Mobile Link Services with MQSeries Everyplace

Sanjay Vivek, Kenneth Tso and David De Roure
Department of Electronics and Computer Science
University of Southampton, UK
{smv99r,kt00r,dder}@ecs.soton.ac.uk

Abstract

The Open Hypermedia model is based upon the separation of hypertext links from documents and treats them as separate entities. Distributed link services take this approach and implement an open hypermedia system above the infrastructure of the World Wide Web. This paper explores the suitability and applicability of extending the architecture of distributed link services by adding a messaging backbone based on secure asynchronous message passing, which provides support for mobile users of distributed hypermedia information systems.

1. Introduction

The development of a large-scale, mobile and highly dynamic network for mobile users has been an ongoing research activity that involves NIST[22], IETF[26] and MANET[19]. Building information systems for mobile users presents tremendous challenges for data management technology, which includes providing data access to mobile users under variable rate and intermittent network connectivity, limitations on bandwidth and device storage. A hypermedia link service can provide practical benefits to information systems[4] by enhanced navigation, and distributed link service provide hypermedia links for users of information environment, in particular, the World Wide Web. This paper explores the use of an asynchronous messaging system as a backbone for providing link services to users with mobile devices, with the focus on extending the distributed link service by integrating Auld Linky, a contextual hypermedia link server, with MQSeries Everyplace(MQe)(trademark of International Business Machines Corporation) [13].

Open Hypermedia Systems (OHS) have been well researched by the hypermedia community[12],[23] and increasingly Web publishing applications adopt the open hypermedia approach[16],[27]. Extracting links from the source documents and managing them separately, i.e. treating links as first class citizens, can facilitate its maintenance. The Distributed Link Service (DLS) adopts this model and treats link creation and resolution as a service, which may be provided by multiple link servers and link resolution engines. OHS are more complex to engineer then closed systems and distributed systems such as DLS more so; there are, for example, important scalability issues to be addressed.

The existing Distributed Link Service architecture is neither sufficient nor appropriate for a mobile computing infrastructure. It assumes there are established network connections to link servers and link databases (linkbases) and this is not always the case in mobile environments. In this paper, we present our ongoing research into providing distributed link services to mobile users. This research focuses on extending the architecture of distributed link services by adding a messaging backbone based on MQe, to provide distributed link services to mobile users.

In section 2, we describe open hypermedia and link services including the Fundamental Open Hypermedia Model (FOHM)[18] and the Auld Linky[17] contextual link server. Section 3 explores the capabilities and security features in MQe to support the development of mobile applications, in particular the mobile link service. We give a scenario related to real-life applications in the next section, followed by a discussion on research issues and further possible extensions on mobile link service. The conclusion can be found after the current related work in section 7.

2. Hypermedia

2.1. Open Hypermedia

An HTML link is defined as follows[24]: "A link is the connection from one Web resource to another." Typical usage of the Web involves embedding links within documents in the HTML format; in this sense the Web can be considered a 'closed' hypermedia system. This embedding strategy, where link data appears at every source anchor, is effective because it enables browsing, but it complicates

both authoring and maintenance. Within hypermedia research and development, the open hypermedia model[7], which looks into abstracting hypertext links away from documents and managed separately, is an established alternative to the embedded link approach. In Open Hypermedia Systems (OHS)[8], a link is a first class entity that is separately stored and managed, like documents they can be stored, transported, cached and searched, and collections of links can be bundled into linkbases. If a destination changes and this affects just one of these links, the separated link data needs updating rather than all of the source documents which refer to that link[5]. This is essentially the idea of the *Open Hypermedia* model, in which the link objects may carry multiple sources, destinations and other information.

Open Hypermedia also supports the notion of *generic links*, in which the association between source anchors and specific locations in specific document is relaxed, so that one source anchor can match many different locations to which it applies. Generic links enable new content to be linked on the fly, and is ideally suited to the pervasive world of dynamic content[28].

