LULEÅ UNIVERSITY OF TECHNOLOGY

2003:04

RESEARCH REPORT

In the Borderland Between Wearable Computers and Pervasive Computing

MARCUS NILSSON
MIKAEL DRUGGE
PETER PARNES

Department of Computer Science and Electrical Engineering
Division of Software Engineering

2003:04 • ISSN: 1402 – 1528 • ISRN: LTU - FR - - 03/4 - - SE

In the Borderland Between Wearable Computers and Pervasive Computing

Marcus Nilsson, Mikael Drugge, Peter Parnes
Centre for Distance-spanning Technology
Department of Computer Science and Electrical Engineering
Luleå University of Technology
971 87 Luleå, Sweden.
{ Marcus.Nilsson, Mikael.Drugge, Peter.Parnes}@cdt.luth.se

March 31, 2003

Abstract

In this paper we will show the ideas that are the foundation for the Borderland architecture. We have looked at two views of how researchers are thinking about computers in the future, wearable computers and pervasive computing. Wearable computers is the view where the user is wearing a computer and by that augment the user's view on his environment with additional digital information. Wearable computers have their strength in that they mostly do not have any requirements on the environment and that the user is in control of his integrity and security. In the pervasive computing view the digital representation of reality is in the physical environment itself. There are few requirements on the user for him to be able to use this digital augmented environment. There is also no restriction of what can be used in this environment as everything can be used for digital representation.

We can see that the problems in one view are often the strengths in the other view. We believe that by combining these views into one we can use the strength of both areas and in that solve some of the problems involved in these two research areas. In this paper we present some of the problems we see in this combined view and also some ideas to solutions that we will evaluate in future work.

1 Introduction

For many years the slogan for Sun has been that ”the network is the computer”. Depending on how you define a computer this can of course be true or false. The traditional view is that the computer is a processing unit with memory, an input device and an output device. If we look at the computer today with this view we would probably define Sun’s slogan as false. If we take away the input and output devices and just keep the processing unit as has been done with IBM’s Meta Pad [24] we will get a computer but not something that the user can use directly. With network capabilities in this basic unit we could really make ”the network is the computer” to a reality. Input and output capabilities would then exist in the network together with all other services that the user would need. This would of course be most useful in a wireless scenario, where the basic processing unit with wireless network capabilities would be worn as a wearable computer and be the link for the user to his digital environment. We should not stop at the input and output, why not make use of the processing powers that can exist in the network. It can be argued that a central processing unit is unnecessary as a wearable computer if there are sufficient resources available over the network. However as Rhodes points out [18] there is a gain for security and privacy in combining pervasive computing and wearable computing. We see that in the society today people often prefer personal equipment instead of public units, for example the mobile phones in contrast to public phones. Personalization of units is important and research in pervasive computing is trying to address this issue, but in the context of pervasive computing [15]. With a personal wearable computer as a controlling unit the other units in the ”computer” will automatically be personalized. The perceived security and integrity will also be increased with a personal controlling computer and this will make it easier for users to adapt to a more pervasive computing environment.

The user in the information era is using several devices with a range of different capabilities from the desktop computer with a big screen and good audio capabilities to the small mobile units with limited sound and/or limited screens. Today we see all of these devices as separate units in most cases. Depending on the circumstances the user will change the device for the best one for the current context. If the user is working with a certain application when this switch should occur, he can choose between three alternatives. The user can finish the work on a device that now is not the optimal for this situation. He could postpone what he is doing until the old device is used again. The last alternative is for the user to change device and transfer the data to the new device where an application with similar functionality exists and continue the work. This data transfer will take time and is not very easy for all devices. All three of these alternatives delays and interrupts the user in his current task. If we instead would see all these devices as part of a bigger and dynamic computer, where all digital devices in the environment are included, we would not have a change of device as all devices are treated as part of the same computer. The central unit in this scenario would be a small and light-weight wearable computer that is worn at all times.

Let’s look at a scenario: You are on your way to work and of course you are wearing your wearable computer that is connected to a

Bluetooth headset enabling audio input and output. As you walk you suddenly get an idea that you just have to write down. You start dictating for your wearable computer and after a while you arrive at your office where you have an ordinary desktop computer. In the scope of this paper we see this as a rather high capacity processing unit connected to several input and output units. The document you started on your way to work has become quite big and you still feel that you have more to add. As you sit down at your desktop computer the different devices recognize that you are starting to use the desktop computer and adds the keyboard and screen as input and output for your ”computer”. You can also use the extra processing power on the desktop computer to speed up or adding more functionality to the applications.

