Profile Driven Data Management for Pervasive Environments *

Filip Perich, Sasikanth Avancha, Dipanjan Chakraborty, Anupam Joshi, Yelena Yesha

Department of Computer Science and Electrical Engineering
University of Maryland Baltimore County
1000 Hilltop Circle, Baltimore, MD 21250, USA
{fperic1, savanc1, dchakr1, joshi, yeyesha}@csee.umbc.edu

.

Abstract. The past few years have seen significant work in mobile data management, typically based on the client/proxy/server model. Mobile/wireless devices are treated as clients that are data consumers only, while data sources are on servers that typically reside on the wired network. With the advent of "pervasive computing" environments an alternative scenario arises where mobile devices gather and exchange data from not just wired sources, but also from their ethereal environment and one another. This is accomplished using ad-hoc connectivity engendered by Bluetooth like systems. In this new scenario, mobile devices become both data consumers and producers. We describe the new data management challenges which this scenario introduces. We describe the design and present an implementation prototype of our framework, MoGATU, which addresses these challenges. An important component of our approach is to treat each device as an autonomous entity with its "goals" and "beliefs", expressed using a semantically rich language. We have implemented this framework over a combined Bluetooth and Ad-Hoc 802.11 network with clients running on a variety of mobile devices. We present experimental results validating our approach and measure system performance.

1 Introduction

The client/proxy/server model underlies much of the research in mobile data management. In this model, mobile devices are typically viewed as consumers of information, while data sources reside on the wired network. The main emphasis is on the development of protocols and techniques to deal with disconnection management and low bandwidth. The aim is often to allow applications built for the wired world (e.g., WWW, databases etc.) to run in wireless domains using proxy based approaches ([7]). In systems based on the 2.5/3G cellular infrastructure, the traditional client–proxy–server interaction is perhaps an appropriate model, where the "client" database can be extremely lightweight [1] or has a (partial) replicate of the main database on the wired side [13].

With the spread of short-range ad-hoc networking systems (such as Bluetooth[1]) that enable devices to spontaneously interact with others in their vicinity, an alternative

* This work was supported in part by NSF awards IIS 9875433 and CCR 0070802, DARPA DAML program, and IBM

[1] http://www.blutetooth.org/

approach will be necessary. Mobile devices (hand-held, wearable or embedded in vehicles), computers embedded in the physical infrastructure, and (nano)sensors will all be able to discover and inter-operate with others in their vicinity. The mobile devices will be able to obtain more context-sensitive data by interacting with peers in their "vicinity" than by contacting a fixed data source on the wired network. In addition to the traditional challenges of mobile networks (i.e., low bandwidth and disconnection), this pervasive paradigm introduces new problems in terms of the environment's stability and accessibility. The connection time to a data source is often limited, as well as the likelihood of reconnecting to the same data source once disconnected. Accordingly, pervasive connectivity will require the mobile devices to be highly adaptive as well.

The objective of this paper is to articulate the requirements for, and present a preliminary implementation of, a robust infrastructure in which independent devices existing in a particular location will be able to collaborate with their mobile peers achieving higher data availability and currency. In this vision of pervasive computing environments, mobile devices are both sources and consumers of information and cooperate with others in their vicinity to pursue their individual and collective information needs.

The remainder of the paper is structured as follows. Section 2 discusses existing work in the area of distributed data management in mobile networks and in the area of user profiles. In section 3 we present new challenges and problems that this scenario introduced to pervasive environments that go beyond distributed database frameworks. In section 4 we describe the framework design and present system level details of its implementation. We conclude with section 5 and describe a future work.

2 Related Work

The problem of data management in a distributed environment has been well researched, both in terms of wired infrastructure and infrastructure-based wireless networks (e.g., MobileIP). The work on distributed and federated databases is well-known in the community [10]. Accordingly, we present work related to data management in wireless networks, and a short discussion of related work in the area of expressing user profiles.