2.2. Link Services

The introduction of separately maintained link data is clearly an additional complexity in the system architecture, and the gain in functionality must be set against the cost of the extra engineering that this implies. As discussed previously, managing link information separately from documents is one of the defining characteristics of open hypermedia. We take this a stage further by abstracting the link information into a link service[4], by which we mean here a network service that encapsulates responsibility for link resolution and maintenance. The idea of a web server and a link server are two distinct services. This separation of responsibilities is the essence of systems that fall into the category of *distributed link services*.

The DLS[4] provides a powerful framework to aid navigation and authoring and addresses some of the issues of distributed information management[10]. Using a proxy approach (also seen in IBM's Intermediaries[1]), the DLS adds links and annotations into documents as they are delivered through a proxy from the original WWW server to the target client browser. It uses a number of software modules to recognize different opportunities for adding various kinds of links to the documents. The interface to these link services is simply HTTP, and the details of the link data model are hidden because the service supports link creation and management via the published interface.

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
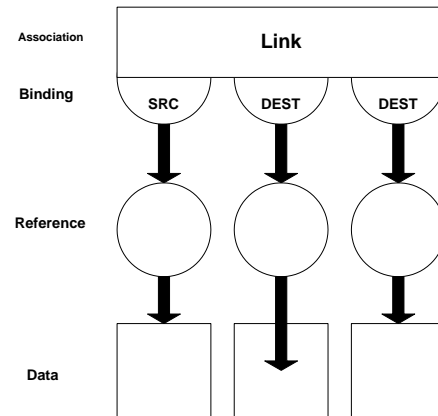

Figure 1. A navigational link in FOHM

Protocol (OHP)[9] effort was followed by the Fundamental Open Hypermedia Model (FOHM), the latter concentrating on the link data model rather than an on-the-wire protocol. FOHM is described in greater detail in the following section.

2.3. FOHM and Auld Linky

The basic FOHM model is constructed from four core objects and they are:

- *Data* objects, which serve as wrappers for any piece of data held outside of the model.

- *Associations*, representing relationships between data objects (or other asso-ciations).

- *References*, which are used to point at data objects and associations.

- *Bindings*, which are used to attach the references to the association structure.

Figure 1 shows a basic navigational link which has a source and two destinations.

In addition to these four first class objects, FOHM also has context objects that can be attached to any of the first class objects. 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.

A contextual structure server has been developed called Auld Linky[17]. It is grounded on the FOHM model and
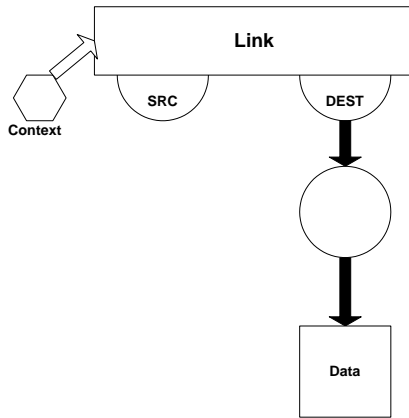
Figure 2. Context placement on FOHM structures

Figure 3. An illustration of the use of MQSeries Everyplace authenticator for WindowsNT

was designed to be a simple, lightweight structure server that serves according to contextual queries. Auld Linky is queried via an XML pattern matching language communicated over HTTP, each query is accompanied by a context object represented using FOHM as shown in figure 2. 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 then 70 KB and consists of a single executable with a simple API, and is ideal for pervasive environments. The development of Auld Linky to date has not addressed security features or any level of guaranteed transaction because that has not been the focus of that group's activity to date. However, recent work at Southampton has begun to investigate mechanisms for securing Auld Linky using MQSeries Everyplace[11].

3 MQSeries Everyplace (MQe)

MQSeries Everyplace (MQe)[21] is a messaging product and also a member of the MQSeries family messaging products. It is designed to satisfy the messaging needs of lightweight devices such as sensors, phones, Personal Digital Assistants (PDAs) and laptop computers as well as supporting mobility and the peculiar requirements that arise from the use of fragile communication networks. One of its major features is the provision of standard MQSeries quality of service, that is, once only assured delivery and exchanges messages with other family members. Since many MQe applications run outside the protection of an Internet firewall, sophisticated security capabilities is also provided.