In the previous scenario the user was actively involved with the computer and in some sense what selection of units that should be part of the computer. We can also look at scenario where the user is not directly active. Mobile health [20] is today a hot topic in research and the Borderland computer view would be useful in this area. By making every sensor a separate smart unit with network capabilities that can reply to service discovery requests by itself we can dynamically make a sensor net on a patient just by putting the needed sensors and a small wearable on the patient. Basic calculation of the sensors’ data can be made by the wearable. But when the patient is near other processing units, these units can be used for more advanced calculations. There will also be a dynamic bit rate for the network connection to the hospital and in some times there may also be no connection at all. The Borderland computer would be able to adjust the amount of data sent to the hospital depending on the connection and it could even store data to send later when no connection is present. When a doctor is examining the patient the different sensors can easily be hooked to the doctor’s wearable computer and be displayed where the doctor wants it, for example in a head mounted display or on a big wall screen. For security reasons the patient would maybe want to be the initiator and the sensors could then be designed to only accept orders from one controlling wearable computer at a time in this case this would be the patients. The patient would in this case be the one who displays the sensor values in a way so that the doctor can examine them.

2 Problems

In the scenarios in section 1 a variety of very different devices are being used and there are several problems that have to be solved for this to work. In this section we will go through the issues that we feel are important and need to be solved.

2.1 Network

The network is a fundamental part of our system and must be considered as a requirement for it to work as expected. There is a clear need of dynamic networks where units are allowed to appear and disappear often. This behaviour is most common in networks which incorporate wireless technologies such as WaveLAN and Bluetooth. The nature of Borderland makes it not acceptable to expect anything static in the network and can therefore not depend on configurations servers [7] to configure the devices in the network.

Many protocols like Appletalk [23] and IPX [26] incorporates technologies to dynamically configure a network without central configurations servers. However many of these network protocols have in the last years been replaced by pure IP networks. Even the key players in these protocols have realized that the future lies in IP networks. Because of that the use of IP in Borderland will help it to easily be adopted into existing networks and devices as no extra infrastructure or technology will have to be implemented for it to work.

The original IP architecture does not incorporate the dynamic and automatic nature that we can see in other protocols which would restrict the uses of Borderland for environment where a static infrastructure with configurations servers are already present. Recently development in the IETF is trying to address this and a standard called Zero Configuration [9] is proposed. One important aspects of this standard is that it is designed to work with existing infrastructure and systems and will therefore make it easy for Borderland to be integrated seamlessly in existing systems and infrastructures.

2.1.1 Service Discovery

Even if the network is important for the system to work, something that is equally important is the ability to discover the different ”parts” of the computer in the network. So when the user is entering a new environment the wearable computer will be able to discover what kind of devices that are available and what services they can provide for the ”computer”. Much research has been done in this area [19]. There are open IETF standards as SLP [14] and also more commercial driven alternatives as JINI [28] and SSDP [11]. Some network technologies such as Bluetooth [3] have implemented their own service discovery protocols.

2.2 Scattered Programs

The Borderland architecture is not restricted to be running on only one device. The main routine of the application will start up on the wearable computer. After the initializing functions the application can be be spread on the network. The UI part can with benefit be executed on the unit that is presenting it if it has enough processing power. If specific calculation will be needed on an input device, like a camera for gesture recognition, this part of the program can be executed directly on that device. Different threads of the program can also be executed on different units. The mobile objects [1, 4] in Borderland are not designed to live by themselves as mobile agents [8, 5], but are thought to be designed for a specific type of unit and only travel to the unit from the controlling wearable computer. On the unit the object should only perform calculations that are relevant to the unit or general calculation if sufficient processing power is available. When there task is finished they should die and be erased.

