CBMC4103
MOBILE COMPUTING FOR MULTIMEDIA

Vaibhav Shukla

OPEN UNIVERSITY MALAYSIA

Project Directors: *Prof Dato' Dr Mansor Fadzil*
Assoc Prof Dr Norlia T. Goolamally
Open University Malaysia

Module Writer: *Vaibhav Shukla*
Acme Learning Private Limited

Moderator: Acme Learning Private Limited in collaboration with
Open University Malaysia

Developed by: *Centre for Instructional Design and Technology*
Open University Malaysia

Printed by: Meteor Doc. Sdn. Bhd.
Lot 47-48, Jalan SR 1/9, Seksyen 9,
Jalan Serdang Raya, Taman Serdang Raya,
43300 Seri Kembangan, Selangor Darul Ehsan

First Edition, April 2015

Table of Contents

COURSE GUIDE

COURSE GUIDE DESCRIPTION

You must read this *Course Guide* carefully from the beginning to the end. It tells you briefly what the course is about and how you can work your way through the course material. It also suggests the amount of time you are likely to spend in order to complete the course successfully. Please keep on referring to the *Course Guide* as you go through the course material as it will help you to clarify important study components or points that you might miss or overlook.

INTRODUCTION

CBMC4103 Mobile Computing for Multimedia is one of the courses offered by the Faculty of Information Technology & Multimedia Communication at Open University Malaysia (OUM). This course is worth 3 credit hours and should be covered over 15 weeks.

COURSE AUDIENCE

This course is offered to students undertaking the Bachelor of Multimedia Technology.

As an open and distance learner, you should be acquainted with learning independently and being able to optimise the learning modes and environment available to you. Before you begin this course, please ensure that you have the right course material, and understand the course requirements as well as how the course is conducted.

STUDY SCHEDULE

It is a standard OUM practice that learners accumulate 40 study hours for every credit hour. As such, for a three-credit hour course, you are expected to spend 120 study hours. Table 1 gives an estimation of how the 120 study hours could be accumulated.

Table 1: Estimation of Time Accumulation of Study Hours

Study Activities	Study Hours
Briefly go through the course content and participate in initial discussion	3
Study the module	60
Attend 3 to 5 tutorial sessions	10
Online participation	12
Revision	15
Assignment(s), Test(s) and Examination(s)	20
TOTAL STUDY HOURS ACCUMULATED	**120**

COURSE OUTCOMES

By the end of this course, you should be able to:

1. Understand the concept of mobile computing, the condition and architecture of mobile software.
2. Familiar with mobile development frameworks such as Java, BREW, Windows CE, WAP, and Symbian.
3. Use XML as document and metadata format for mobile computing.
4. Implement UML as tool for mobile application development.
5. Understand the concept of user interface design for mobile computing.
6. Understand the concept of mobile agents and peer-to-peer architecture of mobile applications.
7. Understand the concept of wireless connectivity technologies for mobile applications.
8. Familiar with synchronization and replication of mobile data.
9. Know the security matters for mobile computing in every aspects of network capabilities and content of mobile applications.
10. Formulate the best design of mobile computing for multimedia contents.
11. Build multimedia contents for mobile computing using appropriate tools.

COURSE SYNOPSIS

This course prepares students with the knowledge of multimedia development for mobile computing. There are many approaches and tools can be used to develop multimedia application for mobile including UML and XML. Students are also exposed to the concepts of graphical user interfaces for mobile application, channels, mobile agents and wireless technology, architecture, design and technologies for mobile computing.

This course is divided into 12 topics. The synopsis for each topic can be listed as follows:

Topic 1 explains about the applications of mobile computing, dimensions of mobile computing, conditions of the mobile users as well as architecture of mobile software applications.

Topic 2 describes the fully centralised framework and tools, n-tier client-server frameworks and tools, windows CE, symbian EPOC, BREW, WAP, Java 2 Micro Edition (J2ME), publishing frameworks of mobile computing and mobile computing tools.

Topic 3 provides explanation about what is XML, key XML technologies for mobile computing, XML and UML, how to put XML to work and also gives a brief introduction to UML.

Topic 4 explains about the user interface development, building generic user interface, using UML for modelling generic user interface components, what are x forms?, how to put it all to work, role of WAP, role of J2ME and the role of BREW.

Topic 5 discusses about voice user interface and mobile applications, multichannel and multimodal user interfaces, software and system architectures for delivering multimodality as well as the evolving definition of multimodality.

Topic 6 explains about the mobile agent, mobile agents for mobile computing, UML extensions for mobile agents, applications for mobile agents and to mobile application and implementation tools, solving mobile application development problem with mobile agents, techniques for agent based software and mobile peer-to-peer computing (MOBI-DIK).

Topic 7 gives explanation about the wireless connectivity, quality of service (QoS), survey of wireless networking technologies, mobile IP and SMS.

Topic 8 describes about the taxonomy of replication and synchronisation, data replication and synchronisation for mobile application, Sync ML and Web Dav.

Topic 9 introduces data acquisition of location information, geographic information systems, building information modelling, location based service applied working, utilising location based services with mobile applications, representing location with UML, security and privacy of location information, internationalisation and localisation.

Topic 10 explains about active computing and wireless infrastructure, practical considerations of building active systems, mobile security in wireless networks, security and ad hoc networking technologies, security threats in ad hoc networks, location information, security and privacy, security – the unsolved problem for mobile agents, difference between privacy and security and modelling security with UML.

Topic 11 explains about the dimensions of mobility, architecture, design and technology selection for mobile applications.

Topic 12 presents voice user interface, problems on building location based services, usage of power, testing mobile applications and also a case study.

TEXT ARRANGEMENT GUIDE

Before you go through this module, it is important that you note the text arrangement. Understanding the text arrangement will help you to organise your study of this course in a more objective and effective way. Generally, the text arrangement for each topic is as follows:

Learning Outcomes: This section refers to what you should achieve after you have completely covered a topic. As you go through each topic, you should frequently refer to these learning outcomes. By doing this, you can continuously gauge your understanding of the topic.

Self-Check: This component of the module is inserted at strategic locations throughout the module. It may be inserted after one sub-section or a few sub-sections. It usually comes in the form of a question. When you come across this component, try to reflect on what you have already learnt thus far. By attempting to answer the question, you should be able to gauge how well you have understood the sub-section(s). Most of the time, the answers to the questions can be found directly from the module itself.

Activity: Like Self-Check, the Activity component is also placed at various locations or junctures throughout the module. This component may require you to solve questions, explore short case studies, or conduct an observation or research. It may even require you to evaluate a given scenario. When you come across an Activity, you should try to reflect on what you have gathered from the module and apply it to real situations. You should, at the same time, engage yourself in higher order thinking where you might be required to analyse, synthesise and evaluate instead of only having to recall and define.

Summary: You will find this component at the end of each topic. This component helps you to recap the whole topic. By going through the summary, you should be able to gauge your knowledge retention level. Should you find points in the summary that you do not fully understand, it would be a good idea for you to revisit the details in the module.

Key Terms: This component can be found at the end of each topic. You should go through this component to remind yourself of important terms or jargon used throughout the module. Should you find terms here that you are not able to explain, you should look for the terms in the module.

References: The References section is where a list of relevant and useful textbooks, journals, articles, electronic contents or sources can be found. The list can appear in a few locations such as in the *Course Guide* (at the References section), at the end of every topic or at the back of the module. You are encouraged to read or refer to the suggested sources to obtain the additional information needed and to enhance your overall understanding of the course.

PRIOR KNOWLEDGE

No prior knowledge required.

ASSESSMENT METHOD

Please refer to myVLE.

REFERENCES

B'Far R. (2005). *Mobile computing principles: Designing and developing mobile applications with UML and XML*. USA: Cambridge University Press.

Ricky Kwok Y. K., & Vincent K. N. Lau. (2007). *Wireless Internet and mobile computing: Interoperability and performance* (Information and Communication Technology Series,). Wiley-IEEE Press.

Reyes, A. et al. (2001). *Flash design for mobile devices*. Wiley Publishing.

TAN SRI DR ABDULLAH SANUSI (TSDAS) DIGITAL LIBRARY

The TSDAS Digital Library has a wide range of print and online resources for the use of its learners. This comprehensive digital library, which is accessible through the OUM portal, provides access to more than 30 online databases comprising e-journals, e-theses, e-books and more. Examples of databases available are EBSCOhost, ProQuest, SpringerLink, Books24x7, InfoSci Books, Emerald Management Plus and Ebrary Electronic Books. As an OUM learner, you are encouraged to make full use of the resources available through this library.

Topic 1 ▶ Introduction to Mobile Computing

LEARNING OUTCOMES

By the end of this topic, you should be able to:

1. Define mobile computing;
2. Recognise the added dimensions in mobile computing;
3. Underline the conditions of mobile users; and
4. Identify the architecture of mobile software applications.

▶ INTRODUCTION

The proliferation of wireless networks and small movable computing devices has headed to the growth of the mobile computing paradigm. Mobile and nomadic users carrying laptops or hand-held computers are able to connect to the Internet through publicly available wireline or wireless networks. In the near future, this trend can only grow as new exciting services and infrastructure delivering wireless voice and multimedia data will be deployed.

What is Mobile Computing?

Mobile Computing is a technology that allows transmission of data, via a computer, without having to be connected to a fixed physical link. Mobile data communication has become a very important and rapidly evolving technology. It allows users to transmit data from remote locations to other remote or fixed locations. This proves to be the solution to the biggest problem of business people on the move – mobility. Mobile voice communication is widely established throughout the world and has had a very rapid increase in the number of subscribers to the various cellular networks over the last few years. An extension of this technology is the ability to send and receive data across

these cellular networks. This is the principle of mobile computing. In the last decade, we have made it possible to increase the complexity of mobile applications and services provided to end-users. This is done through the popularity and evolution of mobile devices like laptops, mobile phones and Personal Digital Assistants (PDA), and the evolution of fast mobile networks. It is also a spectacular growth in multimedia communication especially via the World Wide Web. This chapter explores some of the current technologies of mobile devices, mobile networks and multimedia systems. It is based on the exploration outline of some issues for design and development of mobile multimedia systems in Mobile Communication System. The various generation mobile communication systems will combine standardised streaming with a range of unique services to provide high-quality content (Multimedia) that meets the specific needs of the rapidly growing mobile market. Mobile computing is expanding in 4 dimensions as shown in Figure 1.1.

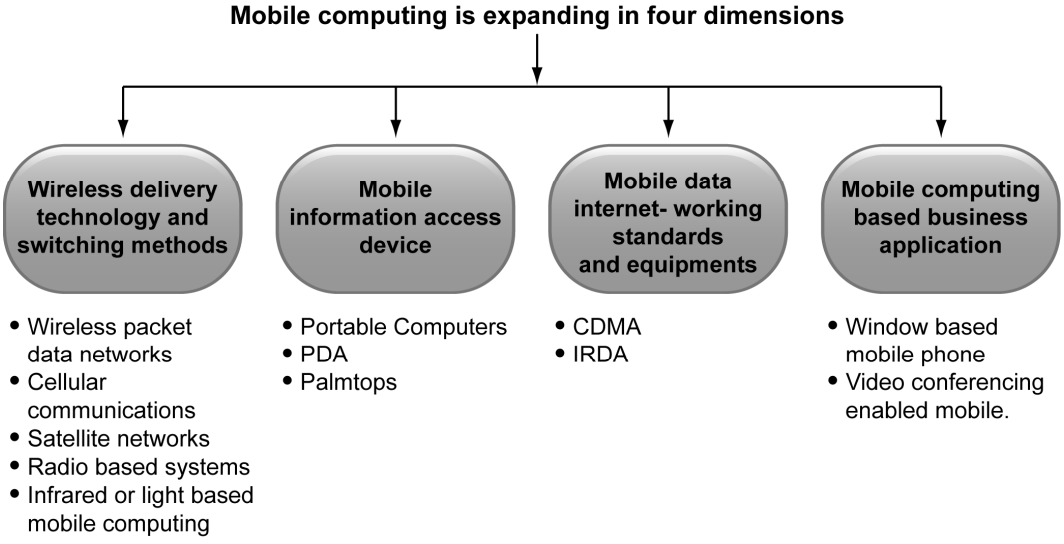


Figure 1.1: Mobile computing is expanding in four dimensions

1.1 APPLICATIONS OF MOBILE COMPUTING

The question that always arises when a businessman thinks of buying a mobile computer is "Will it be worth it?"

In many fields of work, the ability to keep on the move is vital in order to utilise time efficiently. Efficient utilisation of resources (i.e., staff) can mean substantial savings in transportation costs and other non quantifiable costs such as increased

customer attention, impact of on site maintenance and improved inter-communication within the business.

The importance of Mobile Computers has been highlighted in many fields of which a few are described below:

(a) **For Estate Agents**

The working of an estate agent is difficult as they have to either work from home or on field. With the use of mobile computers they can be more productive. They can enquire about the current real estate information by accessing multiple listing services, which they can do from home, office or in while driving their car, when out with clients. They can provide clients with immediate feed back regarding specific homes or neighbourhoods. They can get faster loan approvals, since the applications can be submitted on the spot. Therefore, mobile computers allow them to devote more time to clients.

(b) **Emergency Services**

The mobile computer has the ability to receive information on the move is vital where emergency services are involved. Information regarding the address and other details of an incident can be sent quickly, via a CDPD system using mobile computers, to one or several appropriate mobile units which are in the vicinity of the incident.

(c) **In Courts**

Defence counsels have the authority to carry their mobile computers to court. When the opposing counsel references a case which they are not familiar of, they can use the computer to get direct, real-time access to online legal database services, where they can gather information on the case and related precedents. Therefore, mobile computers allow immediate access to a wealth of information, making people better informed and prepared.

(d) **In Companies**

The mobile computer is much useful for companies as managers can use mobile computers in important presentations to major customers. At a small recess, they can revise the presentation to take advantage of this information. They can communicate with the office about possible new offers and call meetings for discussing responds to the new proposals. The user can also access the latest market share information at anytime and anywhere. Therefore, mobile computers can influence competitive advantages.

(e) **Credit Card Verification**
In a shopping complex when the customer uses a credit card for the payment, the inter-communication required between the bank central computer and the POS terminal, in order to effect verification of the card usage, can take place quickly and securely over cellular channels using a mobile computer unit. This can speed up the transaction process and relieve congestion at the POS terminals.

(f) **Taxi/Truck Dispatch**
Using the idea of a centrally controlled dispatcher with several mobile units (taxis), mobile computing allows the taxis to be given full details of the dispatched job as well as allowing the taxis to communicate information about their whereabouts back to the central dispatch office. This system is also extremely useful to secure deliveries i.e., Securicor. This allows a central computer to be able to track and receive status information from all of its mobile secure delivery vans. Again, the security and reliability properties of the CDPD system shine through.

1.1.1 The Future of Mobile Computing

Today's environment is the computer environment and the technologies are developed in this technological advancements in Artificial Intelligence, Integrated Circuitry and increases in Computer Processor speeds, the future of mobile computing looks bright.

With the emphasis increasingly on compact small mobile computers, it would be possible to have all the practicality of a mobile computer in the size of a hand held organiser.

Use of Artificial Intelligence may allow mobile units to be the ultimate in personal secretaries, which can receive emails and sending messages, understand what they are about, and change the individual's personal schedule according to the message. This can then be checked by the individual to plan his/her day.

With the majority of people working from home rather than commuting, their lifestyle can change. This may be beneficial to the environment as less transportation will be utilised. This mobility aspect may be carried further, even in social spheres, people will interact via mobile stations, discard the need to step outside of the house.

This scary concept of a world full of inanimate zombies sitting, locked to their mobile stations, accessing every sphere of their lives via the computer screen becomes ever more real as technology, especially in the field of mobile data communications, rapidly improves as trends are very much towards mobile computing.

1.1.2 Latest Technologies Used in Mobile Computing

There are various technologies we use in the case of mobile computing some of them are:

(a) **Bluetooth**
Bluetooth is an exciting new technology that allows devises such as mobile phones, laptops computers, digital cameras, PDAS and other portable devices to communicate with each other without using a cable to connect them. The Bluetooth is a technology by which communication between two devices is possible by using short range radio waves.

(b) **DRM**
The DRM is a short-term for Digital Right Management. This is a generic term for access control technology that can be used by the hardware manufacturers and individuals to limit the usage of digital content and devices.

(c) **Short Message Service**
The SMS allows the transmissions of messages up to 160 characters from one device to another device.

(d) **Multimedia Message Service**
The MMS service improves the use of the SMS service. By using the MMS service we can send text as well as a picture, an audio or video. Multimedia Messaging Service (MMS) is a store for forward messaging service that allows mobile subscribers to exchange multimedia messages with other mobile subscribers. As such it can be seen as an evolution of SMS, with MMS supporting the transmission of additional media types:

(i) Text

(ii) Picture

(iii) Audio

(iv) Video

(v) Combinations of the above.

(e) **GPS**

GPS is terms as the Global Positioning System. Basically this is a technology that supports a system which allows us to figure out the location of our destination.

(f) **GSM**

The GSM refers to the Global System for Mobile Communication. This is a digital transmission technique that is widely adopted in Europe and supported in North America. GSM uses 900 MHz and 1800 MHz in Europe.

(g) **CDMA**

Code Division Multiple Access is one of the several digital wireless transmission methods in which signals are encoded using a pseudo random sequence which corresponds to a different communication channel that the receiver also knows and can use to decode the received signal.