The fundamental components of MQe are messages, queues and queue managers. One of its key features is to assure once and once-only synchronous or asynchronous message delivery. In a mobile environment when disconnected from the network, asynchronous message delivery is used. Messages that are sent to a remote device by MQe queues will be temporarily held locally until they can be delivered to their final destination (i.e. when a connection is established). The length of time for messages to remain on the queue is defined by the queue expiry interval (e.g. 5 minutes). Once the time limited is exceeded, the message is marked expired and subsequent action (e.g. deleting it, move it to a dead-letter queue or resending it) is determined by a configurable rule in the queue manager process. Message listeners can be added to the application to listen for events occurring on queues, such as message arrival. Having small execution footprint enables MQe to fit comfortably into modern mobile devices. Using MQSeries-bridge (MQSeries is a registered trademark of International Business Machine Corporation), messages can be exchanged with other MQSeries[14] family members, enabling integration of MQe based services with pre-existing Enterprise applications.

In the security world, there are four major areas: confidentiality, integrity, authentication and non-repudiation. Generally, in MQe, confidentiality of message data is achieved by encryption. Different cryptors are provided, the choice is driven by the cryptographic strength needed to protect the data and complying with national security requirements. The use of SHA1 digest ensures the integrity of message data. Authenticators including authenticator for WindowsNT (WindowsNT is a trademark of Microsoft Corporation) authenticator, and mini-certificates based on WTLS certificate, are used for authentication purposes.

MQe divides security features into four different categories known as local security, queue-based security, message-level security and link security to protect message data. Local security aims to provide protection for message data held by a local queue manager using cryptors. Queue-base security concerns with protecting message data
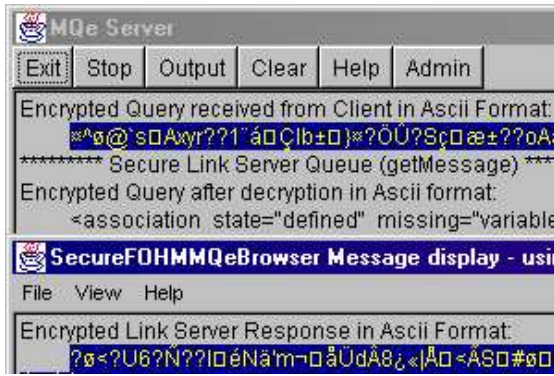
Figure 4. An illustration of the use of MQSeries Everyplace cryptor with encrypted queries and responses highlighted

Figure 5. An illustration of mobile link service providing adaptive information to different users

between an initiating queue manager and a target queue. Message-level security offers protection for message data exchanged between an initiating and receiving MQSeries Everyplace applications. Link security ensures the communication channels between queue managers are protected. Recent work at Southampton[11] has demonstrated the use of MQe to add security features (data encryption and user authentication) to Auld Linky as illustrated in figure 3 and 4.

4 Mobile Link Services

The current DLS is based on mobile agents and relies on a static network, and the communication is synchronous and connection-oriented. However, the mobile link service we propose works beyond this limit. With the asynchronous messaging and once-only message delivery assurance provided by MQe, reliable communication under variable and intermittent network connectivity can be achieved. In addition, both Auld Linky and MQe have small footprints enabling them to fit in devices with limited storage space.

