Report Number: WUCSE-2002-36

2002-09-29

Supporting Context-Aware Computing in Ad Hoc Mobile Environments

Qingfeng Huang

Context awareness is important for many cooperation and coordination scenarios in ad hoc mobile environments. Developing context-aware applications in a dynamic and volatile ad hoc mobile setting is a complex task. The goal of this proposal is to investigate implications of ad hoc mobility on application design and development, to explore design principles and general design frameworks, and to deliver a middleware to facilitate application development for ad hoc mobile systems. The middleware aims to provide a set of toolkits and services for efficient and reliable information discovery and dissemination in ad hoc mobile environments. We believe this research... **Read complete abstract on page 2.**

Supporting Context-Aware Computing in Ad Hoc Mobile Environments

Qingfeng Huang

Complete Abstract:

Context awareness is important for many cooperation and coordination scenarios in ad hoc mobile environments. Developing context-aware applications in a dynamic and volatile ad hoc mobile setting is a complex task. The goal of this proposal is to investigate implications of ad hoc mobility on application design and development, to explore design principles and general design frameworks, and to deliver a middleware to facilitate application development for ad hoc mobile systems. The middleware aims to provide a set of toolkits and services for efficient and reliable information discovery and dissemination in ad hoc mobile environments. We believe this research effort will result in new insights, algorithms and design principles much needed in this area.

Department of Computer Science & Engineering

Washington University in St. Louis
School of Engineering & Applied Science

2002-36

Supporting Context-Aware Computing in Ad Hoc Mobile Environments

Authors: Huang, Qingfeng

September 29, 2002

Abstract: Context awareness is important for many cooperation and coordination scenarios in ad hoc mobile environments. Developing context-aware applications in a dynamic and volatile ad hoc mobile setting is a complex task. The goal of this proposal is to investigate implications of ad hoc mobility on application design and development, to explore design principles and general design frameworks, and to deliver a middleware to facilitate application development for ad hoc mobile systems. The middleware aims to provide a set of toolkits and services for efficient and reliable information discovery and dissemination in ad hoc mobile environments. We believe this research effort will result in new insights, algorithms and design principles much needed in this area.

Department of Computer Science & Engineering - Washington University in St. Louis
Campus Box 1045 - St. Louis, MO - 63130 - ph: (314) 935-6160

Supporting Context-Aware Computing in Ad Hoc Mobile Environments

Qingfeng Huang
Department of Computer Science and Engineering
Washington University
Saint Louis, MO 63130
qingfeng@cse.wustl.edu

September 29, 2002

Abstract

Context awareness is important for many cooperation and coordination scenarios in ad hoc mobile environments. Developing context-aware applications in a dynamic and volatile ad hoc mobile setting is a complex task. The goal of this proposal is to investigate implications of ad hoc mobility on application design and development, to explore design principles and general design frameworks, and to deliver a middleware to facilitate application development for ad hoc mobile systems. The middleware aims to provide a set of toolkits and services for efficient and reliable information discovery and dissemination in ad hoc mobile environments. We believe this research effort will result in new insights, algorithms and design principles much needed in this area.

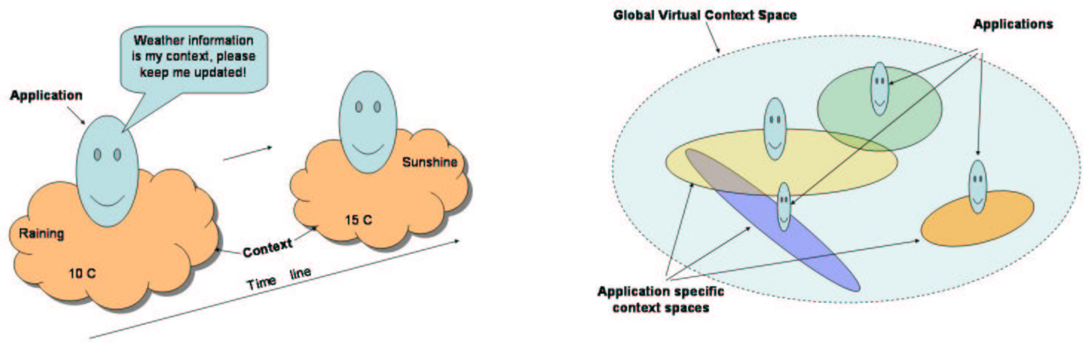
1 Introduction

People have always been consciously and unconsciously using context information in making judgments, in taking decisions, in interacting with each other. For instance, when we talk to friends in a movie theater, we keep our voice lower than we normally do in a park. When a visiting friend asks "which Chinese restaurant is the best?", we know he or she implicitly meant to ask which one is the best "in town". By noticing environmental differences, by understanding implicit assumptions in a situation, we are able to behave and respond in an appropriate manner. In other words, it is the use of "context" that reduces the needed bandwidth in our interactions, makes our communication more efficient, and (in a sense) makes us "more intelligent".