Data Management in Wireless Networks: Chrysanthis *et al* [9] consider disconnected operations within mobile databases by presenting a mechanism, which they call a "view holder", that maintains versions of views required by a particular mobile unit. They also propose an extension to SQL that enables the profile- and capability-based programming of the view holders. Kottkamp and Zukunft [8] present optimization techniques for query processing in mobile database systems that include location information. They present a cost model and different strategies for query optimization incorporating mobility specific factors like energy and connectivity. Bukhres *et al* [3] propose an enhancement to the infrastructure-based mobile network model of Mobile Hosts (MHs) connected over a wireless virtual subnet and Mobile Support Stations (MSSs) connected to a wired static network. They recommend the addition of a mailbox, which serves as a central repository for the MHs that is maintained by the cellular provider and duplicated in all the MSSs. Pitoura [11] presents a replication schema based on augmenting a mobile database interface with operations with weaker consistency guarantees. An implementation of the schema is presented by distinguishing copies into quasi and core; protocols for enforcing the schema are introduced. Finally, Demers *et*

al [5] present the Bayou architecture, which is a platform of replicated, highly available, variable-consistency, mobile databases for building collaborative applications.

We note that in most of the previous work, the wireless networks are supported by the fixed, wireline infrastructure, where query optimization techniques require the support of wireline networks. Our work assumes no support from the fixed infrastructure. When the MH requires instantaneous information (e.g. traffic updates or bad weather warnings), it may be more easily accessible from other "local" MHs than a fixed node. In our work, a mobile device is always in nomadic mode, as defined by [3] and [8].

User Profiles: The data management community of late has been advocating the use of "profiles", especially when dealing with pervasive systems and stream data. For instance, Ren and Dunhamm [12] represent a profile as a collection of continuous location dependent data queries. The location dependent data is described in terms of tuples in a single-relational database (e.g., all hotels and restaurants in a city). A user then specifies her preferences by constructing several SQL queries based on the database schema. In a seminal paper, Cherniak *et al* [4] explore the use of profiles in the area of client/server based data recharging for mobile devices. They discuss the requirements for a successful profile as well as describe the need for a formal language that enables expressing such profiles. Their profile consists of two sections, namely the "domain", which is responsible for the data description, and the "utility", which is a numerical function denoting the data importance in respect to other information. While a step in the right direction, we argue in section 3 that this notion of profile is somewhat limited.

3 Challenges of Data Management in Pervasive Environments

If each entity in pervasive environments is capable of both posing and answering queries, we can describe this model as a type of mobile distributed database. However, it is far more complex than the conventional client-proxy-server based model. We illustrate this by classifying our environment along of four orthogonal axes, i.e., autonomy, distribution, heterogeneity, and mobility ([6]). This system is highly autonomous since there is no centralized control of the individual client databases. It is heterogeneous as we only assume that entities can "speak" to each other in some neutral format. The system is clearly distributed as parts of data reside on different computers, and there is some replication as entities cache data/metadata. Mobility is of course a given – in *ad-hoc* networking environments, devices can change their locations, and no fixed set of entities is "always" accessible to a given device. This is distinct from the issue of disconnection management that is traditionally addressed in the work on mobile databases. In those systems, disconnections of mobile devices from the network are viewed as temporary events and when reconnected, any ongoing transactions between the mobile and the server will either continue from the point of disconnection or be rolled back.

As devices move, their neighborhood changes dynamically. Hence, depending on the specific location and time a particular query is given, the originator may obtain different answers or none at all. Unlike traditional distributed database systems, the querying device cannot depend on a global catalog that would be able to route its query to the proper data source. Additionally, under high mobility conditions where current wireless networking technologies cannot support stable connections, there is no guarantee that the device will be able to access information that resides on neighboring devices.

In other words, data is pervasive – it is not stored in a single repository but is distributed and/or replicated in an unknown manner among a large number of devices and only some of which are accessible at any given time. Querying is by similar reasoning *serendipitous*, if one asks a question to which the answer is stored in the vicinity then the query succeeds. Such a situation seems to leave too much to chance. To improve the chances of getting an answer to a question no matter when it is asked, each device should have the option to cache the metadata (e.g who has what data) and perhaps even the data obtained from neighbors in its current vicinity. To further complicate matters, each data source may have its own schema. Not all possible schema mappings can be done a-priori, and some devices will be unable to translate them due to their limited computational capabilities.

In addition, cooperation among information sources cannot be guaranteed. Clearly, the issues of privacy and trust will be very important for a pervasive environment, where transactions and involved entities are random [14]. Accordingly, there may be an entity that has reliable information but refuses to make it available to others, while another is willing to share information which is unreliable. Lastly, when an entity makes information available to another entity, questions regarding its provenance, as well as protection of future changes and sharing of that information arise.

