Agents, Mobility, and M-services: Creating the Next Generation Applications and Infrastructure on Mobile Ad-Hoc Networks*

Anupam Joshi, Timothy Finin, and Yelena Yesha

Department of Computer Science & Electrical Engineering,
University of Maryland Baltimore County,
Baltimore, MD 21250
`joshi,finin,yeyesha@cs.umbc.edu`

Abstract. With the advent of third generation wireless infrastructure, and the simultaneous emergence of pervasive connectivity for all devices based on Bluetooth like systems for ad-hoc networks, a new vista is open for research in the area of mobile systems. We describe here our ideas for realizing pervasive computing systems based on the cooperation of autonomous, dynamic and adaptive components (hardware as well as software) which are located in vicinity of one another. This is significantly different from "fixed-infrastructure-based", mobile client/server computing between wirelessly connected PDAs and network services. It will enable a new class of applications that effectively exploit mobility and pervasive computing. We address several research problems that span the fields of data management and pervasive computing using multiagent systems as the enabling technology. We propose new agent frameworks and service discovery and delivery protocols most suitable to this new and emerging environment. We are building prototypes and applications in the context of mobile electronic commerce within dynamic communities of ad-hoc mobile services.

1 Introduction

This paper presents a vision and some ideas for realizing next generation mobile systems based on the cooperation of autonomous, dynamic and adaptive components which are located in "vicinity" of one another. These systems will have the property of being composed of a collection of independently designed components that automatically become aware of each other, establish basic wireless communication, exchange information about their basic capabilities and requirements, discover and exchange APIs, and learn to cooperate effectively to accomplish their individual and collective goals.

This represents a fundamental paradigm shift in the way we thing about designing and building complex systems in general, and the way we approach the design of mobile aware applications in particular. We are moving away from

---

* This work supported in part by NSF awards IIS 9875433 and CCR 0070802

B. König-Ries et al. (Eds.): IMWS 2001, LNCS 2538, pp. 106–118, 2002.

the traditional discipline of developing a single, overarching design for a complex system in which all of the parts are carefully engineered to fit together and toward a new approach in which the individual atomic components are designed to be autonomous ("active"), self-describing ("articulate"), highly interactive ("social"), and adaptive ("intelligent").

The aim of our research is to create a robust environment in which independent devices(from computers, PDAs, routers, printers to even home appliances and cars etc.) existing in a location can and will discover, interoperate, and cooperate with the other devices in their vicinity in an as-needed and as-desired basis. The underlying communication technology is presently being developed by consortia such as Bluetooth (http://www.bluetooth.com/), IEEE WLAN/WPAN groups (802.11 and 802.15) and HomeRF (http://www.homerf.org/). An important enabling component of this vision is the work in distributed object-oriented systems which our agent oriented abstraction will build upon. Specifically, systems such as Jini or UDDI which enable the formation of a community of devices/services at the syntactic level make it convenient to add "intelligence" to the system and create communities of "agents" which can then user richer communication languages (ACLs) to communicate information at a semantic level – information about abilities, requirements, commitments, intentions, beliefs, goals, etc.

This idea of "ad-hoc" teams of entities that are dynamically formed to pursue individual and collective goals can be used to create the software infrastructure needed by the next generation of mobile applications. These will use the emerging third and fourth generation broadband wireless systems, as well as short range narrowband systems such as Bluetooth. Heretofore, the software component of mobile computing has lagged behind its hardware (communication, computing and networking) aspects. Much of the research in the software area is often limited to allowing applications built for the wired world (web, databases etc) to run in the wireless domain using proxy based approaches. Our research seeks to move beyond this and provide an agent based framework which uses the power of mobility and ad-hoc wireless connectivity to enable novel applications. The results will point us toward a physical environment in which devices all around us and even on our body are in constant communication and organize themselves to cooperate to do our bidding.

