WS enables service composition patterns to be readily exchanged between tools and thus reused. Although techniques developing business process models based on business requirements are well founded, skilled manual work is still required and is therefore not applicable to dynamic service creation. A semantic gap therefore usually exists between the model of what the user wants to do, and the service composition models that merely express how this may be accomplished. Although work has been done in mapping the behaviour of a system with a high level representation of its requirements, these high level requirements are still complex to express and are typically described by skilled programmers. Adaptive hypermedia techniques have been successfully applied in areas such as e-learning, to dynamically map basic user profile information to hypermedia documents that are based on user's knowledge and preferences[7].

To describe how this can be implemented via BPEL4WS, this paper refers to the tour guide scenario presented in [6], and is described in greater detail in the following subsection.

4. BPEL4WS Description

This section presents a high level abstraction of a BPEL4WS process that depicts the roles that various partners play in a tour guide scenario. The system was prototyped on a Windows 2000 platform, using Auld Linky version 0.72, and Business Process for Web Services Java Runtime. In this scenario, a tourist initiates a query, the query gets processed, and the tourist gets the data requested according to the level of detail and type of tourist. Initially, the middle step will involve sending the query to a Web services enabled Auld Linky application, and the tourist gets back a set of links as specified. From the tourist's point of view, the process will consume their query and reply with a set of tailored links.

In the tour guide scenario, the Web process consists of four partners who play different roles within the process. Their roles

and the type of data they might inquire of an Auld Linky service is described below:

- Tourists** – An overview map is provided to tourists showing the overall layout of a historical site, for example, a castle. A tour will then take visitors through both the physical site as a series of locations and a virtual information space history of the various locations. A location service will notify the application of the visitor's current location and a local map will show the features currently available. Clicking on a feature, such as the entrance, will provide a choice of information describing the entrance at different points in the castle's history. For both adults and children pictures illustrating how it may have looked can be provided. For an adult there may also be descriptions of building materials, why the construction was used and why it changed over time, etc. Children may prefer to have a video description of how the entrance was defended in simpler and more graphic terms. For example, a context object can be attached to a query made to an Auld Linky service that states whether the query is made by an adult or child tourist, and only those structures whose context matches that of the query are returned. In all cases, the detail presented to an historian is unlikely to be the form of information preferred by the tourist.
- Tour guide** - A tour guide does not need a highly visual map of the castle. They are likely to know the layout and the basic history quite thoroughly. Of more interest to them would be a general plan that provides hyperlinks to detailed historical information. This could allow them to look up details in response to a question from a tourist. The URI for the description could then be broadcast to devices in the vicinity (or some other definition of the tour group) in order for those interested to read more or see illustrations. The URI sent need not be for the variant the tour guide sees – the document can be negotiated to reflect the preferences and needs of each user.
- Archaeologists and Historians** - In contrast to the tourist an archaeologist may wish to know where artefacts have been found, and at what levels on a detailed plan of the site, or add details of their own findings. The artefacts may give links to data about the artefacts itself, such as photos. Those with an interest in a particular period in history may rate elements from that period most highly and elements at other times less highly, depending on how far from the preferred period they are. Those visiting from time to time may be most interested in finds since their last visit, overlaid on a general plan of the site. These users are less likely to be interested in artists' impressions of how things may have been or in tourist facilities.
- Tour bus driver nearby** - A bus driver from the tour company may need an itinerary, and could request a full schedule from a Linky service. They may also ask for a road map from the system. The road map may show a simple icon for the castle. This location may be

linked to a web page providing a brief description. The tour guide could inform the bus driver about the ideal location/time to park and pick up passengers, and this may depend on the tourists. The in-car text to voice processor may handle following the link and reading the presentation so that both driver and passengers can learn more about the place they are driving through, maybe even stopping and becoming tourists on the site. The type of tour can be used to control the mode of presentation, density of offered links and level of detail.

Workflow specifications describe the actions that are required to take place during the execution of business processes, and the overall flow of process. Figure 3 describes a tour guide workflow process modelled using a digraph. Each box is an activity (a transaction to be completed), and each activity could be an individual Web Service, described by a WSDL document. The activities are partitioned into 'swim lanes' according to their specific domain, and this describes the flow of control and information between requesting and performing operations of Web Services and can be used by participating partners to figure out how they can interact with a given service, both as service requestors and as service providers.