(h) **GPRS**

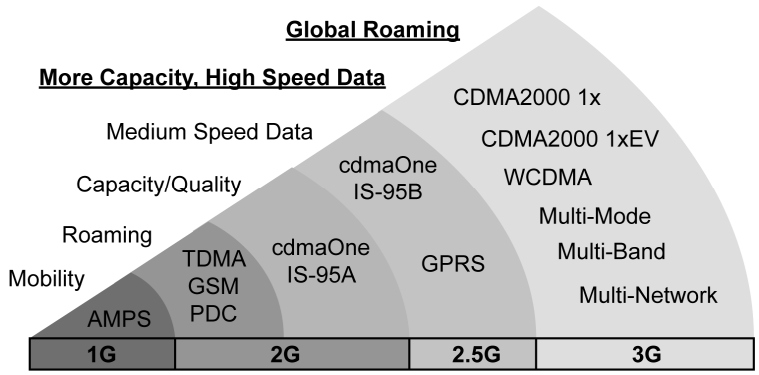
The GPRS refers to the general packet Radio Service. It provides the packet mode transfer for applications that show traffic patterns such as frequent transmission of small volumes or infrequent transmission of small or medium volumes according to the requirement specification.

(i) **WAP**

WAP refers to the wireless application protocol it is a protocol that is developed to allow intelligent transmission of optimised internet content to wireless phones.

(j) **Generations of Mobile Phones**

There are various generations of mobile phones that define the technology improvement in the present day's mobile computing environment.



Source: http://technologycrest.com/wp-content/uploads/2009/05/cdmacdma2000.gif

To name some of the technologies:

(a) **0G** – This is defined as the zeroth generation of mobile telephony technology. This refers to the pre-cell phone mobile telephony technology, such as radio phones. This is frequently used in cars. This generation was launched in the early 1970s in Finland.

(b) **1G** – 1G refers to the first generation wireless telephone technology. This technology was introduced in the early of 80s and refers as the analogue cell phones standards this technology was used until it was replaced by 2G digital phones.

(c) **2G** – The 2G refers to the second generation wireless telephony technology. The 2G services are frequently referred as Personal Communication Service or PCS in the US. The 2G technology was divided in to the TDMA based and CDMA based standards depending on the type of multiplexing. It cannot normally transfer data, such as email or software, other than the digital voice calls itself, and other basic ancillary data such as time and date. Nevertheless, SMS messaging is also available as a form of data transmission for some standards.

(d) **2.5G** – The concept of the 2.5G technology came in existence after development of 2G technologies and before development of 3G technology. In the 2.5G technology we use the concept of Packet switch domain as well as the circuit switch domain. One of the commonly known 2.5G technologies is GPRS.

(e) **3G** – The 3G technology is frequently used in the present day's mobile environment. This service provides the ability to transfer two voices, (telephone call and non voice data) such as downloading information, exchanging Emails, and instant messaging.

1.1.3 Platforms

Mobile solutions on various kind of devices like Window CE, Symbian OS, IOS, etc. and Built the complete expertise in the following Operating System platforms:

(a) **Microsoft Windows CE**
One of the most frequently used operating system in mobile development is Windows CE or Windows Embedded Compact operating system developed by Microsoft for the use of hand held devices such as Pocket PC and other equipment.

(b) **Symbian OS**
The Symbian Operating System is developed by the Symbian a software licensing company. This is the global open industry standard Operating system for advanced, data-enabled mobile phones.

(c) **IOS**
The IOS is an official operating system that is being used in the iPod, iPod touch and iPhones. The IOS are commonly referred to the IOS (Apple). This is used in the iPhones very frequently. There are Cisco IOS which is originally internetwork OS.

(d) **Android**
Android is an operating system developed by Google and is based upon the Linux kernel and GNU software Android is used for mobile devices such as mobile phones, tablet computers and note books. It was initially developed by Android Inc. (a firm purchased by Google) and later positioned in the Open Handset Alliance according to NPD Group, unit sales for Android OS Smartphone ranked first among all smart phone OS handsets sold in the US in the second quarter of 2010, at 33%. BlackBerry OS is second at 28%, and IOS is ranked third with 22%.

(e) **Web OS**
Web OS is an operating system which is being used to provide OS services to wide area applications, including mechanisms for resource discovery, a global name space, remote process execution, recourses management authentication and security. On a single machine, an application developer can rely on the local operating system to provide these abstractions. In the wide area, however, application developers are forced to build these abstractions themselves or to do without. To address these problems Web OS provides basic operating system services needed to build applications that are geographical distributed, highly available and dynamically reconfiguring.

1.1.4 Mobile Architecture Overview

There are three major components to a mobile architecture (Figure 1.2):

(a) A middleware application;

(b) An existing system; and

(c) A handheld application.

Figure 1.2: Mobile architecture

The reason a middleware application is usually needed to provide data transformation, apply business logic, and be a central point of communication for the devices. If a new business system is being developed or rewritten then no middleware may be necessary; the appropriate logic can be built into the system to communicate with the devices from the start. However, most business systems are not rewritten very often and it is economically impractical to rewrite them just to 'mobilise' them. Furthermore a middleware server may also serve as a configuration management server.

SELF-CHECK 1.1

1. The main benefit of mobile computers is that you do not have to bind yourself to a certain place. You can do your work while sitting in a car or a train. You can communicate with other people while sitting anywhere in the world. You can chat online with your friends and family members while sitting on a beach. You can do your office work while sitting anywhere.

   (a) Can you think of some other benefits of mobile computing?

   (b) Explain mobile computing in your own words.

2. When people can do their work while sitting anywhere they will do more work. This will play an important role in the economy of the country and the world.

SELF-CHECK 1.2

Choose one of the following options:

1. Mobile Computing is a technology that allows transmission of data, via .....
   (a) Computer
   (b) Internet

2. Use of ..... may allow mobile units to receive and send emails messages, understand whatthey are about, and change the individual's personal schedule according to the message.
   (a) Artificial intelligence
   (b) Coding

3. ..... is a device which allows devices to communicate with each other without using a cable to connect them.
   (a) MMS
   (b) Bluetooth

4. DRM stands for.....
   (a) Data Right Management
   (b) Digital Right Management

5. The SMS allows the transmissions of messages up to..... characters from one device to another device.
   (a) 160
   (b) 180

6. ..... is a technology allows us to figure out the location of our destination.
   (a) GPS
   (b) GSM

7. GPRS stands for.....
   (a) General Packet Radio Switching
   (b) General Packet Radio Service

(i) Throw some light on the generations of the mobile phones.

(ii) What is the difference between IOS and web OS?

1.2 DIMENSIONS OF MOBILE COMPUTING

The following six points are the dimensions of mobile computing. It is very necessary that the reader understands these dimensions of mobile computing and keeps them in mind throughout the process of design and implementation of the mobile application. Too often, engineers begin with a lot of attention to design and get bogged down in details of the tools that they use and small focused problems within the bigger picture of the system, its design, and its architecture.

The dimensions of mobility are as follows:

(a) **Location Awareness**
The definition of the mobile says the first dimension of mobile computing is the location. As the mobile devises do not work in a particular location the location of the mobile devices may changes in every moment of time so the primary design issue is the location management in the development of mobile devices.

(b) **Network Connectivity Quality of Service (QOS)**
The next issue is related to the network issue. The company focuses that our device is easily compatible for any type of network. The QOS is defined as the "The collective effort of service performances, which determine the degree of satisfaction of a user of this service". The services can have qualitative and quantitative.

(c) **Limited Device Capabilities (Particularly Storage and CPU)**
The important issue is the storage related issues. The company's concern should be regarding the storage capacity of our device and to improve the storage capacity of a mobile.

(d) **Limited Power Supply**
The mobile devices are totally based on battery power which provides less power supply.

(e) **Platform Proliferation**
We need to design such type of device which would support any platform and any network.

1.3 CONDITIONS OF THE MOBILE USERS

There are the billions of mobile users today. Some of them are core users and some of them are native users. As far as mobile application in multimedia is concern there are a few conditions. Wireless networking technologies, as well as the widespread use of mobile devices such as PDAs, cell phones, and laptops, make pervasive computing a reality. Many applications such as email applications can now successfully run in mobile wireless networks. However, there are still many challenges in moving multimedia applications.

There are a few conditions involved for mobile users in terms of multimedia application:

(a) **Delivery of the Content**
The low bandwidth of the wireless network many times creates limitations for the mobile users when they send content in different formats; like text, images, graphics, animations, voice and video.

In the 2G mobile services mobile networks with bandwidth of less than 28.8kbps known as GSM and in the 3G the 2mbps speed is achieved through the UMTS. But some high bandwidth data transfer creates problem. Main problem conditions are:

(i) Less bandwidth, higher transmission latency.

(ii) Time varying error characteristics (high pocket loss reliability).

(iii) Channel capacity

So these are the basic condition that limits the user to deliver the required content.

(b) **Mobility**
Mobility is another arrangement for mobile users. When the user's movement change, the distance between the base station and the mobile host causes disturbance in the wireless network. This means when the user

moves from its home network area the problem of information loss arises. Following are the conditions for mobility:

(i) Rapid and radical changes in available resources capacity such as bandwidth.

(ii) Frequent topology changes.

(iii) Unexpected delays, Packet loss or the loss of service.

(c) **Multimedia**

There are sometimes the multimedia conditions for the mobile users. The application starts hanging when there is increase in the memory storage. A Break in transmission causes connection loss.

(d) **Synchronisation**

Another cause is the synchronisation for the mobile user's means to connect the devices with other devices it creates the problem of synchronisation. Many multimedia application mobiles like 3G and 4G causes the problem with lower version of the mobile software's.

(e) **Access Point Requirement**

User of mobile software application requires the access point to access the internet. The access point is the location setup settings to connect the internet so without the access point user cannot connect to the internet. Sometimes this access point may also be the cause of hacking of information.

(f) **Disclosure of the Information**

For the mobile user it is mandatory to provide his details to the network service provider company. This information includes his/her verification and valid information which is submitted to the provider and finally this information is sent to the government for the security reasons.

(g) **Data Access**

Data access on a mobile device is constrained by unreliable network connections and the hardware constraints of the device itself. When designing data access, consider how low bandwidth, high latency, and intermittent connectivity will impact your design.

(h) **Configuration Management**

When designing, device configuration management, consider how to handle device resets, as well as whether you want to allow configuration of your application over the air or from a host PC.

(i) **Exception Management**
Designing an effective exception-management strategy is important for the security and reliability of your application. Good exception handling in your mobile application prevents sensitive exception details from being revealed to the user, improves application robustness, and helps to avoid your application being left in an inconsistent state in the event of an error.

(j) **Power**
Power is the most limiting factor for a mobile device. Decisions for design should take into account how much power the device consumes and its effect on overall battery life. If you have a choice in devices, consider devices that can draw power from Universal Serial Bus (USB) or other types of data hook-ups.

ACTIVITY 1.1

1. Disscuss about mobile banking.
2. Discuss the WCDMA Mobile Internet in High-Mobility Environment.

SELF-CHECK 1.3

1. Give two reasons why mobile computing is important in the present day computer environment.
2. Explain different conditions for mobile users.

1.4 ARCHITECTURE OF MOBILE SOFTWARE APPLICATIONS

A mobile software application will normally be structured as a multi-layered application consisting of user experience, business, and data layers. When developing a mobile software application, we may choose to develop a thin web-based client or a rich client. If we are building a rich client, the business and data services layers are likely to be located on the device itself. If we are building a thin client, the business and data layers will be located on the server. Figure 1.3 illustrates common rich client mobile software application architecture with components grouped by areas of concern.

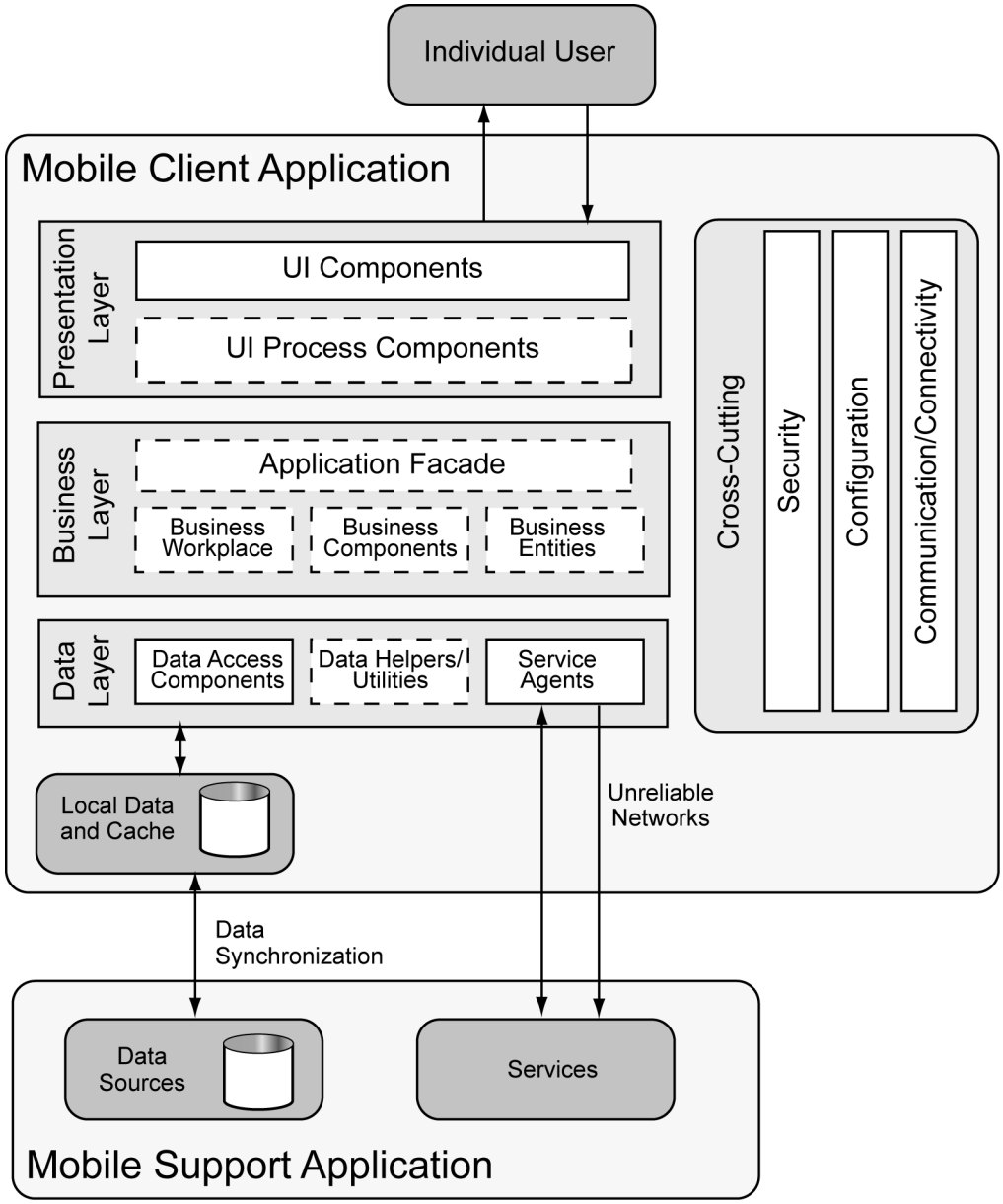


Figure 1.3: Architecture of mobile software applications

1.4.1 Design Considerations

The following design guidelines provide information about different aspects you should consider when designing a mobile application. Follow these guidelines to ensure that your application meets your requirements and performs efficiently in scenarios common to mobile applications:

(a) **Decide if you will build a rich client, a thin Web client, or Rich Internet Application (RIA)**

This is very complex to install and maintain a rich client application. If your application requires local processing and must work in an occasionally connected scenario, consider designing a rich client. We design a thin client if our application can depend on server processing and will always be fully connected. If your application requires a rich User Interface (UI), only limited access to local resources, and must be portable to other platforms, design an RIA client.

(b) **Determine the device types you will support**

Now the second issue is of device type that shows which type of device support our application in the device type selection we consider the following issues we consider screen size, resolution (DPI), CPU performance characteristics, memory and storage space, and development tool environment availability. In addition, factor in user requirements and organisational constraints. You may require specific hardware such as a Global Positioning System (GPS) or a camera, which may impact not only your application type, but also your device choice.

(c) **Design considering occasionally connected, limited-bandwidth scenarios when required**

There are two type of network connectivity the one is the occasionally connected and one is permanently connected. Now if we use mobile device which is a standalone device, we will not need to account for connection issues. When network connectivity is required, mobile applications should handle cases when a network connection is intermittent or not available. It is vital in this case to design your caching, state management, and data-access mechanisms with intermittent network connectivity in mind. Batch communications for times of connectivity. Choose hardware and software protocols based on speed, power consumption, and "chattiness", and not just on ease of programming.

(d) **Design a UI appropriate for mobile devices, taking into account platform constraints**

Mobile devices require a simpler architecture, simpler UI, and other specific design decisions in order to work within the constraints imposed by the device hardware. Keeping these constraints in mind, design specifically for the device instead of trying to reuse the architecture or UI from a desktop or Web application. The main constraints are memory, battery life, ability to adapt to difference screen sizes and orientations, security, and network bandwidth.

(e) **Design a layered architecture appropriate for mobile devices that improves reuse and maintainability**

The multiple layers may be located on the device itself it depends on the application type. To maximise the concept of separation of concerns, and to improve reuse and maintainability for your mobile application we the concept of layers. However, aim to achieve the smallest footprint on the device by simplifying your design compared to a desktop or Web application.