1.1 Hype Vs. Reality for Mobile Applications: M-commerce as a Metaphor

In the past year or two, the research community has seen plenty of hype associated with wirelesss / pervasive / mobile / ubiquitous computing. Mobile Commerce (M-Commerce) in particular was declared as the "killer app" driving the wireless revolution. In what is undoubtedly testimony to the speed at which internet time moves, the last few months have also seen pundits declaring, with equal certainty, that M-Commerce is either a non-starter or that it is dead. In large part, the blame for both the initial hype and the more recent disappointments must rest with the rather narrow vision of m-commerce that some segments of the industry were promoting. In this vision, cell phones (or wirelessly connected PDAs) essentially became mobile storefronts for e-tailers – essentially an incremental change in the present e-tailing idea. We are all familiar with the ads of people buying stuff via their cellphones from the beach. The drawbacks of this idea are not hard to identify, as an increasing number of recent critical commentaries show. This approach essentially treats palmtop devices as consumers (or clients) of goods or information. The information or goods come from servers on the wired side. This approach (typically based on a client–proxy–server model) has been developed by the academia over the last five–six years in contexts such as web access from mobile platforms (for instance[20, 23, 5, 24, 25, 4, 19]) or transaction support for database access[12]. Variants of this approach are now emerging from several commercial companies in the form of systems that allow phone based "microbrowsers" or PDAs to access domain specific internet content (headline news, stocks, sports etc.) from designated portals. The WAP consortium (http://www.wap.com/) is leading efforts amongst telecommunication companies and service providers to evolve a standard mechanism and markup language to support this. There have also been efforts, mostly from startups and e-tailers, to allow people to buy goods using the phone microbrowsers or PDAs. In some sense, one can think of this as a *supermarket approach*, where a few identified service providers exist and the traffic in services is one–way.

Yet viewed in a broader perspective, M-Commerce in particular, and M-Services in general, have yet to be fully articulated or explored, especially in the context of emerging mobile ad-hoc networks. Staying with the M-Commerce idea, consider a move away from the prevailing mobile storefront vision. In the future, instead of just interacting with the "buy–it–yourself" web storefronts, consumers will be able to interact with service providers in a personalized way via automated service-oriented eMarket models (e.g. [17]). The selling of tangible goods will simply be one such service. Other services would include data, information, software components or even processing time/storage on networked machines. There will be no explicit clients and servers – but peers which can be both consumers and providers of different services. M-Commerce will thus be transformed into what we refer to as *Me-Commerce*, or more generally, into *Me-Services*: a personalized view of available services in the neighborhood.

2 A Peek into the Near Future

When Gigi, a graduate student at UMBC walks into the computer science building after a trip to ICDCS, her cell phone starts to ring. It is a call from her personal agent informing her that the notes from the last weeks's research group meeting that she has missed are available for review, and she normally reviews notes before the next meeting which is later today. Gigi walks into her research lab immediately to review these notes. After sitting down in the sofa in the lab, the large screen display automatically shows the notes that Gigi is interested in. She then decides to print out hard copies of the notes. She turns on her PDA and instructs the printing agent in the lab to print out the notes. The printing agent discovers the closest printer from Gigi's current location and submits the print jobs to the printer.

It is 5:30 in the evening, and Jane's panel meeting at NSF in Arlington is just ending. As she steps out of the building to walk the two blocks to her hotel, she decides that after resting for a bit, she'll get ready and have dinner in one of the nearby restaurants. She tells her palmtop of this decision, and asks it to find out nearby restaurants that serve cuisines that she would like to try but can't find in her hometown. The palmtop goes into discovery mode and finds a nearby broker provided by the Arlington Restauranters Association. It sends it Jane's cuisine preferences and price ranges, and asks for a recommendation for a restaurant where she can eat in about 45 minutes from now. The broker has some static information about its local restaurants such as location and menus. Managers in the restaurants are also feeding it dynamic information such as wait times or any discounts/specials that they may have, based on their current situations. Based on both the static and dynamic information, the broker comes up with a list of possibilities. Meanwhile, as Jane is walking back, her PDA asks the PDAs of others in the area for their opinions of good local restaurants and stores them. When Jane is ready to head out, the palmpilot pulls the recommendations from the broker and presents them to Jane. Her choices include Vietnamese, Malaysian, Mongolian and Cambodian restaurants, since these cuisines are similar to the Chinese foods she likes. She selects the Vietnamese restaurant, since it is offering a 20% off coupon and had a couple of good mentions in the information her PDA had gathered from other people. Her PDA communicates this choice to the broker, asks that a reservation be made. It then discovers the local map server that the city of Arlington runs, and gets directions from Jane's hotel to her restaurant.