In general, for pervasive systems to succeed, much of the interaction between the devices must happen in the background, without explicit human intervention. These interactions should be executed based on information in the profile. For instance, a diabetic user's profile can say "Always keep track of the nearest hospital", influencing what data the *InforMa* will seek to obtain and which information sources it will interact with. Of course, the question arises: "what exactly should a profile contain?" As we mentioned, perhaps the best work on profile driven data management is due to Franklin, Cherniak and Zdonik [4]. We argue that their profiles, which explicitly enumerate data and its utility, are not sufficient. A profile should not simply consist of information about utility values of fixed data domains, since in pervasive computing environments both the domains of data which a user may need, as well as its utility, will change with the changing context of the user. We believe that a profile should be described in terms of "beliefs", "desires", and "intentions" of the user, a model which has been explored in multi-agent interactions [2]. The "beliefs" represent information that should be treated as facts, and assigned "utility" and "reliability" values or functions to enable comparison with other information stored in a profile. For example, these may include information about user's schedule or cuisine preferences. The information in the profile of Cherniak *et al.* would in our system be treated as "beliefs". A "desire" represents a wish the user would like to accomplish. Each desire is also assigned "utility" value and function. Lastly, an "intention" represents a set of intended tasks – these can either be deduced from "desires", or be explicitly provided. For example, the profile may contain user's desire to listen to country music as performed by Shania Twain. The system should deduce from this the intention to download Shania Twain MP3s. This intention would then influence the information gathering behavior of the user's PDA. Alternately, a user may explicitly provide an intention to purchase Shania Twain CDs. Upon entering a mall, the device will try to obtain information about new releases from the local music stores. The "intentions" of user, modulated by the "beliefs" as well as

contextual parameters, including location, time, battery power, and storage space, allow the *InforMa* of each entity to determine what data to obtain and its relative worth. We note here that the context information could also be regarded as "beliefs" that were dynamically asserted and retracted.

We believe that a semantically rich language, such as the one currently pursued by the W3C Consortium and DARPA (DAML+OIL[2]), provides a rich framework to represent our enhanced profiles. The advantages of this selection are two fold: By adhering to an already existing language, the syntax and rules do not have to be duplicated by creating a new formal language. Secondly, by utilizing a language used by the Semantic Web, the devices will be able to use the vast resources available on the Internet as well as the resources available in *ad-hoc* networks.

4 Framework Design and Implementation

Our framework is designed to handle serendipitous querying and data management efficiently and scalably in mobile *ad-hoc* environments. The framework consists of multiple instances of the following components: a meta-data representation, a profile, a communication interface, an information provider, and an Information Manager that we call *InforMa*.

4.1 Metadata Representation

The schema (ontology) for every information provider, information instances and even a profile must be understood by other entities in the environment. If this is not true, then the information is useless. At the same time, it is theoretically possible that all schemas are described in a different language. In this worst-case scenario, the existence of a schema translator becomes paramount. We can easily see that this is not a scalable solution and that the translator quickly becomes a bottleneck, preventing smooth exchange of information. We have, therefore, decided to use a common language to describe the schema for any information provider and chosen the semantic DARPA Agent Markup Language (DAML) for this purpose. In addition, instances of information, such as queries and answers, together with a user profile are also described in DAML.

We focused our efforts on developing ontologies that would be most useful for devices in moving vehicles. These include ontologies for describing user profiles, and for describing queries and answers [3]. Common well-known information providers, useful to such devices, are emergency related (e.g., police, medical, and fire department), traffic and road condition related, weather related, and maintenance related (e.g., gas station, towing service etc.). The ontologies are based on and very similar to the DAML-S ontology[4], which attempts to comprehensively describe services for the WWW. Using the DAML-S like description, we are able to match queries with information provider registration information as well as with particular answer instances. Accordingly, a device can describe itself by defining the appropriate service models it implements, the process models that provide the information, and the required inputs to be provided.

[2] http://www.daml.org/
[3] http://mogatu.umbc.edu/ont/
[4] http://www.daml.org/services/

4.2 Profile

Since some entities in the *ad-hoc* environment are required to operate autonomously without an explicit human intervention, the entities must have access to individual rule-based profiles. These profiles are used to determine both the current and the future actions of the entities. We model a profile in MoGATU in terms of "beliefs", "desires", and "intentions", and encode it using DAML-based ontologies as we have previously discussed in Section 3. The "intentions" of user, modulated by the "beliefs" as well as contextual parameters, including location, time, battery power, and storage space, allow the *InforMa* of each entity to determine what data to obtain and its relative worth. We note here that the context information could also be regarded as "beliefs" that were dynamically asserted and retracted.

