Preprint version of paper accepted for
IEEE Workshop on Future Traends of Distributed Computing Systems (FTDCS 2003).

# Towards Middleware Services for Mobile Ad-hoc Network Applications

Thomas Plagemann, Vera Goebel, Carsten Griwodz, and Pål Halvorsen
*University of Oslo, Department of Informatics*
*{plageman, goebel, griff, paalh}@ifi.uio.no*

## Abstract

*Mobile ad-hoc networks are typically very dynamic networks in terms of available communication partners, available network resources, connectivity, etc. Furthermore, the end-user devices are very heterogeneous, ranging from high-end lap-tops to low-end PDAs and mobile phones. Traditionally, middleware is used to abstract from this heterogeneity and enable the application programmer to focus on application issues. We propose to develop middleware services that additionally provide services for information sharing in mobile ad-hoc networks, because the possibility to share information is mission critical for many mobile ad-hoc network applications.*

## 1. Introduction and motivation

In recent years, mobile ad-hoc networking has been recognized as an important research area. Typical application scenarios for mobile ad-hoc networks include emergency situations in which wireless devices are used to coordinate the efforts of rescue personal, and business meetings in which the laptops and other devices are connected in a wireless network to support the collaboration of the participants and increase the meeting's efficiency. These examples indicate already that a common key element in mobile ad-hoc network applications is information access and sharing. This information will not only be represented in the form of textual and numerical data, but also in form of graphics, video, and audio.

Today, the main research emphasis in mobile ad-hoc networks is on routing and service location issues. This is for example reflected by the IETF working group MANET (Mobile Ad-Hoc Networks). There is no doubt that routing and service location problems have to be solved to provide a working infrastructure in mobile ad-hoc networks. However, applications are needed to turn a working infrastructure into a useful infrastructure. There is a magnitude of potentially useful applications for mobile ad-hoc networks, but application development for mobile ad-hoc networks is not easy. These networks are typically very dynamic networks in terms of available communication partners, available network resources, connectivity, etc. Furthermore, the nodes in such networks are very heterogeneous, ranging from high-end lap-tops to low-end PDAs and mobile phones. CPU storage space, bandwidth and battery power represent important resources. Finally, many application scenarios, like coordination of rescue teams, have quite hard non-functional requirements as well, such as fault-tolerance, survivability, real-time, and security. Obviously, solving these issues in every new mobile ad-hoc network application from scratch is not a feasible approach. Instead, we propose to develop a set of generic services, or middleware services, that support the development of applications for mobile ad-hoc networks.

In an infrastructure built on a pure mobile ad-hoc network, only the data that is kept on the mobile devices is accessible. However, for many scenarios, nodes that provide access to other infrastructures will exist within the adhoc network, and thus provide access to additional information. These nodes may be embedded systems and sensors that can can provide applications with localized or environmental information. Other fixed nodes may be connected to the Internet and function as gateways between the mobile ad-hoc network and the Internet. These nodes could provide access to existing information infrastructures, like the World-Wide Web, Content Distribution Networks, and Peer-to-Peer (P2P) Networks. Figure 1 illustrates the typical network environment we envisage for mobile ad-hoc networking applications. In the rest of the paper, we use the term ad-hoc network for mobile ad-hoc networks that might include elements of a fixed infrastructure.

It is the goal of this position paper to (1) analyze the requirements of ad-hoc network applications, (2) evaluate state-of-the-art solutions and identify their deficien-

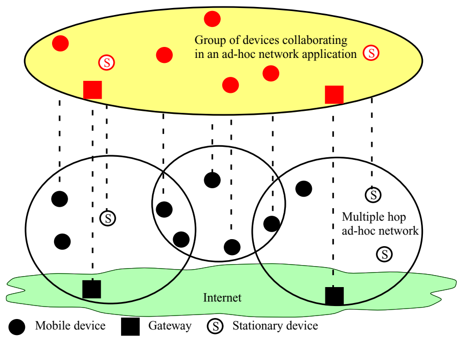


**Figure 1: Network environment**

cies, and (3) to outline our approach to develop middleware services for information sharing.

## 2. Mobile ad-hoc network applications

Application domains for ad-hoc networks include e-learning [1], inter-vehicle communication [2], and collaborative electronic shopping [3]. It is the purpose of this section to identify characteristics and requirements that are common to these and other ad-hoc network applications with respect to information access and sharing.