3 Background and Issues

The basic assumption behind our work is that at the level of computing and networking hardware, we will see dramatic changes in the next few years. Computing will become pervasive – a large number of devices (e.g. phones, PDAs, household appliances) will become computationally enabled, and micro/nano sensors (the so called smart dust) will be widely embedded in most engineered artifacts. All of these devices will be (wirelessly) networked. More specifically, we will assume the emergence of (i) palmtop and wearable/embeddable computers, (ii) Bluetooth like systems which will provide short range, moderate bandwidth connections at extremely low costs, and (iii) widely deployed, easily accessible wireless LANs and satellite WANs. We assume that these will be a mixture of traditional low bandwidth systems and the next generation high speed ones. These developments will lead to wireless networks that will scale all the way from ad hoc body area networks to satellite WANs, and link together supercomputers, "palmstations" and embedded sensors & controllers. There is ongoing research in industry and academia, including work sponsored by NSF, in creating the hardware and low level networking protocols that are needed to realize this vision. Some recent efforts have also started in creating smart spaces and integrated sensor networks.

The scenarios we have described earlier serves to illustrate the technical challenges that we propose to address in our research. In particular, as computing becomes pervasive, an increasing number of entities will be both sources and consumers of data and information. This is a significant change from the present, where mobile devices essentially remain consumers of information, and the research challenge has been to get them the right information in a format suited to the bandwidth and resource constraints of the device[20]. Moreover, existing mobile systems demand interaction from the user. For pervasive systems to succeed, much of the interaction between the plethora of devices will need to happen *in the background*, as it were[29]. In the future we envision, besides the traditional Mobile Support Station based wireless access, many of the devices will communicate using Mobile Ad-hoc Networks (MANET)[16]. Besides obtaining information from “canonical and centralized” sources (such as a yellow pages server for restaurants), a device may obtain information (recommendation about restaurants) or service offers (a discount coupon for dinner) from other devices around it – its present *dynamic community*. The scenario demands that the system have some sense of “vicinity”. The entities in the system must be able to describe themselves as well as discover others and figure how (and whether) to interoperate with them, both at syntactic and semantic levels. Finally, the entities should be able to communicate abstract ideas, (e.g. goals such as what information is it looking for) and be able to negotiate with others for services (I want to know what the traffic conditions are at I 95 Springfield Interchange, in return I can provide traffic conditions in downtown DC.). Of course, the devices participating will be resource constrained and heterogeneous, which poses further problems.

It is pertinent to make the following observations here. First, the scenarios we envision imagine a coexistence of ad-hoc and MSS based networks. Some applications will use one or the other exclusively, while others will need both. Our infrastructure will work in both instances. For example, the restaurant scenario described above uses elements of both types of networks. A totally “centralized” solution, such as always contacting a server like zagat.com, can provide only a subset of the functionalities that our scenario envisions. Secondly, the challenges here go over and beyond those found in heterogeneous and distributed data management, and in ways more subtle than simply handling reduced bandwidth or disconnection, which of course remain important issues. To express this in traditional data management parlance[11], we could say that unlike heterogeneous data access where “schemas” and “catalogs” at least are known in advance, we are talking of a situation where both are highly variable and not known up front. Thus a “query” will return results dependent on where it originates, what location it refers to, and who is around at that time. Finally, note that the personal agent’s call to Gigi is made because of Gigi’s situated location context (the computer science building) has an associated relationship with the problem context and her goal (access to research group meeting notes). The notification action is triggered by the correlations between the contextual information. If either the location context or the problem context/goal has been different, the agent might have not made the call. For example, if Gigi is running low on cat food at home, then it is unlikely the agent would bring this to her attention while she is walking into the computer science building. Similarly, when the printing agent is instructed to print out the notes, it does not require Gigi's explicit interaction to select a printer manually or request the print jobs to be submitted to the closest printer. It has (or can obtain) her printing behavior context that by default, all documents are to be printed to the closest printer and their location shown on her PDA.

