Nimbus: A task aware/context aware mobile computing platform

Kartik Muralidharan and Puneet Gupta
Pervasive IT research
Software Engineering and Technology Labs (SETLabs), Infosys Technologies,
Electronic city, Hosur road
Bangalore, India - 560100
{kartik_muralidharan, puneet_gupta}@infosys.com

Abstract

In a true pervasive world barriers among devices no longer exist. Users can access and manipulate data from anywhere at anytime using a dynamic range of computing devices as they move across different environments. Today's "computer-centric" paradigm however puts increasing demand on user attention impacting overall efficiency. Today, typically a user has to manually figure out the available resources and appreciate the constraints of the new computing environment in order to continue performing his computing tasks in a new device, as he moves across computing devices. This results in the user spending considerable amount dealing with the different nuances of different computing environments and remembering the different contexts and ensuring that all data is in sync as he works on a single task but using different devices and different times. In this paper, we describe Nimbus, an enterprise grade platform we have developed and deployed aimed at providing true anytime anywhere computing, realized by an end to end system designed around the "tasks that the user performs" rather than computing devices the user uses.

1. Introduction

"What information consumes is rather obvious: it consumes the attention of its recipients. Hence a wealth of information creates a poverty of attention, and a need to allocate that attention efficiently among the overabundance of information sources that might consume it." Herbert Simon "Designing Organizations for an Information-rich World"

Over the past five years or so, pervasive computing has emerged as a new computing paradigm with much appeal [1]. Enterprises are increasingly showing interest in deploying pervasive IT infrastructures to realize benefits such as enhanced productivity offered by this new computing paradigm. Pervasive computing is all about putting the user, rather than a particular computing device, in the centre of all computing activities. It is about the concept of "any-time, any-where computing" optimized by means of intelligent environments and context-sensitivity. Essentially, all this boils down to a need for an infrastructure enriched with computing, sensing and communication capabilities and with a very high degree of collaboration between all the infrastructural components, all aimed at enhancing the productivity of end user and providing as optimal a user experience as possible.

Mobility is a critical dimension in enterprise computing today, expected only to increase as the mobile computing market grows and matures. While the diversity of computing devices is helping in widespread adoption of mobile computing, the multi device computing environment that we are faced with presents certain unique challenges such as lack of "contextual continuity" and increased "attention costs". Individual devices (connected or not) operate in isolation from each other. As the user moves from one computing device to another, the work context[1] doesn't move with the user. Also, different devices, owing to difference in capabilities, present constraints that the user needs to be aware of all the time. As a result the user is required to spend his/her "attention bandwidth" more on the computing intricacies rather than focusing on the task[2] at hand.

[1] Context is the information about the environment (connectivity, costs, bandwidth, dependences etc.) associated with an application.

[2] "A task represents a user's computing intention. A task may range from a simple activity such as document editing or

We propose “Nimbus”, a “task aware computing platform” that realizes true any-time any-where computing by placing the user and user tasks at the centre of the computing equation. In Nimbus, the computing devices (mobile or otherwise) are mere instruments that are used to make the tasks available to the user in a context aware manner and customized according to user preferences and device capabilities. The core motivation for this research has been to bridge the gap between research and actual adoption of pervasive computing in the true sense by enterprises. The key contributions of this research are primarily two fold. One is demonstrating how an enterprise grade pervasive computing platform can be created and successfully deployed in organizations today by an appropriate combination of new software and integrating mature and commercially viable mobile and wireless technologies. And two, an innovative architecture in the form of an overlay on the existing computing infrastructure that works without the need to fundamentally alter the underlying computing infrastructure in any way, thereby remaining effectively invisible to the end user and which can be easily extended to create and deploy different pervasive computing solutions. Also, by operating at a higher context level, Nimbus is generic and hence not limited to specific scenarios.

Section 2 describes the problem statement using a task migration scenario as an example as well as the scope of the work in an enterprise context. Section 3 describes the proposed platform architecture that supports the intended computing paradigm. Section 4 explains the workings of the various platform components using the scenario mentioned in Section 2. Section 5 deals with the piloting and deployment related details of the system in an enterprise. Section 6 outlines our approach as well as the current state of work and future research while Section 7 summarizes the results.

2. Introducing ‘Nimbus’

Extended from the fundamental concepts of context aware computing and pervasive computing, Nimbus is a manifestation of the vision of “task aware computing” [2]. In the following we will introduce one of the system scenarios taken as the basis for this work.