4.3 Information Providers

Every device manages a subset of the world knowledge repository that it can provide to itself and possibly to others. Of course, this subset may be inconsistent with the knowledge of other devices and may even be empty. An entity is an information provider when it possesses the capability to accept a query and generate an appropriate response based on the body of knowledge (mainly facts) under its control. These facts could be associated with practically anything in the world, for example the location of gasoline service stations in a certain area and price of gasoline at each station. Moreover, any device in our framework can provide information about more than one class of knowledge. At the same time, some devices may be too resource-limited or otherwise restricted to be able to store or share any information at all. These devices only use information advertised by peers in their environment.

Information providers register themselves with the local instance of *InforMa*. They may also register themselves or be registered with remote *InforMa*s. In the latter case, both information about information providers and the information under the control of those information providers is disseminated to all parts of the ad-hoc environment.

In our implementation, an information provider registers itself upon start-up with its *InforMa* by sending a registration request including a description of its schemas. *InforMa* adds it to its list of local and remote information providers. *InforMa* now knows how to route it any query that this provider is able to answer. On receiving a query, the provider attempts to answer it and sends back its response to the local *InforMa* which routes it back to the source. Renewal of registration information at a remote *InforMa* is the sole responsibility of the provider. *InforMa* simply removes the information related to the provider from its table, once the provider's lifetime has expired.

4.4 *InforMa*: Information Manager

Every entity implements *InforMa*, which is a local metadata repository that includes schema definitions for locally available information providers and particular facts such as queries and answers for local and non-local information providers. Accordingly, *InforMa* stores advertised schema for local information providers and also for those that it believes the device can reach by communicating with other devices in its vicinity. In addition, *InforMa* stores facts that were produced locally or that were obtained from

others. For example, when a device has a local weather information provider and it furthermore knows that it is raining, *InforMa* includes metadata to reflect that knowledge.

Based on their model of interaction with their peers, we can define four basic types of *InforMa* instances. In the most simple form, the *InforMa* maintains required information about information providers locally present on the device. Each entity in the *ad-hoc* environment is required to implement this type of *InforMa*. We believe that this type would be most suitable for resource-limited devices. In addition, this particular form of *InforMa* is most suitable for entities whose environment changes rapidly. Any time a query is posed, *InforMa* is contacted to provide an answer. *InforMa* first attempts to determine whether any local information provider is capable of answering the query and contacts it. Otherwise, the *InforMa* tries to locate some remote information provider and requests its assistance. Finally, when all previous attempts fail, it attempts to contact all of its neighbors to ask them for their help. However, once the query is satisfied, the *InforMa* may choose to forget any information obtained from the other entities, in order to save memory. As an extension to the first type, the *InforMa* may decide to temporarily store the foreign information in the hope that a future query may be answered by reusing it. In this category, knowledge available to *InforMa* still remains restricted to the entity. For more resource-rich devices, the *InforMa* may decide to not only store information related to local information providers (and the ones obtained while answering local queries) but also accept information that was disseminated by other entities in its vicinity. Henceforth, *InforMa* is now more capable and efficient in satisfying queries that originate from its home entity. Finally, the most capable *InforMa* instance makes its knowledge available to all entities in its vicinity by accepting their query requests and by actively advertising its knowledge.

Query Answering: InforMa uses all the information encoded in the DAML metadata to find an appropriate answer or to locate an information provider that could potentially answer the query. In our framework, each *InforMa* first tries to find a valid non-expired answer. Next, it tries to match a local information provider, a remote information provider, or at least some other *InforMa* that could have a richer cache. The matching is done by finding the appropriate process model, and validating all inputs and outputs when necessary. We have implemented the current framework using graph and search techniques; however, it is possible that more capable *InforMa* entities may also utilize more powerful reasoning techniques using Prolog engines.