4 A MultiAgent Approach to Infrastructure

The DARPA Knowledge Sharing Effort [26] has given rise to a model for developing intelligent cooperating agents based on ontologies, content languages and agent communication languages. Although this approach and its associated components are being adopted as a standard (e.g. FIPA) for agent communication by both the research community and the industry, it requires significant extensions to support the interagent interactions and negotiations in a dynamic, mobile ad-hoc environment. In ongoing research, we are looking at a number of key problems, developing theoretically sound solutions for each, building those solutions into our evolving software infrastructure, and experimenting with the results in the mobile ad-hoc environment, particularly with Bluetooth. We seek to design our system to scale across the problem complexities and reaction time needed for applications, ranging from simple but fast reactive cooperation between agents doing network provisioning to the more complex, slower cooperation needed for business to business e-commerce transactions such as supply chain management.

4.1 Conversation-Based Interaction & Negotiation Protocols

The study of agent communication languages (ACLs) is one of the pillars of current agent research. KQML[13] and the FIPA ACL are the leading candidates as standards for specifying the encoding and transfer of messages among agents. While KQML is good for message-passing among agents, directly exploiting it in building a system of cooperating agents leaves much to be desired. After all, when an agent sends a message, it has expectations about how the recipient will respond to the message. Those expectations are not encoded in the message itself; a higher-level structure must be used to encode them. The need for such conversation policies is increasingly recognized by the KQML community, and has been formally recognized in the latest FIPA draft standard [14, 10].

To date, relatively little work has been devoted to the problem of conversation specification and implementation for mediated architectures. Strides must be taken towards facilitating the construction and reuse of conversations. An ontology of conversations and conversation libraries would advance this goal, as would solutions to the following questions: *(i) Conversation specification:* How can conversations best be described so that they are accessible both to people and to machines? *(ii) Conversation sharing:* How can an agent use a conversation specification standard to describe the conversations in which it is willing to engage, and to learn what conversations are supported by other agents? *(iii) Conversation aggregation:* How can sets of conversations be used as agent "APIs" to describe classes of capabilities that define a particular service? We will develop a rich set of negotiation primitives to support various types of negotiations involving different agents (cooperative, competitive, or a mix of both) and in different schemes, including peer-to-peer, mediated, and multi-party (e.g., auctions). These primitives will be based on the syntax and semantics of the FIPA Agent Communication Language, and encoded in XML. With this approach, the defined negotiation primitives will be consistent with a rich agent communication language that is undergoing an international standardization effort. This ensures the generality and practicality of these primitives.

4.2 Agent Development Framework

Our efforts in this direction are represented by the Ronin Agent Framework [8]. At present, this is a simple Jini-based agent development framework that is designed to aid in the development of next generation smart distributed mobile systems. We point out that Jini is simply a convenient starting point for our current implementation – the ideas developed here will work in any service oriented framework. In particular, it is not clear that Jini is the most optimal solution for pure ad-hoc situations.

In the Ronin Agent Framework we introduce a hybrid architecture, a composition of agent-oriented and service-oriented approaches. Ronin contains a number of features that distinguish it from traditional multiagent frameworks and will make it especially suitable for the mobile domain. These include an ACL and network protocol independent communication infrastructure. Ronin has the notion of service discovery (agent discovery) built into the architecture – this will allow us to integrate our new service and information discovery protocols into it.