What if computers were context-aware? What if they could act like real personal or work assistants, rather than passive assistants as they are today? What if an application could change its behavior depending on where it was, who was around, or even what temperature it was? Making computers "context-aware" is an idea that attracts more and more researchers within the computer science communities to the field of context-aware computing. Researchers in the field of *Artificial Intelligence* are trying to formalize the notion of context and formalize reasoning using context [32, 8, 2, 19, 5, 7]. Researchers in the fields of *Human Computer Interaction*, *Mobile Computing*, and *Ubiquitous Computing* are building prototype context-aware systems for use in various settings [45, 46, 4, 1, 14, 11, 3, 31, 40]. For example, Cyberguide is a mobile context-aware tour guide developed at George Tech. By using knowledge about the user's current location as well as a history of past locations, it provides the kind of services that we expect from a real tour guide [1]. These and other prototypes have demonstrated the potential of context-aware applications. We see that human-computer interactions become more pleasant when computers can properly incorporate context into their behavior.

While intelligent context-awareness is desirable, building context-aware systems is a complex task. This is not only because one has to carefully identify the "most relevant" subset of context and work out the appropriate logic of using the context for a specific application, but also because currently one has to develop from scratch the corresponding software for collecting, aggregating, and referring to context. The latter reason exists partly because most of existing context-aware applications are built as monolithic stand-alone systems. A shared conceptual model is missing, and there is little reuse of software components or generalization of designs. Software engineering efforts are under pressure to facilitate the development of context-aware systems. There are already groups proposing solutions to this problem from different perspectives. For example, Dey [15] *et.al.* developed a context toolkit for modular design of context sensors; Hong [21] *et.al.* suggested a service infrastructure approach; and Winograd [48] proposed a blackboard based architecture.

Developing context-aware applications for ad hoc mobile environments is particularly hard because of the additional complexity introduced by the dynamic and volatile nature of ad hoc networks. We view middleware as one of the most effective ways to facilitate the development process. The question is, to what extent can a middleware help? To answer this question, we take a data-centric view and regard context as the set of external data that the designers/programmers decided that would influence the behavior of a corresponding application. This operational view of context is rather narrow. For instance, an application could accumulate observations in its own history and the set of observations can be a context for action too. But this is internal context to a specific application, and is beyond the scope our middleware will consider. The advantage of taking this specific view of context is that it makes clear what we can do to facilitate context-aware programming. We can help the application developers by providing services for discovering the relevant context data available in the environment and deliver the data in a timely fashion. Programming complexity can thus be reduced to a great degree. Figure 1(a) illustrates this idea. The smiley represents a context-aware application. The cloud represents data context. The application tells what it needs by giving its context specification (weather conditions in this case) to the middleware (not shown in the picture). Then the relevant context is provided to it constantly by the middleware service(s). Figure 1(b) is another view of our approach. All the data provided by sensors, both from all the application hosts or from the environments, defines a global virtual context space (GVCS). Each application observes and operates on an application-specific context space, which is a projection of the GVCS. The projections are transparently and continuously maintained by an infrastructure.



(a) (b)

Figure 1: (a) Transparent context maintenance and (b) global view of application specific context

One thing has been omitted in Figure 1. If the infrastructure does not already have the related context sensor for what is specified by the application, application developers need to build the corresponding sensor and plug it into the supporting infrastructure. The requirement on the middleware is to make this process simple, and support hot-swapping.

The main objective of this thesis is to propose and develop a middleware for context discovery and delivery in ad hoc mobile environments. Besides delivering the middleware, the key contributions of this work will be to identify and investigate fundamental design and research issues involved in constructing a middleware that accommodates the dynamic and volatile nature of ad hoc networks. The middleware will have a set of APIs for specifying and accessing the network-wide context, and will contain services that cover distributed context discovery, delivery and maintenance of the needed context for each client on top of ad hoc networks. Scalability, efficiency, and resilience to unannounced disconnections and frequent network topology changes are the key concerns for devising the protocols for the discovery, delivery, and maintenance of contextual information. The middleware will also contain a customizable neighbor monitor service, and a set of toolkits for building adaptive context monitors and for context monitor management. Our approach and middleware will be continuously evaluated by developing and testing a set of sample applications, including a cooperative vehicle traffic safety system.

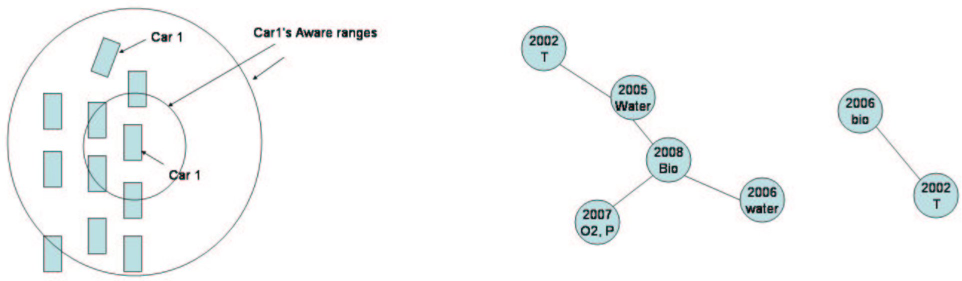
In the next section we will further motivate this work through examples of application scenarios and give a more precise problem definition. In section 3 we present key observations about software design efforts for ad hoc mobile environments and discuss some of the technical challenges we face in achieving our goal. In section 4, we present our approach and preliminary design of the middleware. That is followed by a summary of related work and a short summary of this proposal, in sections 5 and 6, respectively.