Scenario 1*: Helen is working on her desktop computer at work preparing a business proposal. She needs some information and is looking for resources on the internal company knowledge portal. In another task, she is trying to browse through the contents of a book which she wishes to buy from Amazon. Helen is also involved in reviewing a design document created by one of her peers and another document briefly outlining the objectives for her team. While looking for some information on the net, she accidentally comes across a webpage on some exotic Italian recipes. Although this happens to be one of her interests, due to lack of time she is not able to go through it completely. It is almost the end of the day’s work and time for Helen to head home. It would take her another hour to reach and she intends to continue working on her way back (on her PDA) and then again at home on her personal computer.*

In the existing computing environment, one would be required to explicitly port all these tasks through various mediums to achieve continuity between the multiple disconnected devices. For example, a note would have to be manually made about the activities, the Internet links would have to be noted down, while the documents need to be copied onto some suitable storage media. Also, if a particular task is modified (e.g. editing a document) on a particular device, the user would have to ensure that the latest copy of the work is taken when the user moves from one device to another. In a world saturated with Nimbus, these devices would collaborate with each other allowing seamless continuity.

In an enterprise IT context, Nimbus manifests itself as a computing cloud that operates across devices and networks but uniquely characterized by "invisibility", a fundamental character that provides a seamless, distraction free and productive computing experience to end user. Nimbus is context intelligent and understands the limitations and strengths of different devices it works with and appropriately creates customized working environments for the user. The essence is that “the tasks the user is performing” is the central and most important entity. Nimbus gives the user the flexibility to keep working on these tasks effectively and seamlessly, irrespective of the computing device and even remotely! (Using a mobile device). While ubiquitous network connectivity can enhance the over all effectiveness of the pervasive computing environment realized by Nimbus, Nimbus puts no specific requirements

accessing an information service to a complex activity involving a workflow, such as arranging a conference, customer support tasks etc.” [2]

on all devices having network connectivity. The concept of "context carrier" (explained later) ensures contextual continuity even to non-connected devices.

We will now illustrate the working of the above scenario in the presence of Nimbus.

Helen indicates her intention of leaving office to Nimbus on her desktop computer. Transparently (without any user intervention whatsoever) and intelligently, Nimbus appropriately models the tasks that Helen was performing and transfers that work context onto her PDA or any other mobile devices that she is carrying. Helen now leaves. Nimbus is present on the PDA and senses that it is a (wireless) connected device with a reasonable display and some limited memory available. A view of all the complete and incomplete tasks is presented on the device display by Nimbus. A smart task-based view provided by Nimbus, allows Helen to easily initiate any task. Helen can now resume work on the design document, as Nimbus had intelligently (and transparently, without any user intervention) made a decision to carry this document onto the device. Helen can also continue browsing her favorite Italian recipe web page through the GPRS (or any other WWAN) connectivity (Nimbus abstracts the connectivity mechanism from the user so that user can transparently access any networked service) provided on her PDA. She marks this task complete and makes some changes in the other available documents.

Helen finally reaches home and as she is nearing her PC, the Nimbus component on the computer automatically senses her arrival and generates a welcome message on the desktop. Nimbus also offers Helen, to recreate the office environment on her home PC. Helen can now continue working on the same tasks at home in a similar environment/context that was present at work. Nimbus also maintains consistency of the documents, making available any changes that she had made on her PDA on the desktop PC.

Rooted in Nimbus are some of the following fundamental concepts that are important to appreciate to understand the applicability of Nimbus in a specific context. These are:

- Task aware computing: We interpret task-aware computing as being about designing computing solutions, centered around the concept of the "tasks the user is performing" and then providing a computing fabric in such a way that the specific computing device that is being used to perform a task becomes less significant and the focus is always on the user task. A significant ramification of this concept is "computing device independence"- an essential characteristic in true mobility.
- True any-time any-where computing: Nimbus propagates the concept of truly ubiquitous systems, where the focus is on enabling the execution of a user task any time and any where. The concept of "any where" here extends even beyond the place where the user is present. So, if it makes sense for a user task to be executed on a remote computing device, the user should be able to do so.
- Context aware computing: The computing activity should be context aware. The behavior of an application in a specific or computing system in general, should change with the changes in the environment contexts in which the user is operating [3]. Related ramifications of this are the ability of system (applications) to exhibit intelligence in behavior and implicit inputs, thereby providing a distraction free user interaction.