The design goal of Ronin is to define an open framework that specifies the infrastructure requirement and the interface guideline for the interaction and communication between agent-oriented components. It models services as agents. Each service consists of two parts: an Agent Deputy and an Agent. An Agent Deputy acts as a front-end interface for the other agents in the system to communicate with the Ronin Agent it represents. Ronin Agent is a service that can be realized as software or hardware. Ronin does not define how an Agent Deputy should communicate with a Ronin Agent. However, it does define the interface for the communication between the Ronin Agent and Agent Deputy. Specifically, each Agent Deputy must implement a `deliver` method. This delivery abstraction means that depending on their connectivity and network QoS, agents can deploy deputies that will provide features of transcoding or disconnection management that we have developed and implemented as a part of this research. Other network and medium specific deputies can be developed and added as the need arises. The messages that are interchanged between Ronin Agents are embedded within `Envelope` objects during the delivery process. This meta-level approach allows Ronin Agents to interchange messages with arbitrary content message types under a uniform communication infrastructure [30]. Within each `Envelope` object, the type of content message and the ontology identifier of the content message are also stored.

There is a set of attributes associated with each Ronin Agent. These attributes can be further divided into two subsets. The first set of attributes, Agent Attributes, define the generic functionality of an agent in domain independent fashion. For example, an agent could be a broker, or a service provider. Ronin framework defines the types and the semantics of Agent Attributes. The second set of attributes, Agent Domain Attributes, define the domain specific functionality of an agent. For example, in a financial domain, an agent could be a stock quote server. The framework neither defines the Domain Attribute types nor their semantics. While domain attributes will allow us to create agents that understand a domain specific ontology, agent attributes provide a common base from which interaction amongst agents from heterogeneous domains can be bootstrapped.

4.3 Agent Based Service Discovery and Information Management in Ad-Hoc Systems

In order for an entity to cooperate with others in its vicinity, it first needs to discover other entities as it moves into a new location. This problem of "service discovery" has recently been explored elsewhere as well in the context of distributed systems. Several interesting issues are also open in terms of service discovery in ad-hoc networks. We use the term service broadly here – it could be a computational component which executes, data/information, or even a physical thing that one entity is willing to provide to the other. As hosts move, the software/hardware/information services/components in their vicinity change. The resource poor nature of many of the tetherless/embedded devices also presents a challenge in terms of what software components they can execute locally. Thus there is a need to develop mechanisms which allow for components to describe themselves (at a semantic level) and their "requirements", as well as for other components to locate them.

State of the art systems such as Jini[1], Salutation[28], UPnP[28], IETF's draft Service Location Protocol[15], E-Speak[18] and Ninja[9], provide for networked entities to advertise their functionality. However, these systems are either tied to a language (Java/Jini), or describe services entirely in syntactic terms as interface descriptions. This not only limits interoperability, but forces a client to know a-priori how to describe a service it needs in terms of an interface. Moreover, they return "exact" matches and can only handle equality constraints. This leads to a loss of expressive power in the component description. For example, the Jini discovery and lookup protocols are sufficient for service clients to find a service that implements the method `printIt()`. However, they are not sufficient for clients to find a printer service that has the shortest print queue, that is geographically the closest, or that will print in color but only within a prespecified cost constraint. Clearly, this can create problems in the mobile and ad-hoc environment, as we have described in [7]. Our ongoing work in this area is to create a smarter service discovery infrastructure. We build on top of Jini, and essentially extend Reggie, Jini's matching engine, to create *DReggie*. Components register their capabilities (what services they can provide) and constraints/requirements (what software/hardware they need to run, how much is the cost to run them) using a DAML (DARPA Agent Markup Languaue, http://www.daml.org) ontology. DReggie (and its earlier version, XReggie[21]) still provides the simple, syntax oriented match of Jini that compares interfaces the component implements. In addition, it will match on DAML descriptions for a component and handle required constraints (is a component mobile? will it run on a dragonball processor? etc.) This matching is fuzzy and inexact. It can either be done by DReggie itself, or by an outside service such as a Prolog engine[8]. As mentioned earlier, in some pure ad-hoc scenarios a Jini based approach is unlikely to work. To show the feasibility of our semantic service discovery ideas, we have developed a similar DAML based extension to Bluetooth SDP[2].