EXERCISE 1.1

1. Explain the data communication.

2. Explain the basic mobile computing architecture by using a suitable diagram.

SUMMARY

- This topic shows that the GSM is the most important second generation digital cellular network. GSM was primarily designed for voice transmission. To avoid even higher implementation cost UMTS tries to reuse as much infrastructure as possible while introducing new services and higher data rates based on CDMA technologies. The initial installation will basically use the GSM/GPRS infrastructure and offer only moderate rates. Today CDMA providers are making more realistic estimates of around five times as many users. In future the GSM becomes the most successful 2G mobile communication system.
- There are various applications of the distributed system technologies. The use of distributed system technology to enable users who are not fixed in a single physical position to communicate with computers which form part of a network; more often than not these computers act as some form of server. This is one of the major development areas in distributed computing, with many manufacturers attempting to embed the same functions found in a normal computer into hand-held devices such as mobile phones.

KEY TERMS

AMPS	Advanced Mobile Phone Services
BS	Base Station
CDPD	Cellular Digital Packet Data
DCS	Digital Communication System
GSM	Global System Mobile
MSC	Mobile Switching Centre
NMT	Nordic Mobile Telephone
PCS	Personal Communication System
PSTN	Public Switched Telephone Network

SELF-TEST

1. How is data communication related to mobile computing?
2. What are the applications of mobile computing?
3. Define in your own words the mobile software application architecture.
4. Define in brief about Bluetooth.
5. Write short note on interpretation and active transation.

Topic 2 ▶ Introduction to Mobile Development Frameworks and Tools

LEARNING OUTCOMES

By the end of this topic, you should be able to:

1. Describe fully centralized frameworks and tools;
2. Define N-tier client server architecture;
3. State the symbian operating system for mobile computing;
4. Underline the concept of BREW, WAP, and window CE;
5. Describe the publishing framework of mobile computing; and
6. List mobile computing tools and its uses.

▶ INTRODUCTION

The path-breaking technology of the 90s, the Internet, has heralded a dramatic transformation in the business landscape and touched the life pattern of common people. Closely following it, the wireless Internet, or more specifically, mobile computing, is having a profound impact on our lives now. Unlike most technologies whose impact was first felt by the developed world, Internet and mobile computing are benefiting the developing countries by jump-starting them into the digital economy. In spite of the infrastructure and bandwidth limitations, developing nations, like Malaysia, are making the most of their early starter's advantage to catch up with the developed countries. A new technology's

successful adoption often depends on its development tools. Good tools help new developers more easily get started and make experienced developers more productive. For instance, the success of Microsoft's programming environment is closely associated with the success of its Visual Studio tools.

2.1 FULLY CENTRALIZED FRAMEWORK AND TOOLS

Mobile development framework and tools includes fully centralized framework and tools and client server framework and tools which have been discussed below.

Fully centralized framework and tools is defined by the following points:

(a) Fully centralized framework and tools involves designed clients;

(b) This framework is implanted in nature; and

(c) This framework is designed to do only one thing.

Fully centralized framework and tools is applied on the following:

(a) QOS;

(b) Restricted power supply;

(c) Active communications; and

(d) Location awareness.

Also there are some platforms on which fully centralized framework do not apply, which are defined below:

(a) Platform propagation;

(b) Restricted device capabilities; and

(c) Support for multiplicity of user interfaces.

Examples of which fully centralized framework

(a) Call centers;

(b) Battlefield systems; and

(c) Grocery store.

2.2 N-TIER CLIENT-SERVER FRAMEWORKS AND TOOLS

We can use a number of web applications. All the web applications follow the N-Tier client-server architectures. The client server architecture has an application server, a large number of clients, and a database. N-Tier architecture is a Client-Server architecture combined with the Layered architecture (Figure 2.1). The reason of combining Client-Server and N-Tier here is because they are very much related.

In simple definition we can say that Client-Server system is one in which the server performs some kind of service that is used by many clients. The clients take the lead in the communication. The basic Client-Server architecture has 2 tiers (Client and Server). We will basically explain the 3-tier architecture here, which is an extension to the 2-tier architecture.

The first layer in the client server architecture is the presentation tier (Figure 2.1), the client or front-end, deals with the interaction with the user. Usually, there can be any number of clients which can all access the server at the same time. Currently the clients are mostly thin clients, which mean they do not contain a lot of application code (in contrast to fat clients). Clients process user input, send requests to the server, and show the results of these requests to the user. A common client is made up of a number of dynamic HTML pages that one can access with a web browser.

In the client server architecture the second, layer is of application tier, the server, or the back-end, or middleware, processes the requests of all clients. It is the actual web application that performs all functionality specific to the web application. However, it does not store the persistent data itself. Whenever, it needs data of any importance, it contacts the database server.

The third or database tier contains the database management system that manages all persistent data.

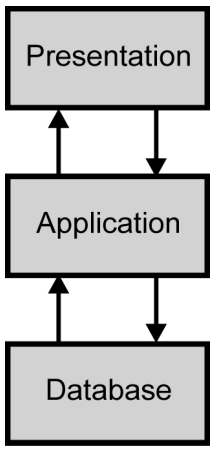


Figure 2.1: Tier client server architecture

In client server architecture there are multiple clients. That is what client-server computing is all about. However, in the second and third tier there can also be multiple instances of the same application. If this is the case, it is because of scalability, load-balancing and redundancy. Which means the system is important, so let us add extra equipment that does the same thing. This makes the server a very powerful system, but also introduces synchronisation problems.

Example: In the web application we use the three tier client server architecture in the web applications the first tier is the presentation-tier, the second tier is the application tier and the third tier is the database tier.

2.2.1 Where Does it Come From?

At the advance multitasking operating systems in the 1960s, it became possible to access a single computer (the server) from different terminals (clients). However, the distance between the clients and the server became bigger and the number of clients increased. At that time the application and database tiers were still integrated. It is called client-server computing.

In the 1990s, the booming of the Internet and e-commerce, and the architecture became much important, and much time and money was invested in it. As other good architectures have shown, it is a good idea to separate the application code from the data. This principle was applied to the client-server architecture. Companies created application servers to ease the creation of web applications.

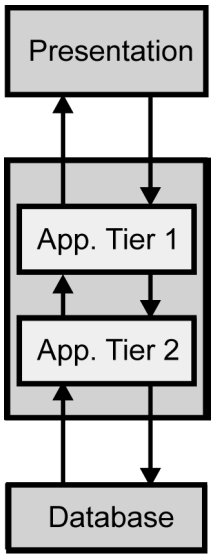


Figure 2.2: N-Tier Client-server architecture

The N-tier client server architecture is nothing new it is really 3 tier architectures in which the middle tier is split into new tiers. In the N-tier architecture the application layer is split into separate parts. These parts differ from system to system. Figure 2.2 shows it, where application layer is split into application tier 1 and application tier 2.

2.2.2 Client Server Architecture of Mobile Phones

The concept of client/server was first used in 1980s in reference to the Personal Computer (PCs) on the network. The actual client/server model started gaining acceptance in the late 1980s. The client/server software architecture is a versatile, message-based and modular infrastructure that is intended to improve usability, flexibility, interoperability, and scalability as compared to centralised, mainframe, time sharing computing.

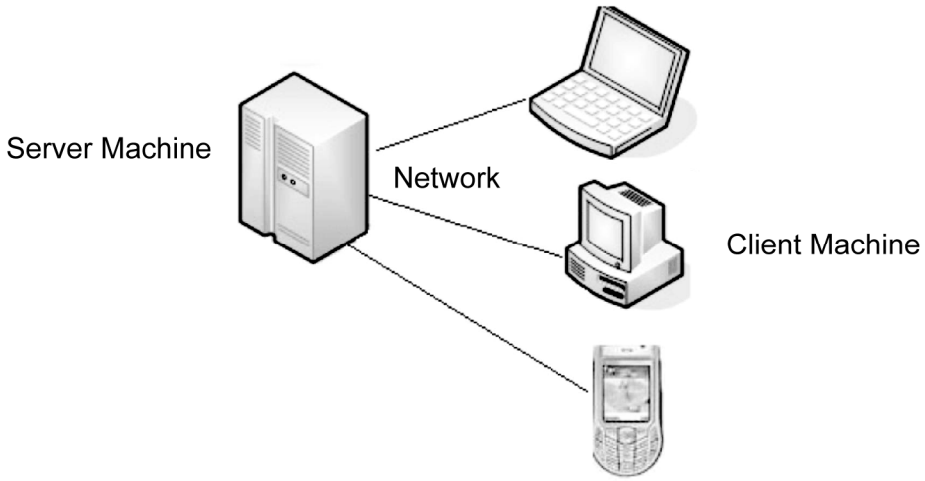


Figure 2.3: Client server architecture

As shown in Figure 2.3 in the client server architecture there is one client computer and on the server computer in which the client sends a service request to the server and the server gives the response to the client. In simple words we can say Client server architecture is architecture of a computer network in which many clients (Remote processors) request and receive service from a centralised server (host Computer). Client computers provide an interface to allow a computer user to request services of the server and to display the results the server returns. Servers wait for requests to arrive from clients and then respond to them. Ideally, a server provides a Standardised transparent interface to clients so that clients need not be aware of the specifics of the system (i.e., the hardware and software) that is providing the service.

Today, clients are often situated at workstations or on personal computers, while servers are located elsewhere on the network, usually on more powerful machines. This computing model is especially effective when clients and the server each have distinct tasks that they routinely perform. In hospital data processing, for example, a client computer can be running an application program for entering patient information while the server computer is running

another program that manages the database in which the Information is permanently stored. Many clients can access the server's information simultaneously, and, at the same time, a client computer can perform other tasks, such as sending e-mail. Because both client and server computers are considered intelligent devices, the client-server model is completely different from the old "mainframe" model, which utilised a centralised mainframe computer that performed all the tasks for its associated "dumb" terminals.

2.3 WINDOWS CE

One of the most frequently used operating system in the mobile development is Windows CE or Windows Embedded Compact operating system developed by Microsoft for use with hand held devices such as the Pocket PC and other equipment. Like Linux, Windows the Windows CE (CE stands for embedded compact) is an operating system, which are being developed for using in the mobile devises. However, it is used only in portable systems like cell phones or PDAs. It is ideal for embedded systems, where memory is a constraint and processors Windows CE is an operating system adopted by small devices and based on window 95. The development for this operating system under the code name Pegasus began in 1995. Specially designed for micro-computers, the abbreviation CE stands informal for "Compact Edition". These microcomputers are known as handheld computer or Personal Digital Assistant (PDA). The first version of Windows CE requires as a minimum 4 MB of ROM, 2 MB of RAM and a processor of the SuperH3, MIPS 3000 or MIPS 4000 architecture. One of the first devices for Windows CE 1.0 which was the HP 300 LX came on the market on 16th November 1996. The resolution of the touch screen is 640 240 pixels and corresponds to the half-VGA resolution. For synchronising data between mobile device and desktop computer, the software "Handheld PC Explorer" is used.

Field of Application:

(a) Handhelds and similar mobile devices;

(b) Data exchange between stationary and mobile computers; and

(c) Dates mobile available, management of tasks and contacts.

SELF-CHECK 2.1

1. Write down the N-tier client-server architecture for mobile computing.
2. Write the short note on the window CE.

True and False

1. Tiers are commonly and physically separated from each other.
2. In the client server architecture the second, layer is of presentation tier.
3. The client/server software architecture is a versatile, message-based and modular infrastructure that is intended to improve usability.
4. Client server architecture is architecture of a computer network in which many clients (Remote processors) request and receive service from a centralised server (host Computer).
5. One of the most frequently used operating system in the mobile development is Windows CE.

2.4 SYMBIAN EPOC

2.4.1 About Symbian

The Symbian Operating System is developed by the Symbian, software licensing company. This is a global open industry standard operating system for advanced, data-enabled mobile phones. Symbian has licensed mobile phone manufacturers including Motorola, Nokia, Samsung, Siemens and Sony Ericsson. Publicly announced Symbian OS to the world's leading products based on Symbian OS include the Benq P30, Motorola A920 for 3, Samsung SGH-D700, Siemens SX-1, NTT Do Como FOMA F2102V and F2051 built by Fujitsu, Sony Ericsson P800 Smartphone, Nokia 9200 Communicator range.

This is one of the powerful platforms that are being used in the development of palmtop and wireless applications. Its robust object orientated architecture makes efficient use of the reduced processing power and memory available on portable devices.

The developer has three options with which to exploit EPOCs power, OPL, Java and C++. Each language provides the ability to develop and deliver fully featured robust applications; however, each language involves a compromise between the access to EPOC functionality, performance and development time.
One of the basic languages that appeared on the Psion Organiser in 1991 is the OPL. OPL is the obvious choice where development time needs to be minimised; optimal performance is not critical and direct access to all of EPOCs functionality is not essential. However, despite its limitations OPL is capable of delivering rich and functional applications.

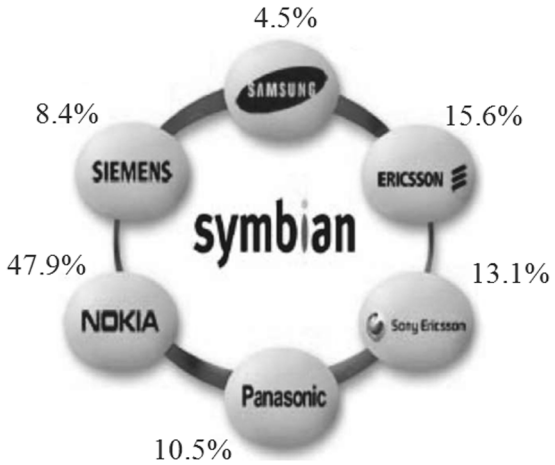


Figure 2.4: Symbian OS for different mobile companies

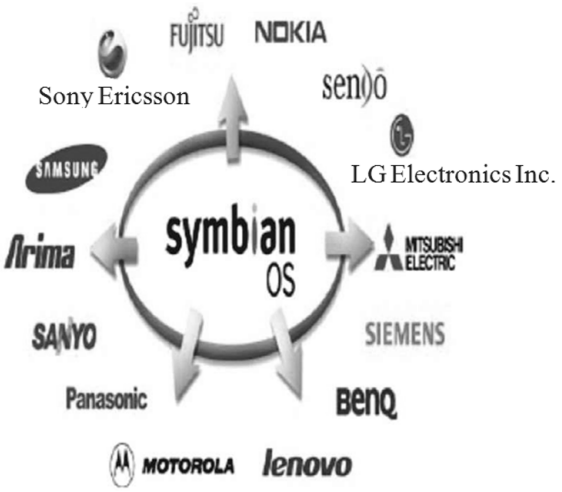


Figure 2.5: Different uses of Symbian OS in laptops and PDA

SELF-CHECK 2.2

1. In your own word explain which type of operating system we use in mobiles.
2. Give the two reasons why the Symbian OS is used in mobile.

Fill in the blanks:

1. ..... is one of the powerful platforms that are being used in the development of palmtop and wireless applications.
2. The developer has three options with which to exploit EPOCs power, ... , ... and ..... .
3. One of the basic languages that appeared on the Psion Organiser in1991 is the.....
4. OPL, Java and C++ provides the ability to develop and deliver ..... robust applications.

2.5 BREW

(a) The BREW is a technology that is used in wireless communication. BREW stands for Binary Runtime Environment for Wireless. From the developer point of view BREW can be viewed as a set of APIs that enable developers to create software applications for wireless devices (wireless phones for now), and a means of selling and delivering applications to end-users.

(b) The Mobile phones BREW lies between a software application and the Application Specific Integrated Circuit (ASIC) level software working as a thin client in between these. Thus, developers can write to BREW without knowing or caring about the device's chipset or air interface. While it's true that CDMA is Qualcomm's specialty, BREW is equally capable of running on devices that employ other air interface standards. Figure 2.6 shows the conceptual layering of software on a wireless device.

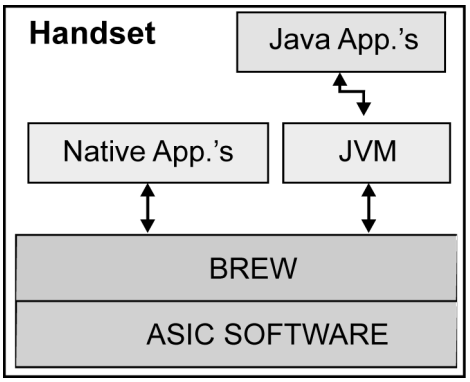


Figure 2.6: Basic layer srchitecture of BREW

(c) One of the major components of BREW is the BREW Distribution System (BDS). The BDS encompasses end users' ability to shop for, purchase, download, and install software over the wireless carrier's network.

(d) The BREW Shop lets users browse the carrier's Application Download Server to see what applications are available for purchase or trial. The entire transaction is completed over the air. The carrier generates a billing record for each purchase and a corresponding charge appears on the subscriber's monthly phone bill.

BREW is a platform developed by QUALCOMM for mobile phones. BREW can support GSM/GPRS, UMTS, and CDMA because it is air interface independent. When BREW was first introduced it was solely developed for CDMA handsets. Standing for Binary Runtime Environment for Wireless, basically BREW is a software platform that can download and run small programs for playing games, sending messages, sharing photos, etc. Main advantage of BREW platform is that the application developers can easily install their applications between all the Qualcomm Application Specific Integrated Circuits (ASICs). The BREW runs between the application and the wireless device's chip operating system; therefore, BREW enables a programmer to develop applications. Without needing to code for system interface or understand wireless applications. The software's that are used in the BREW enabled handsets are developed in C or C++ using the BREW SDK. The BREW SDK contains a BREW Emulator that can be used for testing during the development process. BREW applications must be digitally Signed unlike the Java ME platform, where any developer can upload and execute software on any supported handset.

Only content providers or authenticated BREW developers have the tools necessary to create a digital Signature because BREW gives complete control over the handset hardware. The BREW API is more standard across supported phones than the J2ME API, which canbe considerably different depending on the phone model. Also graphics tricks are easier with Brew and have direct access to the screen buffer.