Nimbus is an evolving research program with multiple evolving scopes. The research program has been so designed so that Nimbus can be viewed from different perspectives. This could be

- An end to end system/computing paradigm over which specific solutions can be built.
- A platform for designing and developing solutions around the concept of "task awareness" as proposed by the platform.
- Technologies for integrating pervasive computing capabilities in IT application- Increasing the PQ [4] (Pervasiveness Quotient) of applications/solutions/IT infrastructure
- A collection of individual technologies/system capabilities that can be adopted in existing applications/solutions or around/using which new applications/solutions can be created
- A test bed for demonstrating specific concepts/paradigms in a real world end-to-end system thereby evaluating the validity/value of a concept/technology and then adopting them in different solutions

3. Platform description

As a technology platform, Nimbus is intended to be a system that is deployable in a typical enterprise context. So, besides the above mentioned pervasive computing concepts there are other enterprise IT related architectural considerations that have been taken into account in developing the platform. These enterprise IT specific considerations are briefly mentioned below to put this in context.

- A technology integration approach [4], utilizing commercially available infrastructural technologies
- Deployability considerations: Architecting the solution in such a way that individual devices (PCs or mobile) can be easily Nimbus-enabled
- Security requirements typical in any enterprise IT deployment
- Incremental modification of the enterprise IT [4]. The solution should act as an over-lay on the existing IT infrastructure without requiring modifications (or minimal modifications) to the core IT infrastructure.
- Scalability and extensibility: The architecture should be scalable (increasing the extent of enterprise deployment) as well as easily extensible allowing more functionality to be added without requiring complicated roll-outs.

This section provides a fundamental overview of the different architectural components that work together in forming the backbone of the Nimbus platform. The key focus elements of Nimbus are context sensitivity, contextual continuity, device independence and task awareness. Note that we extend the typical definition of context (See Note 1) to include information about user tasks, task dependencies (including computing resources) and other task parameters that are required for effective task execution.

The components are best viewed as individual capabilities that can be leveraged in creating a system that exhibits one or more characteristics as described in the section above.

Fundamentally, the platform has three key component categories.

1. Components on individual computing devices (PCs, Laptops, other types of computing devices including PDAs, smart phones)
2. Context carrier and associated components
3. Network components

The broad architectural components of Nimbus are shown in Figure 1. The different components of the architecture are explained below.

----INSERT FIGURE 1 ABOUT HERE-------

Platform Manager: An environment at a given location is the set of capabilities that can be used from that location. These capabilities include those that can be accessed directly e.g. Local printer, as well as those that can be used via some communication network e.g. Remote File Server. Platform manager essentially models, analyzes and monitors the environment in real time. The Platform Manager is responsible for

- Matching the requests issued by the Task Handler (described next) to the set of local or remote services
- Maximizing overall utility of a requested set of capabilities in the face of limited resources and according to user preference. For example, modifying the structure of the user context to reduce its size without significant loss of information to handle mobile devices of limited memory resources.
- Retrieving and restoring the configuration and execution state of services
- Proactive guidance. When a task can't be executed because of resource restrictions, the system offers suggestions to the user

Task Handler: It is the responsibility of the Task Handler to proactively request the Platform Manager to set up the capabilities that support the user tasks. The output of the Task Handler is the computing context. The Task Handler assumes that the Platform Manager holds explicit information about what the local environment can offer– its available capabilities and resources – as determined by local devices, communication infrastructures, and applications. Furthermore, upon dynamic changes in the available capabilities or resources, the Task Handler negotiates the adjustments that best serve the person's intention with the Platform Manager.

Context Monitor: This component is responsible for observing the physical context surrounding people. For instance, what is the physical location of the person, what are the current devices available.

This awareness can be made in the form of certain triggering conditions such as:

- An explicit indication from the person wanting to move to another environment
- An event originated by the activity itself e.g. closing a window on a window based interface

Context carrier and associated components: A fundamental component of the system is the "context carrier", typically a mobile device (PDA or a mobile phone) that is associated with an end user and acts as her/his proxy as the user moves from one computing context to another. The context carrier ensures that all computing context changes in any device that the user is using (including any resources) are seamlessly transferred, as the user moves from one computing device to another, thus maintaining task consistency. High level context carriers such as PDA's would allow continuity in user computing whereas low level devices like a mobile phone or USB-stick would behave only as a transitory device between environments.