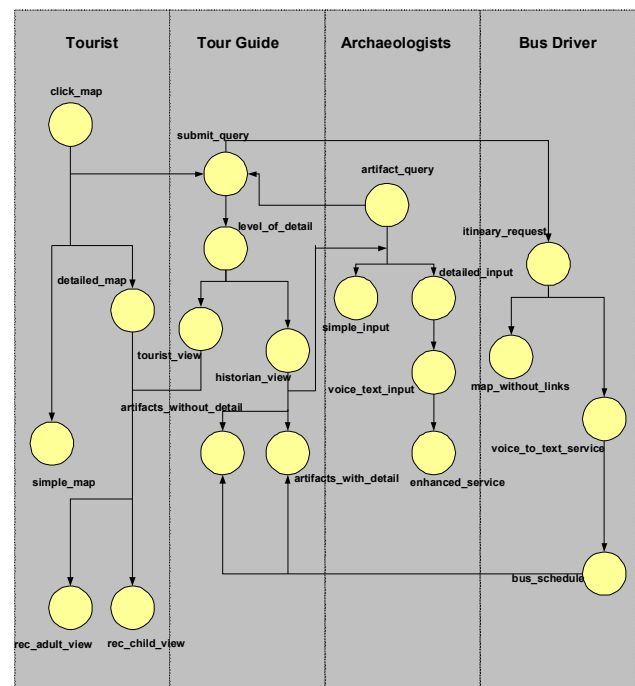


Fig.3. Public flow model of the tour guide process

To constrain the order in which operations at a service provider are to be invoked, a *flow model* has to be defined. A flow model is a directed graph that connects the activities, via control links. The control links determine the invocation order of activities by pointing from an activity to its successors. An activity is related to an operation of a port type of the service provider that is constrained by the flow model. In BPEL4WS, every business process has both a private and public part[8]. The private part of a business process is known as its *private interface*, likewise, the

public part is its *public interface*. Only the partner/service provider that implements a particular process has access to the private interface. Flow models that describe the function of a particular business process are found in the private interface. BPEL4WS enables individual parts of a process, such as the point at which a tourist initiates a query about a particular historical landmark, to be exported as part of the business processes public interface.

This public flow model is most likely to be the first document produced during the development of an inter-organizational Web process, and creates a *global model*, i.e., a new Web service that aggregates the services of multiple service providers.

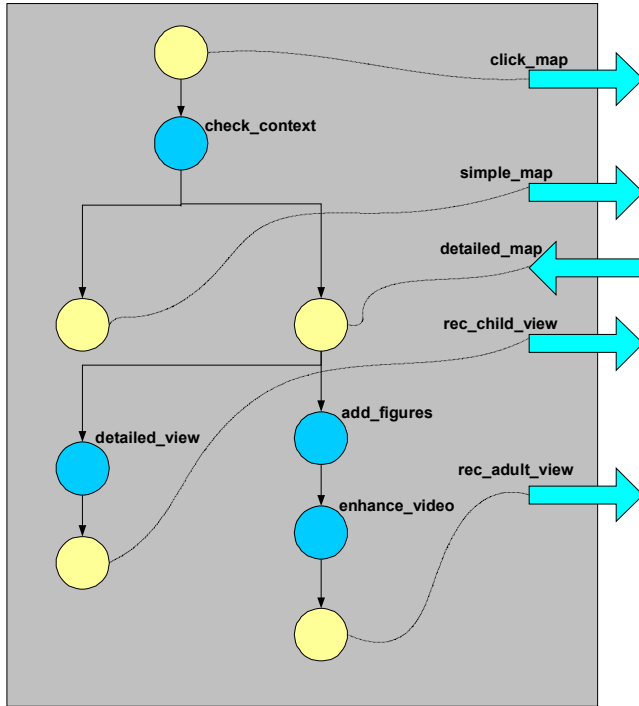


Fig.4. Private interface for the tourist domain

The next step would be splitting this flow model into domain-specific public parts, featuring the mapping of operations and interfaces between the participating partners. The description of public flow interfaces and the mapping of the interaction between the participating service providers to achieve a particular business goal, calls for the usage of BPEL4WS global models. The flow models can be used to define the behaviour of a Web service implemented by one of the partners. They complement the definition of the external interface of the service, which describes operations requesting services from service providers or offering services to service requestors. Figure 3 shows an implementation of the four public parts as flow models, and the overall global model of the tour guide example process. Technically, the partners implement a collection of port types covering the required services as its operations; these services are made available via corresponding ports. The operations between partners, that must communicate in order to perform the required functions, are wired together via *plug links*. A plug link identifies pairs of operations that communicate with each other, and describes which operation initiates this communication.

