Middleware for Mobile Applications Beyond 3G

Kimmo Raatikainen
Nokia Research Center and University of Helsinki

Abstract Context-awareness and adaptability to changes in the execution and communication environment will be the key enablers for future applications that try to realise the "always-on" vision. This paper addresses fundamental research challenges and issues in middleware for mobile computing in systems beyond 3G. The key areas covered include adaptability, wireless communication, distribution of functionality, dynamic end-user systems, proximity, and open standards. In each area we identify crucial research issues. It should be noted that the division is not orthogonal: The same or similar issues are met in different key areas. We also outline a roadmap to construct a middleware solution to support future applications that will enable seamless service provisioning in heterogeneous, dynamically varying computing and communication environments.

Keywords: CORBA, Java, Internet protocols, adaptation, wireless communications, reconfiguration, interoperatibility, open standards

"Any technology distinguishable from magic is insufficiently advanced," Gregory Benford.

1. INTRODUCTION

Context-aware applications will be of fundamental importance in mobile Internet that will be much more than Internet access from mobile devices. In fact, the Internet will be almost invisible since people will use mobile services and their favourite applications.

The transition to the Mobile Internet will be much more demanding than the transition to mobile phones in voice services. The primary reason is heterogeneous demands of various services and applications on the

The original version of this chapter was revised: The copyright line was incorrect. This has been corrected. The Erratum to this chapter is available at DOI: 10.1007/978-0-387-35584-9_19

O. Martikainen et al. (eds.), *Smart Networks*

underlying computing and communications infrastructure. Direct use of existing Internet applications in a mobile environment has usually been unsatisfactory. Web browsing, for example, over GSM Data Service using the standard HTPP/TCP/IP/PPP stack typically can utilize only about one third of the nominal bandwidth. Therefore, services and applications need to take into account the specific characteristics of mobile environments.

In principle, one could rewrite each service and application for Mobile Internet. This, however, would be both cumbersome and expensive hence slowing down development and deployment of services and applications. Therefore, essential generic functionality for mobile applications should be implemented in the infrastructure.

Middleware is a widely used term to denote a set of generic services above the operating system. Although the term is popular, there is no consensus of a definition. Nevertheless, middleware is used to denote a set of generic services above the operating system. A good summary can be found in the IETF RFC2768 [1]. Typical middleware services include directory, trading and brokerage services for discovery, transactions, persistent repositories, and different transparencies such as location transparency and failure transparency. Examples of middleware include Common Object Request Broker Architecture (CORBA) [2], Java 2 Enterprise and Micro Editions (J2EE and J2ME) [3, 4], Distributed Common Object Model (DCOM) [5], and Wireless Application Environment (WAE) [6]. Characteristically, the competing middleware specifications provide many similar but slightly different services. In order to overcome the problems due to different specifications, the Parlay Group [7] has specified a set of UML models and corresponding APIs that can be implemented in CORBA, Java and DCOM environments.

Context-awareness is considered as a fundamental property of future mobile applications. In essence, context-awareness means that the behaviour of an application depends on the current context. The context includes user preferences, device characteristics, properties of connectivity, geospatial location (time and space), state of service (session), usage history. Context-awareness also means that changes essential enough in the context trigger changes in the application behaviour. This kind of functionality is often called as adaptability.

In this paper we examine functional requirements of middleware supporting context-aware applications. We start by briefly discussing future mobile applications, their communication characteristics and their composition. The functional requirements are presented as research challenges in Section 3. The research space is divided into six research areas, each containing a set of key functional requirements that enable seamless service provisioning in heterogeneous, dynamically varying computing and

communication environments. In Section 4, we outline a roadmap, which tries to show a way from the current state-of-practice to the envisaged state-of-future.

2. FUTURE MOBILE APPLICATIONS

2.1 Communication Characteristics

The most significant feature in communication needs of applications in the Mobile Internet Era will be diversity. All kinds of applications will be in use. Their Quality-of-Service requirements (bandwidth, latency, reliability) as well as communication patterns will numerous. Some applications will also adjust their behaviour according to the properties of connectivity. Future mobile terminals will have a few, say 3-6, applications simultaneously active. Some terminals will also be able to use different access technologies either simultaneously or one at a time. Therefore, the most important property of any communication system is its ability to handle mixtures of flows and traffic characteristics in a reasonable way.