A simplified scenario,use mobile link services to manage railway information and filter out irrelevant information for mobile users. The information system used in the railway industry may contain different types of information required by different users. For instance, passengers may want to find out information such as the schedule for all the trains going from one station to another on a particular day or the information about the departure and arrival time of a particular train. On the other hand, a train inspector may need information such as the service history and updating the inspection record about the trains when carrying out inspection and maintenance work. The rail company's manager requires access to all the information available includ-
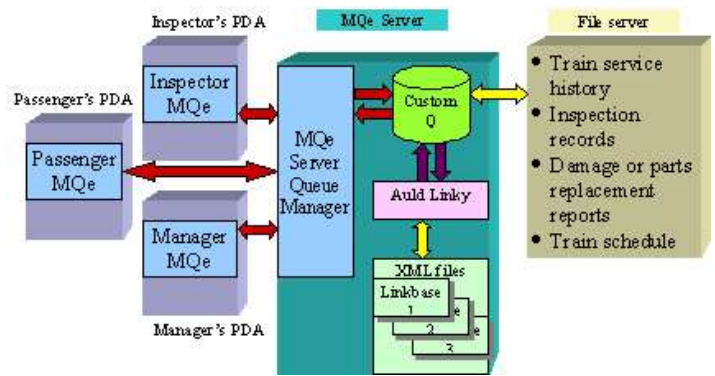
ing train service histories, inspection records and reports, and train schedule. This enables him to view the reports and records, update the train schedule and monitor the train service.

By structuring the information using the FOHM model and storing it in linkbases, mobile users (passengers and train inspectors) could initiate different contextual queries to obtain the relevant information they need; another way to achieve the same result is via the authentication process where the user can be identified as a passenger or an inspector. This concept is illustrated in figure 5.

Users are required to enter their login name and password (see figure 3) on their PDAs before sending their encrypted contextual queries (see figure 4) to the MQe server. After the MQe server has authenticated the users, the queries will be decrypted in the custom queue and then forward to Auld Linky. Responses from Auld Linky will tell the custom queue where to retrieve or update the relevant information stored in the file server. Finally, responses will be encrypted and sent back to the user's PDAs for where the responses will be decrypted and displayed in a GUI. From the scenario described previously, the irrelevant or unauthorized information will be filtered out before delivering to the inspector and the passenger. In addition to the advantages described in[11], the FOHM model, illustrated in figure 6, provides an easy way of accessing and more importantly maintaining the railway information to inspectors, passengers and managers.

Tailored information is delivered to mobile users based on the contextual queries they sent. This helps to eliminate the possibility of over-whelming users with unnecessary information. One of the advantages is that the user will save time by having the right information delivered in the shortest time possible,and does not need to browse through the whole database to get the right information. The open hy-
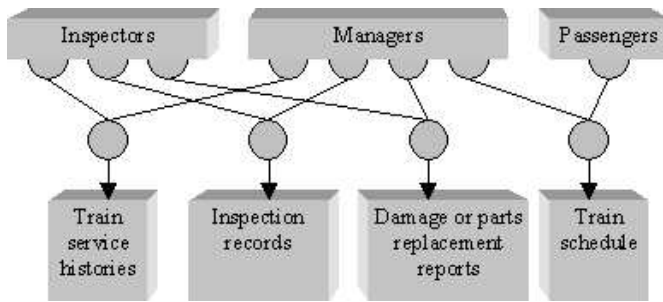
Figure 6. An illustration of using FOHM model to facilitate maintenance and reuse of links

permedia systems (the link server and link bases), act as a facilitator between public transport operation and management.

5 Issues

5.1 Further Possible Extensions

The mobile link service is an example of the integration between link services and MQe. We identify and discuss two possible ways of enhancing the mobile link service.

One of the most popular topics in the mobile computing world is peer-to-peer system, which is suitable for decentralized systems sharing data. For instance, a peer maintaining a linkbase on a mobile device may want to exchange data stored in another peer's linkbase. The mobile link service can be enhanced to fulfill this need.

If needed, security can be taken to a higher level by using more sophisticated cryptors and certificates issued by legitimate certificate authorities. Utilizing role-based access control provides authenticated users with different levels of access and control to the protected data, depending on the authority allocated to the group to which the users belong.

5.2 Implementation

The system has been prototyped using MQSeries Everyplace version 1.0, Auld Linky, and MQSeries version 5.2. We have provided proof of concept for the architectures described in the previous sections. We have also prototyped the system described in section 5.4.