Communication here means the exchange of messages, for example, sending a request message and receiving a response message.

Figure 4 shows the private interface of the tourist domain, and implemented into a BPEL4WS flow model. The newly added activates, represented in the private model as grey, and retains conformity with the public interface to ensure that the semantics of both private and public flows do not differ. Designing the private part of the business process requires all partners to adhere to the interface and port definitions given in the global model, thus preserving interoperability among the partners in compliance to the protocol agreed upon earlier.

BPEL4WS presents the means of developing and describing both public and private workflows of distributed business processes. The fact that any OHS application adhering to the respective interface definition can interact with other OHS conformant systems, allows service providers to fulfil these roles provided its compatible with the BPEL4WS flow model for that process.

5. Using Semantic Web technologies

The service-oriented infrastructure is implicit in the Semantic Web, where the emphasis on automation demands machine processable content and service descriptions. In fact the Resource Description Framework (RDF) of the Semantic Web is itself a candidate for expressing hypermedia relationships, having many of the properties of an Open Hypermedia System[11], and one can envisage an implementation of Open Hypermedia using Semantic Web technologies.

Having established that we can put together a link resolution process using Web Services composition, we built a small system to explore the implementation of Open Hypermedia using both Web Services and RDF. We chose RDF as a more general and more powerful means of expressing associations than XLink – for example we can express the FOHM data model for hypermedia links[22].

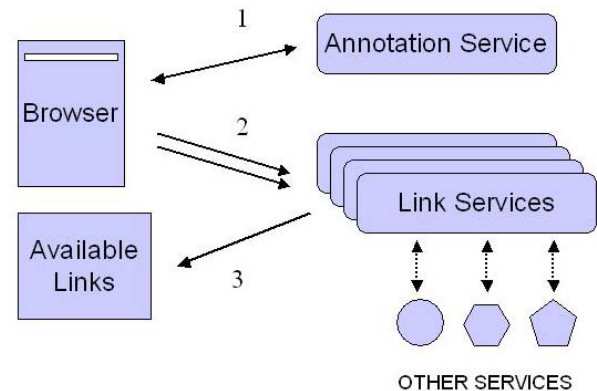


Fig.5. The open hypermedia prototype

We implemented this prototype in the Water language. Water is a language designed to provide a unified, homogeneous and simple platform to capture the benefits (and overcome limitations) of Web Services and XML. This new language was created by Christopher Fry and Mike Plusch[17] and was influenced by many other languages such as Scheme, Java,

HTML, Lisp, Self, Smalltalk, Basic, ML, and Dylan – it provides a multiple-inheritance, prototype-based object system and supports an XML Syntax called ‘Concise XML’.

The prototype system used Annotea[18] as a client and is illustrated in Figure 5. When Annotea loads a document, it consults annotation servers for available annotations (of various types, expressed in RDF) and indicates the location of these via the browser interface. An annotation service was written in the Water language and used to identify and present ‘buttons’ to the user when a document is loaded (mimicking the Microcosm interface[16]). Subsequently the user can use the buttons to initiate link resolution via link services encoded as Web Services in Water (effectively initiating Water method calls), or make a selection in the document and use Annotea’s annotation interface to initiate an XPointer-based query. The prototype link service is a linkbase which loads link data stored very simply as RDF triples, closely following the encoding for annotations and also drawing on the example encoding for XLink as RDF.

This prototype provided an effective demonstration that this collection of technologies can be used to implement a prototype open hypermedia system with some ease. It also provided proof of concept that RDF lends itself to linkbase representation.

6. Further Work

Ultimately, accurate mapping of user level requirements to service composition requires modelling of the real world in the form of the user’s context. 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 defines XML-based standards that support the development of the Semantic Web[2], and aims to make the content of the web more amenable to processing by automated agents. As part of this initiative, DAML-S[9] uses the DAML+OIL ontology language to provide semantic mark-up of web services. Modelling the links between a service and real world, which are mapped to semantic descriptions of the real world, can be accomplished by using ontologies for expressing the inputs, outputs, preconditions and effects of a service invocation.