When the application interacts with a human end-user, interaction mode is a good classifier of applications. Most of the applications fall into one of the following three categories:

- ***Messaging***. These applications are non real-time. The underlying network can use store-and-forward, store-and-retrieve, or store-and-push mechanisms. The applications are not delay sensitive. The delays can be in the scale of seconds, minutes, or even hours. However, the delivery must usually be error free. The message size varies from a few bytes to several Megabytes. This category calls for efficient use of network resources.
- ***Interactive content retrieval***. These applications are nearly real-time. The users do not tolerate long delays. Some content formats, such as audio and video, are delay sensitive but tolerate losses. In these applications playback buffers can be used to balance delay varition. Some other content formats, text in particular, are not delay sensitive but require error free transmission.
- ***Rich call***. These applications require real-time communication. An example of today is voice call and that of the future is a multiparty video conference. These "calls" will be annotated by drawings, text files, etc.

When machine-to-machine applications, that is applications without human interactions, are also considered, then two main categories of applications can be identified: control and command applications and management applications:

- ***Control and command.*** In these applications short messages (payload from 10 to 10,000 bytes) are typical. The message delivery must be timely, reliable and error free. The delays can seldom be more than one second. The usual interaction pattern is either a simple request-reply or a single notification.
- ***Management.*** In management applications the interaction patterns can be quite complicated involving a dialog of several messages. Management applications require reliable and error free transmission but the time scale of delay bounds is in tens of seconds.

2.2 Application Composition

Figure 1 depicts a highly abstracted vision of how a service application is distributed among various application servers, network elements and terminal or end-user systems. It should be noted that, for simplicity, the figure only shows a single terminal device although multi-party applications will be much more important and challenging than one-party applications such as information browsing. In addition, we must also be ready to cope with end-user systems based on body area networks and home communication systems.

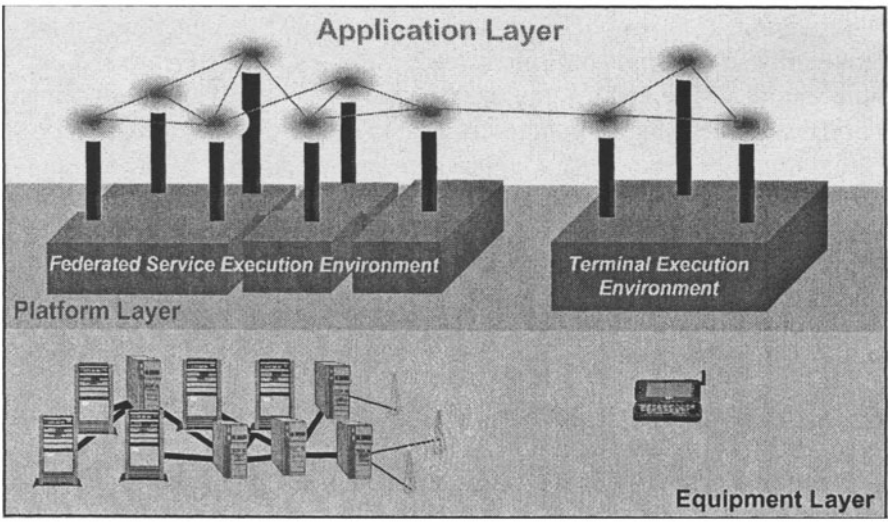

Figure 1. Partitioning and Distribution of Application Logic

The execution environments or the platform layer consist of middleware, operating systems and protocol stacks that should support fast service

development and deployment. The platforms should make it easy to divide the application logic into co-operating parts—someone may call them components, to distribute and configure these components as well as to redistribute and reconfigure them. Additional requirements for future mobile applications include adaptability to changes in the execution and communication capabilities, efficient use of available communication resources, dynamic configuration of end-user systems as well as ultimate robustness, high availability and stringent fault-tolerance.

The requirements for data accessed by these applications are quite similar. The execution environment should provide consistent, efficiently accessible, reliable and highly available information base. This implies a distributed and replicated worldwide "file system" that also supports intelligent synchronization of data after disconnections.