Automatic user detection: This component enables the system to detect the presence of the user and begin interacting with the user. In the current system, we have realized automatic user detection using Bluetooth, Infrared and cradle based connectivity modes independently so that a wide range of devices can be automatically detected.

XML modeler: An important component of our system is the representation and management of user tasks. In order for computers to process tasks, we have to be able to represent a task. The XML Modeler works closely with the Task manager structuring the computing context into an XML task description as specified by *Nimbus*. Figure 2 shows a subset of task representation as modeled by the XML modeler.

----INSERT FIGURE 2 ABOUT HERE------

User avatar: The 'User Avatar' is a logical entity associated with every user that runs in the IT environment and operates on behalf of that user, managing user contexts and preferences. The component maintains information about the user in the form of user profiles, aiding in making certain decisions. Nimbus is so architected, that users need to only contact their user avatars for any operation from any device. User avatars provide complete abstraction of the IT infrastructure to the end user/device, making the solution very flexible and extensible. User avatars can also interact with each other, there by providing high levels of system/user collaboration.

Intentional Resource Information Service [IRIS]: IRIS is a technology under development at our lab that allows resource discovery procedure in networked environments based on user intentions. IRIS springs from the concept that end-users should not be bogged down with network names when looking for a resource on the network, and should have the liberty to freely sketch and describe *what* they want and not *where* to go about finding it. The purpose of IRIS is to have the user focus on the intent rather than details of the computing paradigm.

Messaging engine: This component provides the framework for reliable and efficient synchronous messaging between the various components of the Nimbus Architecture. The **Session manager** maintains the state of events by keeping track of previous transactions. Messaging engine provides the messaging infrastructure that Nimbus uses for device-user avatar, device-device and "user-avatar"-"user-avatar" interactions.

4. Nimbus in Action

We now use Scenario 1 described in the Section 2 to illustrate how the Nimbus architecture achieves seamless continuity of 'user tasks', focusing on the interactions among the individual components explained in Section 3.

While Helen is working on her PC, the Task Handler continuously monitors the tasks she is working on. When her intention of leaving is indicated to Nimbus, the Task Handler supported by the XML Modeler generates a snapshot of the working context. This XML snapshot provides sufficient information to restore the user context on a remote device. The Task Handler then requests the Context Monitor to provide details on the physical context. Here, the physical context would include information on the type of context carrier (in this case a PDA) available, how the carrier is connected with the PC and characteristics of the carrier (available memory, display, connectivity options). The Task Handler then synchronizes the relevant data with the carrier, proactively making intelligent decisions when needed. One of these decisions is determining which resources need to be carried along with the carrier. Due to memory limitations of the context carrier, large resources often cannot be carried along. The Task Handler places these 'large resources' in the Transient Data Storage associated with Helen's User Avatar.

The Platform Manager on the PDA is aware of the carrier capabilities, allowing Helen to restore only those tasks that are possible in the given environment. In addition, Nimbus allows Helen to remotely execute certain tasks on her office PC and determine the status of these tasks via her User Avatar. A manifestation of this feature is the ability of the user to indicate Nimbus via her/his PDA to recreate the work environment on a remote device. A scenario example could be Helen interacting with Nimbus on her PDA telling it to recreate her latest work context on her office PC just before she reached office. This could be specifically useful in situations where certain tasks require considerable time to setup (e.g. a task requiring synchronization of data from multiple locations). As before, the Task Handler notes any changes made to Helen's working context and resources.

At home the Automatic User Detection component identifies the carrier through Bluetooth™ detection. The Task Handler on the carrier interacts with that on the home PC transferring the user context and resources along with the changes. The Task Handler on the home PC then co-ordinates with the Platform Manager to restore the user context in the given environment. If the environment is suitable the Platform Manager makes available the 'large resources' that were not present on the PDA by interacting with the user's User Avatar.