DAML-S facilitates the automation of Web service tasks including automated Web service discovery, execution, composition and interoperation. A DAML-S service is characterized by three types of knowledge; a Service Profile, Process Model, and Service Grounding. A Service is described by a Service Model, which captures a Process Model of how the Service works; i.e. how it is composed, and what happens when it is invoked. A Process can either be atomic or composite, and both atomic and composite processes can be expressed in an abstract form called Simple Process. This enables a more abstract version of a process to be defined, hence ensuring only relevant pre-conditions and control constructs are enabled by a particular application. The Service Grounding specifies the details of how an agent can access a service. It specifies a communication protocol, and service-specific details such as port numbers used in contacting the service. Used together, the Process Model and Grounding describe how agents and services access, and interoperate with each other.

DAML+OIL’s roots in description logics and frame systems support the usage of a range of automated reasoning techniques in applying adaptive hypermedia techniques to service composition[20]. Among the techniques that can be applied are the usage of rule engines that reason on simple conditions during service input and output, performing limited automated service composition[26], using situation calculus to reason about semantic information in DAML-S specifications, and tools that aid developers in the simulation, verification and automated composition of web services[24]. The abstraction of existing services as DAML-S Simple Processes could be used to populate Auld Linky linkbases. Simple Process expressions could also be used to capture the users required inputs, output, preconditions and effect in terms that relate to ontologies that model the real world, e.g. the RosettaNet Ontology for computer equipments (www.rosettanet.org). Candidate service composition can therefore be expressed as DAML-S composite processes made up from Simple Processes from the Auld Linky linkbases. 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. tour agencies could exchange successful candidate compositions combining voice services and specific tour services, such as video services and tour guides for the blind.

Our work on the integration of adaptive hypermedia techniques with service composition is based upon a pervasive computing environment[13]. We are interested in what we call the *pervasive information fabric*, by which we mean the middleware for future information applications in this pervasive computing setting[29]. The service-oriented approach extends naturally into this setting. Pervasive computing is seen as an ideal application area for semantic web technologies, since similar problems in terms of the number of potential services and huge range of potential compositions exist. The application of Semantic Web technologies to Open Hypermedia can be taken much further. We have already investigated ontology services[4]. The relationship between Semantic Web and taxonomic hypertext, and the role of inference, are rich areas for investigation. Meanwhile at an infrastructure level, Semantic Web technologies can be applied to facilitate service description and discovery.

One of the drawbacks of defining service composition candidates at an abstract level is that some mismatch may exist between the concrete services mapped by Auld Linky and the semantics implemented by the concrete services. In some cases, this mismatch may mean that a candidate composition is not ideal for a user’s current context and an alternative composition is required. 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. One solution to this problem would be automatically generating XSLT scripts that adapt communications between services based on a Topic Map expression of their semantics[7].

A close analogy can be found between the definition of composition in hypermedia authoring languages and Architectural Description Languages, with hypermedia nodes mapping to components in ADLs, which implement services, and links mapping to the concept of connectors, which represent

communication between components[23]. In both cases, connectors to components in ADLs, and the corresponding links between nodes in Open Hypermedia models are considered first class objects. Connectors are the ideal abstraction for representing the semantic adaptation functions described above, both to provide support for representing semantic adaptors with an Auld Linky composition, and to allow such semantic connectors to be stored in linkbases, and reused by 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 reused as sub-components within composition candidates.

7. Related Work

The emergence of Web Services has led to more research into Web Services composition, which is an active area of research, with many concepts and methodologies being proposed by different research groups. However, very little work has been done on the use of Web Services composition to address the information perspective, in particular the hypermedia perspective.

The SWORD[26] project looks at various techniques at composing Web Services. In SWORD, a service is represented as a logical rule that expresses the inputs and outputs associated with it. A rule-based expert system is used to determine automatically whether a process is realised with the predetermined services. It also returns a process plan that describes the composition.

The WSMF[15] project provides a conceptual model for the development and description of Web Services technology. There are four key objects in WSMF: *ontology*, *goal repositories* that describe issues that can be solved by Web Services composition, *Web Services descriptions* that define the various aspects of Web Services, and *mediators*, which resolve interoperability issues among different service providers.