3. RESEARCH CHALLENGES IN MOBILE COMPUTING

In this section we identify the key research challenges in the future software systems enabling seamless service provisioning in heterogeneous, dynamically varying computing and communication environments. The division is not orthogonal; same or similar research topics and issues appear in more than one research challenges. We have divided the research space into the areas of adaptability, wireless communication, distribution of functionality, dynamic end-user systems, proximity, and open standards. Other divisions can be found in [8-16].

3.1 Adaptability

Adaptability is one of the key research areas in nomadic computing. The basic principle of adaptability is simple. When the circumstances change, then the behaviour of an application changes according to the desires of a user—or more precisely according to principles ascribed to her.

Let us take an example: 1) Important messages or their shortened versions (at most 200 characters) should be sent me at any price. 2) Voice messages should be sent me if the cost is 10 cents, at most. 3) Emails are to be sent if the cost does not exceed 5 cents. Attachments may be dropped but the message cannot be shortened. Email traffic is not allowed to disturb reception of important messages or voice messages. 4) Only the reception of important messages can slow down the delivery of voice messages. 5) Only the reception of very important messages can delay the delivery of important messages.

Although the example above concerns only messaging, it is quite long and far from a precise definition. In that sense it is typical and sets an important research topic: How to translate the wishes of users, which are almost always inaccurate, incomplete and sometimes even contradictory, into a set of rules precise enough for processing to be automated with sufficient reliability?

Learning the wishes and desires of a user is a crucial part of adaptability. Fully automated learning may be a utopia. Starting from an empty set of knowledge would take too much time. The learning path would necessarily contain too many malfunctions that upset the user who will be ready to discard the gadget.

Instead, starting from teaching is much more fruitful. In the same way as a child in a family is taught the habits of the family in a matter of years, the personal agents are taught the habits of the user in a few days. In this process the user interface is crucial: How and in which forms are users willing to give feedback? Most probably the feedback system cannot be unified but must be personalized for each user. The common attitude of a layman user—"I give feedback as I will"—must be the starting point of the user interface design. In particular, obtaining positive feedback is problematic. In some cases, missing feedback can be interpreted as a correct selection of action. However, in other cases that interpretation is questionable or even misleading.

Another crucial problem is the size and computational complexity of the knowledge base. Most probably the knowledge needs to be partitioned so that each subset is small enough. The partitioning, however, is not alone sufficient. The models presenting the subsets of the knowledge must be combined in different ways for different purposes. It should be noted that this is not the only application of partitioning in nomadic computing.

Adaptability cannot only be reactive. When the battery dies or the connectivity breaks, many actions are impossible. However, something could have been done beforehand. Therefore, adaptation must also be proactive, which, in turn, requires predictability of the near future. An important question in predictions is to distinguish between the situations in which the user behaviour seems to be predictable and those being unpredictable.

The objective should not be a perfect system since that would take forever. Instead, the goal must be a system that often (or at least sometimes) behaves correctly but almost never behaves incorrectly. Doing nothing is quite often a lesser evil than taking the wrong action.

3.2 Wireless Communication

As a communication channel, air is problematic. In the last ten years the progress in coding has significantly increased the capacity of wired channels. Unfortunately these fruits cannot fully be utilized on wireless channels. The applied coding is always a compromise between information density and redundancy providing robustness against interference. The basic problem with wireless links is instability in the sense that the level of interference varies in time and place, and according to environmental conditions.

It is very often said that the speed of 2 Megabits per second, which is assumed to be available in the 3G, will be fast enough. However, in the history of computing, the spare capacity has never been left unused. In addition, one should notice that the capacity and speed of wired connections has increased much faster. For each magnitude of improvement in wireless communications there has been an improvement of 2-3 magnitudes in wired communications.

The problems of wireless links are not uniform. We have wireless LANs, satellite links, cellular networks, and short-range radio links. Each poses specific problems of its own. Therefore, wireless communications must be regarded as a polymorbid patient.

Each of the access technologies mentioned above differs, at least in some aspect, from the others. Hence, the support system of a nomadic user must be able to support communication links of different kinds. It must enforce the higher layers of communications to adapt to the situation at hand. However, the adaptation of communication is not sufficient, the behaviour of applications also needs to be adapted.