----INSERT FIGURE 3 ABOUT HERE-------

5. Nimbus in the Real World

At SETLabs, Infosys, the Nimbus initiative is being actively deployed in different scenarios. Apart from the horizontal Scenario 1 described earlier, currently three other pilots, each focusing on different application scenarios as well as evaluating specific concepts and features of the technology platform are being developed. A typical sample size of each pilot is minimum ten users. The collaborative application (described later) is being piloted on a larger scale (>30 users across the organization) however. The deployment includes desktop computers, notebook computers, pocket PC PDAs, Palm PDAs and Symbian OS based smart phones, Java based simple mobile handsets, Wireless LAN (802.11b), Bluetooth adaptors and other standard IT infrastructural components.

In the rest of the section, we describe an optimized enterprise collaboration application realized using Nimbus using a scenario description.

Note: The other two pilots currently in deployments deal with a healthcare scenario, aiming to demonstrate a pervasive computing environment realized by Nimbus in a hospital scenario and a field force automation scenario, demonstrating extension of today's field force automation solutions by incorporating pervasive computing characteristics enabled by Nimbus.

Scenario 2: Helen is in a meeting discussing some aspects about an upcoming project with some team members. Before coming to the meeting, she was working on a client response document. This document is a collaborative effort of Helen and some other members of the managerial team. James, who is also in the managerial team, requires the document to make some urgent modifications and submit it for review. James calls Helen and asks for her to send it over to him. Helen asks NIMBUS (available on her PDA), to make the document available to James on his computer so that he can modify the same. The document pops up on James screen where he modifies the document. Nimbus takes care that all other stakeholders see the latest document. Helen also asks NIMBUS to make the document available on the nearest printer in the conference room which is the only computing resource available so that she can also share the same document with other team members involved in the meeting.

6. Work in progress

In Nimbus we are exploring scalable enterprise pervasive systems. Although our setup is similar to that of Aura [5, 6, 7], our focus is on how existing enterprise infrastructure setup can be made pervasive and extensible with personalized mobile software.

In an enterprise pervasive context users are often on the move using different computing entities at different times as they move along. Current software applications are however designed to run on a single computer and never relocated. There are several options for moving the desktop along as explained in [8]. We use an approach similar to that of the WAIFARER approach [8], wherein wrapping techniques are used to capture the interface exported by an application. This information is then stored in an XML file later used to restore the application state. Most component based applications, such as those using Microsoft COM, export such API's that allow the wrapper to extract sufficient information. Using these COM objects, Nimbus clients capture enough application state for restoration of the user tasks. This scheme does not assume that identical sets of applications are available on all computing entities, but similar applications may or may not exist on each computing device. The main disadvantage of this approach is that the user tasks are currently limited to applications that export this functionality. Further, system level API calls are used to capture information of the capabilities of the current computing device, such as network connectivity as well as available resources to support context aware computing.

Broadly speaking, Nimbus is evolving in four dimensions. These are

- Introducing new pervasive computing capabilities in Nimbus
- Vertical specific application scenarios demonstrating relevance and value of the pervasive computing paradigm.
- Enhancing intelligence and improving proactive decision making capabilities of Nimbus
- Addressing the issues of security and privacy.

Most of the research work is aimed at improving the portfolio of capabilities along these dimensions. Some examples of the research projects are

- IRIS: Intentional Resource Information Service, an architecture enabling intention based resource discovery.
- Environment aware computing: This initiative is focusing on a specific aspect of pervasive computing and that is about environment awareness. Prototypes involve ability of mobile devices to be aware of other computing devices in the environment and using these device capabilities to execute tasks optimally.
- Application level network abstraction: Technology components that allow automatic detection and selection of the most appropriate underlying network connection (WLAN, GSM, GPRS Bluetooth) depending on the user and task context.
- Biometrics: Inclusion of voice tags, for execution of tasks in Nimbus.

7. Conclusion

Contextual continuity and user centered computing is essential to realize true mobility. The platform as outlined above can be manifested into multiple applications/solutions that can take advantage from the concept of task-aware/context aware computing. The applications of the platform are by no way only restricted to mobile device based applications. Without any doubt, mobility is a central construct in the platform. However, in certain cases, such as enterprise "collaboration" applications, mobility may be desired at the task execution level, without having the mobile devices as an essential part of the scope. Having mobile devices, though, as part of the computing infrastructure can greatly enhance the overall value of a solution/application because the concepts such as "user context carrier" can be effectively leveraged.