2.6 WAP

WAP stands for wireless application protocol. WAP is a technology that lets cell phone users order goods and send emails from their cell phones. WAP or wireless Application Protocol, adds a new dimension to the Internet i.e. mobility. With a mobile phone or a laptop that supports WAP you can book tickets, order food and check your bank account at any time. You can play games while stuck in traffic; get the stock market news in the elevator, complete mobility, the world wide web with you all the time. It also includes services such as downloading entire phone books and new, improved ways of controlling outgoing and incoming calls will make mobile telephony even easier. To access WAP services you need a WAP product. Besides the WAP compatibility, WAP products have a large full graphic display and include a micro browser. The basic function of WAP is for mobile phones to be able to communicate with a server installed in the mobile phone network. It brings together the web and telecommunications. Of course, as all new technologies, WAP also comes with its questions and hesitations.

Basically, the mobile WAP device is attached to the mobile network such as GSM, CDMA, etc. This then dials the modem which is attached to a dial-server i.e. RAS or Remote Access Service. This server gives the WAP device access to the protocols it needs. These are the same lower level protocols as a normal Internet Service Provider will give you. We call this the PPP or Point-to-Point Protocol. In Figure 2.7, you can see the working system of WAP.

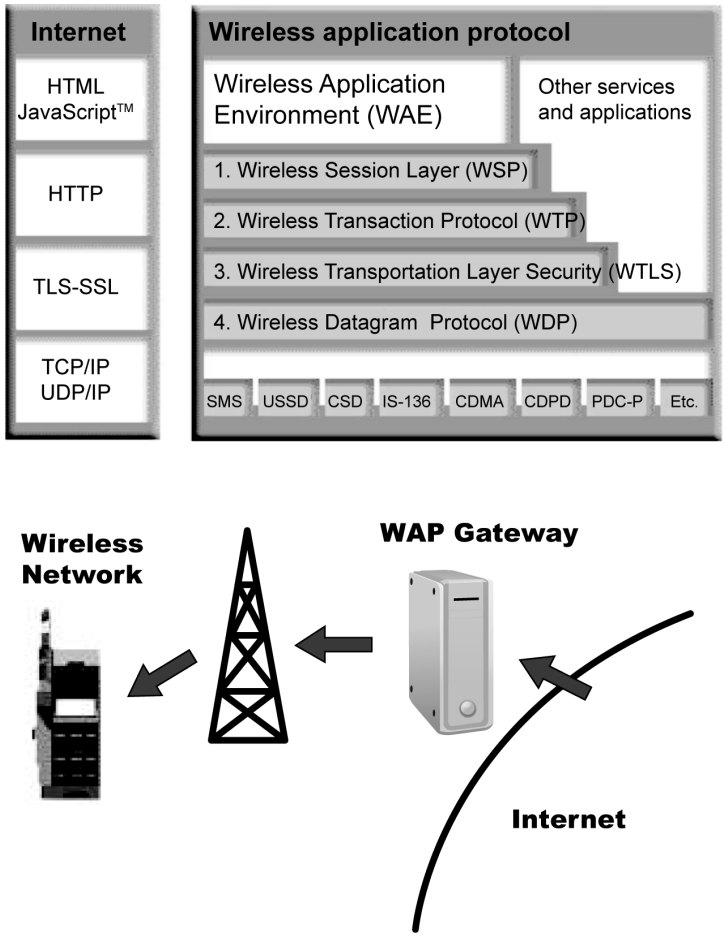


Figure 2.7: Working system of WAP

These protocols are used to access the next step in the chain, which is the WAP gateway. The WAP gateway is the link between the wireless and the wired world, basically giving the WAP device or the mobile phone the access to the common internet.

2.7 JAVA 2 MICRO EDITION (J2ME)

Java is one of the most frequently used programming languages which are used in the mobile applications development. Java is becoming a standard across wireless devices because of its application portability, endorsement by virtually all wireless technology vendors, integrated safeguards for Network delivery, sandboxing, and it's a powerful object oriented programming language.

Java runtime environments are standard or readily available for Symbian (e.g., Nokia Communicator 9200), embedded Linux (Sharp Zaurus), Pocket (Compaq's iPAQ), Palm (KVMs from Sun, Esmertec, Kada, and IBM) and real-time operating systems. The modular design of the J2ME architecture enables an application to be scaled based on constraints of a small computing device. J2ME architecture doesn't replace the operating system of a small computing device. Instead, J2ME architecture consists of layers located above the native operating system, collectively referred to as the Connected Limited Device Configuration (CLDC). The CLDC, which is installed on top of the operating system, forms the run-time environment for small computing devices. The J2ME architecture comprises three software layers. The first layer is the configuration layer that includes the Java Virtual Machine (JVM), which directly interacts with the native operating system. The configuration layer also handles interactions between the profile and the JVM. The second layer is the Profile layer, which consists of the minimum set of Application Programming Interfaces (APIs) for the small computing devices. The third layer is the Mobile Information Device Profile (MIDP). The MIDP layer consists of Java APIs for user network connections, persistence storage and the user interface. It has access to CLDC libraries and MIDP libraries Small Computing Device Requirements There are minimum resource requirements for a small computing device to run a J2ME application. First the device must have a minimum of 96 × 54 pixel display that can handle bitmapped graphics and have a way of users to input information, such as a keypad, keyboard or touch screen. At least 128 kilobytes (KB) of non-volatile memory is necessary to run Mobile Information Device (MID), and 8 KB of non-volatile memory is needed for storage of persistent application data. To run JVM, 32KB of volatile memory must be available. The device must provide a two way connectivity. A MID let is an application that requires a device that implements Java ME, MIDP to run. Like other Java programs, mildest have a "compile once, run anywhere" potential. All the files necessary to implement a MID let must be contained within a production package called the Java Archive (JAR) file. These files include MID let classes, graphic images and the manifest file. The manifest file contains a list of attributes and related definitions that are used by the application manager to install the files contained in the JAR file onto the small

computing device. Along with this, a Java Application Descriptor (JAD) file can also be included within the JAR file.

A JAD file is used to provide the application manager with additional content information about the JAR file to determine whether the MID let suite can be implemented on the device. Symbian are modules of java code that run in a server application to answer client requests. Since servlets are written in the highly portable Java language and follow a standard framework, they provide a means to create sophisticated server extensions in a server and operating system independent way. To create a wireless mobile application, the MID lets are deployed on a mobile phone (client) which supports java applications and The servlets are deployed on a server. The client application is then tied into the server via An HTTP connection. The application will send requests to the server for data and receive responses that will be mapped into the appropriate fields of the client application.

2.8 PUBLISHING FRAMEWORKS OF MOBILE COMPUTING

The publishing frame work of mobile computing follows the three-layered scheme common to information systems. The Presentation Layer, which includes the graphical interface, enables the user to view data sent by the publisher. The one of the Application Logic Layer encloses the objects representing domain concepts that fulfil application requirements (Domain Objects) and the supporting services (Service Objects). The second layer which is Storage Layer encloses database or files that contain the information to be disseminated by the publisher and received by subscribers. The fundamental domain objects are Transmitter and Tuner. The framework also includes a set of service objects: Positioning System Manager, Replication Manager, Web Server, Event Listener and Event Manager. The transmitter is responsible for keeping the definitions of the channels disseminated by the publisher. It stores information about the associations between tuners and channel items, which is necessary to the notification mechanism. This component is responsible for interacting with the tuners, and for obtaining the information to disseminate from file systems or databases. The tuner is responsible for the channel items replication, according to the settings of the corresponding subscriptions. This component includes an Event Listener, which receives the incoming notifications sent by the transmitter, and passes them to the tuner. The Positioning Service is responsible for supplying the geographic position of the subscriber to the tuner. The Replication Manager, an object of the tuner, uses positioning data to discriminate the items covering the current position, establishing an update priority for the remaining. This priority establishment is especially important for mobile computing

operations because, as connectivity may be poor, the most relevant/important items are more likely to be updated using this technique. In addition, the replication manager may decide not to update low priority items. The Replication Manager is the object in our reference architecture responsible for maintaining the herons among the items data. It has two parts, a subscriber-side, and a publisher-side. The publisher-side creates a new file containing the data representation of an item whenever its changes on the publisher's database. At the time defined in the schedule, the client tries to contact the server and perform the updates. In the case of a database source, the file is downloaded and the contents loaded into the local database; in the case of a set of files, they are downloaded to the corresponding directory on the subscriber.

EXERCISE 2.1

I. Match the Following:

1.	BREW	(a)	configuration layer
2.	Send e mails through phones	(b)	CDMA
3.	J2ME communication	(c)	Technology used in wireless
4.	Qualcomm's specialty	(d)	BREW
5.	Set of API's	(e)	WAP
6.	First layer of J2ME	(f)	Object oriented programming language

II. Fill in the blanks:

1. The Mobile phones BREW lies between a …… application and the ……. level software working as a thin client in between these.
2. One of the major components of BREW is the …… system.
3. The BDS encompasses end users' ability to ……. for, ……., ……., and …….. software over the wireless carrier's network.
4. Main advantage of BREW platform is that the application developers can easily …… their applications between all the Qualcomm Application Specific Integrated Circuits (ASICs).
5. The software's that are used in the BREW enabled handsets are developed in C or C++ using the …… .
6. WAP stands for…… .
7. WAP products have a large full graphic display and include a ……...
8. RAS stands for …… .
9. The 3 layers of J2ME are …….. layer, …….. layer and ……. Layer.

2.9 MOBILE COMPUTING TOOLS

The obtainable options for staying associated while on the go are numerous such as: smart phones, netbooks, laptops, and a broad choice of other tools process the Internet by means of cellular-based handy hotspots and mobile broadband cards, along with wi-fi which is gradually more obtainable wherever people assemble. Simultaneously, the tools that we hold are becoming increasingly more competent, and the restrictions among them more and more imprecise. In the urbanized world, mobile computing has become an essential part of everyday life in the workers, and a key driver is the growing easiness and pace with which it is likely to access the Internet from practically anywhere in the world through the ever-expanding cellular group.

Mobiles as a type are established as a more appealing and more competent with each passing year, and prolong to be an expertise with new revelations. The mobile market nowadays has almost 4 billion subscribers, over two-thirds of whom exist in developing nations. More than a billion new phones are formed each year, a run of constant improvement and modernism that is exceptional in recent times. The fastest-growing vending section is related to smart phones which means that a gigantic and growing number of people all over the world now possess and carry a computer that is suitable to fits in their hand and is competent to connect to the system wirelessly from nearly anyplace. Thousands of functions intended to sustain a broad variety of tasks on almost any smart-phone operating system are willingly obtainable, with more entering the market constantly. These mobile computing tools have become acknowledged support in every day life, providing us on-the-go access to devices for business, video/audio capture and necessary editing, sensing and measurement, geolocation, social networking, personal efficiency, references, just-in-time learning – certainly, practically anything that can be performed on a desktop.

Users now days anticipate anytime, anywhere process to data and services that in the earlier period were obtainable only while sitting in front of a computer connected to the network through a cable. Additionally to the usual software for email, message, and calendaring, new tools permit consumer to handle personal information (like Evernote, Nozbe, Wesabe, and TripIt), work together and easily access and share files (Dropbox and CalenGoo are two of many probable examples), or keep side by side of social networks (Limbo, Facebook, Foursquare, Whrrl), and usually make inspection and updating work, school, or personal information flows something simply done on the fly.

For many citizens in the world, but particularly in developing countries, mobiles are ever more the access summit not only for ordinary devices and interactions, but also for all type information, training materials, and much more. A continually more frequent pattern is for people to gaze to mobile computing proposals as their tool of alternative, as they are frequently far economical as compare to desktop or laptop computers. For this set, mobile computing tools are more reasonable, more reachable, and simpler to use than desktop computers, and supply loads of functionality to provide as their main computing tool.

For those who require a slightly more suppleness and authority from a mobile platform, it comprises netbooks, smartbooks, or other specific devices. Minor and lighter as compare to a laptop, this group of tools can use the Internet through numerous networks. Netbooks access usual production and communications applications, using a standard keyboard and a compact laptop-like design. More particular devices, like ebooks, email readers, and others are modified for a particular reason. The benefits they propose are storage and portability; the Kindle, for example, makes it simple to hold a library full of reading stuff, where as the Peek email reader conveys email access on a very condensed tool.

The transportability of mobile devices and their capability to attach to the Internet about anywhere proves them perfect as a stock up of reference stuff and learning experiences, along with general-use tools for fieldwork, where they can be used to trace annotations by means of voice, text, or multimedia, and access reference sources in actual time.

The mobile computing tool is certainly one of the attractions of this century. With modernizations endlessly rolling up, the mobile computing device is speedily improving the diverse features of our world, accomplishing every component achievable. It has had a broad involvement to the global economy, to the education scheme, to both mainstream and exceptional amusement, and still up to the communal bunch. The conjecture of all is that it does not end there. With boundless competencies that the mobile device can willingly provide to the bigger masses, it is only perhaps true that it has turn out to be a standard of our time.

2.9.1 Wi-Fi Technology, the Latest in Mobile Computing

Furthermore, by using the technology of wi-fi, you are associated to the rest of the world anywhere you may be. Public computer shops and private desktops have by now been effects of the earlier period. Mobile computers are by now providing people the chance to attach with whomever they wish even if they are continents spaced out.

The mobile computers are transportable which means the whole lot you may require for the day further on is enclosed in one gadget that is suitable in your pocket. Since it is a handy gadget, the mobile device is always with you, making it more opportune for you all through times that you have imperative tasks to accomplish.

SUMMARY

- Windows CE which is officially known as Windows Embedded Compact or Windows Embedded CE post version 6.0, and sometimes abbreviated WinCE) is defined as an operating system which is developed by Microsoft for portable computers and embedded systems. Windows CE is a distinct operating system and kernel, rather than a trimmed-down version of desktop Windows.

- The Symbian is an software licensing company that develops and licenses Symbian OS, which is used as an open industry standard operating system for data enabled mobile phones.

- The BREW which is Binary Runtime Environment for Wireless may be defined as an application development platform created by Qualcomm, originally for CDMA mobile phones, but in present days mobile computing environment it is frequently used in GSM mobile phones also.

- The Wireless Application Protocol (WAP) is an open, global specification that empowers mobile users with wireless devices to easily access and interact with information and services instantly.

- Java 2 Micro Edition J2ME is a set of technologies and specifications developed for small device like mobile phones. J2ME uses a subset of J2SE (Java 2 Standard Edition) components such as a smaller virtual machine and leaner APIs. These are similar to J2SE but with highly reduced functionality. Although it has some unique features not present in J2SE.

KEY TERMS

CSMA-CD	Carrier Sense Multiple Access with Collision Detection
DSMA-CD	Digital Sense Multiple Access with Collision Detection
IS	Intermediate System
MDIS	Mobile Data Intermediate Systems
M-ES	Mobile End Systems
MNLP	Mobile Network Location Protocol
OSI-CLNS	Open Systems Interconnection-Connectionless Network Services

SELF-TEST

1. Define the N-tier client-server architecture in details.
2. What do you mean by the client server applications?
3. Differentiate between WAP and BREW.
4. What are mobile computing tools?
5. What do you understand by publishing framework?

Topic 3 ▶ Mobile Computing and UML

LEARNING OUTCOMES

By the end of this topic, you should be able to:

1. Define XML, XML web services;
2. Identify XML at work;
3. Underline the key XML technologies for mobile computing;
4. Describe the structure of UML; and
5. Express the behaviour model and component model of UML.

▶ INTRODUCTION

Mobile computing contains a host of portable technologies that makes anytime Internet access not only possible, but integral to everyday life. From notebook computers to personal digital assistants like the Backberry® and iPhone®, to standard cell phones, mobile computing has developed into an indispensable way of life. Mobile voice communication is extensively established throughout the world and has had a very rapid increase in the number of subscribers to a variety of cellular networks over the last few years. An extension of this technology is the ability to send and receive data across these cellular networks. This is the principle of mobile computing. Mobile data communication has become a very important and rapidly evolving technology as it allows users to transmit data from remote locations to other remote or fixed locations. This proves to be the solution to the biggest problem of business people on the move – mobility.

Mobile laptop and notebook computers can use one of the two types of wireless access services when away from the home or office. The most commonly used and least expensive is WiFi®. WiFi uses radio waves to broadcast an internet signal from a wireless router to the immediate surrounding area. If the wireless network is not encrypted, anyone can jump on. WiFi is commonly used in public places to create "hotspots." Metadata is actually useful for searching and controlling content. For example, consider metadata on web pages. Search engines often place a higher priority on metadata tags such as page title, keywords and description than they do on the actual contents of the page. To those searching the Web, this metadata is useful for finding relevant pages. Metadata is also important for faster and more accurate database search and retrieval and for information stored in data warehouses.

3.1 WHAT IS XML?

XML stands for Extensible Mark-up Language which is similar to HTML. It improves the functionality of the Web by letting you identify your information in a more accurate, flexible, and adaptable way. The main objective of XML is to carry the data; however, the main objective of HTML is display the data. Unlike HTML, the tags here are not predefined; one must define their own tags. It is self-descriptive and a W3C recommendation.

It is extensible because it is not a fixed format like HTML (which is a single, predefined markup language). Instead, XML is a metalanguage – a language for describing other languages – which lets you design your own markup languages for limitless different types of documents. XML can do this because it's written in SGML, the international standard metalanguage for text document markup (ISO 8879).

3.1.1 XML Web Services

XML Web services are defined as the basic building blocks in order to scattered computing on the Internet. Open principles and the concentration on interaction and association between people and applications have produced an atmosphere where XML Web services are becoming the podium for application incorporation. Applications are created using numerous XML Web services from a variety of sources that effort together despite of where they exist in or how they were put into practice.