3.3 Distribution of Functionality

Situations, in which a user moves with her end-device and uses information services, are challenging. Moreover, the nomadic user of tomorrow will not appreciate a static binding between her and an access device; not even in the case of multi-mode access devices that can handle several access technologies including wireless LAN, short-range radio, and packet radio. It must be possible to move a service session (or one end-point of a service session) from one device to another.

In these situations the partitioning of applications and the placement of different co-operating parts is a research challenge. The support system of a nomadic user must distribute, in an appropriate way, the parts among the end-user system, network elements and application servers. In addition, when the execution environment changes in an essential and persistent way, it may be beneficial to redistribute the co-operating parts. The redistribution

or relocation as such is technically quite straightforward but not trivial. On the contrary, the set of rules that the detection of essential and persistent changes is based on is a challenging research issue.

Another research issue of fundamental importance in distribution is fault-tolerance. Replication, which is a commonly used method to achieve fault-tolerance in traditional distributed systems, is not sufficient alone. The baseline applications must remain operational, at least in a tolerable manner, even if some services of the underlying execution environment cannot be utilized. Here we have a nice additional requirement for those who research modularisation of application and adaptability.

3.4 Dynamic End-User Systems

The end-user devices of today are primarily integrated units like PDAs, laptops, and mobile phones. However, the situation will change in the future. The successor of the current mobile phone, or at least the successor of its successor, will be quite different. It will not disappear or loose its importance but its role will be very different. This "*FuturePhone*" will be the core of the personal computing and communication system. The *FuturePhone* probes its surroundings looking for suitable peripheral devices such as displays, input devices, processors, fast access memories and access points to communication channels. It dynamically builds up the most appropriate end-user system that can be auto-configured. The set of rules how the appropriateness of different configurations is evaluated is far from trivial. In addition, we must remember that the FuturePhone must be able to remain operational even if it cannot find any suitable peripherals. Moreover, the FuturePhone must be able to establish different kinds of ad-hoc networks that are simultaneously operational.

In the dynamic configuration we have a huge space of research items. On the conceptual level there are research issues related to profiles, various kinds of context also including the social context, roles and trust. On the technical level we must solve the problems related to authentication, authorization, and delegation.

3.5 Proximity

Location was identified already in the NII white paper [12] as one of the key issues in nomadicity. The baseline location-based services are not anymore a research issue. Simple location-based queries and notifications/advertisements are ready to be deployed. Perhaps, the privacy issues still need further studies, in particular the enforcement of regulatory

rules that differ from one country to another. The challenge is to design a flexible policy mechanism that allows fast changes in applicable policies.

Another challenge, not necessarily a research issue, is the way to cope with a plethora of location standards. The Location Interoperability Forum (LIF) [17] was established to an industry-wide initiative for promoting a secure, simple, ubiquitous, and interoperable location services solution to improve technology and maximize business. One of the underlying observations was a need for a global forum to:

- address the complexity and multiplicity of current solutions and market situation;
- define, develop, and promote an interoperable location services solution that is open, simple, and secure; and
- allow user appliances and Internet-based applications to obtain location information from the wireless networks independent of their air interfaces and positioning methods.

A research issue, however, is the prediction of end-users' future locations. The predictions may, for example, be used to schedule an appointment. An important aspect of location predictions is to combine information from different sources—from models of user movements and behavior, calendar, and to-do-lists, for example.

To include time and location into queries may not be straightforward. As an end-user I would like to post a request to "book a table for me and my wife in a nearby restaurant today at 7 pm." When the booking service starts to process my request, it meets several challenges:

- Where I am going to meet my wife today, sometime between now and 7 pm?
- How much we are ready to walk to a suitable restaurant? Do we prefer a moderate restaurant within ½ mile over a better one within 2 miles?
- What kind of restaurant we prefer? My preferences may be different from those of my wife? How to balance between our preferences to meal, to walk for a good meal, etc?

The example above demonstrates that the personal profile indicating personal preferences and matching rules is not trivial. Therefore, research on personal profiles and on matching rules is crucial. Furthermore, the semantics of nearby or rating value (single number) of proximity is of major importance.