8.References

[1] Guanling Chen and David Kotz, "A Survey of Context-Aware Mobile Computing Research", *ACM Technical Report*: TR2000-381, 2000

[2] Z. Wang, D. Garlan, "Task-Driven Computing", *Technical Report CMU-CS-00-154*, Carnegie Mellon University School of Computer Science.

[3] Glenn Judd and Peter Steenkiste, "Providing Contextual Information to Pervasive Computing Applications", *Proceedings of the First IEEE International Conference on Pervasive Computing and Communications,* March 23-26, 2003

[4] Puneet Gupta, Deependra Moitra, "Evolving a Pervasive IT Infrastructure, A technology integration approach", *ACM Personal and Ubiquitous Computing, February 2004 Volume 8 Issue 1*

[5] David Garlan, Daniel P. Siewiorek, Asim Smailagic, Peter Steenkiste, "Project Aura: Towards Distraction-Free Pervasive Computing.", *IEEE Pervasive Computing*, 2002

[6] Joao Pedro Sousa, David Garlan, "Project Aura: An Architectural Framework for User Mobility in Ubiquitous Computing Environments", In *Proceeding of the 3rd Working IEEE/IFIP Conference on Software Architecture,* Montreal, 2002.

[7] Joao Pedro Sousa and David Garlan, "From Computers Everywhere to Tasks Anywhere: The Aura Approach", School of Computer Science, Carnegie Mellon University, http://www.cs.cmu.edu/~aura/

[8] Dag Johansen, Havard Johansen, Robbert van Renessem, "Environment Mobility- Moving the Desktop Around", *2nd Workshop on Middleware for Pervasive and Ad-Hoc Computing, ACM 2004*