8. Conclusions

The paradigm of Web Services is being promoted and standards are being worked out to ease the adoption of Web Services. Web Services readily enable linkservices to be published, deployed, and invoked by compatible services on both a global scale on the Internet, and also in a local-area peer-to-peer and pervasive scale. This paper outlines how adaptive hypermedia techniques can be applied to form linkservices via web services composition. A tour guide scenario is outlined that builds on an existing adaptive hypermedia engine, Auld Linky, to develop service compositions to match tourist requirements based on abstract models of existing web services and known compositions. Auld Linky is then used to deploy concrete service composition on demand for a particular tourist type, based on a combination of the abstract service composition, the availability of pre-defined services and the tourist's current context. We then describe how this architecture can be made more flexible and expressive 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 linkservice compositions.

Though the Web's linking model is largely based on links that are embedded within documents, the infrastructure has become increasingly sympathetic to the open hypermedia approach, for example with the development of XPointer, XLink and the Resource Description Framework (RDF). Part of our motivation for this work was to find out how easy it is to build open hypermedia systems now, ten years after our first endeavours with the early Web infrastructure; i.e. what would such a system look like now, and how easy is it to build? Our experiences suggest that Web Services and Semantic Web technologies do greatly facilitate the construction of Open hypermedia Systems.

9. ACKNOWLEDGMENTS

We are grateful to our colleagues Danus Michaelides and David Millard from the Intelligence, Agents and Multimedia Research Group at University of Southampton for helpful discussions. Auld Linky was developed under EPSRC projects LinkMe (GR/M25919) and Equator (GR/N15986/01), and the scenario in this paper is related to the Equator 'City' project. This research was partly supported by IBM under the Faculty Partnership Award programme and by the EPSRC Comb-e-Chem e-Science pilot project (GR/R67729/01)

10. REFERENCES

- [1] Aalst, W., Hee, V., and Houben, G. (1994). Modelling Workflow Management Systems with High-level Petri Nets. Proceedings of the second Workshop on Computer Supported Cooperative Work, Petrinets and related formalisms, pp. 31-50.
- [2] Berners-Lee, T., Hendler, K., Lassila, O. (2001), 'The Semantic Web', Scientific American, pp 35-43, Issue 284 (3), 17th May 2001
- [3] Carr, L.A., De Roure, D., Hall, W. and Hill, G.J. 'The Distributed Link Service: A Tool for Publishers, Authors and Readers'. In Proceedings Fourth International World Wide Web Conference: The Web Revolution, (Boston, Massachusetts, USA), 1995, pages 647-656.
- [4] Carr, L.A., Hall, W., Bechhofer, S. and Goble, C.A. 'Conceptual Linking: Ontology-based Open Hypermedia'. In Proceedings of the Tenth International World Wide Web Conference, Hong Kong, May 1-5, 2001, pages 334-342.
- [5] Casati, F., Ilnicki, S., Jin, L., Krishnamoorthy, V., and Shan, M (2000). Adaptive and Dynamic Service Composition in eFlow. Proceedings of the International Conference on Advanced Information Systems Engineering, Stockholm, Sweden, pp. 13-31.
- [6] Chalmers D, Michaelides D, Carr L, Dulay N, Hall W, DeRoure D, Sloman M (2003). Contextual Information and Physical Environments.
- [7] Conlan, O. (2002), 'Enabling "Smart Space" Service Mobility Negotiation by Ordinary Users', Proceedings of Eurescom Summit 2002 Heidleberg, Germany.
- [8] Cubera, F., Goland, Y., Klein, J., Leymann, F., Roller, D., Weerawarana, S. (2002). Business Process Execution Language for Web Services. <http://www.ibm.com/developerworks/library/ws-bpel/>