3.6 Open Standards

The assumption that the services are available anytime and anywhere calls for open standards. The time when a closed proprietary solution was sufficient is history. The baseline services must be available and operational

everywhere. Therefore, open worldwide standards are necessary. However, this does not mean that closed proprietary solutions are not developed anymore. They are still produced but they must be interoperable with the rest of the world. The proprietary solutions will have an important but specific scope. They will be used as means to customize and differentiate services and products.

4. ROADMAP

When our research agenda and those of the other groups—a summary can be found in [18]—are put together, then the first conclusion must be that there is a long to-do list before ubiquitous mobile information society is real. The research groups devoted to mobile, nomadic, pervasive or ubiquitous computing have taken different paths in order to achieve their goals. However, many groups have adopted the traditional paradigm of constructive computer science: To implement prototype(s) based on the proposed solutions in order to demonstrate the feasibility of the proposal.

Prototype implementation is fun. You can quite easily hire hackers to do the programming. However, we are not quite sure how productive, in a long run, prototype implementations are in a research process. Therefore, we have downgraded the role of prototyping from the first class to the economy class. In essence, prototyping is only a tool in verification and validation but not a driving force.

4.1 All IP Networks and Middleware

The current trend in developing forthcoming telecommunication networks is to utilize Internet protocols. An immediate implication is that IP is the layer 3 protocol and that the addresses are IP addresses. However, this is not sufficient. Other solutions—both above and below the IP protocol—are also needed to meet the requirements for next generations of telecommunication networks. Issues under study in the Internet community and in various standardization bodies, forums and consortia of telecommunications include mobility, Quality-of-Service, security, management of networks and services, discovery, ad-hoc networking, dynamic configuration, and geospatial location.

Another significant trend is the requirement of ever-faster service development and deployment. An immediate implication has been the introduction of various service/application frameworks/platforms. Middleware is a widely used term to denote a set of generic services above the operating system. Although the term is popular, there is no consensus of

a definition [1]. However, typical middleware services include directory, trading and brokerage services for discovery, transactions, persistent repositories, different transparencies such as location transparency and failure transparency.

The benefits of middleware software, such as CORBA, J2EE, J2ME , and DCOM, are obvious; see e.g. [19-21]. The most significant advantage—when compared to a pure IP-based socket programming approach—is in the improved programming model. The middleware solutions are usually based on object-oriented programming and method invocations. The invocations are based on strongly typed interfaces that provide both compile and run time error checking. They also hide many implementation details. Therefore, middleware-based application development is much faster than the Internet based one.

Although the middleware solutions provide superior programming environments over the Internet solution, they also pose serious shortcomings. The fundamental problem of the current middleware specifications is that they only take advantage of a narrow subset of useful Internet protocols. The current middleware specifications were born in a time when the Internet protocols were a synonym of the TCP/IP transport. Later they have developed solutions of their own for Quality-of-Service, directory, discovery, and so on, independently from each other and from the IETF specifications. The fundamental research challenge is the question of how the developments in Internet protocols and in different middleware solutions could be harmonized.

By harmonization we mean two things. Firstly, we need to solve the problem of incorporating evolving Internet solutions of Quality-of-Service, mobility, discovery, and security into the existing middleware specifications without breaking those specifications. Secondly, we need to find solutions to how different middleware solutions can become interoperable in the sense that components of an application can be executed on different middleware platforms.

Figure 2 outlines a framework for our roadmap. It decomposes the abstract notion of execution environment in Figure 1 into a layered architecture. One should notice that the operating system is not shown in the figure because it would introduce another dimension. The roadmap is the way how we assign the research issues identified in our research agenda into various functional blocks in Figure 2 and further into various standardization activities in relevant forums including IETF [22], W3C [23], 3GPP [24], OMG [25], Parlay [26].