### 2.1. Characteristics

There are three generic characteristics that ad-hoc network applications share: (1) information access and sharing is mission critical, (2) three classes of information sources have to be integrated, and (3) cooperation is necessary, but not always desired.

The simplest ad-hoc network applications have no other purpose than sharing some resources, e.g., a printer. However, application such as those mentioned above are based on access and sharing of information. In e-business applications, more and correct information might lead to better deals. E-learning applications are only possible if teaching material is accessible. In emergency and disaster management, fast access to data can even save human lives.

In these and other application scenarios, an ad-hoc network is likely interconnected to other infrastructures. For example, it may include stationary nodes that are also connected to the Internet and provide gateways. This means that three different types of information sources may be available:

- **Mobile end-user devices**: each end-user device stores some data, some of which may represent useful information for other end-users in the ad-hoc network. It is important to note that this information may not be directly accessible to all other nodes at all times because of the mobility of end-user device. End-user devices with new information may enter the ad-hoc network, while others may leave and render some information inaccessible. Generally, it cannot be foreseen which end-user devices with which information will be part of a ad-hoc network. Information about physical location and user intervention may alleviate this problem.
- **Isolated stationary devices**: in this context, only such stationary devices that can communicate with mobile devices are considered. Conceivable devices range from regular PCs to embedded systems and sensors. Typically, such devices are not associated with an end-user but are aware of their location. As such, they may provide information about their environment, make additional resources temporarily available to ad-hoc network applications. They may also provide means such as mailboxes for asynchronous communication between mobile devices.
- **Gateways**: mobile or stationary devices might act as gateways, e.g. to the Internet, to enable access to a larger body of information. It is not possible to download all relevant information for a certain task onto a device in advance, because the amount of the corresponding data can be quite large and it cannot be foreseen which information is needed.

The third characteristic of ad-hoc network applications is concerned with cooperation, which is necessary but not always desired from all nodes. Without a cooperation of nodes, ad-hoc networks cannot work. This is valid at the network layer for routing issues, at the middleware layer for resource sharing issues, and at the application layer for information sharing issues. However, end-users do not always desire to cooperate. One reason is that each node has to contribute consumable resources to the cooperation, such as CPU cycles and battery power to perform compute intensive tasks or to replicate data. Another reason is that certain end-users might be in competition with each other. In the business meetings scenario, two groups may negotiate contract terms. In order to work efficiently, they are willing to share resources freely, but they share only selected information. In application scenarios like medical rescue operations, it is also important to protect the privacy of the information, for example from outsiders such as journalists.

### 2.2. Requirements

Information is stored and managed in the system as data. Based on the discussion above, we identify the following functional high-level requirements of application with respect to data access and sharing:

- **Dynamic access to all relevant data and**

**resources**: resources from other devices might be used to forward data e.g., the other device has better connection to the Internet, to cache important data on other devices because the own device has not enough storage space, and to perform compute intensive tasks on other devices. In this context, it is important to consider the highly dynamic nature of ad-hoc networks. Otherwise, a node that keeps a unique copy of mission critical data could leave the range of the ad-hoc network and the data is "lost."

- **Dynamic collection and structuring of accessible data**: in many situations too much data, especially if it is unstructured, is not helpful for human user - it might even decrease the efficiency of the user. Thus, the accessible data has to be structured in such a way that the end-user is able to access quickly the important data. Since the data is collected from sources that might be only temporary part of the ad-hoc network it is important to find solutions for the dynamic nature of data availability.
- **Access control for resources and data according to policies**: in scenarios where all entities follow a common goal, it is natural to share everything. In situations where different entities follow different and sometimes contradicting goals, like two negotiating groups in a business meeting, it is also natural that not everything should be shared.

The access to data should be provided by the follwing three interaction types: retrieval, event notification, and client-to-client. In retrieval mode, the user or application is active and requests the system to retrieve data. The data might be specified by it's name, like a URL, or specify a query to search for data. In event notification mode, a subscriber, e.g., user or application, is informing the middleware that it is interested in an event. The subscriber specifies which event it is interested in with the help of a channel or a filter. It should be noted that independent entities and groups of entities can subscribe for events. In client-to-client interaction, users or input/output devices communicate in real-time with each other, like in a video conference. Client-to-client communication may be performed in a 1-to-1 relationship, but also as 1-to-many and many-to-many relationship.