- [9] DAML-S: Semantic Markup for Web Services', The DAML Service Coalition, <http://www.daml.org/services/>, October 2002
- [10] Davis, H.C., Millard, D.E. and S. Reich, "OHP - communicating between hypermedia aware applications", in Proceedings of the Workshop 'Towards a New Generation of HTTP', held in conjunction with the 7th International WWW Conference (J. Whitehead, ed.), Irvine, CA, USA, University of California, Irvine Department of Information and Computer Science, Apr. 1998.
- [11] De Roure, D., Carr, L.A., Hall, W and Hill, G. J. 'A Distributed Hypermedia Link Service'. In Proceedings of the Third International Workshop on Services in Distributed and Networked Environments (SDNE96), pages 156-161.
- [12] De Roure, D., Walker, N. and Carr, L.A. (2000) Investigating Link Service Infrastructures. In Proceedings of ACM Hypertext 2000, pages 67-76.
- [13] DeRoure D.C, Tso K, Vivek S. Mobile Link Services with MQSeries Everyplace. In Proceedings of the IEEE Annual Conference on Pervasive Computing and Communications (PerCom). Fort Worth, Texas, USA. March 23-26 2003
- [14] E. Christensen, F. Curbera, G. Meredith, S. Weerawarana. Web Services Description Language (WSDL) 1.1, W3C Note 15 March 2001. <http://www.w3.org/TR/wsdl.html>
- [15] Fensel, D. and Bussler, C. The Web Service Modeling Framework. <http://www.cs.vu.nl/~dieter/ftp/paper/wsmf.pdf>, September 2002.
- [16] Fountain, A., Hall, W., Heath, I and Davis, H.C. (1990) 'Microcosm: An Open Model for Hypermedia with Dynamic Linking'. In Rizk, A and Streitz, N and Andre, J, Eds. Proceedings Hypertext: Concepts, Systems and Applications, Proceedings of ECHT'90, Paris, November 1990, pages 298-311.
- [17] Fry, C., Plusch, M. and Lieberman, H. 'Static and Dynamic Semantics of the Web'. In 'Spinning the Semantic Web: Bringing the World Wide Web to Its Full Potential', Edited by Dieter Fensel, James A. Hendler, Henry Lieberman and Wolfgang Wahlster. MIT Press. February 2003.
- [18] Kahan, J. and Koivunen, M. 'Annotea: an open RDF infrastructure for shared Web annotations'. Proceedings of the tenth international conference on World Wide Web, 2001, Hong Kong, Hong Kong . Pages 623 - 632.
- [19] Leymann, F. Web Service Flow Language (WSFL). IBM Technical Document, <http://www4.ibm.com/software/solutions/webservices/pdf/WSFL.pdf>, 2001.
- [20] McIlraith, S.A., Son, T.C., Honglei Zeng, H. (2001), 'Semantic Web Services', IEEE Intelligent Systems, 16(2), March/April 2001.
- [21] Michaelides, D.T., Millard, D.E., Weal, M. J. and De Roure, D. 'Auld Leaky: A Contextual Open Hypermedia Link Server'. Proceedings of the 7th Workshop on Open Hypermedia Systems, ACM Hypertext 2001 Conference. Aarhus, Denmark.
- [22] Millard, D.E., Moreau, L.A.V., Davis, H.C. and Reich, S. FOHM: A Fundamental Open Hypertext Model for Investigating Interoperability Between Hypertext Domains. In Proceedings of the '00 ACM Conference on Hypertext, May 30 - June 3, San Antonio, TX (2000), pp. 93-102.
- [23] 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.
- [24] Narayanan, S., McIlraith, S.A., (2002) "Simulation, Verification and Automated Composition of Web Services", Proceedings of 11th World Wide Web Conference, May 7-11, 2002, Honolulu, Hawaii, pp 77-88.
- [25] P. A. S. Sinclair, K. Martinez, D. E. Millard, and M. J. Weal. Links in the Palm of your Hand: Tangible Hypermedia using Augmented Reality. In Proceedings of the '02 ACM Conference on Hypertext, May 30 - June 3, Maryland, 2002.
- [26] Shankar, P. and Fox, A. (2002) SWORD: A Developer Toolkit for Web Service Composition. Proceedings of the Eleventh International World Wide Web Conference, Honolulu, HI.
- [27] Sheth, A., Kochut, K., Miller, J., Worah, D., Das, S., Lin, C., Lynch, J., Palaniswami, D., and Shevchenko, I. (1996). Supporting State-wide Immunization Tracking using Multi-paradigm Workflow Technology. Proceedings of the 22nd International Conference on Very Large Databases, Bombay, India, pp. 263-273.
- [28] Thatte, S. XLANG: Web Services for Business Process Design Author. Technical report, Microsoft, 2001.
- [29] Thompson, Mark and Roure, David De and Michaelides, Danis (2000) Weaving the Pervasive Information Fabric. In Reich, S. and Anderson, K.M., Eds. Proceedings Open Hypermedia Systems and Structural Computing, 6th International Workshop, OHS-6, 2nd International Workshop, SC-2, San Antonio, Texas, USA, May 30-June 3, 2000 Proceedings 1903, pages 87-95.