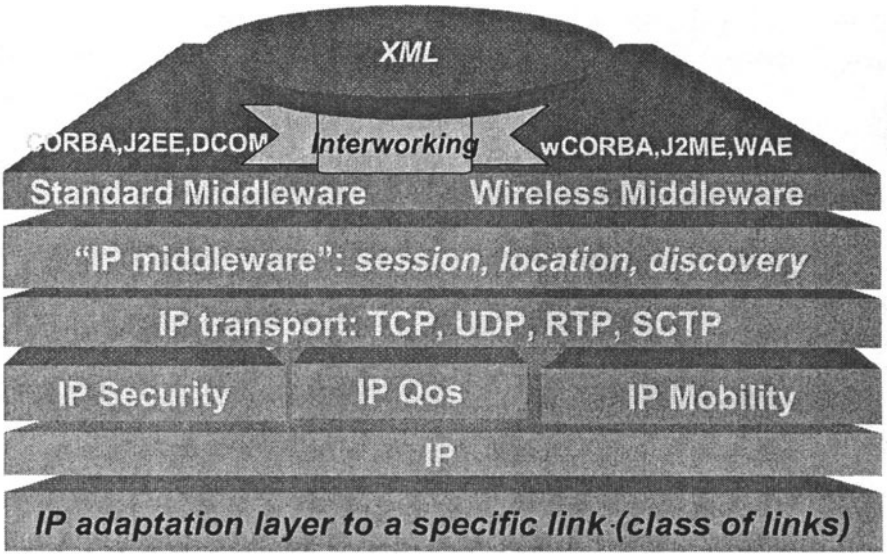

Figure 2. Layered Architecture of Middleware and Internet protocols

We need to work on details in various working groups of standardization bodies. We also need to have a vision where to go. Here we need both practitioners—ones who know all the details—and visionaries—ones who know almost nothing in details but almost anything in general.

4.2 Interoperability

The increasing diversity of devices—terminals, network elements, and application servers—imply that different middleware solutions will be in use. It is highly improbable that there will be, in a near future, a single dominant middleware platform which would be good enough for different devices and purposes. This heterogeneity requires interoperability on two levels: between middleware platforms and between parts of an application running on different middleware platforms.

The interoperability between different platforms is quite mature. In particular, the OMG has been the leading forum in specifying interoperability bridges between CORBA and other middleware platforms. In contrast, the interoperability between parts of an application running on different middleware platforms is still immature. There are practically no tools available to support this kind of interoperability. The burden of interoperability is totally left to application developers.

The OMG, however, has started a comprehensive development of a new architecture denoted as Model Driven Architecture (MDA) [27, 28]. The objective is to interrelate IDL specifications, UML modeling [29], Meta-Object Facility [30], and XML Metadata Interchange (XMI) [31]. The forthcoming MDA might provide a useful starting point for tools supporting interoperability between parts of an application running on different middleware platforms. A sketch of an interoperability framework based on UML is outlined in Figure 3.

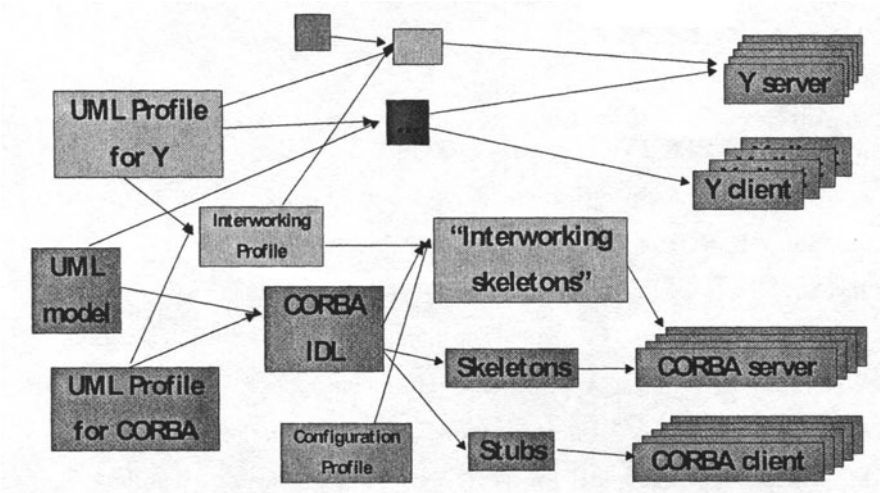

Figure 3. A UML-based Interoperability Framework

5. CONCLUSIONS

Almost ten years ago Mark Weiser [8] identified the target in future computing to be invisible computing, to hide computers and communications. The research agenda of Weiser [9] was later essential extended by Alan Demers [10] and by Leonard Kleinrock [11, 12]. Today we mainly use in our research agenda the terms and concepts originating in the pioneering work in Xerox PARC [8-10]. This does, however, not mean that the research has done almost any progress in the last ten years. On the contrary, accumulated shared knowledge over the problem space as well ass pros and cons of different proposed solutions have been utilized. We have been able to progress both in details and in two or three party interactions.