## 3. State-of-the-Art

The Integrated Mobile Ad-hoc QoS Framework (iMAQ) [4] aims at a generic middleware that includes application-aware adaptation, configuration management, and data dissemination and replication. For now it has only produced results on routing in ad-hoc networks. JINI [5] allows devices to offer and access services through a lookup service, which is provided by dynamically discovered devices. SLP [6] works in a similar way, but concentrates exclusively on protocol issues. Proem [7] and JXTA [8] define protocols for P2P communication that integrate service offering, binding and usage with routing functionality. JXTA peers can optionally provide access to hardware resources or routing services to subgroups of its peers. It lacks a means to identify its contents beyond the lifetime of a group. 7DS (seven degrees of separation) [9][10] enables information dissemination and sharing among mobile hosts in a P2P fashion. It runs as an application on heterogeneous devices. Participants can obtain data, cache them and exchange them with other interested participants. The SyncML protocol [11] is designed to achieve the synchronization of mobile devices and relies on client-server semantics.

Replicated systems with volatile connectivity face challenges in assigning unique names, in location, and in replication. Naming and location are frequently considered in combination. URLs [12] tie an object location to a host location, and rely on DNS [13]. The Globe system, on the other hand, assigns global unique IDs to replicatable, movable objects [14]. Location services in P2P systems are often based on distributed hash tables (DHTs) that translate keys to locations. These keys are well-known names of entities. The names may belong to mobile hosts [15] or files [16][17][18], but may just as well refer to objects in an application [19] or processes. Typical location services rely on well-known hierarchies of directories [13][18]. The location mechanisms of DHT P2P systems provide a possible approach for implementing event notification schemes such as publish/subscribe, middlemen or rendezvous. Many replication systems have their origin in distributed file system research. Examples that rely on a central server are AFS [20], Coda [21], Rumor [22], Roam [23] and Ficus [24]. Napster [25] and Gnutella [26], which allow the data exchange among peer hosts, are bound to a fixed network because they require the coordination of a fixed server. Still, the hoarding and reintegration mechanisms of Coda provide a viable approach for replication in the face of predictable connection loss. JetFile [27], Farsite [28] and PAST [29] are serverless file systems but require a complete dissemination of updates. LIME [30] and XMIDDLE [31] do not require such complete dissemination. LIME is based on Tuple Spaces [32], but lacks data structures. This is a central issue in XMIDDLE, which focuses on replication and reconciliation of data. Reconciliation of data is a severe problem when information has changed on disconnected hosts or groups of hosts. XMIDDLE provides middleware support to perform this task in application-dependent ways.

To support service discovery in a ad-hoc network, one cannot rely on traditional, centralized directory services. Existing approaches for service discovery employ different topologies or ontological structures, e.g., hierarchical rings [1] and multi-layer clusters [33] of service nodes. In the hierarchical rings approach, nodes that offer thematically coherent services are conceptually connected in rings where one node is selected as an access point. The hierarchy of rings is formed by building other rings of the access point nodes. The multi-layer cluster approach considers nodes that are physically close (in radio range) and semantically close (offer similar services), and groups them in clusters. The various hierarchical layers are then formed using the physical and semantical closeness of the sub-clusters. The search for a service is initiated on a lower layer and recursively broadened to higher layers if no satisfactory service is found.

Tuple structures are very simple and general and do not support the complex data organization of our application scenarios. XML and especially DOM are much more suitable to model hierarchical data structures and to address specific paths [34][35]. A more advanced approach supporting user and location dependent services is SDL (Service Description Language) [36]~~, SDS (Secure Service Discovery Service) [37] and [38][39][40]~~. Descriptions with more semantics have been developed in the areas of ontologies, XML, and RDF [41]. An important example is DAML (DARPA Agent Markup Language) [41], which uses XML and RDF to describe entities and the relations between them. This enables the users to develop ontologies and annotate information such as service descriptions. In the context of ad-hoc networks, information systems related issues are only very rarely addressed so far [35][42][43][44][45][46].

## 4. Ad-hoc InfoWare approach

In order to understand whether mobile handheld devices are able to run standard middleware solutions, like CORBA, with satisfying performance, we have performed experimental studies. We measured the performance of two CORBA implementations, MICO and Orbix/E, on different platforms, and used the following wireless network technologies: Wi-Fi 802.11b, Bluetooth, and IrDA . Figure 2 summarizes the results of one experiment that are representative for all results. In this experiment, we measured the average response time for a simple method invocation when client and server reside on the same node. As nodes we used an IPAQ with a SA-1110 206 MHz CPU and 64 MB memory, a PC with a Pentium II 350 MHz CPU with 512 MB memory (PC1); and a PC with a Pentium III 500MHz CPU with 384 MB memory (PC2).