There are perhaps as many descriptions of XML Web Service as there are companies constructing them, but nearly all descriptions have these objects in general:

(a) XML Web Services depicts valuable functionality to Web users all the way through a standard Web protocol. Usually the procedure or protocol used is SOAP.

(b) XML Web services offer a method to portray their interfaces in sufficient aspect to permit a user to construct a consumer application to converse to them. This portrayal is typically offered in an XML document known as a Web Services Description Language (WSDL) document.

(c) XML Web services are recorded in order that impending users can locate them effortlessly. This is performed with Universal Discovery Description and Integration (UDDI).

Web services are the major applications in business-to-business, business-to-customer, and enterprise applications integration solutions. As the mobile Internet turns out to be one of the major methods for information delivery, mobile Web services are considered as a crucial phase of e-business architecture. Here, we anticipated a mobile Web services middleware that translates typical Internet services into mobile Web services. We executed a WSDL (Web Service Description Language) constructer that transforms HTML/XML into WSDL and a SOAP (Simple Object Access Protocol) message processor. The former diminishes the overhead price of reconstructing mobile Web services and facilitates flawless services among wired and wireless Internet services. The latter improves SOAP processing performance by removing the Servlet container (Tomcat), a needed component of usual Web services execution. Our system can wholly support standard Web services protocol, diminishing interaction overhead, message processing time, and server overburden.

SOAP is an XML-based messaging protocol which is used for interfacing two processes on individual platforms, where the platform is of either the same type or another type as the other process. To perform heavy lifting, XML services like SOAP can be used. This will show presentation as parsing of XML is more strong.

SOAP is a new mobile electronic device, which can be used to control electronic devices, external or internal to the SOAP, in an instinctive, expedient, and contented manner. For instance, a SOAP may provide as an option to input devices like a mouse. A SOAP device may comprise a core, and a hull that at least moderately includes the core. The core contains a tracking part competent of tracking movement relative to the hull.

XML technologies in mobile computing need fresh content. Please keep in mind all the technologies must be written in keeping mobile computing in mind.

3.1.2 Key XML Technologies for Mobile Computing

(a) XHTML

(b) VXML

  (i) intended for voice user interfaces

  (ii) permits requirement of a command-based voice dialogvia a markup language

(c) WML

(d) Xforms

(e) CCXML

(f) XML Pipeline

(g) WBXML

(h) SSML

(i) RDF

3.1.3 CCXML

(a) CCXML is termed as Call Control Extensible

(b) It is a Markup Language

  (i) Application of XML for organizing voice calls

  (ii) It concentrates on steering the calls and connecting calls (in contrast to VXML)

  (iii) It is associated with Java Telephony APIs (JTAPI).

3.1.4 XML Pipeline

(a) It identifies how to process a variety of XML resources. It can be considered in two different circumstances.

(b) It identifies the flow of giving out instructions that are pertained to one or more specified documents inhabited on the host.

It identifies the flow of giving out instructions that are applied to a variety of XML documents, residing at various hosts.

It is familiar with type of processes:

(i) Constructive processes create new information

(ii) Enlarging processes insert new types (definitions) of information

(iii) Examination processes gaze at the content of a document

(iv) Extraction processes copy a part of the document that they look into

(v) Packaging processes are dispersed processes that address the processing of distributed resources.

3.1.5 WBXML

(a) WAP Binary Extensible Markup Language;

(b) Defines a method to characterize XML in 0's and 1's as a substitute of text; and

(c) KXML (parse WBXML).

3.1.6 SSML

(a) Synthetic Speech Markup Language; and

(b) It is used for the infrastructure of the voice user interface.

3.1.7 RDF

(a) Resource Description Framework.

(b) Produced particularly:

(i) To permit discovery of a variety of resources.

(ii) Indexing them.

(iii) Establishment of resources that are made up of other RDF resources by basically nesting the RDF descriptions.

(c) RDF is part of Semantic Web.

ACTIVITY 3.1

Discuss with your friends and teacher, what are the difference between XML and HYML.

Example of XML Document

XML documents use a self-describing and simple syntax:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<note>
   <to>Tove</to>
   <from>Jani</from>
   <heading>Reminder</heading>
   <body>Do not forget me this weekend!</body>
</note>
```

The first line is the XML declaration. It defines the XML version (1.0) and the encoding used (ISO-8859-1 = Latin-1/West European character set).

The next line describes the root element of the document (like saying: "this document is a note"):

```
<note>
```

The next 4 lines describe 4 child elements of the root (to, from, heading, and body):

```
<to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Do not forget me this weekend!</body>
```

And finally the last line defines the end of the root element:

```
</note>
```

You can assume, from this example, that the XML document contains a note to Tove from Jani.

Dont you agree that XML is pretty self-descriptive?

XML Documents form a Tree Structure

XML documents must contain a root element. This element is "the parent" of all other elements.

The elements in an XML document form a document tree. The tree starts at the root and branches to the lowest level of the tree.

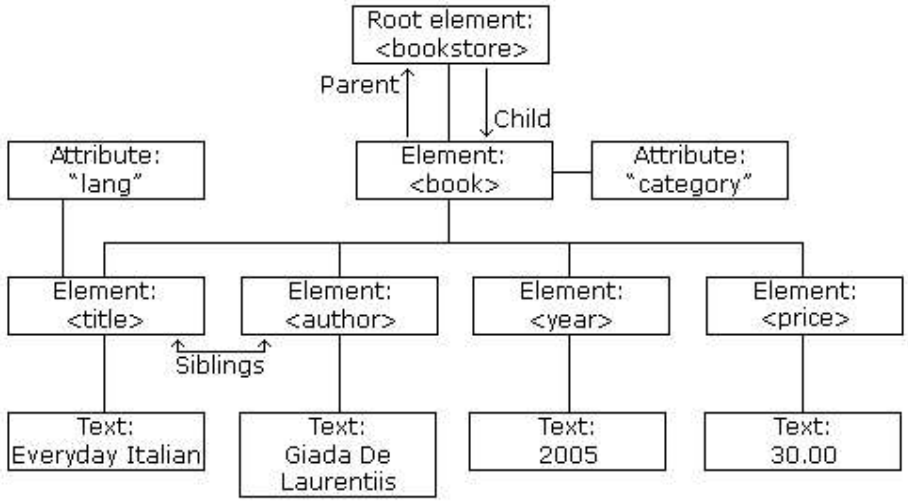
All elements can have sub elements (child elements):

```
<root>
  <child>
    <subchild>.....</subchild>
  </child>
</root>
```

The terms parent, child, and sibling are used to describe the relationships between elements. Parent elements have children. Children on the same level are called siblings (brothers or sisters).

All elements can have text content and attributes (just like in HTML).

Example: The root element in the example is <bookstore>. All <book> elements in the document are contained within <bookstore>.



The <book> element has 4 children: <title>,< author>, <year>, <price>.

The syntax rules of XML are very simple and logical. The rules are easy to learn, and easy to use.

All XML Elements must have a Closing Tag

In HTML, elements do not have to have a closing tag:

```
<p>This is a paragraph

<p>This is another paragraph
```

In XML, it is illegal to omit the closing tag. All elements must have a closing tag:

```
<p>This is a paragraph</p>

<p>This is another paragraph</p>
```

Note: You might have noticed from the previous example that the XML declaration did not have a closing tag. This is not an error. The declaration is not a part of the XML document itself, and it has no closing tag.

XML Tags are Case Sensitive

XML tags are case sensitive. The tag <Letter> is different from the tag <letter>.

Opening and closing tags must be written with the same case:

```
<Message>This is incorrect</message>

<message>This is correct</message>
```

Note: "Opening and closing tags" are often referred to as "Start and end tags". Use whatever you prefer. It is exactly the same thing.

XML Elements must be Properly Nested

In HTML, you might see improperly nested elements:

```
<b><i>This text is bold and italic</b></i>
```

In XML, all elements must be properly nested within each other:

```
<b><i>This text is bold and italic</i></b>
```

In the example above, "Properly nested" simply means that since the <i> element is opened inside the <b> element, it must be closed inside the <b> element.

XML Documents must have a Root Element

XML documents must contain one element that is the parent of all other elements. This element is called the root element.

```
<root>
  <child>
    <subchild>.....</subchild>
  </child>
</root>
```

XML Attribute Values must be Quoted

XML elements can have attributes in name/value pairs just like in HTML. In XML, the attribute values must always be quoted.

Study the two XML documents below. The first one is incorrect, the second is correct:

```
<note  date=12/11/2007>
  <to>Tove</to>
  <from>Jani</from>
</note>
```

```
<note  date="12/11/2007">
  <to>Tove</to>
  <from>Jani</from>
</note>
```

The error in the first document is that the date attribute in the note element is not quoted.

Entity References

Some characters have a special meaning in XML.

If you place a character like "<" inside an XML element, it will generate an error because the parser interprets it as the start of a new element.

This will generate an XML error:

```
<message>if salary < 1000 then</message>
```

To avoid this error, replace the "<" character with an entity reference:

```
<message>if salary & lt; 1000 then</message>
```

There are 5 predefined entity references in XML:

```
&lt;     <    less than

&gt;     >    greater than

&amp;    &    ampersand

&apos;   '    apostrophe

&quot;   "    quotation mark
```

Note: Only the characters "<" and "&" are strictly illegal in XML. The greater than character is legal, but it is a good habit to replace it.

Comments in XML

The syntax for writing comments in XML is similar to that of HTML.

<!-- This is a comment -->

White-space is Preserved in XML

HTML truncates multiple white-space characters to one single white-space:

HTML:	Hello Tove
Output:	Hello Tove

With XML, the white-space in a document is not truncated.

XML Stores New Line as LF

In Windows applications, a new line is normally stored as a pair of characters: Carriage Return (CR) and Line Feed (LF). In Unix applications, a new line is normally stored as a LF character. Macintosh applications also use an LF to store a new line.

XML stores a new line as LF.

An XML document contains XML Elements. What is an XML Element? An XML element is everything from (including) the element's start tag to (including) the element's end tag.

An element can contain other elements, simple text or a mixture of both. Elements can also have attributes.