2.2.1 State Preservation

The dynamic nature of the wireless mobile network will make it likely that the network will break and units that are running the program will lose connection with the central controlling wearable computer. The research of today is focusing on recovery of data and state with the assumption that a new connection with the network and the unit will after a time be established and that the data will then still be valid. This is not sufficient for Borderland as the calculation done on the external unit can have strict timing requirements, as the data in a gesture recognition unit. The local nature of the network architecture in Borderland as discussed in section 2.1 does not at all guarantee that a new network connection will make it possible to connect to the disconnected unit, and by that way receive the lost information. The important research question here is then how this kind of behaviour can be handled and hidden for the user so that a consistent experience without abruptness in the work flow will be achieved.

2.3 User Interfaces

As the user moves between different contexts during the day, he will be involved in a number of situations with unique requirements on the user interface. For example, the user presented in the scenario earlier only had access to audio input and output while walking. However, upon entering the office the user interface was extended with a keyboard for input and a large monitor for visual output. Even though the task of writing a document is the same, the ways of performing it can vary depending on the circumstances. Sometimes there may be very subtle nuances in what constitutes a good user interface. For example, when driving a car a head-mounted display and a keyboard may not be the optimal choice, while for a passenger in the same car it may well be. When the car gets stuck in a traffic jam for a prolonged period of time, the driver may pick up the keyboard to do some quick editing.

2.3.1 Adaptiveness

The key point in this discussion is that an application should be able to dynamically adapt the user interface to deal with the devices currently available. It should be easy to add and remove devices, and there should be no restriction as to when and how they can be used. For visual output, this adaptiveness could mean allowing a user with a small handheld computer to transfer its graphic representation to a high-resolution monitor by walking up to it. For text input, it could mean dictating by use of voice but doing editing with a keyboard when gaining access to one. The task and application remains the same but it can now be more easily performed by utilizing the best and most suitable equipment available.

2.3.2 Technical Challenges

The main technical problem in this area is how to describe the user interface so it can be presented correctly and be usable with such a large number of diverse input and output devices. In addition, the application itself needs to find ways of presenting the same material in different ways, such as through vision, audio or even tactile feedback. There is ongoing research in this area; the eXtensible Interface Markup Language (XIML) [17] is an XML-based language enabling a framework for defining interface components. XIML supports, user interfaces that automatically adapt to the screen area available, as well as distributing components of a single user interface onto multiple target devices.

2.4 Privacy and Security

With the use of a wearable computer as the central unit in this scattered computer, we will increase privacy and security for the user. The user will have full control over what information is passed to other units in the network as the wearable computer is configurable to the users wishes and needs for the current situation.

As objects of the application will be spread around the network, techniques that will preserve the object on a foreign device have to be implemented [21, 27, 30]. There is also a need to validate the object running on other units so that no third parties software can try to behave like an object of the running application and receive data that is not meant for it or send false information to other parts of the system.

The owner of the units used by the application must also have a way to determine what an object is allowed to do on the unit [10]. It is also important to implement some kind of system to be able to determine how much of the resources that can be used by the individual objects running on the unit [2].

2.4.1 Anonymity

As the network is configured automatically and dynamically the identity of the unit in the network will not be traceable to a specific user as long as this identity is not used to fetch or send this kind of information in such a way that it is readable by a third party. The action taken with this identity can of course be logged but a connection between actions can only be traced as long as the user is using the same identity. Because of the dynamic nature of the network the user can choose to change identity for his unit at any time as long as other units are not depending on this to reach the unit with required information.

3 Borderland Today

The Borderland architecture is under development and the central controlling unit is today a Pocket PC device with WaveLAN and Bluetooth. With the help of the jacket technology that Compaq have integrated in their iPAQ series we have the possibility to add technologies that need more speed than a wireless network can give (for example the graphical data to a head mounted display). This is good for the independence of the user as the wearable can be used as an ordinary wearable computer when the pervasive infrastructure is not present at his current location. The architecture is not dependent on Pocket PC devices and could be implemented on any wearable computer.

We believe that the handheld computer is well suited for this kind of system as it has enough processing power to function on its own but is also light-weight and small so that it can be carried around all day.

We are developing the Borderland architecture in Java [12] but we are trying to make the architecture independent of language and Operating System.

4 Related Work

Related work can be divided in two different fields, distributed systems and single systems or client and servers solutions. In distributed systems the running applications can be divided into smaller pieces and be executed on different units. On single systems the applications are restricted to one unit and must have some way of communicating with other units to use them.