**Figure 2: Response times of MICO and Orbix/E**

The measurement results show that there is a clear performance gap between the IPAQ and the PCs. However, it is also obvious that the implementation itself has a strong influence on the performance. Generally, we learned that handheld devices today can be quite powerful and performance is only one reason to use new solutions for middleware service for ad-hoc networks instead of standard middleware solutions.

The other reasons can be broadly classified into two groups: (1) multimedia data is not properly supported in standard middleware solutions, and (2) ad-hoc networks are of highly dynamic nature, because nodes can join and leave the network at any time. For both problem classes it is important to provide a flexible solution, because a single static solution cannot solve all problems, especially in such hetrogenous environments like ad-hoc networks. We apply two principal methods to achieve flexibility: the separation of mechanisms and policies and dynamic configuration.

### 4.1. Multimedia support

As a starting point for multimedia support we use the MULTE-ORB. The MULTE-ORB supports the dynamic configuration of protocols and flexible signalling. The overall architecture of the MULTE-ORB is illustrated in Figure 2.

The application entity is either some kind of application or a higher-level middleware component, like IIOP, QIOP or the stub and object adapter. The management toolkit and the data protocol is part of the ORB core of the middleware, using CORBA terminology. The management toolkit consists of an extensible set of signalling modules for various tasks like connection management, QoS negotiation, etc. A script is interpreted from the coordinator at runtime. The script specifies which particular signalling protocol should be used

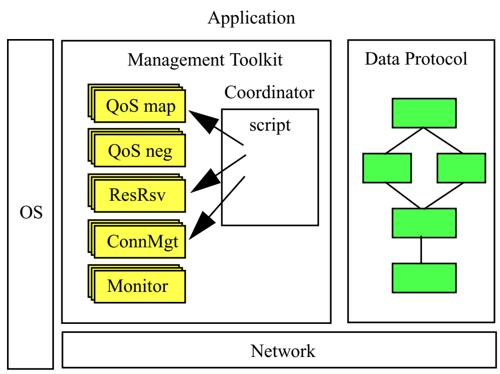


**Figure 3: MULTE-ORB Architecture**

in which order. The data protocol is configured at runtime out of fine-granular protocol functions. The operating system (OS) exposes different services, like access to resources and OS functionality, through well defined interfaces (APIs). One class of these APIs provides access to network services, but we regard the network as a separate entity in the architecture, because its resources and services are distributed in the network itself. The data protocol and management toolkit use the network through the OS APIs to communicate with their peers.

So far, our work on the MULTE-ORB has focussed on configuration and flexibility in end-systems. For ad-hoc networks, we have to extend it to support the dynamic configuration of functionality on intermediate nodes. For example, an intermediate node might be used to scale down the resolution of a video such that it can be presented on the screen of a PDA.

### 4.2. Managing dynamic information

Each node in an ad-hoc network is potentially able to contribute with data, capabilities, and physical resources to the application. Obviously, it is necessary to decribe the data that is on a node and the middleware has to keep track which data is where. The same is valid for capabilities, like particular scaling functions that are implemented on a node, and for the physical resources.

To describe these three types of contributions, we use XML, which has become a quasi standard for data exchange. In order to understand some of the semantics of data, we will apply meta-data ontologies for the different application domains. **We need a better explanation here!**

Distributed directories with a certain degree of redundancy are used to keep track of data, capabilities, and physical resources. The degree of redundancy should be defined by policies, like each node has to store all meta-data about its direct neighbours. There are basically two approaches to update the distributed directories: event based and demand based. In the event based approach, a new node joining the network represents an event and all its meta-data is broadcasted to its direct neighbours. In the demand based approach, a user initiated query might trigger the execution of a location protocol that tries to find out about new nodes and retreives their meta-data.

## 5. Outlook

**Needs to be written!** In this short paper, we have briefly discussed the requirements of ad-hoc network applications, given an overview on related work and outlined how we approach the development of middleware services that fulfill the application requirements.

In the full version of this paper we will give a more detailed discussion and analysis of requirements and related work, and especially elaborate more in detail on our approach towards middleware services for information sharing in ad-hoc networks.

## 6. References


[1] B. König-Ries, M. Klein, "Information Services to Support E-Learning in Ad Hoc Networks", *1st International Workshop on Wireless Information Systems (WIS 2002)*, April, 2002, Ciudad Real, Spain.