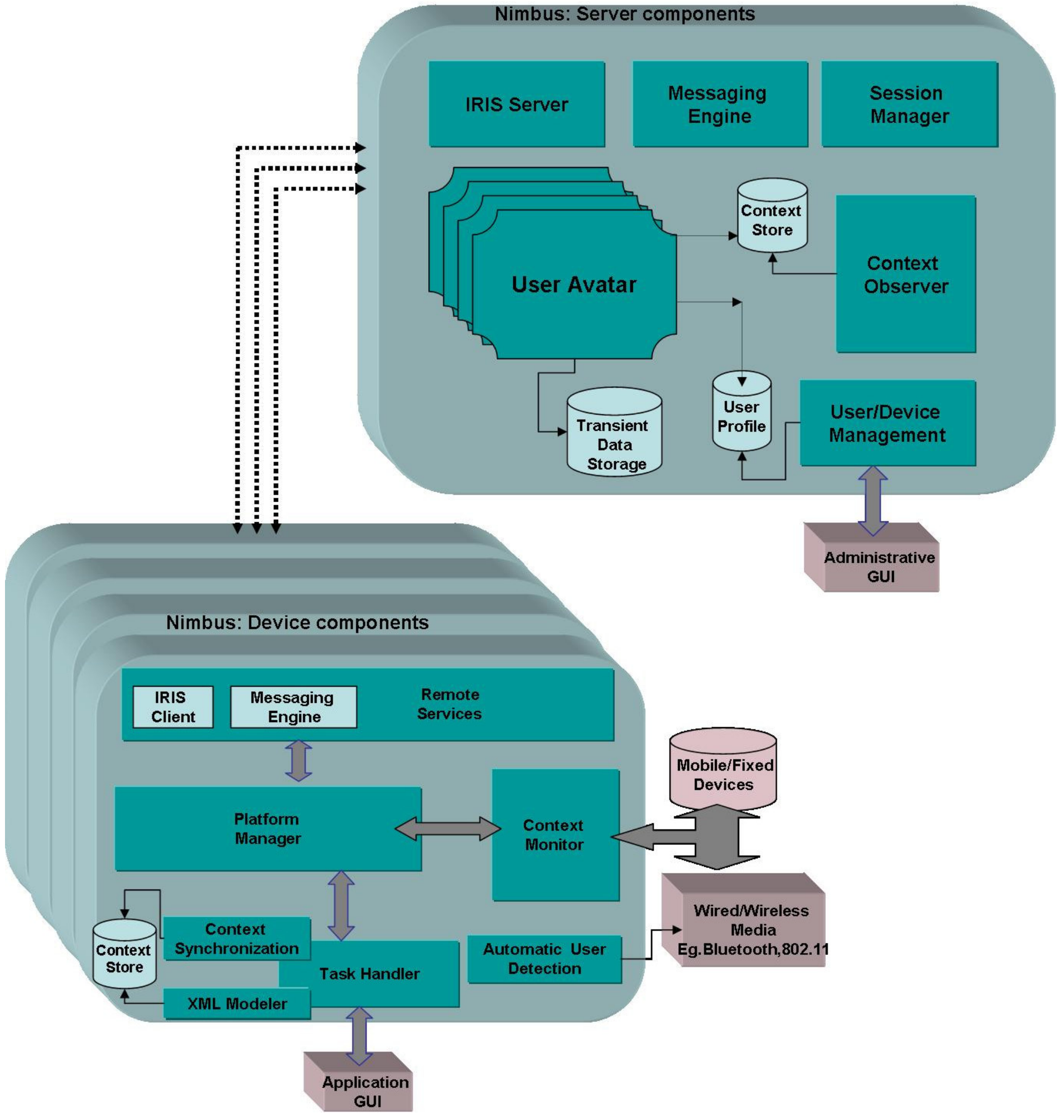
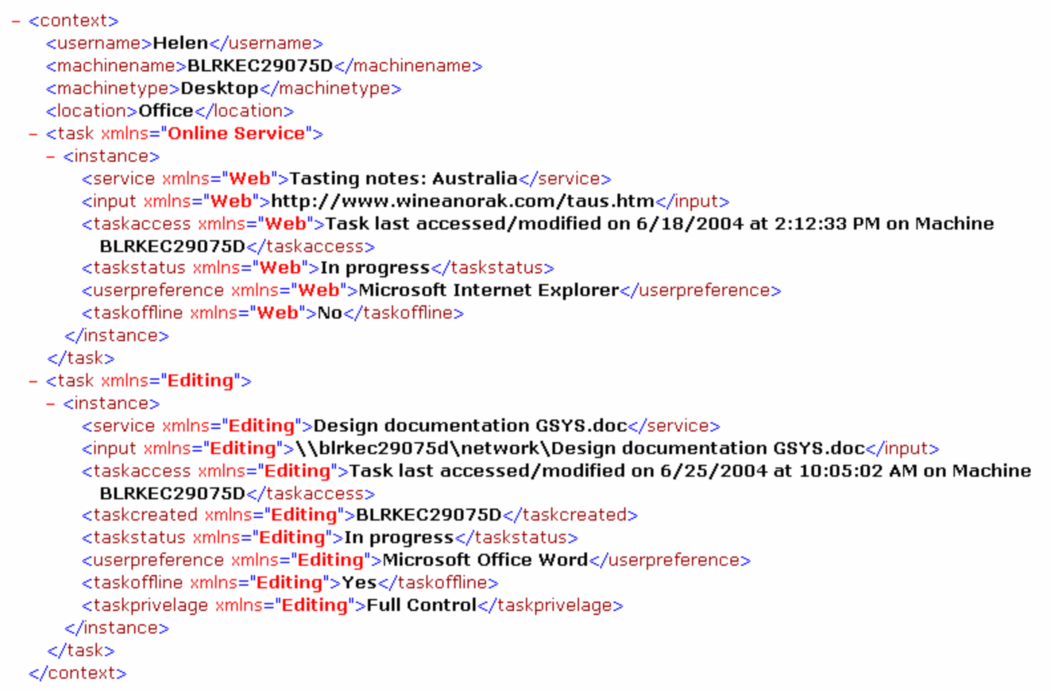