The distributed system do not really handle the security and integrity part that Borderland do in that personal information will travel on the network more often. The information may also be stored in locations that the user does not directly have access to and in that controll over. Hive [16] is such a system where applications are a gathering of mobile agents who can roam on the network and execute on units that have the services that are of interest. These mobile agents do not have a central controlling unit and do not require a trusted hardware to function and as so do not deal in the best way with the integrity and security for the user. The Hive project have been experimenting with adding wearable computers to the system. However the wearable computer is not looked upon as a controlling unit as in Borderland but just as another part. Other projects in this area are The Open Agent Architecture [6] and one.world [13].

In the client and server systems we find Intel-Research personal server [29] that share many ideas with Borderland, but in its current implementation is mostly a mobile file server and does not incorporate mobile code ideas and the use of many different external units. A slightly different view in this area can be found in InfoPad [25] and Parctab [22]. Both are taking the approach with thin clients where mobile units are used to connect to central servers to function. By that they really depend on an infrastructure to work and also makes much of the personal data to be located somewhere where the user in most cases is not in control over it.

5 Conclusions

We have presented the ideas behind the Borderland architecture, which is being developed at Luleå University of Technology. We believe that there are good aspects from both wearable computers and pervasive computing. However we also see that there are problems involved in both. The strength in one area is often the problem in the other. By combining these two areas we believe that we can take the strength from both views and in that way create a new architecture that will solve both views problems by incorporating the strengths from the other area's view. The integrity and control of wearable computers will be preserved but will be target for more threats as more information will leave the user into the environment. However the user will have full control of leaving this communication at any time. As we still will be able to use the digital environment the strength from pervasive computing will still be valid and in some way even better as no infrastructure for units need to present for them to function in Borderland.

We believe that many of the wearables today are made in a way that an average person will not use. A computer that the user is required to wear need to be small and light-weight. Handheld devices are designed in a way that we believe is something that the ordinary user would wear and use on a daily basis. Today these devices may not have the processing power that some application in wearable computer require, but as Borderland is capable of using the power in its environment this will not be a problem in most cases.

6 Future Work

Borderland is a recently started project and therefore we have most of the work in the future.

Network is really important and we see great opportunities in spontaneous network that do not need special servers to function. In Borderland we want to evaluate the proposed standard zero configuration and also see if more things can be done to support these kind of networks.

We will also develop a software architecture to deal with mobile code and the privacy and integrity concern involved in this. This architecture will of course be designed to take advantage of the fact that a wearable computer is always worn by the user. A big topic in this area is how to make loss of connection to important unit as invisible as possible for the user.

User interfaces are important and new ways to describe the user's digital environment must be researched and evaluated. We also see a great need for dynamic and adaptive user interfaces in the future with many different input and output devices coming and going as the user is moving around during the day.

Privacy and integrity are important topics and will be present in all work that will be done in Borderland.

7 Acknowledgements

This work was sponsored by the Centre for Distance-spanning Health care (CDH) and the Centre for Distance-spanning Technology (CDT) at Luleå University of Technology, and by the Swedish Foundation for Strategic Research (SSF).

References

[1] Lorenzo Bettini, Viviana Bono, and Betti Venneri. Coordinating mobile object-oriented code. In *Coordination Models and Languages*, pages 56–71, 2002.

[2] S. Brandt, G. Nutt, and K. Klingenstein. A discrete and dynamic approach to application/operating system qos resource management, 1998.

[3] Jennifer Bray and Charles F Sturman. *Bluetooth: Connect Without Cables*. Prentice-Hall, Upper Saddle River, 2001.

[4] Antonio Carzaniga, Gian Pietro Picco, and Giovanni Vigna. Designing distributed applications with a mobile code paradigm. In *Proceedings of the 19th International Conference on Software Engineering*, Boston, MA, USA, 1997.

[5] David Chess, Colin Harrison, and Aaron Kershenbaum. Mobile Agents: Are They a Good Idea? Technical Report RC 19887 (December 21, 1994 - Declassified March 16, 1995), IBM Research, Yorktown Heights, New York, 1994.

[6] Adam Cheyer and David Martin. The open agent architecture. *Journal of Autonomous Agents and Multi-Agent Systems*, 4(1):143–148, March 2001. OAA.

[7] R. Droms. Dynamic host configuration protocol. Technical report, IETF, 1997.