2 Problem Definition and Motivations

The dominant theme of this proposal is to develop a middleware to support context-aware computing in ad hoc mobile environments. It is reasonable for one to ask why context-awareness in ad hoc mobile environment is interesting in the first place? Some of the application scenarios we present next address specifically this point.

2.1 Application Scenarios

Figure 2 shows two motivating examples for the sharing of contextual information in ad hoc mobile systems.



(a) (b)

Figure 2: Examples of cooperative traffic safety (a) and incremental planet exploration (b)

Figure 2(a) is an example of cooperative traffic safety system in which vehicles can assess the safety of their own state by incorporating knowledge of other vehicles' location, velocity and acceleration information. To each vehicle's safety-aware applications, the motion profile information of itself and other vehicles around are the context for making the safety assessment. Obviously this sharing of context is indispensable for such applications to work. Figure 2(b) is a hypothetical example of planet exploration by robots. Assume sometime this year, NASA sends two robots equipped with temperature sensors to Mars. In 2005, they decide to send a robot with water probes to team work with the temperature robots, so they can do a correlation study of water and temperature distribution on Mars. And in subsequent years, they send new water-probing robots, Oxygen and atmosphere pressure sensing robots, life-probing robots to team work with previous robots to carry out new studies and explorations. Note that such a distributed robot team system is more reliable than a monolithic robot with all the sensors needed for exploring an unknown planet where anything could happen. It is also less expensive if later robots can reuse resources already in place. Further, this scenario admits incremental deployment of a system, new subsystems can be added subsequently when new technologies are available, when the exploration suggests new directions.

Note that the data exchange/sharing between hosts can also be achieved through a fixed supporting infrastructure. But a fixed-infrastructure approach for the above two scenarios is unreasonable. The ad hoc network based approach is more natural.

2.2 Problem Definition

Our operational view of context serves as the basis of our middleware approach. Context in this thesis is defined as:

> *the set of external data that designers/programmers decided that would influence the behavior of a corresponding application.*

Hence the main task of the middleware is to

> *provide toolkits and services to facilitate the discovery and delivery of context data in an efficient manner.*

More specifically, we aim to support the discovery and delivery the context data that may exist beyond immediate network neighborhood in the ad hoc mobile environment. Figure 3(a) shows the world view we study. Smiley faces represent mobile units. They form ad hoc mobile networks. The communication links are represented by edges.

Mobile units may or may not carry their own sensors, which are represented by hexagons in the figure. The small solid circles represent sensors in the environment. We assume all environmental sensors are wireless. When the mobile units come close to any environmental sensor, they may pick up their signals. Note that although we use the same icon to represent all sensors on the mobile units, they can actually be different kinds of sensors.

We assume that all our applications and related software components are running on the mobile units. From the software component's point of view, the environmental sensors do not exist. What exists are the signals in the environment. Virtual sensors on the mobile units pick up these signals, as shown in Figure 3(b). In a way, Figure 3(b) is an abstract model of the kind of systems we are aiming to support. Henceforth we will no longer talk about environment sensors, and solely focus on this reduced model.

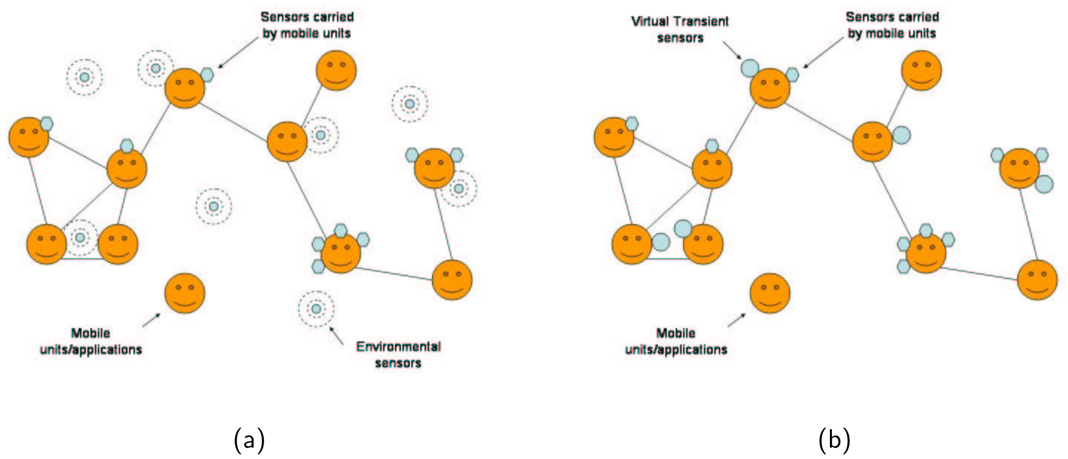


Figure 3: The abstract model

3 Key Observations and Research Issues

Our preliminary investigation into the problem of middleware support for context-aware computing in ad hoc mobile environments uncovered many research issues that must be tackled along with the middleware design and development. Next we highlight some of the issues relating to the way we plan to design the middleware:

1. The nature of ad hoc mobility and wireless communication
2. The relative system dynamics between the applications, the network, and the sensors.
3. The balance between application-awareness and application-transparency.
4. The use of spatial, temporal constraints, and mobility information to improve efficiency.