Caching: Every *InforMa* has a limited cache in which it stores registration information, queries and answers for a short period of time. Depending on its mode of operation, the cache size, the arrival rates of registration information and queries, and the lifetime of the registration information, *InforMa* may or may not be able to answer a certain query. In order to increase the chances of responding positively to a certain query and to decrease response time, *InforMa* can employ various cache-replacement algorithms to cache responses to previous queries together with information provider advertisements. We have implemented both hybrid Least Recently Used (LRU) and Most Recently Used (MRU) replacement algorithms utilizing a priority-based scheme allowing *InforMa* to determine what information to retain and what to discard. These algorithms work like the standard LRU and MRU algorithms; however, they assign the highest priority to local information providers first, followed by remote information

providers, and the lowest priority to answers to previous queries. Additionally, we have implemented a preliminary semantic cache replacement algorithm that uses a profile. The user profile is used to generate standing queries as well as to determine the utility value of all cached information. The device uses these queries to contact other devices in its vicinity in the hope of obtaining information that the user may require in near future. The semantic replacement algorithm applies the utility values of all cached and incoming information to manage the limited cache size. Accordingly, along with time considerations of LRU and MRU algorithms, the semantic-based algorithm also covers other contextual dimensions.

Advertisement and Solicitation: In addition to supporting the registration of information providers, our framework also supports the concept of solicitation of information about information providers. Henceforth, every *InforMa* can periodically send solicitation requests to its peers. When a new information provider is discovered, *InforMa* caches the information if possible. Similarly, every *InforMa* can advertise its information providers to all its neighbors in the vicinity. One important point to emphasize here is that solicitation of information providers from remote *InforMa*s and broadcast-based advertisement is restricted to 1-hop neighbors only. This prevents unnecessary flooding of this information across the network.

Multi-Hop Networks and Routing: It is possible for a query to travel multiple hops within our framework. Every *InforMa* knows either the final destination of a particular message or a route to it. It obtains the information as follows: when a remote query or registration arrives on the Bluetooth or the Ad-Hoc 802.11 interface, *InforMa* stores the address of the remote device in its routing table. When, on the other hand, it receives a remote forwarded query or registration request, it notes in its routing table that the source of the message can be reached through the forwarder, unless it already knows how to reach the source. To facilitate this routing mechanism, we ensure that every message contains the Bluetooth or the 802.11 device address of the source.

4.5 Experimental Results

For evaluating our framework, we simulated the scenario of devices exchanging data while passing by one another. This was done with the use of several laptops and iPAQ 3870 devices. Connectivity was provided by Bluetooth (embedded in the iPAQs, Ericsson cards connected on serial ports on the laptops) and 802.11 cards in Ad-Hoc mode. Our first set of experiments simply validated the working of the system. A device was able to discover information from nearby devices, both directly connected and those more than a hop away. It was also able to cache data and respond to queries. The test consisted of four devices and was executed over a simulated period of 100 minutes. Device *A* was able to communicate with device *B*, which in turn was in range of devices *C* and *D*. Device *A* provided weather information and device *D* had information about locations and prices of nearby gas stations. We evaluated the system by randomly selecting a query and assigning it to one of the four devices while monitoring information present at each cache. For example, when device *A* asked for the closest gas station, it was able to deduce that device *D* contains the required information and that the query should be routed through device *B*. Moreover, when device *B* received the query, it was able to immediately return a cached answer instead of routing the request to device D.

We next studied the impact on the performance by varying the cache size of each *InforMa*. Since our present implementation must linearly scan the cache to see a matching DAML structure, increasing cache size increases the amount of time spent on this scan. However, even for a 30K cache, the processing time was on average 5ms per query after 100 runs. The network transmission time completely dominates this in Bluetooth environments (4.56s to transfer a 1.0KB "query" and to send the response). Even for the much faster 802.11 devices, the time needed to send the query and receive the answer over the network was five times longer (27ms combined round trip time)! We expect to incorporate ongoing research into creating indices for DAML statements in future versions of *InforMa* so as to increase the processing speed.

Next, we compared the performance of cache replacement policies, namely LRU, MRU and the semantics-based algorithm that we propose. Intuitively, the algorithm that can use the information in the profile to know that type of information the user needs should do a better job of caching. To validate this, we considered a scenario where the device is mobile and receiving two types of advertisements - one from local restaurants and the other from clothing stores over a simulated period of 100 minutes. The probability of the device receiving either advertisement type was 0.5. The profile indicated that the utility of the restaurant information was 9 times that of the clothing store information. As such, the simulated user queried about restaurants 9 times as often as about clothing stores. The goal of the device was to anticipate the user's future demand as dictated by the user's profile, consequently ensuring that the needed information was cached (if previously available). The worst performer in this scenario was MRU with success rate of 0.37. This was followed by LRU with success rate of 0.63. The best performance was the semantic replacement algorithm, with success rate of 0.87. The success of the semantic replacement algorithm is attributable to the high accuracy of the predictions based upon the profile.