When the services or components in the above scenario are essentially information sources, then several new research issues arise. As we alluded to in our example, in ad-hoc networks the dynamic community of an entity – other entities in its vicinity, are also sources (and potentially consumers) of information. So rather than a client server model, we have a peer–peer interaction. For some queries, instead of getting information from a centralized source, it may be advantageous to "ask around". Consider for example finding the nearest gas station (another car in your vicinity might know), or wanting to find a good Indian restaurant in downtown DC at the end of an NSF review panel (other panelists, or the program manager, might know). Note that this is not merely querying for location dependent data, at least as the term is presently used. Over and above issues associated with traditional distributed databases, there are new challenges here:

- *The Data sources available vary with location and time.* As entities move, their dynamic neighborhood changes, and so depending on when and where a query is given, it may get different answers. There may even be no answer, if no-one in the neighborhood knows. In other words, there is no fixed catalog of information providers that a client can query. We propose to use our service discovery mechanisms to address this problem. Each source describes what information it can provide and whom it is willing to share it with. This will allow dynamic catalog building where the devices are moving slowly, or their registration by a stationary "catalog server". Where the movement is fast, broadcasting the query to all neighbors may be the only solution.
- *Cooperation amongst information sources cannot be guaranteed.* In particular, at any given time, we may have in our neighborhood entities which have

reliable information that we are looking for but refuse to provide it (due to privacy concerns) as well as those who have unreliable information but are willing to share it. Thus users must be able to specify to their devices what information they are willing to share. We propose to use privacy based policy mechanisms to achieve this, and draw upon XML based standards such as P3P from the Web Consortium. Moreover, since there is no prior catalog, no guarantee can be made of the trustworthiness of the information sources – they may intentionally provide misleading information. Distributed trust maintenance is an interesting area of research in itself which some of us are examining (EECOMS project from NIST), but for which we do not seek support in this effort.

- *The query may be explicit or implicit.* For example, if the user normally eats out and likes Indian restaurants, the agent keeps asking others in the vicinity on their behalf and prefetches this information. This requires that the users be able to indicate to their devices what information they normally want. This problem is similar to the "what information I don't want to share" problem. Analogous to the approach there, we will provide XML based mechanisms for the user to specify policy on what they are interested as well as what they would like to ignore. The information may be pulled via a query, or may be passively received as a push from others. The agent on the car computer may listen for information other cars around it are broadcasting (about traffic conditions). It may even be that other devices in the vicinity specifically push information to this device when it is in their vicinity – as Jane passes in front of a pizza hut, the store's agent decides to offer her an on–the–spot coupon for 10% off.
- *Since information sources are not cataloged a-priori, schema translations cannot be done beforehand.* Moreover, due to the limited compute power on palmtop type devices, a dynamic schema translation may not be possible. Our approach will be to use ontologies, and assume an agreement on a base ontology (agent attributes) amongst all entities. Using the base ontology, agents could decide that they do not share the same domain specific schema/vocabulary. They could then seek to discover an entity willing to do translation. Alternatively, one of the agents could obtain the ontology/schema that the other is using and do the mapping itself. For common domains (restaurants, travel, banking, shopping etc.), we assume that domain specific standards will evolve, perhaps pushed by merchants associations and trade groups. For example, the fast food restaurants associations could provide a standard ontology that all its constituents and their agents would use. We assume that these ontologies will be encoded in DAML (DARPA Agent Markup Language), and be published from well know sites. Agents could then inform another of the ontology they are using by simply pointing to the appropriate DTD.

These issues, key in creating the infrastructure that applications will use, remain the focus of our ongoing efforts.