```
<bookstore>
  <book category="CHILDREN">
    <title>Harry Potter</title>
    <author>J K. Rowling</author>
    <year>2005</year>
    <price>29.99</price>
  </book>
  <book category="WEB">
    <title>Learning XML</title>
    <author>Erik T. Ray</author>
    <year>2003</year>
    <price>39.95</price>
  </book>
</bookstore>
```

In the example above, <bookstore> and <book> have element contents, because they contain other elements. <author> has text content because it contains text.

In the example above only <book> has an attribute (category="CHILDREN").

EXERCISE 3.1

1. Write short note on HTML and XML.
2. What are the basic difference between XML and HTML?

3.2 KEY XML TECHNOLOGIES FOR MOBILE COMPUTING

In the present day's environment to meet the new requirements of web publishing, W3C has developed the Extensible Mark-up Language (XML). The goal was to develop a language that has the flexibility of SGML and the simplicity of HTML. XML was designed to deliver structured, possibly complex content over the web while still being easy to implement. XML is primarily a meta-language for describing other mark-up languages. It neither specifies a fixed tag set nor the semantics of the tags, but allows users to define their own set of tags and the structural relationships between the tags. A Document Type Definition (DTD) may be associated with the document, but it is not required. XML as such does not contain any functionality, but is used as a data description, interchange, and storage format.

Two basic properties of XML documents are well formedness and validity. The well formedness constraints control of proper syntax and structure of the document. Moreover, an XML document is valid if it is well-formed and it complies with the constraints expressed in an associated document type declaration. XML linking require the two mark-up languages Link and X Pointer. X Link is an XML language, which uses Uniform Resource Identifiers (URIs) to describe links between different files. X Pointer complements Link with the ability to address specific parts of elements or data in an XML document. Together, they allow multidirectional inks, links to multiple resources, link databases, links that point specific places inside the documents, and links to and from read-only documents.

3.2.1 XSL

The XSL is defined as the Extensible Stylesheet Language (XSL). This is the style language of XML. In the XML tags we do not have any predefined semantics; XSL is used to describe how the elements are presented. XSL specification is divided in two separate parts. The actual XSL specification defines a vocabulary of formatting objects (XSL-FOs) that have the necessary base Semantics. XSL Transformations (XSLT) specification defines a language for transforming the original XML document to the document that is composed of the elements having formatting semantics. The conversion of XML to the presentation structure is done by XML/XSL processor.

It takes an XML document, and constructs the source tree from the document. Using the XSL style sheet, it constructs a separate tree, the result tree, which is composed of formatting elements. The result tree does not have tope composed of formatting objects introduced in XSL specification, but it can consist of any XML elements. However, the document resulting from the transformation must always be a well-formed XML document. The basic building block of XSL transformations is the template rule, specified with: template element describing how the original XML element node is converted into the element node that can be formatted, styled, and displayed It consists of two parts, a matching part and a formatting part. The matching part identifies the XML node in the source document to be formatted, and the formatting part produces part of the result tree by applying formatting to the nodes.

An XSL formatting object represents a particular kind of formatting information, which is applied to the content of the formatting object. The formatting vocabulary is built on the basis of Cascading Style Sheets (CSS) and Document Style Semantics and Specification Language (DSSSL). Over 90 percent of the XSL formatting properties are defined in CSS. However, XSL extends CSS, e.g., to allow pagination and frame based structure. The extensions are done by adding new values to CSS properties, by splitting CSS properties into several new properties, or by creating completely new properties.

SELF-CHECK 3.1

Give the two reasons, why we use XML technologies in the present days mobile computing?

3.3 XML AND UML

Unified Modelling Language (or UML) is an Object Management Group (OMG) standard and a successor to many of the object-oriented methods developed in the 1980s and 1990s. The idea of using UML to model XML documents is not new.

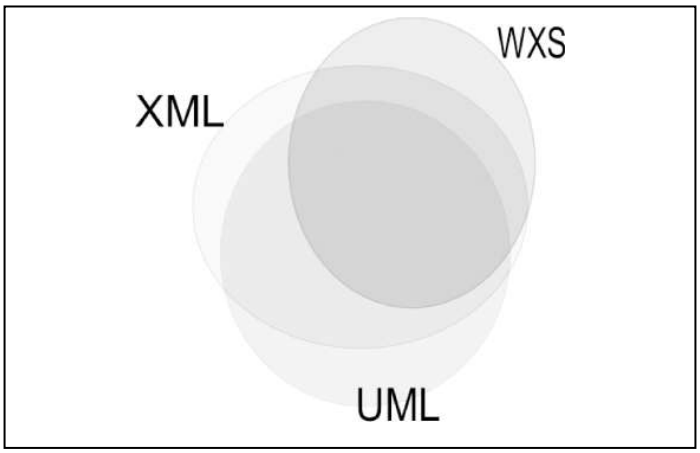
There are two different levels at which UML and XML can be mapped:

(a) The UML diagram is a higher-level view of the schema, and the schema by itself is the main delivery. The UML diagram needs to be able to control exactly how each schema structure is described. Specific stereotypes and parameters are often added to customise the level of control. UML can be used to model an XML schema.

(b) UML can be used to model the structure of XML documents directly. XML schemas can be generated for the purpose of validating the documents, but they are provided as a convenience for application developers. UML does not worry about schema details. Their style and modularity are not their most important features. The algorithm for producing these schemas is focused on expressing validation rules that make the XML data match the UML diagram as closely as possible.

The one of the point concerning XML and UML is that it's not simple to generate DTDs and W3C XML Schemas from UML. When generating DTDs or W3C XML schemas from UML, you have to cope with the restrictions of these languages, notably those related to unordered content models. Unordered content models are a natural fit for UML, in which the attributes of a class are unordered. The limitations of DTDs and W3C XML Schemas create problems when UML attributes are serialised as XML elements.

The another important points that appear clearly in all the work related to this topic is that it is quite easy to map UML objects into XML or to use UML to describe classes of instance documents. The most difficult issue when doing so is that UML operates on and graphs, XML is a tree structure. Some links need to be either removed or serialised using techniques to make the mapping happen cleanly (you can use X Link, but it is not built into XML 1.0). Except for this issue, the relationship between UML and XML is quite natural in both directions: UML provides a simple language to model XML documents and XML provides natural serialisation syntax for UML objects.



The issue when modelling W3C XML schemas in UML is that the model needs to describe the XML instances and the schema itself. This is where all the complexity of W3C XML schemas enters the UML world. While there is a good overlap between UML and XML, the overlap is not so good between XML and W3C XML schemas. W3C XML schemas have in some ways enriched XML with their own expectations, and their expectations do not match those of UML.

Overlaps between XML, UML, and W3C XML schema.

EXERCISE 3.2

1. Differentiate between UML and XML.
2. How XML Technologies implemented in Mobil Computing?

3.4 PUTTING XML TO WORK

XML provides a text format for transmitting structured information. XML is designed to store data. Microsoft extends the usefulness and power of XML in form of XAML by making it a .NET programming language. Starting with Internet Explorer 4.01, Microsoft has shipped their own XML parser, MSXML, as a COM component. XAML syntactically inherited from XML is a scripting programming language used to write applications that covers both Windows and Web worlds. Now as a developer, you may find yourself scripting XAML tags to write Windows or Web applications. Besides the easy to use and understand, XAML is flexible enough to provide options to use any .NET programming

language as code behind. Not only you can write the UI definitions, such as size, color, and layout of Windows but you can also write the events and methods in XAML files. XAML takes advantage of existing XML syntaxes. Extensible Application Markup Language (XAML) is a new scripting language based on XML that allows Longhorn developers to build and managed UI applications via scripting. If you remember Windows Forms model, the settings of a Form and its child controls were used to store in a .cs file with all the attributes of the form and controls. Now, all these attributes can store in a xaml file and that can run without any additional code. A window has properties such as width, height, background color, foreground color and so on. These property values can be set at design-time as well as at run-time. In case of design-time, the values are stored in a file that is associated with a window. When a C# compiler compiles the code, the values of window properties get stored in the executables. In case of XAML, a panel represents a window. The values of the panels are stored in the XAML itself in form of XML attributes and elements. When you open an XAML file in browser (Windows Longhorn), the operating system is smart enough to read the XAML tags, elements, and attribute values and creates a window with the given attribute and element values. All controls or windows in XAML have a predefined tag and each of this tag has properties related to that control or window. For example, the following code for Button tag sets a few properties of Button control: <Button Width="20" Height="10">OK Button</Button>

3.5 INTRODUCTION TO UML

UML stands for Unified Modeling Language. UML, or Unified Modeling Language, is a specification language that is used in the software engineering field. It can be defined as a general purpose language that uses a graphical designation which can create an abstract model. UML is an extensible language. Much of the semantics for this language were greatly improved with the UML 2.0 update. In 1996, a group of developers came together to create the UML Partners.

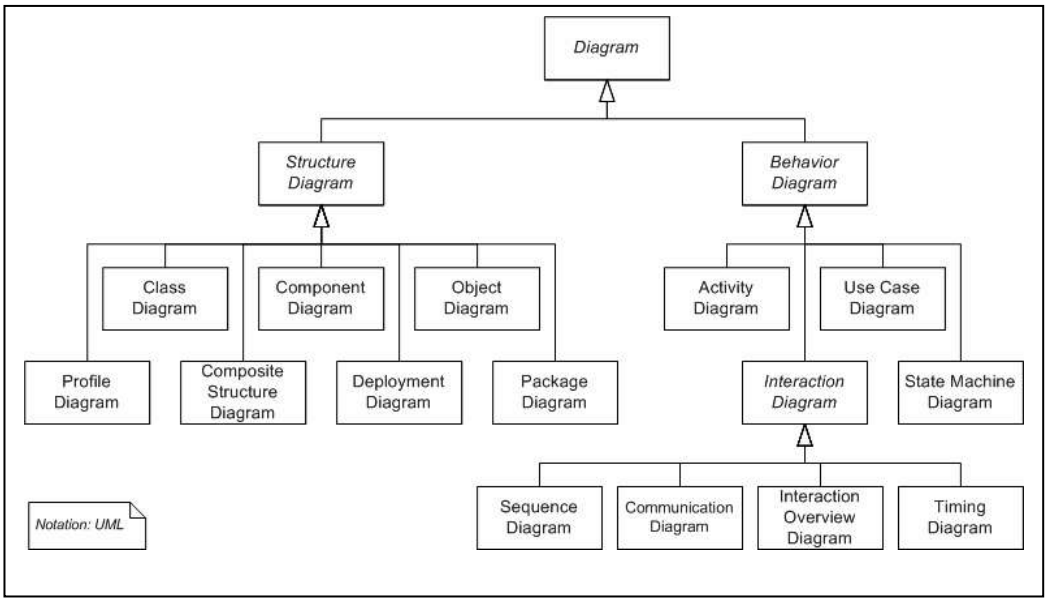
It has two elements that are used for customization, and these are stereotypes and profiles. UML was meant to be a unifying language enabling IT professionals to model computer applications.

Since UML is not a methodology, it does not require any formal work products (i.e., "artifacts" in IBM Rational Unified Process® lingo). The Object Management group is responsible for defining UML, and they do this via the UML Meta model. XMI can be used to serialize the UML model. UML is not limited simply modeling software. It can also be used to build models for system engineering, business processes, and organization structures. The Unified Modeling Language

is important for a number of reasons. First, it has been used as a catalyst for the advancement of technologies which are model driven, and some of these include Model Driven Development and Model Driven Architecture. When UML was created, one of the goals of the developers was to create a language that could support every object oriented approach. Some of the features which UML supports includes time analysis, data analysis, object oriented structure design, and state charts.

3.5.1 User View of UML

There are 14 types of diagrams divided into two categories. Out of fourteen diagrams seven diagrams represent structural information, and the other seven represent general types of behaviour, including four that represent different aspects of interactions. These diagrams can be categorised hierarchically as shown in the following class diagram.



UML element may appear on almost all types of diagrams; this flexibility has been partially restricted in UML 2.0. UML may define additional diagram types or extend existing diagrams with additional notations.

In keeping with the tradition of engineering drawings, a comment or note explaining usage, constraint, or intent is allowed in a UML diagram.

3.5.2 Structure View

The Structure diagrams are a primarily concern on the things that must be present in the system being modelled. Since structure diagrams represent the structure used extensively in documenting the architecture http://uk.ask.com/wiki/ Software_Architecture?qsrc=3044 of software systems.

(a) **Class Diagram**
The class diagram is used to describe the structure of a system by showing the system's classes, their attributes, and the relationships among the classes.

(b) **Composite Structure Diagram**
The composite structure diagram is used to describe the internal structure of a class and the collaborations this structure makes possible. Figure 3.4 shows a composite structure diagram.

(c) **Component Diagram** http://uk.ask.com/wiki/Component_diagram?qsrc=3044:
Th e Component Diagram is used to describe how a software system is split up into components and shows the dependencies among these components.

(d) **Deployment Diagram**
The deployment diagram is used to describe the hardware used in system implementations and the execution environments and artifacts deployed on the hardware.

(e) **Object Diagram**
The object diagram is used to show a complete or partial view of the structure of a modelled system at a specific time.

(f) **Package Diagram**
The package diagram is used to describe the methods that define how a system is split up into logical groupings by showing the dependencies among these groupings.

(g) **Profile Diagram**
Operates at the Meta model level to show stereotypes as classes with the <<stereotype>> stereotype, and profiles as packages with the <<profile>> stereotype. The extension relation (solid line with closed, filled arrowhead) indicates what Meta model element a given stereotype is extending.

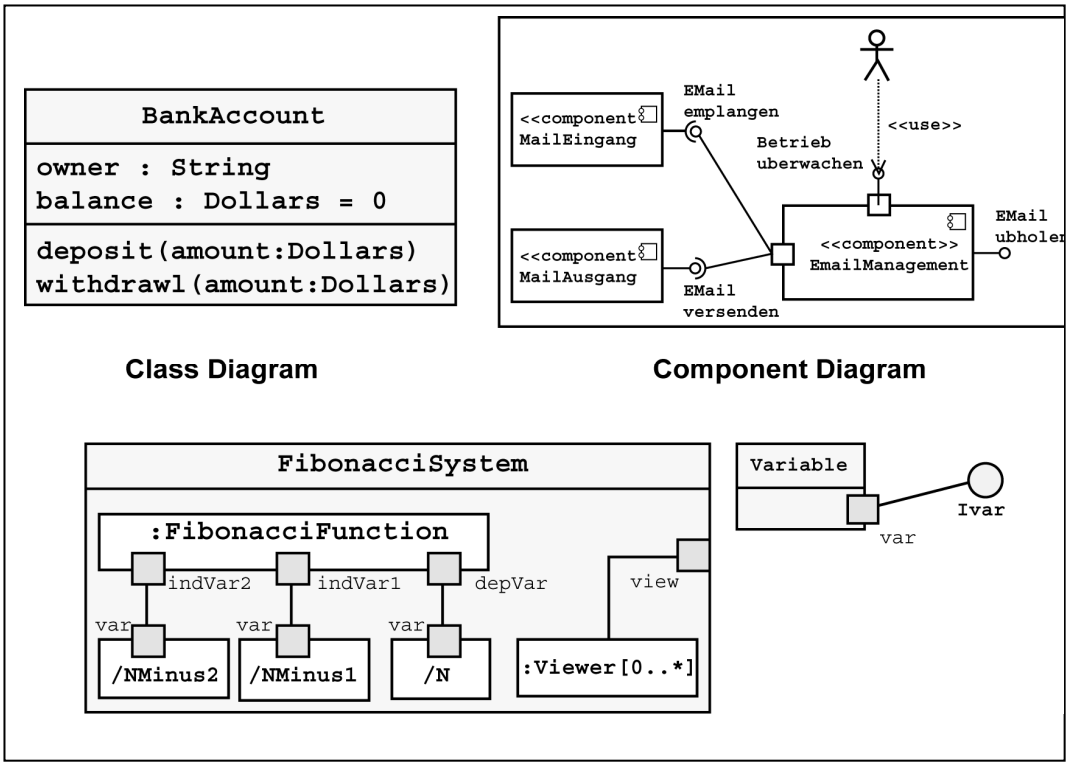


Figure 3.1: Composite structure diagrams

3.5.3 Behavior View

The Behaviour model is a primary concern on what must happen when the system is being modelled. The Behaviour view is focused on the behaviour of a system; hence they are used extensively to describe the functionality of software systems.

(a) **Use Case Diagram-http://en.wikipedia.org/wiki/Use_case_diagram**
The Use Case Diagram describes the functionality provided by a system in terms of actors, their goals represented as use cases, and any dependencies among those use cases.

(b) **http://en.wikipedia.org/wiki/Activity_diagram**
The Activity Diagram describes the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of the control.

(c) **UML Stat Machine Diagram-http://en.wikipedia.org/wiki/UML_state_machine**

The UML State Machine Diagram: describes the states and state transitions of the system.

EXERCISE 3.3

1. Explain the XML and UML.
2. Write a short note on users view and behaviour view of UML.

3.5.4 Implementation View – Component Diagrams

Overview

Component diagrams are used to model physical aspects of a system. This diagram is different in terms of nature and behaviour.

The question that arises here is that what are these physical aspects? Physical aspects are the elements like executables, libraries, files, documents, etc., which resides in a node.

The component diagrams are used to visualise the organisation and relationships among components in a system. These diagrams are also used to make executable systems.

Purpose

The purpose of component diagram is different from all other diagrams discussed so far. It does not describe the functionality of the system but it describes the components used to make those functionalities. Component diagram is a special kind of diagram in UML.

The component diagrams are used to visualise the physical components in a system. These components are libraries, packages, files, etc.

Component diagrams can also be described as a static implementation view of a system. Static implementation represents the organisation of the components at a particular moment.

The single component diagram cannot represent the entire system but a collection of diagrams are used to represent the whole.

So the purpose of the component diagram can be summarised as:

(a) The diagram construct executables by using forward and reverse engineering.

(b) This diagram visualise the components of a system.

(c) This diagram describes the organisation and relationships of the components.

How to Draw Component Diagram?

The component diagrams are used to describe the physical aspects of a system. This aspect includes files, executables, libraries, etc.

The fundamental purpose of the component diagram is different. The component diagrams are used during the implementation phase of an application. It is prepared well in advance to visualise the implementation details.

Initially the system is designed using different UML diagrams and then when the aspects are ready component diagrams are used to get an idea of the implementation.

One of the important diagrams is the component diagram. The component diagram is an important diagram. This diagram is very important because without it the application cannot be implemented efficiently. A well prepared component diagram is also important for other aspects like application performance, maintenance, etc.

So before drawing a component diagram the following artifacts are to be identified clearly:

(a) First we check the files used in the system.

(b) Libraries and other artifacts relevant to the application.

(c) Relationships among the artifacts.

Now after identifying the artifacts the following points needs to be followed:

(a) Use a meaningful name to identify the component for which the diagram is to be drawn.

(b) Prepare a mental layout before producing using tools.

(c) Use notes for clarifying important points.

The Figure 3.2 shows the component diagram for order management system. Here the artifacts are files. Here the diagram shows the files in the application

and their relationships. In actual the component diagram also contains libraries, folders, etc.

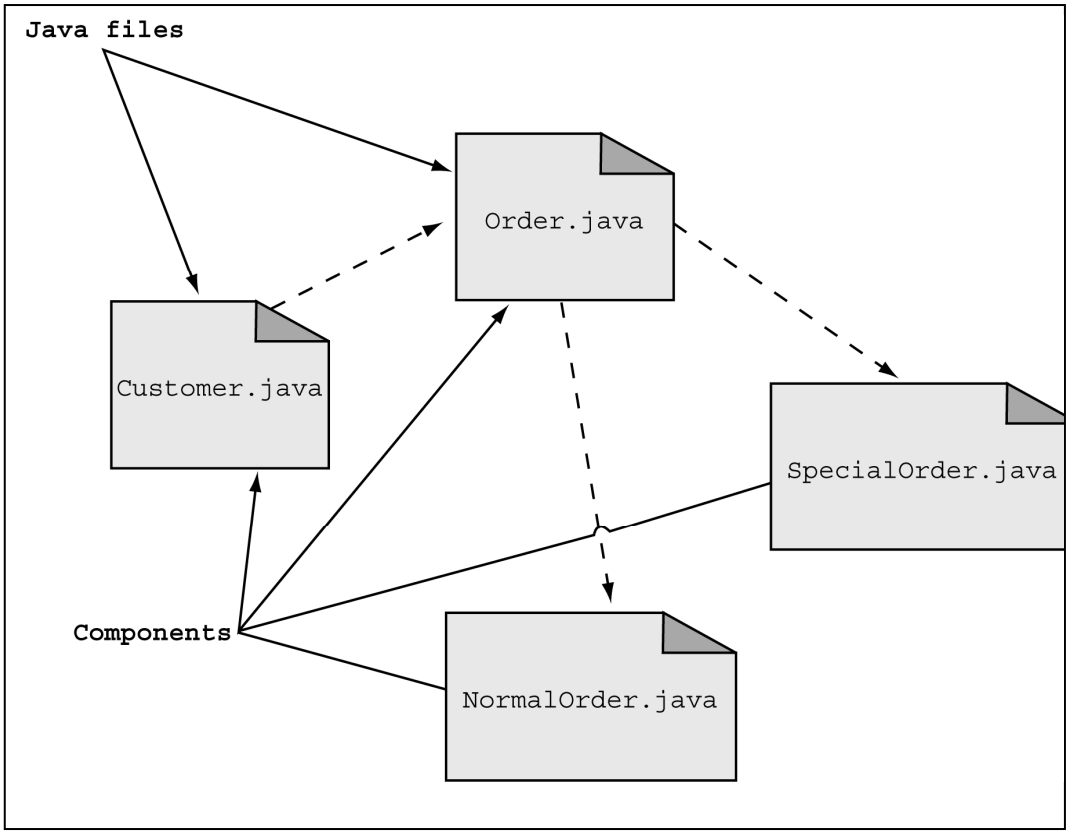


Figure 3.2: Component diagram of an order management system

In the Figure 3.2 four files are identified and their relationships are produced. Component diagram cannot be matched directly with other UML diagrams discussed so far. Because it is drawn for a complete different purpose.

So the component diagram has been drawn considering all the points mentioned above.

Where to Use Component Diagrams?

To visualise the static implementation view of the system we describe the component diagram. Component diagrams are special type of UML diagrams used for different purposes. These diagrams show the physical components of a system. To clarify it, we can say that component diagrams describe the organisation of the components in a system.

Organisation can be further described as the location of the components in a system. These components are organised in a special way to meet the system requirements. As we have already discussed those components are libraries, files, executables etc. Now before implementing the application these components are to be organised. This component organisation is also designed separately as a part of project execution.

Component diagrams are very important from implementation perspective. So the implementation team of an application should have a proper knowledge of the component details.

Now the usage of component diagrams can be described as:

(a) Model the components of a system;

(b) Model database schema;

(c) Model executables of an application; and

(d) Model system's source code.

SELF-CHECK 3.2

1. Design the structural, behaviour and user view of UML.
2. Design the component diagram for banking system.

SUMMARY

- Extensible Mark-up Language (XML) is a set of rules for encoding documents in machine-readable form. It is defined in the XML 1.0 Specification produced by the W3C, and several other related specifications, all gratis open standards.

- One of the important diagrams that are a primarily concern on the things that must be present in the system being modelled is the structural diagram. Since structure diagrams represent the structure they are used extensively in documenting the architecture http://uk.ask.com/wiki/Software_Architecture?qsrc=3044 of software systems.

- The Behaviour views are very frequently used to describe the functionality of software systems. Component diagrams are used to model physical aspects of a system. This diagram is different in terms of nature and behaviour.

KEY TERMS

HTML Hyper Text Mark-up Language

UML Unified Modelling Language

XML Extensible Mark-up Language

XSL Extensible Style sheet Language

SELF-TEST

1. What are the basic difference between XML and HTML?
2. How XML technologies implemented in mobile computing?
3. Differentiate between UML and XML.
4. Is there any relation defines in structured and behavior view of XML?
5. Write short notes on key XML technologies.

Topic 4 ▶ Generic User Interface Development

LEARNING OUTCOMES

By the end of this topic, you should be able to:

1. Define user interface development;
2. Identify how to build generic user interface;
3. Recognise the usage of UML for modeling graphic user interfaces;
4. Describe X forms; and
5. Underline the role of WAP, J2ME, BREW and Microsoft platforms for mobile GUI's.

▶ INTRODUCTION

A GUI permits a computer user to move from one application to another. A good GUI makes an application easy, practical, and efficient to use, and the marketplace success of today's software programs depends on a good GUI design. Consider the Macintosh and the IBM-PC. Computer users view Apple's Macintosh computers as having the best GUI. Likewise, their positive view of the Macintosh system is almost double of the Windows users'. The development of a new software is extremely expensive. With the success or failure of a product and maybe the entire company dependent on the GUI's reception in the marketplace, a good GUI design is extremely important. Unfortunately, it is not easy to define if an application's GUI is easy, practical, or efficient. These are the attributes that do not lend themselves to counting. The marketplace does attempt to access these attributes, however, but even after over 10 years of GUI development, there are still questions concerning what is a good GUI design.

4.1 USER INTERFACE DEVELOPMENT

The User Interface (UI) is everything designed into an information device with which a human being may interact including display screen, keyboard, mouse, light pen, the appearance of a desktop, illuminated characters, help messages, and how an application program or a website invites interaction and responds to it. The user interface can arguably include the total "user experience," which may include the aesthetic appearance of the device, response time, and the content that is presented to the user within the context of the user interfacing information technology. In early computers, there was very little user interface except for a few buttons at an operator's console. The user interface was largely in the form of punched card input and report output.

The user interface was a nearly blank display screen with a command line, a keyboard, and a set of commands and computer responses that were exchanged. This command line interface led to one in which menus (list of choices written in text) predominated. And, finally, the graphical user interface GUI arrived, originating mainly in Xerox's Palo Alto Research Centre, adopted and enhanced by Apple Computer, and finally effectively standardised by Microsoft in its Windows operating systems.

The Visual part of computer application or operating system through which a user interacts with a computer or software. It determines how commands are given to the computer or the program and how information is displayed on the screen. Three main types of user interfaces are (1) Command language: the user must know the machine and program-specific instructions or codes. (2) Menus: user chooses the commands from lists displayed on the screen. (3) Graphical user Interface (GUI): user gives commands by selecting and clicking on icons displayed on the screen.

Over the past decade, the rapid prototyping of user interfaces has become quite a common technique in the industry; however, prototyping remains weak in linking the application domain with the user interface, and automated prototyping is mostly limited to database-oriented integration in the overall requirements engineering process.

Recently, the use of scenarios for requirements acquisition has gained a lot of attention in the research community. Yet the applications most importantly, the prototyping and the scenario approaches lack transition from scenarios to formal specifications, the target of the requirements engineering process, remain ill-defined.

Here, we suggest a requirement engineering process that generates a user interface prototype from scenarios and yields a formal specification of the application. Scenarios are acquired in the form of collaboration diagrams as defined by the Unified Modelling Language (UML), and are enriched with User Interface (UI) information. These diagrams are automatically transformed into UML State chart specifications of the UI objects involved. From the set of obtained specifications, a UI prototype is generated that is embedded in a UI builder environment for further refinement. Based on end user feed back, the collaboration diagrams and the UI prototype may be iteratively refined, and the result of the overall process is a specification consisting of the State chart diagrams of all the objects involved, together with the generated and refined prototype of the UI.

4.2 BUILDING GENERIC USER INTERFACE

It is difficult to develop an application for multi-platform deployment without duplicating development effort. Advances in electronics, communications, and the fast growth of the Internet have made use of a wide variety of computing devices an everyday occurrence. Furthermore, users expect to remotely access their data from any of these devices (e.g., desktop computer, laptop, handheld phone, etc.). Developers now face the daunting task to build user interfaces that must work in multiple devices. To address the need for a uniform language for building multi-platform applications, we have developed the User Interface Mark- up Language (UIML). We discuss our research in creating such languages, with some of the support tools available, and describe our approach towards creating a new user interface design methodology to build multi-platform user interfaces. UIML is a single language for building user interfaces for any device. UIML emphasises the separation of concerns of an interactive application in such a way that moving one program from one platform to another might require only small or no changes at all.

Furthermore, because it is based on XML, it is easy to write transformations that take the language from one abstract representation to a more concrete representation. The tools built around UIML extend the language with the use of transformations that afford the developer the creation of user interfaces with a single language that will execute in multiple platforms.

The approach taken with UIML is that of building an application using a generic vocabulary that could then be rendered for multiple platforms. Using a generic vocabulary for desktop applications, for example, the developer can write a program in UIML only once and have it to be rendered for Java or HTML. And, within HTML, it can be rendered to different versions of HTML and/or browsers. We

present some details of the generic vocabulary and the advantages of this approach in this paper. Based on our experience with UIML, we have learned that the generic vocabulary is not sufficient to build interfaces for widely varying platforms, such as Voice XML, handhelds (e.g., Palm PDAs) and desktops. In this paper, we describe the preliminary results of a multi-step process of building multi-platform user interfaces. It involves programming in UIML at different levels of abstractions with the support of transformations that move the developer's work from one representation to the next. This paper presents our initial results on this area. The language design for UIML was done in 1997 by a group of individuals that went on to start the company Harmonica. At that time, the language was deemed too early for commercial adoption, thus, the company development plans were put in hiatus for a few years. Meanwhile research in UIML began at the Computer Science Department at Virginia Tech. At the end of 1999, with the expansion of the Internet and the adoption of HTML and other mark-up languages, it was deemed appropriate to commercialise UIML. Since then, Harmonica has been creating commercial tools to support the deployment of UIML to commercial environments. In parallel, research continues on UIML at Virginia Tech. Support for UIML continues to grow beyond these two organisations. For example, there was a conference devoted to UIML in March 2001 held in France. More information on UIML can be found at http://www.uiml.org.

ACTIVITY 4.1

Discuss with your friends and teacher about User Interface and Building Generic User Interface.

SELF-CHECK 4.1

Fill in the blanks:

1. A ..... permits a computer user to move from one application to another.
2. The ..... enables interactivity.
3. The user interface was largely in the form of ..... .
4. UML stands for ..... .
5. ..... is a single language for building user interfaces for any device.

Choose one: UIML, User Interface Development, HTML, punched card input and report output, GUI, Unified modeling language.

4.3 USING UML FOR MODELLING GENERIC USER INTERFACE COMPONENTS

4.3.1 The UML Profile for Generic User Interface Layout

Graphical user interfaces (GUIs) are essential components of software systems. They comprise of two main components: a dynamic or behavior part and a static or layout part. This partition is frequently reproduced in expansion processes, which gives away each part to experts, either software engineers or screen designers. This partition of labor results in troubles and disagreements throughout the early stages of the design process. As both areas make use of modeling, we have decided to bridge the gap by generating a general language for both groups to use when emerging GUIs. We regard a model of GUI layout the omitted link among the two mostly disconnected worlds of graphics design and software development.

While dynamics and performance can be represented using obtainable UML diagrams layout cannot, as all available UML diagrams are not layout-aware. Thus, we have formed the UML Profile for GUI Layout, a UML 2.0 profile that uses Diagram Interchange to stock up layout information when staying entirely conventional with the standard. The profile's metamodel pays typecast classes

that are connected by controlled relations, as explained by the UML 2.0 expansion methods.

The main notion of the outline is the Screen Area, a section of screen space that provides a particular purpose. Each can either enclose other Screen Areas or present a portion of functionality to the user, like Images, Texts, Links and Forms. Position and size of each Screen Area is located using the Diagram Interchange elements Point and Dimension, respectively.

4.3.2 UIML

The declarative XML-based language that can be used to define user interfaces is UIML. The designing goals of UIML are to "reduce the time to develop user interfaces for multiple device families". A related design rationale Behind UIML is to "allow a family of interfaces to be created in which the common features are factored out" One of the primary design goals of UIML is to provide a canonical format for describing interfaces that map to multiple devices. In this section we present some of UIML's language features.