- A sound software development process requires first to understand the nature of environments the software is intended for, and its impact on software design. For instance, in a highly volatile network, one should avoid using any synchronous and connection-oriented style communication. Otherwise, the resulting middleware can be either very fragile, or ends up making assumptions that are not reflecting the intended reality. Since our main goal is to facilitate context information discovery and delivery in ad hoc mobile networks, we are specifically interested in the following two questions:
  - What is the information capacity of ad hoc mobile networks?
  - How do different levels of mobility and capacity impact strategies for information discovery and delivery?

  Regarding the first question, recent theoretical results have shown that the capacity of ad hoc networks is very limited because all hosts have to serve as routers for others, and ad hoc networks are only scalable for relatively localized communication. Yet, much work needs to be done to understand how mobility impacts capacity, even though we empirically know that it will further reduce it. Regarding the second question, previous studies in ad hoc routing have shown that the efficiency of a routing protocol depends on the mobility model of the underlying system. A protocol that performs well in one scenario may perform badly in another. So far we have not found any related work addressing the issue of how mobility impacts the performance of distributed discovery protocols, such as breadth-first-search. While answers to these two questions are key to both efficiency and reliability of the software, an extensive study is beyond the scope of this proposal. Instead, we will concentrate on the following subproblems:

  - How to define the mobility level of an ad hoc mobile system?
  - What is the impact of mobility on the performance of traditional search algorithms such as breadth-first-search?
  - How does the performance of distributed protocols degrade when the search depth increases? (More network resources are used when search depth increases. That can have a negative impact on performance.)

  Knowledge gained from studying the questions above will further our understanding of ad hoc mobility, will advise system designers, including middleware and application developers, on how to accommodate the nature of ad hoc mobile environments.

- Because a middleware acts as a medium between different components in a complex system, the relative dynamics of system components will affect the way the middleware is designed. A middleware tailored to accommodate the relative dynamics of the system components usually exhibits higher efficiency. An example may help us see why this is the case. Figure 4 is a simple producer-consumer example. As shown in the figure, the sensor samples the data with a frequency of 10 readings per second, while the application that needs to use the corresponding sensor data can only process it at the rate of 1 reading per second, or only need 1 reading per second. A naive context delivery protocol might deliver the context event every time a new datum is produced by the sensor. This will unnecessarily waste network resources. A smarter context management system may allow clients to define their needs more precisely and adjust the delivery strategy accordingly. This is a generic problem for any producer-consumer type system. It just becomes a more difficult issue in

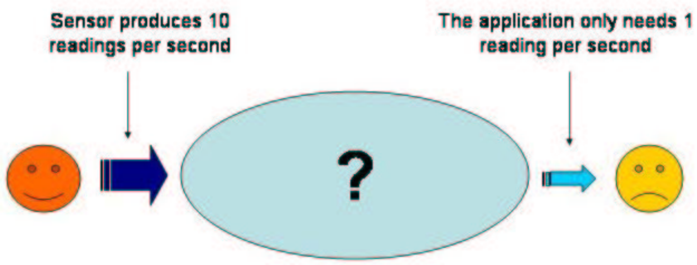


Figure 4: To deliver or not to deliver?

  the ad hoc mobile environment, where bandwidth is a more precious resource. The consideration of efficiency and dynamics leads us to address the following questions:

  - How to characterize the dynamics of the application, network, sensors trio in a simple way?
  - How should a middleware accommodate different relative dynamics of such a trio system?

  Good characterization suggests good solution strategies. We believe that once we can characterize the system in a clear way, we will see what interaction and communication protocols are suitable for each scenario.

- Furthermore, we have learned from past research in mobile computing that application awareness is essential to cope with the diversity of applications and unpredictability of environments, while application transparency can reduce some complexities that applications face. We seek to strike a delicate balance between application-awareness and application-transparency in our middleware effort. Note that here the awareness only refers to the awareness of mobile network conditions. The questions are:

  - What conditions are of interest to most context-aware applications?
  - How to detect changes and notify clients in an efficient and prompt manner?

  By balancing application-awareness and application-transparency, the middleware will both reduce the complexity for programmers while keeping applications informed of critical information that can be used for transition to a more efficient state.

- Finally, we see that in ad hoc mobile environment, relevant context information is usually spatially and temporally close to the application host. This observation motivates and justifies the use of spatial and temporal constraints to limit the scope and increase efficiency of information discovery and dissemination. The questions are:

- How to use these constraints to increase efficiency while avoiding adversely affecting the semantics of the application?
- Who should be the manager of these constraints? Individual applications or the middleware service?

To accommodate application diversity, we need to let applications to define their particular constraints. But the middleware must have some control over them. For instance, when a network is relatively volatile or in a state of congestion, a request for a wide range discovery should be postponed or refused by the middleware service. Again, a delicate balance is needed for the system to work reliably and predictably. Furthermore, we see that the host motion profile (such as location, velocity information) can be used to anticipate disconnections. By being able to anticipate disconnection we may be able to increase the efficiency of information discovery and delivery by proper routing of messages. Again, the question is what are the best ways to do it in different scenarios?