5 Conclusion

We have sketched a vision that integrates exciting new developments in ad-hoc mobile systems (such as Bluetooth) and distributed systems for realizing new kinds of mobile systems based on the cooperation of autonomous, dynamic and adaptive components which are located in vicinity of one another. These systems will have the property of being composed of a collection of independently designed components that automatically become aware of each other, establish basic wireless communication, exchange information about their basic capabilities and requirements, discover and exchange APIs, and learn to cooperate effectively to accomplish their individual and collective goals. The result will point us toward an physical environment in which devices all around us and even on our body are in constant communication and organize themselves to cooperate to do our bidding.

We have also described some ongoing efforts of our eBiquity research group here at UMBC (http://research.ebiquity.org/) towards creating an agent based infrastructure to support this vision. We have focused this paper on the agent infrastructure and service discovery. Our related efforts include securing such pervasive infrastructures using distributed trust based approaches[6], new communication protocols that work on Bluetooth and IR short range networks[3], and developing applications such as secure m-services in a smart space[22, 27] that use this infrastructure.

References

[1] Ken Arnold, Ann Wollrath, Bryan O'Sullivan, Robert Scheifler, and Jim Waldo. *The Jini specification.* Addison-Wesley, Reading, MA, USA, 1999.

[2] Sasikanth Avancha, Anupam Joshi, and Timothy Finin. Enhancing the bluetooth service discovery protocol. Technical Report TR-CS-01-18, CSEE Department, University of Maryland Baltimore County, Aug 2001.

[3] Sasikanth Avancha, Vladimir Korolev, and Anupam Joshi. Transport protocols in wireless networks. In *Proc. IEEE Intl. Conf. Computer Communications and Networks*, Phoenix, October 2001. (to appear).

[4] Harini Bharadvaj, A. Joshi, and Sansanee Auephanwiriyakyl. An active transcoding proxy to support mobile web access. In *Proc. IEEE Sumposium on Reliable Distributed Systems*, October 1998.

[5] E.A. Brewer, R.H. Katz, Y. Chawathe, A. Fox, S.D. Gribble, T. Hodes, G. Nguyen, T. Henderson, E. Amir, H. Balakrishnan, A. Fox, V. Padmanabhan, and S. Seshan. A network architecture for heterogeneous mobile computing. *IEEE Personal Communications Magazine*, 5(5):8–24, 1998.

[6] Andrej Cedilnik, Lalana Kagal, Filip Perich, Jeff Undercoffer, and Anupam Joshi. A secure infrastructure for service discovery and access in pervasive computing. Technical Report TR-CS-01-12, CSEE Department, University of Maryland Baltimore County, Aug 2001.

[7] Harry Chen, Dipanjan Chakraborty, Liang Xu, Anupam Joshi, and Tim Finin. Service discovery in the future electronic markets. In *Proc. AAAI 2000 Workshop on Knowledge Based Electronic Markets*, 2000.

[8] Harry Chen, Anupam Joshi, and Tim Finin. Dynamic sercice discovery for mobile computing: Intelligent agents meet jini in the aether. *Baltzer Science Journal on Cluster Computing*, 4(4), Oct 2001.

[9] Steven E. Czerwinski, Ben Y. Zhao, Todd D. Hodes, Anthony D. Joseph, and Randy H. Katz. An architecture for a secure service discovery service. In *Fifth Annual International Conference on Mobile Computing and Networks (MobiCom '99)*, pages 24–35, Seattle, 1999.

[10] Ian Dickenson. Agent standards. Technical report, Foundation for Intelligent Physical Agents, october 1997.

[11] M. Dunham, P. Chrysanthis, A. Joshi, V. Kumar, K. Ramamritham, O. Wolfson, and S. Zdonik. Issues in wireless data management. Discussion of the wireless data management panel at NSF IDM PIs Meeting, March 2000.

[12] M. Dunham, A. Helal, and S. Balakrishnan. A mobile transaction model that captures both the data and movement behavior. *ACM/Baltzer Journal of Mobile Networks and Applications*, 2(2):149–162, 1997.