[2] Franz, W.; Eberhardt, R.; Luckenbach, T., "FleetNet - Internet on the Road", *8th World Congress on Intelligent Transportation Systems (ITS 2001)*, Sydney, Australia, Oct. 2001

[3] L. Capra , W. Emmerich and C. Mascolo. "Middleware for Mobile Computing: Awareness vs. Transparency (Position Summary)", *8th Workshop on Hot Topics in Operating Systems (HotOS-VIII)*, Schloss Elmau, Germany, May 2001.

[4] iMAQ University of Illinois- Urbana Champaign, USA. http://cairo.cs.uiuc.edu/adhoc/

[5] W. Keith Edwards, *Core Jini*, Prentice Hall, June 1999

[6] E. Guttman, C. Perkins et al, "Service Location Protocol, Version 2", Internet Engineering Task Force, RFC 2608, June 1999

[7] Korteum, G., Schneider, J., Preuitt, D., Thompson, T.G.C., Fickas, S., Segall, Z., "When Peer-to-Peer comes Face-to-Face: Collaborative Peer-to-Peer Computing in Mobile Ad Hoc Networks", *International Conference on Peer-to-Peer Computing (P2P 2001)*, Linkoping, Sweden, Aug. 2001

[8] D. Brookshier, *JXTA: Java P2P Programming*, ISBN 0672323664, 2002

[9] Papadopouli, M., Schulzrinne, H.: "Seven Degrees of Separation in Mobile Ad HOc Networks", *IEEE GLOBECOM*, San Fransisco, CA, USA, Nov./Dec. 2000.

[10] Papadopouli, M., Schulzrinne, H.: "Performance of Information Discovery and Message Relying in Mobile Ad Hoc Networks", Columbia University Technical Report CUCS-004-02, Feb. 2002

[11] The SyncML Initiative, SyncML Sync Protocol, Version 1.1, February 2002

[12] T. Berners-Lee, L. Masinter, M. McCahill: "Uniform Resource Locators (URL)", IETF RFC 1738, December 1994.

[13] A. Tanenbaum, M. van Steen, *Distributed Systems Principles and Paradigms*, Prentice Hall 2002.

[14] S. Ballintijn, M. van Steen, and A. S. Tanenbaum, "Scalable naming in global middleware", *International Conference on Parallel and Distributed Computing Systems (PDCS)*, August 2000, pp. 624–631

[15] K. Hildrum, J. D. Kubiatowicz, S. Rao, and B. Y. Zhao, "Distributed object location in a dynamic network", *ACM Symposium on Parallel Algorithms and Architectures (SPAA)*, Winnipeg, CA, August 2002, pp. 41–52

[16] S. Ratnasamy, P. Francis, M. Handley, R. Karp, S. Shenker, "A scalable content addressable network", *ACM SIGCOMM 2001*, San Diego, CA. USA, 2001, pp 161–172.

[17] I. Stoica, R. Morris, D. Karger, M.F. Kaashoek, H. Balakrishnan, "Chord: A scalable peer-to-peer lookup service for internet applications", *2001 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, pp 149–160, 2001.

[18] M. van Steen, F.J. Hauck, P. Homburg, and A.S. Tanenbaum, "Locating objects in wide-area systems", *IEEE Communications Magazine*, 36(1):104–109, January 1998.

[19] B. Y. Zhao, J. Kubiatowicz, and A. Joseph, "Tapestry: An infrastructure for fault-tolerant wide-area location and routing", Technical Report UCB/CSD-01-1141, University of California, Berkeley, 1996.

[20] R. Campbell, *Managing Andrew File System (AFS)*, Prentice Hall PTR, 1998.

[21] M. Satyanarayanan, J.J. Kistler, P. Kumar, M.E. Okasaki, E.H. Siegel, and D.C. Steer, "Coda: A Highly Available File System for a Distributed Workstation Environment", *IEEE Transaction on Computers*, 39(4), April 1990.

[22] R. Guy, P. Reiher, D. Ratner, M. Gunter, W. Ma, G. Popek. "Rumor: Mobile Data Access Through Optimistic Peer-to-Peer Replication", *Workshop on Mo-bile Data Access*, November 1998.

[23] D. Ratner, P. Reiher, and G. Popek, "Roam: A Scalable Replication System for Mobile Computing", *Workshop on Mobile Databases and Distributed Systems (MDDS)*, September 1999.