Lastly, we studied the performance of our framework in one-hop and multi-hop networks. As we have described earlier, the transmission speeds for a single hop were 4.56s seconds in Bluetooth environment and 27ms seconds for 802.11 based devices. The measured results indicate that Bluetooth connectivity is fast enough to allow exchanges and interactions in relatively stable environments (e.g. a person in a mall, cars traveling in the same direction on a highway) which cover many of the pervasive computing scenarios. However, if the relative speed of objects in motion is high, then current networking technology clearly precludes the use of serendipitous querying.

5 Conclusions and Future Work

We have addressed the issues of mobile data management in pervasive environments by proposing a framework that is capable of handling a heterogeneous set of mobile devices. We have presented the need for a robust framework enabling serendipitous querying in mobile *ad-hoc* environments. Each device is represented by one information manager, *InforMa*, and a number of information providers (data sources). Each device may also contain a profile reflecting the user's "beliefs", "desires", and "intentions". The profile and other information is encoded in a semantically rich language. Our implementation concurrently operates over both Bluetooth and 802.11 Ad-Hoc networks. The *InforMa* collects information about the current environment, and uses

it in conjunction with the profile to predict its user's future requirements. We experimented and tested three alternative cache replacement policies as well as query routing over multi-node paths. Our results show that the semantic cache replacement algorithm outperforms both LRU and MRU replacement policies. Additionally, our results show that both Bluetooth and 802.11 support routing over multiple nodes.

In this paper, our focus was to address the issues concerned with querying and processing, which are similar to read-only mode operations in information access systems. We will extend our model by adding write-mode data access patterns allowing for full transaction capabilities.

References

1. C. Bobineau, L. Bouganim, P. Pucheral, and P. Valduriez. PicoDBMS: Scaling down Database Techniques for the Smartcard. In *VLDB*, 2000.
2. M. Bratmann. *Intentions, Plans, and Practical Reason*. Harvard University Press, 1987.
3. O. Bukhres, S. Morton, P. Zhang, E. Vanderdijs, C. Crawley, J. Platt, and M. Mossman. A Proposed Mobile Architecture for Distributed Database Environment. Technical report, Indiana University, Purdue University, 1997.
4. M. Cherniak, M. Franklin, and S. Zdonik. Expressing User Profiles for Data Recharging. *IEEE Personal Communications*, July 2001.
5. A. Demers, K. Petersen, M. Spreitzer, D. Terry, M. Theimer, and B. Welch. The Bayou Architecture: Support for Data Sharing among Mobile Users. In *Proc. IEEE Workshop on Mobile Computing Systems & Applications*, 1994.
6. M. Dunham and A. Helal. Mobile computing and databases: Anything new? *ACM SIGMOD Record*, 24(4), December 1995.
7. A. Joshi. On Proxy Agents, Mobility and Web Access. *ACM/Baltzer Journal of Mobile Networks and Applications*, 2000.
8. H. Kottkamp and O. Zukunft. Location-Aware Query Processing in Mobile Database Systems. In *Proc. of the ACM Symposium on Applied Computing*, Feb. 1998.
9. S. Lauzac and P. Chrysanthis. Utilizing Versions of Views within a Mobile Environment. In *DEXA*, pages 408–413, Aug. 1998.
10. M. Tamer Oezsu and Patrick Valduriez. *Principles of Distributed Database Systems*. Prentice Hall, Inc., New Jersey, 2nd edition, 1999.
11. E. Pitoura. A Replication Schema to Support Weak Connectivity in Mobile Information Systems. In *DEXA*, 1996.
12. Q. Ren and M. Dunham. Using Semantic Caching to Manage Location Dependent Data in Mobile Computing. In *ACM MobiCom'00*, 2000.
13. C. Tait, H. Lei, S. Acharya, and H. Chang. Intelligent File Hoarding for Mobile Computers. In *ACM MobiCom'95*, 1995.
14. J. Undercoffer, F. Perich, A. Cedilnik, L. Kagal, and A. Joshi. A Secure Infrastructure for Service Discovery and Access in Pervasive Computing. *ACM Monet: Security in Mobile Computing Environments*, Spring 2002.