[13] Timothy Finin, Don McKay, Rich Fritzson, and Robin McEntire. Kqml: An information and knowledge exchange protocol. In *Proc. Intl. Conf. Building and Sharing of Very Large Knowledge Bases*, Tokyo, 1993.

[14] FIPA. FIPA 97 specification part 2: Agent communication language. Technical report, FIPA - Foundation for Intelligent Physical Agents, october 1997.

[15] E. Guttman, C. Perkins, J. Veizades, and M. Day. "rfc2068: Service location protocol, version 2". ftp://ftp.isi.edu/in-notes/rfc2608.txt, 1999.

[16] Z. J. Has. Panel report on ad hoc networks - MILCOM'97. *Mobile Computing and Communications Review*, 2(1), January 1998.

[17] A. Helal, M. Wang, A. Jagatheesan, and R. Krithivasan. Brokering based selforganizing e-service communities. In *Fifth International Symposium on Autonomous Decentralized Systems (ISADS)*, Dallas, Texas, March 2001.

[18] Hewlett-Packard. *E-Speak Architectural Speciification*, version beta 2.2 edition, December 1999.

[19] A. Joshi, S. Weerawarana, and E. N. Houstis. Disconnected Browsing of Distributed Information. In *Proc. Seventh IEEE Intl. Workshop on Research Issues in Data Engineering*, pages 101–108. IEEE, April 1997.

[20] Anupam Joshi. On proxy agents, mobility and web access. *ACM/Baltzer Journal of Mobile Networks and Applications*, 2000. (accepted for publication, also availbe as UMBC CS TR 99-02).

[21] Anupam Joshi and Liang Xu. A jini based framework for a component recommender system. In *Proc. Sixteenth IMACS World Congress*, August 2000.

[22] Lalana Kagal, Vladimir Korolev, Harry Chen, Anupam Joshi, and Tim Finin. Centaurus: A framework for intelligent services in a mobile environment. In *Intl. Workshop on Smart Appliances and Wearable Computing, Workshop of* 21^{st} *IEEE ICDCS*, April 2001.

[23] R. H. Katz, E. A. Brewer, E. Amir, H. Balakrishnan, A. Fox, S. Gribble, T. Hodes, D. Jiang, G. T. Nguyen, V. Padmanabhan, and M. Stemm. The bay area research wireless access network (barwan). In *Proceedings Spring COMPCON Conference*, 1996.

[24] M. Liljeberg, M. Kojo, and K. Raatikainen. Enhanced services for world-wide web in mobile wan environment. http://www.cs.Helsinki.FI/research/mowgli/mowgli-papers.html, 1996.

[25] B. D. Noble, M. Satyanarayanan, D. Narayanan, J. E. Tilton, J. Flinn, and K. R.Walker. Agile application-aware adaptation for mobility. In *Proceedings of the 16th ACM Symposium on Operating System Principles.*

[26] Ramesh S. Patil, Richard E. Fikes, Peter F. Patel-Schneider, Don McKay, Tim Finin, Thomas Gruber, and Robert Neches. The DARPA knowledge sharing effort: Progress report. In Bernhard Nebeld, Charles Rich, and William Swartout, editors, *Principles of Knowledge Representation and Reasoning: Proceedings of the Third International Conference (KR92)*, pages 777–788. Morgan Kaufman, 1992.

[27] Olga Ratsimor, Vladimir Korolev, Anupam Joshi, and Timothy Finin. Agents2go: An infrastructure for location-dependent service discovery in the mobile electronic commerce environment. In *Proc. ACM MOBICOM Workshop on Mobile Commerce*, Rome, July 2001.

[28] Rekesh John. UPnP, Jini and Salutaion - A look at some popular coordination framework for future network devices. Technical report, California Software Labs, 1999. Available online from.

[29] Mark Weiser. Some Computer Science Issues in Ubiquitous Computing. *Comm. ACM*, 36(7):75–85, July 1993.

[30] H. Chen X. Luan. A meta protocol for agent communication. Technical report, University of Maryland Baltimore County, 1999.