4 Approach

While many of the questions in the previous section remain to be answered, our current understanding of the problem and our goal for a flexible and modular middleware suggest a design of the middleware along the lines shown in Figure 5. This middleware framework will also serve as a test-bed for many of the ideas that will arise from our investigation into research issues discussed earlier.

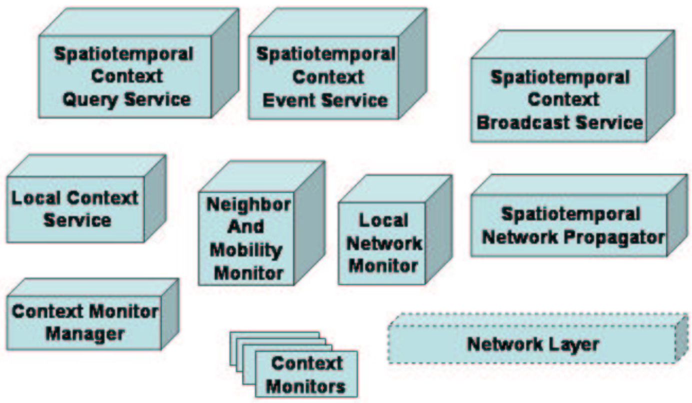


Figure 5: Overview of the middleware design

The middleware contains the following components:

- A set of spatiotemporal context services catering to different characteristics of the application, network, and sensor behaviors. They are:
  - A **Spatiotemporal Context Event Service (SCES)**: it provides context event registration and delivery subject to a spatial (both physical space and network space) constraint relative to the client. This service also features two temporal options. One is event registration time interval, the other is the preferred rate for the data context. The first temporal option enables system self-healing (self cleaning) in case of unexpected disconnection. The second provides a way for system optimization via proper scheduling and sensor adaptation. Further, unlike traditional event services which select their event server by names (IP address , object name, etc.), this new service identifies the event provider(s) by their spatial properties. The set of providers changes as hosts move. This provider set is dynamically and transparently maintained by the service. This event service is expected to be useful for applications running on

ad hoc networks where topology changes are relatively slow with respect the dynamics of sensors, and where applications have persistent need for sensor data.

  - A **Spatiotemporal Context Query Service (SCQS)**: it provides one-time context state (value) query service within a spatial bound. The query will persist for a specific amount of time. Similar to SCES, the information providers are dynamically determined according to the constraints given by clients. This service is expected to be useful for applications that only sporadically need context data from sensors.
  - A **Spatiotemporal Context Broadcast Service (SCBS)**: it provides services to push context to a space (area) specified by clients, and these events will persist within a time bound. This time bound, acting as a physical time-to-live (PTTL), is expected to eliminate the need for the root to re-broadcast sensor data when new arrivals appear within the spatial bounds. Note that in traditional communication systems, TTL is logical, usually represented by a maximum hop count threshold. Here we use a physical TTL (Similar effect of traditional logical TTL is encapsulated in our logical spatial constraints in stead). This broadcast service is expected to be useful for time-critical systems such as cooperative vehicle safety monitors, where the initialization latency of query or event registration is unbearable (even half second advantage in such systems is expected to save many lives).

- A **Spatiotemporal Network Propagator**: this is a spatial and temporal transportation layer used by the above spatiotemporal context services. Besides managing the propagation of data throughout a specified space, it also contains a temporal data manager that removes "stale" data, queries, and event registrations.
- A **Local Context Service**: this service provides local context information. It aggregates context data from local context monitors and serves as an interface for accessing them.
- A **Neighbor and Mobility Monitoring Service**: this service provides neighbor arrival and departure information to its client. It also provides a mobility indicator of the ad hoc mobile network. It is used by the spatiotemporal propagator and can be independently used by other clients too.
- An **Ad Hoc Network Monitor Service**: this service monitors local network traffic status by monitoring the latency in neighbor message deliveries. It can be used by both middleware and applications for adaptation.
- A set of **toolkits** for building and managing adaptive context monitors.

This middleware framework features flexibility and extensibility through the separation of concerns between context request/presentation, context discovery/delivery, and context sensing. This separation of concern makes it easier to incorporate new protocols, change interfaces etc., when necessary.

As real-life evaluation of the middleware requires large-scale deployment of ad hoc mobile systems in various mobility scenarios, this is unlikely to be available during this dissertation effort. So we seek to evaluate the middleware and corresponding distributed protocols via simulation. Currently we are in the process of identifying appropriate simulation tools.

We expect the implementation and evaluation of the middleware will, in turn, raise new questions and provide new insights leading to the design refinements (or may even demand a total redesign).

5 Related Work

While there have been some work dedicated to supporting context-aware computing from various perspectives, little has been done in the community to study the implications of mobility on the design of supporting software, especially in the context of ad hoc mobility. Next we turn our attention to related work in context-aware computing, mobile computing and ad hoc networking.

5.1 Supporting Context-Aware Computing

Schilit, *et al.* [41, 42] proposed a system architecture to support location-based context-aware computing. The architecture is composed of three types of dynamic environment servers: user agents, device agents, and active maps. Each user agent encapsulates the information about a user; each device agent monitors and managers a device; and the active map stores public information about a geographic region and serves as a location-oriented directory service. The active map service allows queries and subscriptions for six types of questions regarding the location of an object [42]. One drawback of the region-based active map service, as pointed out by Schilit himself, is that clients need to manage the handoff between regions themselves. In our view, the concept of fixed regional server is a good abstraction mostly for nomadic mobile computing.