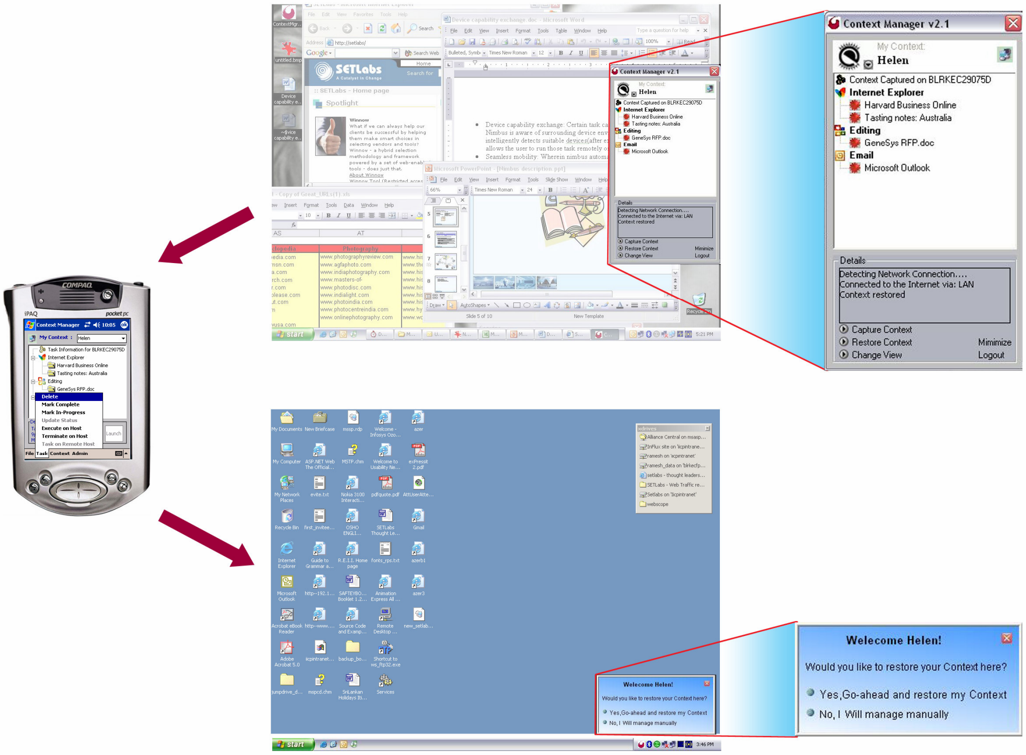
Figure 1. Nimbus- Key architectural components[3]

[3] *The term device in "Device components" in Figure 1 refers to any kind of computing device such as PCs, Laptops, PDAs, phones etc. Notice the one to many relationship between the"Nimbus server components" and multiple device components*



```
- <context>
    <username>Helen</username>
    <machinename>BLRKEC29075D</machinename>
    <machinetype>Desktop</machinetype>
    <location>Office</location>
  - <task xmlns="Online Service">
    - <instance>
        <service xmlns="Web">Tasting notes: Australia</service>
        <input xmlns="Web">http://www.wineanorak.com/taus.htm</input>
        <taskaccess xmlns="Web">Task last accessed/modified on 6/18/2004 at 2:12:33 PM on Machine
          BLRKEC29075D</taskaccess>
        <taskstatus xmlns="Web">In progress</taskstatus>
        <userpreference xmlns="Web">Microsoft Internet Explorer</userpreference>
        <taskoffline xmlns="Web">No</taskoffline>
      </instance>
    </task>
  - <task xmlns="Editing">
    - <instance>
        <service xmlns="Editing">Design documentation GSYS.doc</service>
        <input xmlns="Editing">\\blrkec29075d\network\Design documentation GSYS.doc</input>
        <taskaccess xmlns="Editing">Task last accessed/modified on 6/25/2004 at 10:05:02 AM on Machine
          BLRKEC29075D</taskaccess>
        <taskcreated xmlns="Editing">BLRKEC29075D</taskcreated>
        <taskstatus xmlns="Editing">In progress</taskstatus>
        <userpreference xmlns="Editing">Microsoft Office Word</userpreference>
        <taskoffline xmlns="Editing">Yes</taskoffline>
        <taskprivelage xmlns="Editing">Full Control</taskprivelage>
      </instance>
    </task>
  </context>
```

Figure 2. Nimbus- XML Task Representation.



Context Manager v2.1

My Context:
Helen

Context Captured on BLRKEC29075D
Internet Explorer
Harvard Business Online
Tasting notes: Australia
Editing
GeneSys RFP.doc
Email
Microsoft Outlook

Details
Detecting Network Connection....
Connected to the Internet via: LAN
Context restored

Capture Context
Restore Context
Change View
Minimize
Logout

Welcome Helen!

Would you like to restore your Context here?

● Yes,Go-ahead and restore my Context
● No, I Will manage manually

Figure 3. Snapshots of a Nimbus based application demonstrating Scenario 1. Here we see the seamless transition of user tasks as shown in the application GUI from one desktop (above) to another (below). The context carrier is a high-end PDA which allows the user to continue working on the same tasks from the device. Also seen below is the 'automatic user detection' welcome message.