[24] T.W. Page,Jr., R.G. Guy, G.J. Popek, J.S. Heidemann., "Architecture of the Fi-cus scalable replicated file system", Technical Report CSD-910005, UCLA, USA, March 1991.

[25] Napster, http://www.napster.com

[26] Gnutella, http://www.gnutella.com

[27] B. Groenvall, A. Westerlund and S. Pink, "The Design of a Multicast-based Distributed File System", *Third Symposium on Operating Systems Design and Implementation (OSDI 99)*, New Orleans, Louisiana, February, 1999, pp. 251- 264.

[28] W. J. Bolosky, J. R. Bouceur, D. Ely, M. Theimer, "Feasibility or a Serverless Distributed File System Deployed on an Existing Set of Desktop PCs", *ACM Sigmetrics*, June 2000.

[29] A. Rowstron, P. Druschel, "Storage Management and Caching in PAST, A Large-scale, Persistant Peer-to-peer Storage Utility", *ACM SOSP*, October 2001.

[30] Picco, G.P., Murphy, A.L., and Roman, G.-C., "Lime: Linda meets Mobility," *21st International Conference on Software Engineering*, May 1999, pp. 368-377.

[31] L Capra, C. Mascolo, S. Zachariadis, W. Emmerich "XMIDDLE: A Data-Sharing Middleware for Mobile Computing", Dept. of Computer Science, University College London, Dept. of Computer Science, Gower Street, London, WC1E 6BT UK

[32] D. Gelernter, "Generative Communication in Linda", *ACM Transactions on Programming Languages and Systems*, 7(1), 80-112, 1985

[33] M. Klein, B. König-Ries: "Multi-Layer Clusters in Ad-hoc Networks - An Approach to Service Discovery", *International Workshop on Peer-to-Peer Computing*, Pisa, Italy, Mai 2002

[34] G. Shegalov, M. Gillmann, G. Weikum: "XML-enabled Workflow Management for E-Services across Heterogeneous Platforms", *The VLDB Journal*, Special Issue on E-Services, 2001.

[35] Sanchez, C. and L. Gruenwald, "An Agent-Based Architecture Using XML for Mobile Federated Database Systems", 2*nd International Conference on Mobile Data Management*, Hong Kong, January 2001.

[36] Hodes, T., Katz, R., Servan-Schreiber, E., Rowe, L.: "Composable Ad-hoc Mobile Services for Universal Interaction", *3rd ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom)*, Budapest, Hungary, September 1997

[37] Czerwinski, S., Zhao, B., Hodes, T., Joseph, A., Katz, "R.: An Architecture for a Secure Service Discovery Service", *5th ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom)*, Seattle, WA, USA, August 1999

[38] Q. Ren and M. H. Dunham, "Using Semantic Caching to Manage Location Dependent Data in Mobile Computing", *Mobicom 2000* .

[39] M. H. Dunham and V. Kumar, "Impact of Mobility on Transaction Management," *International Workshop on Data Engineering for Wireless and Mobile Access*, August 1999, pp 14-21.

[40] Zhang, J. and L. Gruenwald, "Spatial and Temporal Aware, Trajectory Mobility Profile Based Location Management for Mobile Computing", *International*

*Database and Expert Systems Applications (DEXA) Workshop on Mobile Database and Distributed Systems*, September 2002.

[41] DARPA Agent Markup Language (DAML), http://www.daml.org

[42] Chung, I., L. Gruenwald, C., S. Chung, "Adaptive Transaction Management Protocols for Mobile Client Caching DBMSs", *Journal of Information Science and Engineering*, Volume 18, Number 3, May 2002.

[43] Gruenwald, L., S. Banik, "A Power-Aware Technique to Manage Real-Time Database Transactions in Mobile Ad-Hoc Networks", *4th International Workshop on Mobility in Database and Distributed Systems*, part of the International Conference on Database and EXpert systems Applications (DEXA), September 2001.

[44] Gruenwald, L., S. Banik, "Energy Efficient Transaction Management for Real-Time Mobile Databases in Ad-hoc Network Environments", *2nd International Conference on Mobile Data Management*, Hong Kong, January 2001.

[45] A. Theobald, G. Weikum, “The XXL Search Engine: Ranked Retrieval of XML Data using Indexes and Ontologies” (Demo Paper), *ACM SIGMOD International Conference on Management of Data*, Madison, Wisconsin, June 2002.

[46] A. Theobald and G. Weikum, “Adding Relevance to XML, The World Wide Web and Databases”, *3rd International WebDB Workshop*, Dallas, Texas, May 2000