Dey, *et al.* [15], proposed a conceptual framework to support the ability to collect and transform contextual information, with a focus on design reuse. The framework contains five categories of components: context widgets, interpreters, aggregators, services and discoverers. Widgets encapsulate the sensor details away from applications.

Interpreters are used to convert context from one representation to another. Aggregators gather logically related information and make it available in a single component for applications. The context services are responsible for controlling or changing state information in the environment using actuators. Discoverers maintain a registry of widgets, interpreters, aggregators and services for applications to lookup.

Hong, *et al.* [21], suggested a service infrastructure approach designed to simplify the task of creating and maintaining context-aware systems. It is designed to have four different services: a context event service, a context query service, a sensor management service and an automatic path creation service. The authors argue for the need to have a query specification language and for locating context via automatic path creation.

Castro, *et al.* [9], are working on a service infrastructure for sensor data fusion with a focus on the creation of new fusion services which extract and infer context information from sensor data using evidential reasoning techniques, and on optimizing the implementation in terms of performance goals and resource constraints.

Welling, *et al.* [47], proposed a framework for building environment-aware applications. Their architecture is based on an event delivery mechanism that separates event detection from event delivery. They emphasize the architectural flexibility and extensibility by this separation of concerns.

Mandato, *et al.* [30], developed a modular mobile internet portal enhanced with context-aware features. It emphasizes the automatic configuration of some services according to the user's context. It contains a module supporting mobile ad hoc services based on the Jini [17] technology.

Dix, *et al.* [16, 14, 11], presented a framework to support the design of interactive mobile systems based on an understanding of location and space within these systems.

Our approach to supporting mobile context-aware computing complements the efforts above, with a focus on ad hoc mobile environments. We expand the scope of the application context beyond its immediate neighborhood and provide a set of spatiotemporal context services to simplify the development of advanced context-aware applications. Realizing the diversity of ad hoc mobile environments and applications, we seek to provide a set of solutions catering to the different application needs.

5.2 Mobile Computing

An emerging theme in mobile computing is that application-aware adaptation is essential for coping with wide variations in network conditions and limited local resource availability brought by mobility [34, 35]. The need for network-awareness and services that expose mobility to the application was identified by many others [23, 13, 22, 43]. Katz, for instance, noted the need for adaptation of mobile systems to a variety of network environments [23]. Davies et al. emphasized the need for protocols to provide feedback about the network to applications in a vertically integrated application environment [13]. Joseph et al. pointed out that applications operating in the harsh conditions of a mobile environment must often be aware of and adapt to those conditions to excel [22].

5.3 Ad Hoc Networking

Discovery and sending information beyond the immediate neighborhood in ad hoc networks inevitably needs to deal with message routing. Much work has been done to develop new routing protocols for the mobile ad hoc environment. Protocols such as Dynamic Source Routing(DSR) [6], Ad hoc On-demand Distance Vector(AODV) [38], Temporally Ordered Routing Algorithm (TORA) [37] are among the unicast routing protocols currently being evaluated by the IETF MANET working group. A common characteristic of these protocols is their on-demand construction and maintenance of routes, designed to reduce the cost of stale information. More recently, location-aware routing protocols have been proposed, e.g., Location-Aided Routing (LAR) [24], Scalable Location Updated-based Routing [49]and Grid [28]. By using location information about the mobile nodes, routing paths can be constructed more efficiently. Going beyond the assumption of a connected network, Epidemic routing [44] and disconnected transitive communication [10] use pair-wise message exchanges among mobile hosts to move a message through the network, enabling communication among hosts which may never co-exist in a connected network. Similar to the efforts to provide new unicast mechanisms, new multicast protocols [18, 12, 39, 27] also are being proposed for ad hoc networks. Yet, a very recent study [36] has shown that simple flooding performs comparatively better than two representative multicast protocols, ODMRP [27] and MAODV [39]. We see that much deeper understanding of ad hoc mobility is required to devise better protocols. Worth special attention are several novel multicast models such as Geocast [33, 25, 26], and content-based multicast [50]. They are being proposed as variants of the IP-based multicast. In Geocast, the multicast message is sent to hosts residing in a specific geographic area. In content-based multicast, the receiver set for information changes dynamically based on the content of the information being sent as well as the mobility of the receivers themselves. In addition, the landmark work by Gupta, Kumar and others [20, 29] has shown that ad hoc networks are inherently unscalable for general communications, suggesting they are only likely to be used for relatively localized communications. We believe their work will dramatically change design strategies and methodologies for investigating problems in ad hoc networks.

Our work is inspired by many of the ideas presented here and elsewhere. We integrate some of them in our software engineering approach to address the issue of supporting context-awareness on the backdrop of ad hoc networks.