5.3 Asynchronous Link Processing

From an OHS perspective, asynchronous service interaction readily enables selective and asynchronous link processing. When considering link resolution in a distributed link service across multiple link services, where lock-stepped coordination between services is unlikely to be achieved, decoupling document content from resolved links is desirable. There are cases where the results of link resolution queries may no longer be required - perhaps the user is no longer reading the document - and thus the ability to propagate message cancellation on queues between application and services where the queries may not have yet been delivered to services, or the responses back to the client is desirable. If the user browses to another document before the links have been processed they must be processed. Consequently, the links have to be displayed separately, in a manner independent of the document that the user is viewing.

5.4 Hypermedia and Web Services

We suggest that a service-oriented architecture, such as that offered by Web Services, readily enables hypermedia services to be published, deployed, and invoked by other services on both a global scale on the Internet, and also in a local-area peer-to-peer and pervasive scale. To enable asynchronicity between services, we have begun research into using store-and-forward middleware messaging systems, such as MQe, and thus provide the levels of communication decoupling required to meet this agenda.

Distributed service-oriented architectures help create a distributed environment in which any number of services, regardless of physical location, can interoperate seamlessly in a platform and language neutral manner. The success of any distributed service architecture is not only dependent on its ability to seamlessly integrate new and existing services, but also to function during periods of intermittent network connectivity. Breaking the traditional synchronous nature of interactions between OHS components would engender reliability and scalability of services in this environment. Within the eBusiness community, Web Services have emerged as a set of open standards, defined by the World Wide Web consortium, and ubiquitously supported by IT suppliers and users. They rely on the syntactic framework XML, the transport layer SOAP[3], the XML-based language WSDL[6] to describe services, and the service directory UDDI[29].

In the OHS approach, an open hypermedia system typically consists of three types of components: the client, a link or structure service, and a hyperbase or a linkbase. In the OHS architecture the interfaces between these components are clearly defined, and this allows each interface to be clearly defined as a Web Service. Any application or hypermedia system conforming to the respective interface definition can integrate with other OHS conformant systems. For example, any hypermedia system implementing the OHS client interface can use OHS linking and navigation services provided by any OHS conformant link server.
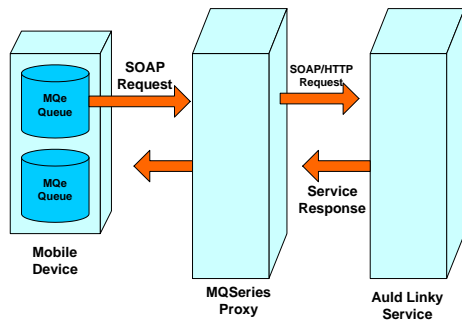
Figure 7. Invoking an Auld Linky Service

The Web Service architecture would allow these OHS services to be published, deployed, and invoked by other like-minded services on a global scale.

We propose the addition of MQe as a messaging backbone in a Service-Oriented Architecture (SOA) that increases reliability, scalability, fault tolerance, and the loose coupling of providers and requestors. In figure 7, Service A may wish to invoke the Linky service but the user could be on a mobile device that holds an MQe queue. The user of the client device works offline and stores the SOAP call as a message in an MQe queue on the client device. During network access, the message is sent to a separate MQSeries input queue on the server. The MQSeries proxy retrieves the data from the MQSeries input queue, translating them to HTTP requests, and subsequently forwards it to the Linky service. The response from the Linky service is returned to the proxy, and places it in the output queue. MQSeries later sends the result of the query to the MQe queue on the client device using a queue synchronization process.

Certain issues crop up with the usage of an asynchronous method of transport. The length of time a queue holds the message is among these issues. MQe queues can be defined with an expiry interval, and this function ensures that any message that has remained for a period longer then specified will be deleted. The type of service that is invoked is important in this aspect. If for example, a user queries an Auld Linky service that responds with a set of autobiographical links of an author, the issue of message expiry is not paramount. However, it may be the case that the links returned are of critical importance requiring the client to be informed with haste. This raises the question on how and when to update the links, and how to manage the liveness of asynchronous queries between client and services. MQe does facilitate the concept of filtering which allows it to perform powerful search functions, thus allowing the client to receive messages with higher priority first (or with a shorter expiry time). The links resolved would have to be pre-tagged before being send to the MQe queue.