```
<?xml version="1.0" ?>

<!DOCTYPE uiml PUBLIC "-//Harmonia//DTD UIML 2.0 Draft//EN"
"UIML2_0g.dtd">
<uiml>

<head>...</head>

<interface>

<structure>...</structure>

<content>...</content>

<behavior>...</behavior>

<style>...</style>

</interface>

<peers>...</peers>

<template>...</template>

</uiml>
```

The UIML does not contain any platform-specific or metaphor-dependent tags.

For example, there is no tag like <window> that is directly linked to the desktop metaphor of interaction. UIML uses about thirty generic tags instead. Platform specific renderers have to be built in order to render the interface defined in UIML for that particular platform. Associated with each platform-specific renderer is a vocabulary of the language widget-set or tags that are used to define the interface in the target platform.

At the highest level, a UIML document comprises of four components: <Head>, <interface>, <peers> and <template>. The <interface> is the only component that is relevant for this discussion. Information on the others can be found elsewhere. The heart of the UIML document in terms of representing the actual user interface is the<interface>. All the UIML elements that describe the UI are present within this element. There may be multiple <interface> elements.

The four main components are the following:

(a) **<Structure>**
The relationships between the various UI elements within the interface, is represented using. <Structure>: The physical organisation of the interface, including each <structure> is comprised of different <part>s. Each part represents the actual platform-specific UI Element and is associated with a single class of UI elements. The particular category of UI elements in UIML is represented by "class" in UIML. Different parts may be nested to represent a hierarchical relationship. There might be more than one structure in a UIML document representing different organisations of the same UI.

(b) **<style>**
A list of properties and values used to render the interface are stored in style. The properties are usually associated with individual parts within the UIML document through the part-names. Properties can also be associated with particular classes of parts. The properties associated with parts for Graphical User Interfaces (GUIs) could be the background colour, foreground colour, font, etc. We can also implement multiple styles within a single UIML document associated with multiple structures or even the same structure. This facilitates the use of different styles for different contexts.

(c) **<content>**
The actual content associated with the various parts of the interface is represented by content. A clean separation of the content from the structure is useful when different content is needed under different contexts. This feature of UIML is very helpful when creating interfaces that might be used in multiple languages. An example of this is a UI in French and English, for which separate content is needed.

(d) **<behaviour>**
A set of conditions and associated actions within rules specified the behaviour of an interface. UIML permits two types of conditions. The first condition is when an event occurs, while the second is true when an event occurs and the value of some data associated with the event is equal to a certain value.

There are four kinds of actions that occur the first action is to assign a value to a part's property. The second action is to call an external function or method. The third is to fire an event and the fourth action is to restructure the interface. Currently, there are platform-specific renderers available for UIML for a number of different platforms. These include Java, HTML, WML, and Voice XML. Each of these renderers has a platform-specific vocabulary associated with it to describe the UI elements, their behaviour and layout. In this way, the generic vocabulary is not a lowest-common-denominator approach. The generic UIML file would then contain three <style> elements. One is for cross-platform style, one for HTML, and one for Java UIs:

```
< uiml>

...

<style id="all Platforms">

<property id="g: title">My User Interface</property>

</style>

<style id="only HTML" source="all Platforms">

<property id="h: link-colour">red</property>

</style>

<style id="only Java" source="all Platforms">
```

```
<property id="j: resizable">red</property>

</style>

...