6 Conclusion

The dynamic nature of mobile networks presents us with an unprecedented context-rich environment that computing devices and applications must face. At the same time, the richness of this context brings to life many research opportunities and challenges. The software engineering community is under pressure to facilitate the development of context-aware applications for the mobile environment. While much effort has been dedicated to supporting context-aware computing from various perspectives, we are unaware of specific system designs for supporting context-awareness in ad hoc mobile environments, especially when the context needed goes beyond the immediate neighborhood. Our work is among the first to study the implications of ad hoc mobility to the design of supporting software and to deliver a middleware tailored to the nature of ad hoc mobility. We expect our investigation to lead to new insights into the design principles for ad hoc mobile systems.

References

[1] G. Abowd, C. Atkeson, J. Hong, S. Long, R. Kooper, and M. Pinkerton. Cyberguide: A mobile context-aware tour guide. *ACM Wireless Networks*, 3:421–433, 1997.

[2] Varol Akman and Mehmet Surav. Steps toward formalizing context. *AI Magazine*, 17(3):55–72, 1996.

[3] Abhaya Asthana, Mark Cravatts, and Paul Krzyzanoowski. An indoor wireless system for personalized shopping assistance. In *Proceedings of IEEE Workshop on Mobile Computing Systems and Applications*, pages –84, Santa Cruz, California, December 1994.

[4] Frazer Bennett, Tristan Richardson, and Andy Harter. Teleporting - making applicatoins mobile. In *Proceedings of IEEE Workshop on Mobile Computing Systems and Applications*, pages 82–84, Santa Cruz, California, December 1994.

[5] Paolo Bouquet and Luciano Serafini. Two formalizations of context: A comparison. In *CONTEXT*, pages 87–101, 2001.

[6] J. Broch, D.B. Johnson, and D.A. Maltz. The dynamic source routing protocol for mobile ad hoc networks. Internet Draft, March 1998. IETF Mobile Ad Hoc Networking Working Group.

[7] Sasa Buvac. Propositional logic of context. In Richard Fikes and Wendy Lehnert, editors, *Proceedings of the Eleventh National Conference on Artificial Intelligence*, pages 412–419, Menlo Park, CA, 1993. AAAI Press.

[8] Sasa Buvac. Quantificational logic of context. In *AAAI/IAAI, Vol. 1*, pages 600–606, 1996.

[9] Paul Castro and Richard Muntz. Managing context data for smart spaces. *IEEE Personal Communication*, pages 44–46, October 2000.

[10] X. Chen and A.L. Murphy. Enabling disconnected transitive communication in mobile ad hoc networks. In *Proceedings of the Workshop on Principles of Mobile Computing*, pages 21–27, August 2001.

[11] K. Cheverst, N. Davies, K. Mitchell, A. Friday, and C. Efstratiou. Experiences of developing and deploying a context-aware tourist guide: The GUIDE project. In *Proceedings of MobiCom*, pages 20–31. ACM Press, 2000.

[12] C. Chiang, M. Gerla, and L. Zhang. Adaptive shared tree multicast in mobile wireless networks, 1998.

[13] N. Davies, G. Blair, K. Cheverst, and A. Fridy. Supporting Adaptive Services in a Heterogeneous Mobile Environment. In *Proc of the Workshop on Mobile Computing Systems and Applications*, pages 153–157, Dec 1994.

[14] Nigel Davies, Keith Cheverst, Keith Mitchell, and Adrian Friday. Caches in the air: Disseminating tourist information in the guide system. In *Proceedings of the 3rd International Symposium on Wearable Computers (ISEC'99)*, pages 21–28, New Orleans, Louisiana, October 1999. IEEE Computer Society Press.

[15] Anind K. Dey, Daniel Salber, and Gregory D. Abowd. A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. *Human Computer Interaction Journal*, 16, 2001. Special issue.

[16] Alan Dix, Tom Rodden, Nigel Davies, Jonathan Trevor, Adrian Friday, and Kevin Palfreyman. Exploiting space and location as a design framework for interactive mobile systems. *ACM Transactions on Comupter-Human Interaction*, 7(3):285–321, September 2000.

[17] K. Edwards. *Core JINI*. Prentice Hall, 1999.

[18] M. Gerla, C. Chiang, and L. Zhang. Tree multicast strategies in mobile, multihop wireless networks, 1998.

[19] F. Giunchiglia, C. Ghidini, and M. Semantics. Local models semantics,or contextual reasoning = locality + compatibility. In *Proceedings of the Sixth International Conference on Principles of Knowledge Representation and Reasoning (KR'98)*, pages 282–289, Morgan Kaufmann, 1998.

[20] P. Gupta and P.R. Kumar. The capacity of wireless networks. *IEEE Transactions on Information Theory*, 46(2):388–404, 2000.

[21] J. Hong and J. Landay. An infrastructure approach to context-aware computing. *Human Computer Interaction*, 16, 2001.

[22] A.D. Joseph, J.A. Tauber, and M.F. Kaashoek. Mobile computing with the rover toolkit. *IEEE Transactions on Computers*, 46(3):337–352, March 1997.

[23] R. H. Katz. Adaptation and Mobility in Wireless Information Systems. *IEEE Personal Communications*, 1(1):6–17, 1994.

[24] Y. Ko and N. Vaidya. Location-aided routing (LAR) in mobile ad hoc networks. In *Proceedings of MobiCom*, pages 66–75, 1998.