The worldwide research community has provided some insight into ubiquitous mobile information society. There is, however, a huge lot to be done. We believe that we can decompose the research and problem space

into manageable R&D initiatives. Long-term interactions between identified subsystems are, perhaps or most probably, manageable.

Our primary concern is the overall picture. It is easy to draw a few PowerPoint slides. However, the road to a product or to a prototype is long and rocky. The divide-and-conquer principle is a necessity. However, very many can divide a whole into pieces, almost anybody can manage a well-defined piece but to conquer the whole is the challenge.

REFERENCES

[1] B. Aitken et al., "Network Policy and Services: A Report of a Workshop on Middleware," RFC 2768, February 2000.

[2] OMG, "CORBA 5. Specification," http://www.omg.org/.

[3] JavaSoft, "Java 2 Enterprise Edition," http://java.sun.com/j2ee/.

[4] JavaSoft, "Java 2 Micro Edition", http://java.sun.com/j2me/.

[5] Eddon, G. and Eddon, H. "Inside Distributed Com," Microsoft Press, 1998

[6] WAP Forum, "Wireless Application Environment," WAP Specification WAE-

[7] Parlay Group, "Parlay 2.1 Specification," http://www.parlay.org/.

[8] M. Weiser, "The Computer for the Twenty-First Century," Scientific American, September 1991, 94-104.

[9] M. Weiser, "Some Computer Science Issues in Ubiquitous Computing," Communications of the ACM, July 1993, 74-84.

[10] A.J. Demers, "Research Issues in Ubiquitous Computing," Proc. ACM PODC'94, August 1994, 2-8.

[11] R. Bagrodia, W.W.Chu and L. Kleinrock, "Vision, Issues, and Architecture for Nomadic Computing," IEEE Personal Communications, December 1995, 14-27.

[12] Cross-Industry Working Team, "Nomadicity in the NII," available from http://www.lk.cs.ucla.edu/LK/lkxiwt/.

[13] M. Satyanarayanan, "Fundamental Challenges in Mobile Computing," Proc. ACM SigMobile, 1, 1, pp. 1-7, April 1997.

[14] G. Banavar et al., "Challenges: An Application Model for Pervasive Computing," in Proc. MobiCom'2000, August 2000, 266-274.

[15] M. Satyanarayanan, "Pervasive Computing: Vision and Challenges," IEEE Personal Communications, 8, 4, pp. 10-17, August 2001.

[16] L. Kleinrock, "Breaking Loose," Commun. of the ACM, 44, 9, pp. 41-45, September 2001.

[17] Location Interoperability Forum, http://www.locationsforum.org/.

[18] K. Raatikainen, "Functionality Needed in Middleware for Future Mobile Computing Platforms," Proc. ACM Advanced Topic Workshop: Middleware for Mobile Computing November 16, 2001, Heidelberg, Germany

[19] K. Raatikainen, "Middleware Solution for All IP Networks," Proc. 3Gwireless workshop, March 23-26, 2001, London, UK, pp. 335-340.

[20] K. Raatikainen, "Middleware for Future Mobile Networks," Proceedings of IEEE International Conference on 3G Wireless and Beyond, May 30 – June 1, 2001, San Francisco, Calif., pp. 722-727.

[21] K. Raatikainen, "Middleware," Chapter 3.2.7 in MITA Handbook, IT Press, November 2001.

[22] IETF Home Page, http://www.ietf.org/

[23] W3C Home Page, http://www.w3.org/

[24] 3GPP Home Page, http://www.3gpp.org/

[25] OMG Home Page, http://www.omg.org/

[26] Parlay Home Page, http://www.parlay.org/

[27] R. Soley, ed. An OMG discussion paper on Model Driven Architecture, version 3.2. OMG document omg/2000-11-05, November 2000.

[28] OMG MDA Home Page, http://www.omg.org/mda/

[29] OMG, UML Specification. OMG document formal/2000-03-01, March 2000.

[30] OMG, MOF Specification. OMG document formal/2000-04-03, April 2000.

[31] OMG, XMI Specification. OMG document formal/2000-11-02, November 2000.