We have begun to explore the addition of MQe as a mes-

saging backbone that increases reliability in a service oriented architecture, focusing initially on application to link services. The goal of this infrastructure is to provide reliable asynchronous hypermedia services to mobile users, and ensures that a transaction is completed once initiated.

6 Related Work

Besides the LinkMe[15] project in which Auld Linky was developed, there has not been much research done into providing distributed link services for mobile users. Our research is based upon extending an asynchronous messaging backbone to the DLS architecture to provide secure and reliable hypermedia link services to mobile users. Our research also explores the addition of a messaging backbone that increases reliability in a service-oriented architecture, focusing on application to mobile link services.

The majority of hypermedia systems that we work with are intranet based with a traditional client-server model based on large servers accessed by many clients and interconnected by a static network. Technologies such as Bluetooth (Bluetooth is a trademark owned by Bluetooth SIG)[2] and Jini (Jini and all Jini-based marks are trademarks of Sun Microsystems, Inc.)[25] are part of the emerging information infrastructure for the mobile and pervasive computing environment but they do not address the information perspective, in particular the hypermedia perspective.

Agent frameworks such as the Southampton Framework for Agent Research (SoFAR)[20] provide similar communication interactions as discussed previously. Many existing agent frameworks require supporting infrastructure to be manually deployed and configured; e.g. agent registries, access controls for mobile code. Mechanisms for automating platform provision, discovery and maintenance are ongoing research topics.

7 Conclusion

Current distributed link services are not designed for mobile environments and are based on synchronous, connection-oriented communications. In this paper we have described in detail the extension of the current DLS architecture by adding a messaging backbone, based upon IBM's MQSeries Everyplace, which provides secure and reliable link services to mobile users. We believe this extension is an interesting approach that addresses many of the shortcomings of the present DLS. We have presented evidence to justify this belief by illustrating how the architecture can be used to provide secure and reliable hypermedia services to mobile users during periods of intermittent network connectivity.

8 Acknowledgements

References

[1] Barrett, R. and Magilo, P.P. Intermediaries: An approach to manipulating information systems. IBM Systems Journal, (38):629-41, 1999.

[2] Bluetooth SIG. Specification of the Bluetooth system, Dec. 1999. http://www.bluetooth.com

[3] Box, D., Ehnebuske, D., Kakivaya, G., Layman, A., Mendelsohn, N., Nielsen, H. F., Thatte, S. and Winer D. Simple Object Access Protocol (SOAP) 1.1, W3C Note 08 May 2000. http://www.w3.org/TR/SOAP

[4] Carr, L., De Roure, D., Hall, W., Hill, G., (1995) The Distributed Link Service: A Tool for Publishers, Authors and Readers, World Wide Web Journal 1(1), 647-656.

[5] Carr, L.A., Hollom, R.J., Hall, W., Davis, H.C., The Microcosm Link Service and its Application to the World Wide Web, In the Proceedings of the First World-Wide Web Conference, pp 25-34, CERN, May 1994.

[6] Christensen, E., Curbera, F., Meredith, G. and Weerawarana, S. Web Services Description Language (WSDL) 1.1, W3C Note 15 March 2001. http://www.w3.org/TR/wsdl.html

[7] Davis, H. C., Lewis, A., and Rizk, A. OHP: A Draft Proposal for a Standard Open Hyper-media Protocol. In Proceedings of the 2nd Workshop on Open Hypermedia Systems, ACM Hypertext '96. UCI-ICS Technical Report 96-10, Department of Information and Computer Science, University of California, Irvine, 1996, pp. 27-53.

[8] Davis, H. C., Millard, D. E., Reich, S., Bouvin, N., Grnbk, K., Nrnberg, P. J., Sloth, L., Will, U. K., AND Anderson, K. M. Interoperability between hypermedia systems: The standardisation work of the OHSWG. In Hypertext '99, The 10th ACM Conference on Hypertext and Hypermedia, Darmstadt, February 21-25,1999 (Feb. 1999), ACM, pp. 201-202.