[25] Y. Ko and N. H. Vaidya. Geocasting in Mobile Ad Hoc Networks: Location-based multicast algorithm. In *IEEE WMCSA*, New Orleans, LA, Feb 1999.

[26] Y. Ko and N. H. Vaidya. GeoTORA: A Protocol for Geocasting in Mobile Ad Hoc Networks. In *8th International Conference on Network Protocols (ICNP)*, Osaka, Nov 2000.

[27] S. Lee, W. Su, and M. Gerla. Demand multicast routing protocol in multihop wireless mobile networks, 2000.

[28] J. Li, J. Jannotti, D. Decouto, D. Karger, and R. Morris. A scalable location service for geographic ad-hoc routing. In *Proceedings of the Sixth Annual ACM/IEEE International Conference on Mobile Computing and Networking*, pages 120–130, August 2000.

[29] Jinyang Li, Charles Blake, Douglas S. J. De Couto, Hu Imm Lee, and Robert Morris. Capacity of ad hoc wireless networks. In *Proceedings of the 7th ACM International Conference on Mobile Computing and Networking*, pages 61–69, Rome, Italy, July 2001.

[30] D. Mandato, E. Kovacs, F. Hohl, and H. Amir-Alikhani. CAMP: A Context-Aware Mobile Portal. *IEEE Communications Magzine*, pages 90–97, Jan 2002.

[31] N. Marmasse and C. Schmandt. Location-aware information delivery with comMotion. In *Proceedings of the Second International Symposium on Handheld and Ubiquitous Computing*, pages 157–171, September 2000.

[32] John McCarthy and Saša Buvač. Formalizing Context (Expanded Notes). In A. Aliseda, R.J. van Glabbeek, and D. Westerståhl, editors, *Computing Natural Language*, volume 81 of *CSLI Lecture Notes*, pages 13–50. Center for the Study of Language and Information, Stanford University, 1998.

[33] Julio C. Navas and Tomasz Imielinski. Geocast-geographic addressing and routing. In *ACM/IEEE MobiCom*, Budapest, Hungary, Sept 1997.

[34] B. Noble. System support for mobile, adaptive applications. *IEEE Personal Communications*, 7(1):44–49, February 2000.

[35] Brian D. Noble, Morgan Price, and Mahadev Satyanarayanan. A programming interface for application-aware adaptation in mobile computing. In *Proceedings of the Second USENIX Symposium on Mobile and Location-Independent Computing*, Ann Arbor, Michigan, April 1995.

[36] Katia Obraczka, Gene Tsudik, and Kumar Viswanath. Pushing the limits of multicast in ad hoc networks. International Conference on Distributed Computing Systems(ICDCS), 2001.

[37] V. Park. and M.S. Corson. Temporally-ordered routing algorithm (TORA) version 1: functional specification. Internet Draft, August 1998. IETF Mobile Ad Hoc Networking Working Group.

[38] C. Perkins and E. Royer. Ad hoc on-demand distance vector routing. In *Proceedings of the Second IEEE Workshop on Mobile Computing Systems and Applications*, pages 90–100, February 1999.

[39] Elizabeth M. Royer and Charles E. Perkins. Multicast operation of the ad-hoc on-demand distance vector routing protocol. In *Mobile Computing and Networking*, pages 207–218, 1999.

[40] N. Ryan, J. Pascoe, and D. Morse. Fieldnote: A handhelod information system for the field. In *First International Workshop on TeloGeoProcessing*, 1999.

[41] B. Schilit, N. Adams, and R. Want. Context-aware computing applications. In *IEEE Workshop on Mobile Computing Systems and Applications*, 1994.

[42] William Noah Schilit. *A System Architecture for Context-Aware Mobile Computing*. PhD thesis, Columbia University, New York, 1995.

[43] D. Terry, M. Theimer, K. Petersen, A. Demers, M. Spreitzer, and C. Hauser. Managing update conflicts in bayou: a weakly connected replicated storage system. In *Proceedings of the 15th ACM Symposium on Operating System Principles*, pages 173–183, Dec 1995.

[44] A. Vahdat and D. Becker. Epidemic routing for partially-connected ad hoc networks. Technical Report CS-2000-06, Duke University, 2000.

[45] R. Want et al. An overview of the PARCTab ubiquitous computing environment. *IEEE Personal Communications*, 2(6):28–33, December 1995.

[46] Roy Want, Andy Hopper, Veronica Falcao, and Jonathan Gibbons. The active badge localtion system. *ACM Transactions on Information Systems*, 10(1):91–102, January 1992.

[47] Girish Welling and B. R. Badrinath. A framework for environment aware mobile applications. In *International Conference on Distributed Computing Systems*, 1997.

[48] Terry Winograd. Architecture for context. *Human Computer Interaction Journal*, 16, 2001. Special issue.

[49] S.-C. Woo and S. Singh. Scalable routing in ad hoc networks. Technical Report TR00.001, Oregon State University, March 2000.

[50] H. Zhou and S. Singh. Content based multicast(cbm) in ad hoc networks. In *Proceedings of the ACM/IEEE Workshop on Mobile Ad Hoc Networking and Computing (MOBIHOC), Boston, MA, August 2000*, 2000.