[8] A. Duda and S. Perret. Mobile agent architecture for nomadic computing. In *International Conference on Computer Communications*, 1997.

[9] Guttman Erik. Autoconfiguration for ip networking. *IEEE Internet Computing*, May 2001.

[10] J. Feigenbaum and P. Lee. Trust management and proof-carrying code in secure mobile-code applications, 1997.

[11] Yaron Y. Goland, Ting Cai, Paul Leach, and Ye Gu. Simple service discovery protocol. Technical report, IETF, 1999.

[12] James Gosling, Bill Joy, Guy Steele, and Gilad Bracha. *The Java Language Specification*. Sun Microsystems, Inc., `http://java.sun.com/docs/books/jls/second_edition/html/j.title.doc.html%`, second edition, 2000.

[13] Robert Grimm, Janet Davis, Eric Lemar, Adam MacBeth, Steven Swanson, Tom Anderson, Brian Bershad, Gaetano Borriello, Steven Gribble, and David Wetherall. Programming for pervasive computing environments. Technical report, University of Washington, `http://one.cs.washington.edu/papers/one.world.pdf`, 2002.

[14] Erik Guttman. Service location protocol: Automatic discovery of IP network services. *IEEE Internet Computing*, 3(4):71–80, 1999.

[15] Trevor Jonathan, David M. Hilbert, and Bill N. Schilit. Issues in personalizing shared ubiquitous devices. In Gaetano Borriello and Lars Erik Holmquist, editors, *UbiComp 2002: Ubiguitous Computing*. Springer, 2002.

[16] Nelson Minar, Matthew Gray, Oliver Roup, Raffi Krikorian, and Pattie Maes. Hive: Distributed agents for networking things. In *First International Symposium on Agent Systems and Applications (ASA'99)/Third International Symposium on Mobile Agents (MA'99)*, Palm Springs, CA, USA, 1999.

[17] Angel Puerta and Jacob Eisenstein. Ximl: a common representation for interaction data. In *Proceedings of the 7th international conference on Intelligent user interfaces*, pages 214–215. ACM Press, 2002.

[18] Bradley J. Rhodes, Nelson Minar, and Joshua Weaver. Wearable computing meets ubiquitous computing: Reaping the best of both worlds. In *ISWC*, pages 141–149, 1999.

[19] M. Robert. Discovery and its discontents: Discovery protocols for ubiquitous computing, 2000.

[20] I. Sachpazidis. @Home: A system for real time monitoring of patients' vital parameters. In *Advanced Medical Image Processing. Proceedings*, June 2002.

[21] Tomas Sander and Christian F. Tschudin. Protecting mobile agents against malicious hosts. *Lecture Notes in Computer Science*, 1419:44–??, 1998.

[22] Bill N. Schilit, Norman Adams, Rich Gold, Michael M. Tso, and Roy Want. The PARCTAB mobile computing system. In *Workshop on Workstation Operating Systems*, pages 34–39, 1993.

[23] Gursharan S. Sidhu, Richard F. Andrews, and Alan B. Oppenheimer. *Inside AppleTalk*. Addison Wesley, Massachusetts, 1990.

[24] Thomas Staudter. The core of computing. *IBM Think Research*, 2002.

[25] Thomas E. Truman, Trevor Pering, Roger Doering, and Robert W. Brodersen. The infopad multimedia terminal: A portable device for wireless information access. *IEEE Transactions on Computers*, 47(10):1073–1087, 1998.

[26] Paul Turner. Netware communications processes. In *Netware Application Notes*, pages 25–81. Novell Research, September 1990.

[27] G. Vigna. Protecting mobile agents through tracing. In *Third Workshop on Mobile Object Systems*, 1997.

[28] Jim Waldo. The Jini architecture for network-centric computing. *Communications of the ACM*, 42(7):76–82, 1999.

[29] Roy Want, Trevor Pering, Gunner Daneels, Muthu Kumar, Murali Sundar, and Light John. Personal server: Changing the way

we think about ubiquitous computing. In Gaetano Borriello and Lars Erik Holmquist, editors, *UbiComp 2002: Ubiguitous Computing*. Springer, 2002.

[30] Bennet S. Yee. A sanctuary for mobile agents. In *Secure Internet Programming*, pages 261–273, 1999.