[9] Davis, H., Reich, S. and Millard, D. A proposal for a common navigational hypertext pro-tocol. Techical report, Dept. of Electronics and Computer Science, 1997. Presented at 3.5 Open Hypermedia System Working Group Meeting. Aarhus University, Denmark. September 8-11.

[10] De Roure, D., Carr, L., Hall, W. and Hill, G. (1996) A Distributed Hypermedia Link Service, In Proceedings SDNE96, IEEE Computer Society Press.

[11] De Roure, D.C., Tso, K. K. K. and Lambert, H. Securing a Open Hypermedia System (OHS) Using MQSeries Everyplace (MQe). OHS2002, Maryland,USA.

[12] Grnbk, K., Sloth, L. and Orbk P. (1999) Webvise: Browser and Proxy Support for Open Hypermedia Structuring Mechanisms on the WWW. In Proceedings of the Eighth International World Wide Web Conference, 253-268.

[13] IBM MQSeries Everyplace, http://www-3.ibm.com/software/ts/mqseries/library/ manualsa/manuals/mqsev12.html

[14] IBM MQSeries Family. http://www-4.ibm.com/software/ts/mqseries/

[15] LinkMe: Distributed Link Services for Mobile Environments, http://www.iam.ecs.soton.ac.uk/projects/linkme/

[16] Lowe, D. and Hall, W. (1998) Hypertext and the Web: An Engineering Approach, J. Wiley and Son

[17] Michaelides, D. T., Millard, D. E., Weal, M. J., and De Roure, D. C. D. Auld leaky: A contex-tual open hypermedia link server. Proceedings of the 7th Workshop on Open Hypermedia Systems, ACM Hypertext 2001 Conference. Aarhus, Denmark.

[18] Millard, D. E., Moreau, L., 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.

[19] Mobile Ad Hoc Networking (MANET), http://www.sensorsmag.om/articles/ 0101/18/main.shtml

[20] Moreau, L., Gibbins, N., De Roure, D., El-Beltagy, S., Hall, W., Hughes, G., Joyce, D., Kim, S., Michaelides, D., Millard, D., Reich, S., Tansley, R. and Weal, M. SoFAR with DIM Agents: An Agent Framework for Distributed Information Management. In The Fifth International Conference and Exhibition on The Practical Application of Intelligent Agents and Multi-Agents, pages 369-388, Manchester, UK, Apr. 2000.

[21] MQSeries Everyplace Performance Report, [Online], Available: http://www-3.ibm.com/software/ts/mqseries/txppacs/ep01.html. July 25, 2002

[22] National Institute of Standards and Technology (NIST), http://w3.antd.nist.gov/wahnmahn.shtml

[23] Osterbye, K. and Wiil, U. (1996) The Flag Taxonomy of Open Hypermedia Systems, In Proceedings of the 1996 ACM Hypertext Conference, 129-139.

[24] Raggett, D., Le Hors, A., and Jacobs, I. HTML 4.01 Specification. World Wide Web Consortium, Recommendation REC-html401-19991224, December 1999.

[25] Sun Microsystems, The Jini Lookup Service Specification (LU), version 1.0.1 available at http://www.sun.com/jini/specs/lookup101.html, November 1999

[26] The Internet Engineering Task Force (IETF) - Mobile Ad-hoc networks (manet), http://www.ietf.org/html.charters/manet-charter.html

[27] Thistlethwaite, P. (1997). Automatic Construction and Management of Large Open Webs, Information Processing and Management, 33(2), 161-173.

[28] Thompson, M.K. and De Roure, D.C. Towards the Pervasive Information Fabric: Issues in Pervasive Hypermedia. To be submitted to Mobile Data Management, MDM2003, Melbourne, 21-24 January, 2003.

[29] Universal Description, Discovery and Integration (UDDI), http://www.uddi.org