</uiml>
```

In the example above, both a web browser and a Java frame have a title, which is My User Interface. However, only web browsers can have the colour of their links set, so property h: link-colour is used only for HTML UIs. Similarly, only Java UIs can make themselves non-resizable, so the j: resizable property applies only to Java UIs. When the UI is rendered, the renderer will choose exactly one <style> element.

SELF-CHECK 4.2

1. The declarative XML-based language that can be used to define user interfaces is ..... .
2. GUI comprises of two main compnent ..... and ..... .
3. The UIML does not contain any platform-specific or ..... tags.
4. At the highest level, a UIML document comprises of four components ....., ..., ... and ..... .
5. The relationships between the various UI elements within the interface, is represented using ..... .
6. A list of properties and values used to render the interface are stored in ..... .
7. The feature of UIML that is very helpful when creating interfaces that might be used in multiple languages is ..... .

4.4 WHAT ARE X FORMS?

The simple X forms defines how our form looks, what our form does, where the form is of any type developed on any language. Traditional HTML Web forms do not separate the purpose from the presentation of a form. This allows for flexible presentation options, including classic XHTML forms, to be attached to an XML form definition.

The following Figure 4.1 illustrates how a single device-independent XML form definition, called the X Forms Model, and has the capability to work with a variety of standard or proprietary user interfaces.

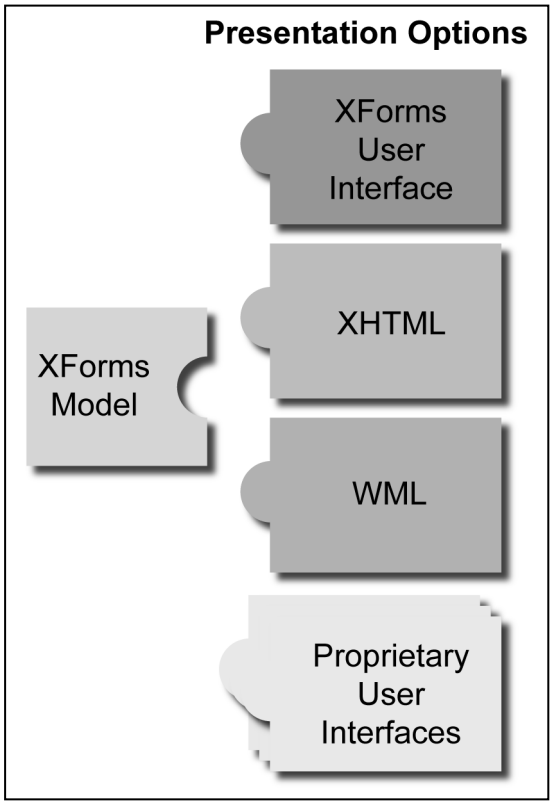


Figure 4.1: X Form

The main concept behind the X forms is that forms collect data, which is expressed as XML instance data. The X Forms Model describes the structure of the instance data. This is important, since like XML, forms represent a structured interchange of data. Workflow, auto-fill, and pre-fill form applications are supported through the use of instance data.

The X Forms User Interface provides a standard set of visual controls that are targeted toward replacing today's XHTML form controls. These form controls are directly usable inside XHTML and other XML documents, like SVG. Other groups, such as the Voice Browser Working Group, may also independently develop user interface components for X Forms.

Finally, there needs to be a channel for instance data to flow to and from the X Forms Processor. For this, the X Forms Submit Protocol defines how X Forms send and receive data, including the ability to suspend and resume the completion of a form.

The following Figure 4.2 summarises the main aspects of X Forms.

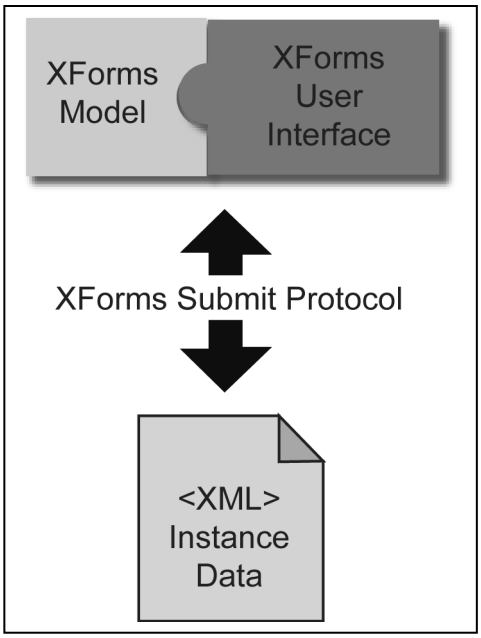


Figure 4.2: The main Aspects of X Form

4.4.1 Key Goals of X Forms

(a) The X form provide support for structured form data.

(b) Advanced forms logic without server round-tripping.

(c) The X forms provide dynamic access to server data sources during form execution.

(d) Decoupled data, logic and presentation.

(e) Seamless integration with other XML tag sets.

(f) Richer user interface to meet the needs of business, consumer and device control applications.

(g) The X forms support for handheld, television, and desktop browsers, plus printers and scanners.

(h) The X forms provide improved internationalisation and accessibility.

(i) The X form provide facility for multiple forms per page, and pages per form.

(j) The X forms provide the capability of Suspend and Resume.

EXERCISE 4.1

1. Write a short note on X forms.
2. What do you mean by user interface development?

4.5 PUTTING IT ALL TO WORK

As we know that forms are used for collecting data, so it's not we can say that the most important concept in X Forms is "instance data", an internal representation of the data mapped to the familiar "form controls". Instance data is based on XML and defined in terms of X Path's internal tree representation and processing of XML.

It might seem strange at first to associate X Path and X Forms. X Path is best known as the common layer between XSLT and X Pointer, not as a foundation for web forms. As X Forms evolved, however, it became apparent that forms needed greater structure than was possible with simple name-value pairs. With structured data comes the need to reach into the instance data to connect or "bind" form controls to specific parts of the data structure, hence X Path.

Since X Forms and XSLT are related through X Path, it is interesting to compare the two technologies. XSLT is usually described in terms of three trees, usually produced by parsing XML documents: XSLT with labelled 'source tree', 'style sheet tree', and 'result tree' and arrows.

(a) From input sources, typically documents, a "source tree" and "style sheet tree" are parsed into memory.

(b) Processing of these two trees forms a third tree, the "result tree".

(c) Upon completion of processing, the result tree is serialised, typically to a new XML document.

X Forms processing is similar, but combines input and output into the same tree: a single tree, labelled 'instance data', with arrows in and out.

(a) From an input source, either inline or an XML document on a server, "instance data" is parsed into memory.

(b) Processing of the instance data involves interacting with the user and recording any changes in the data.

4.6 ROLE OF WAP

WAP permits wireless device users to provide information from internet over wireless network, without wasting a much time and cost. WAP permits mobile devices to interact with application servers and information databases via the internet. Applications can be animatedly downloading, uploaded and run on WAP devices.

WAP is useful in numerous ways. Major profits of WAP are as below:

(a) WAP benefits the developers that develop wireless application.

(b) WAP benefits the companies that produce wireless devices.

(c) WAP benefits the network operators or service providers.

(d) WAP benefits the end-users that use applications.

4.6.1 Benefits to Developers

WAP permits developers to generate applications with more features and raises the price of application for the customer. WAP authorize developers to build up applications that can be work across diverse types of devices, browsers, gateways and networks. The wireless Transport Layer Security requirement of WAP permits developers to simply integrate protection features into their applications.

4.6.2 Benefits to Device Manufactures

WAP also supply support to WAP device inventers. WAP device producers have multiplicity of WAP sustaining micro browsers like blazer and embider and WAP operating systems like symbian and palmOS. By using WAP sustaining micro browsers and operating system in to WAP device, manufactures can extensively improve the worth of devices and users of WAP devices can access a variety of applications and services produced by a big society of web developers on WAP tools. Simultaneously, the producers are guaranteed that the micro browser and WAP operating system will function across a broad series of WAP gateways and networks without making any physical and rational modify in hard ware of WAP tool.

4.6.3 Benefits of Service Providers

WAP service provides also gain by cheering their client base to use WAP tools and supplying support of WAP gateways themselves. They permit their clients to a large number applications and content obtainable on the internet. Because service providers manage the WAP gateway, they can also manage the home page of their own WAP site. Bulky wireless content providers are recognized to expend a vast quantity to get a space on the home pages of the service providers. So that the WAP users can also access other wireless sites along with service provider's site.

4.6.4 Benefits to End-users

Clients perhaps benefit mainly between the dissimilar classes of WAP consumers. They can use any web content via HTML-to-WML formatting services. A big numbers of application developers are making content and services specially for consumers using WML. End users with WAP supporting devices can use the application or services provided by their network operator.

This is a major value intention for the consumer as they don't have to get to cooperate with interfaces. There are some negligible differences in dissimilar micro browsers.

Consumers are able to use the content and services in spite of of the service provider or the network. Moreover, the safety features of WAP permit consumers to do contented deal of perceptive information such as credit card numbers and passwords on wireless network.

4.7 ROLE OF J2ME

J2ME GUI permits you to extend mobile applications having the appearance-and- feel, in addition to functionality of desktop applications. Mobile Java developers are well conscious of the harsh boundaries of the integral user interface offered by J2ME. The interface is totally at the sympathy of the virtual machine with small or no control left to the developer. This is well for throwaway small applications, but most saleable and bigger applications need precise organization over their user interfaces, even if only for artistic functions.

Developers only have two real alternatives: Produce their own user interface by means of low-level graphical directions, or use a third-party mobile GUI. There

are only a small number of mobile GUI libraries principles permitting for, but the majority of them fall victim to some of the following troubles:

(a) Many mobile GUI libraries offer components which are as visually unattractive as that of the included user interface of J2ME.

(b) Numerous compatibility issues with many of the GUI libraries. Some declare to be companionable with approximately all mobile handsets while in realism they only work on a small subset of handsets.

(c) A small number of GUI libraries with reasonable interfaces are naturally very huge and are in fact more geared towards newer handsets. Their huge JAR file sizes depart little room, if any, for the real application and some undergo from performance issues on older handsets.

(d) Some depend on supplementary API's that are not obtainable on all handsets.

Because of these problems, many developers still end up developing their own GUI libraries from scratch, potentially delaying their projects for months. Our company faced the same problem when we started to develop our mobile applications. Since we were unable to find a suitable mobile GUI library, we decided to develop our own – the J2ME GUI library.

4.8 ROLE OF BREW

Mobile phones are an ideal means of interaction with other populace together with serving us to preserve our contact lists, to do lists etc. With time supplementary features on these devices twisted them into Smart phones which further offering phone functionality also acted as a Personal Digital Assistant. A Smart phone also permits for setting up of new applications. In the light of ever increasing list of functionalities, a Smart phone desires an efficient device to organize them all – an operating system. As of current circumstances there are a number of smart phone operating systems like Symbian, Microsoft Windows Mobile, Palm OS etc. BREW is also the one from this class.

4.8.1 BREW Overview

(a) It means Binary Runtime Environment for Wireless. BREW functions on top of hardware platform and is transportable to a variety of hardware platforms.

(b) The product by Qualcomm Internet Services – a division of Qualcomm.

(c) Currently integrated to all Qualcomm handsets, however is portable to non-Qualcomm handhelds.

(d) Compatible with any other mobile operating system.

(e) Small size – about 150kB – smaller than the rest of rival operating systems in the market.

(f) It can work on CDMA or GSM/GPRS or UMTS handhelds. But at present it runs only on CDMA devices.

(g) Applications development languages C, C++, Java using a Java Virtual Machine developed for BREW platform.

(h) BREW can be incorporated into vendors' devices by means of the BREW Porting kit granted by Qualcomm.

BREW application environment has an event driven architecture such as Microsoft Windows, Macintosh and other operating systems have. Two types of applications can be formed on BREW. One of them is Applet and the second is Extension. Applet is a GUI application with user interface and reusable code summarized in one or more classes. An expansion is a reusable code printed by OEMs to insert functionalities to the handsets.

A BREW application is a single Module object and one or more applets and classes. An Applet or Class must be allocated a unique 32 bit identifier known as Class ID. Class IDs are required to be used during expansion procedure.

Each BREW application must be transported with a matching Module Information File connected with it. MIF comprises essential information about a BREW application:

(a) Supported classes

(b) Supported applets

(c) Applet details

(d) Class IDs assigned to applets and classes

Module Information file is produced with BREW MIF Editor tool incorporated in SDK. Resource files can comprise speech and tool specific data for BREW applications therefore facilitating the localization.

4.8.2 Role of Microsoft Platform

At its current Mobility Developer discussion, Microsoft laid out its procedure for facilitating developers to generate next-generation location-aware applications. The new platform involves some existing hardware, software, and services, in addition to new server products designed to reduce the time to advertise for corporations that desires to provide products relying on the platform to their consumers.

Mobility Landscape

With the numeral new PCs procured each year (as divergent to those procured to replace obtainable systems) commencing to reject, Microsoft known that the next conflict will be fought for the software that is used by the millions of transportable devices that will be procured over the next decade. Cell phones, BlackBerrys, pagers, Palm-powered PCs, Pocket PCs, and most laptop and tablet PCs will make up the huge mass of new computing tools. And they all have a common process- they're de intended to function when not attached to the corporate group. Furthermore, the huge mass of the tools will also not be processing a Microsoft operating system. This means that Microsoft has to discover a way to produce returns from the tools, and it has to offer its consumers with a method to use its software to offer services on the tools.

SELF-CHECK 4.3

1. Is the Microsoft platform good for present days mobile computing environment justify your answer by giving suitable examples.
2. In which field do we use J2ME in mobiles?

Hardware

There are three versions of Windows Mobile for various hardware devices: Windows Mobile Professional runs on 'Windows Phones' (Smartphone) with touch screens.

(a) 'Windows Phones' runs Windows Mobile Standard on with regular screens. (b) 'Windows Mobile Classic devices' (Pocket PCs) runs on Windows Mobile

(b) Classic.

SUMMARY

- The goal of a user interface design is to make the user's interaction as simple and efficient as possible, in terms of accomplishing user goals – what is often called user – centred design.
- The goal of user interface design User interface design or user interface engineering is the design of computers, appliances, machines, mobile communication devices, software applications, and websites with the focus on the user's experience and interaction. X Forms may be defined as an XML format for the specification of a data processing model for XML data and user interface(s) for the XML data, such as web forms, etc.

KEY TERMS

FSK Frequency Shift Keying

MDBS Mobile Data Base Station

MDIS Mobile Data Intermediate Systems

MSK Minimum Shift Keying

SELF-TEST

1. Define the term GUID.
2. What are X forms? Define in detail.
3. How are X forms related to the UML?
4. What are the key goals for the X forms?

Topic 5 ▶ VUIs and Multimodal User Interfaces

LEARNING OUTCOMES

By the end of this topic, you should be able to:

1. Define quality of speech, voice transcription and voice recognition;
2. Describe text-to-speech technologies;
3. Underline the concept of multimodal content;
4. State the term internationalization and localization; and
5. Recognise the evolving definition of multimodality.

▶ INTRODUCTION

In the previous topic, we have studied the Generic User Interface Development, User Interface Development, building Generic User Interfaces, and using UML for Modeling Generic User Interfaces Components, X Forms and developing Mobile GUIs. Now, in this topic, we would focus on VUIs and Mobile Applications.

With mobile technology taking center stage in today's world, organisations are eager on making major advancements in the mobile space. As mobile technology keeps the customers stay connected for almost all the time, enterprises are moving beyond the desktop world to keep attuned to their customer's needs. An Interactive Voice Response (IVR), processes inbound phone calls, plays recorded messages including information extracted from databases and the internet, and potentially routes calls to either in house service agents or transfers the caller to

an outside extension. It enhances our phone system's current robust features of predictive dialing, ACD, and digital call recording. VUI (Voice User Interface) is the term used to describe the interaction with computers through voice/speech platform in order to initiate an automated service or process. Voice User Interface (VUI) systems can be used for many different phone applications.

5.1 VOICE USER INTERFACE AND MOBILE APPLICATIONS

The VUI is the interface to any speech application. Voice User Interface Design discloses the design principles and practices to produce commercial success in a period when effective ASRs (automated speech recognition) are not toys but competitive necessities. VUI (Voice User Interface) is commonly defined as the following:

"A Voice User Interface (VUI) is the term used to describe the interaction with computers though voice/speech platform in order initiate an automated service or process."

VUIs have become more commonplace, and people are taking advantage of the value that these hands-free, eyes-free interfaces provide in many situations, with advances in technology. VUIs need to respond to input reliably, or they will be rejected and often ridiculed by their users. Voice User Interface (VUI) systems can be used for many different phone applications.

5.1.1 Quality of Speech

The evolution of speech quality contains several problems (Jekosh 1993, Mariniak 1993). Speech quality is a multi-dimensional term. The evaluation methods of speech quality are usually designed to test speech quality in general, but most of them are suitable also for synthetic speech. It is very difficult, almost impossible; to say which test method provides the correct data. In a text-to-speech system not only the acoustic characteristics are important, but also text pre-processing and linguistic realisation determine the final speech quality. Separate methods usually test different properties, so for good results more than one method should be used and how to assess the test methods themselves.

The evaluation procedure is usually done by subjective listening tests with response set of syllables, words, sentences, or with other questions. The test material is usually focused on consonants, because they are more problematic to synthesise than vowels. Especially nasalised consonants (/m/ /n/ /ng/) are usually considered the most problematic (Carlson et al. 1990). When using low

bandwidth, such as telephone transmission, consonants with high frequency components (/ f/ /th/ /s/) may sound very annoying. Some consonants (/d/ /g/ /k/) and consonant combinations (/dr/ /gl/ /gr/ /pr/ /spl/) are highly intelligible with natural speech, but very problematic with synthesised one. Especially final /k/ is found difficult to perceive. The other problematic combinations are for example, /lb/, /rp/, /rt/, /rch/, and /rm/ (Goldstein 1995).

Some objective methods, such as Articulation Index (AI) or Speech Transmission Index (STI), have been developed to evaluate speech quality (Pols et al. 1992).

These methods may be used when the synthesised speech is used through some transmission channel, but they are not suitable for evaluating speech synthesis in general. This is because there is no unique or best reference and with a TTS system, not only the acoustic characteristics are important, but also the implementation of a high-level part determines the final quality. However, some efforts have been made to evaluate objectively, for example, the quality of automatic segmentation methods in concatenative synthesis.

To improve the test results we repeated the test procedure to the same listening group, the test results may increase significantly by the learning effect which means that the listeners get familiar with the synthetic speech they hear and they understand it better after every listening session. Concentration problems, on the other hand, may decrease the results especially in segmental methods. Therefore, the decision of using naive or pro listeners in listening tests is important.

During last decades several individual test methods for synthetic speech have been developed. Some researchers even complain that there are too many existing methods which make the comparisons and standardisation procedure more difficult. On the other hand, there is still no test method to give undoubtedly the correct results. The most commonly used methods are introduced in this topic.

Also some computer software's have been developed for making the test procedure easier to perform. One of these is for example the SAM SOAP (A Speech Output

Assessment Package) which is implemented in PC-environment and contains several different test methods (Howard-Jones et al. 1991).

Segmental Evaluation Methods

To test a single segment or phoneme intelligibility is tested by segmental evaluation methods. The very commonly used method to test the intelligibility of synthetic speech is the use of so called rhyme tests and nonsense words. The rhyme tests have several advantages. The number of stimuli is reduced and the test procedure is not time consuming. Also naive listeners can participate without having to be trained and reliable results can be obtained with relatively small subject groups, which is usually from 10 to 20. The learning effects can also be discarded or measured. With these features the rhyme tests are easy and economic to perform.

The obtained measure of intelligibility is simply the number of correctly identified words compared to all words and diagnostic information can be given by confusion matrices. Confusion matrices give information how different phonemes are misidentified and help to localise the problem points for development.

However, rhyme tests have also some disadvantages. With monosyllabic words only single consonants are tested, the vocabulary is also fixed and public so the system designers may tune their systems for the test, and the listeners might remember the correct answers when participating in the test more than once.

Rhyme tests are available for many languages and they are designed for each language individually. The most famous segmental tests are the Diagnostic and Modified Rhyme Tests described below. Some developers or vendors, such as Bell core and AT&T have also developed word lists for diagnostic evaluation of their own.

Haskins Sentences

Haskins sentences are also developed to test the speech comprehension in a sentence or word level. Unlike in Harvard sentences, the test material is anomalous which means that the missed items cannot be concluded from context as easily as with use of meaningful sentences (Pisoni et al. 1980). As in Harvard sentences, a fixed set of sentences is used and due to learning effect the test subjects can be used only once for reliable results. The first five sentences of the test material are (Allen et al. 1987):

(a) The wrong shot led the farm

(b) The black top ran the spring

(c) The great car met the milk

(d) The old corn cost the blood

(e) The short arm sent the cow

It is easy to see that these sentences are more difficult to perceive than Harvard sentences and they are not faced in real life situations.

Semantically Unpredictable Sentences (SUS)

The SUS-test is also an intelligibility test on sentence level (Goldstein 1995, Pols et al. 1992). The words to be tested are selected randomly from a pre-defined list of po ss ibl e c an did ate s. Th ese a re mos tl y m on o-s yll ab ic words wi th so me expectations. The test contains five grammatical structures described withexamples in Table 5.1 below. As in Haskins sentences, the missed item cannot be concluded from textual context.

Table 5.1: Grammatical Structures in SUS-test (Jekosh 1993)

	Structure	Example
1	Subject - verb - adverbial	The table walked through the blue truth.
2	Subject - verb - direct object	The strong way drank the day.
3	Adverbial - verb - direct object	Never draw the house and the fact.
4	Q-word - transitive verb - subject - direct object	How does the day love the bright word.
5	Subject - verb - complex direct object	The plane closed the fish that lived.

In the actual test, fifty sentences, ten of each grammatical structure, are generated and played in random order to test subjects. If the test procedure is run more than once, a learning effect may be observed. But because the sentence set is not fixed, the SUS-test is not as sensitive to, for example, the learning effect as previously described test sentences.

Comprehension Tests

Most of the test methods above are used to test how the single phoneme or word is recognised. In comprehension tests a subject hears a few sentences or paragraphs and answers to the questions about the content of the text, so some of the items may be missed (Allen et al. 1987). It is not important to recognise one

single phoneme, if the meaning of the sentence is understood, so the 100% segmental intelligibility is not crucial for text comprehension and sometimes even long sections may be missed (Bernstein et al. 1980). No significant differences were obtained in understanding between natural and synthetic voice (Goldstein 1995).

Only with prosody and naturalness the differences are perceptible which may also influence to the concentration of test subjects.

Prosody Evaluation

Evaluation of the prosodic features in synthesised speech is probably one of the most challenging tasks in speech synthesis quality evaluation. Prosody is also one of the least developed parts of existing TTS systems and needs considerable attention for the research in the future. Prosodic features may be tested with test sentences which are synthesised with different emotions and speaker features.

The listeners' task is to evaluate for example with five level scales how well the certain characteristic in speech is produced. Evaluation may be made also by other kind of questions, such as "Does the sentence sound like a question, statement or imperative".

5.1.2 Voice Transcription

The concept of transcription says that it is the process of taking data in one format and transferring it to another. In many cases, the audio dictation is captured digitally, or with audio tape, and transferred to either a paper or electronic document. In the case of legal documents, there are several advantages of using a legal transcription company in your practice.

The voice transcription industry changes rapidly. Technological advances as well as widening security concerns, as evidenced by regulatory requirements, are converging to bring about a new level of sophistication and integration of voice technology into everyday use. Outsourcing Transcription Services, moving in line with the changing and demanding requirements of global clients, has been able to stay ahead of our competitors in the industry, by providing professional Voice Transcription services.

The scope of voice transcription outsourcing is infinite when considering the mission-critical nature of various tasks carried out in industries such as Health Care, Medical, Insurance, Legal, and Media.

5.1.3 Voice Recognition

Voice or speech recognition is the ability of a machine or program to receive and interpret dictation, or to understand and carry out spoken commands. In other words, we can say that the process of Voice recognition is defined as the ability of a machine or program to identify words and phrases in spoken language and convert them to a machine-readable format. Rudimentary speech recognition software has a limited vocabulary of words and phrases and may only identify these if they are spoken very clearly. More sophisticated software has the ability to accept natural speech.

For use with computers, analogue audio must be converted into digital signals. This requires analogue to digital converter. For a computer to decipher the signal, it must have a digital database, or vocabulary, of words or syllables, and a speedy means of comparing this data with signals. The speech patterns are stored on the hard drive and loaded into memory when the program is run. A comparator checks these stored patterns against the output of the A/D converter.

ACTIVITY 5.1

1. Explain the term Quality of speech.
2. What do you mean by voice transcription and voice recognition? Write short note on voice transcription and voice recognition.

5.1.4 Text to Speech Technologies Converting Written Language to Spoken Language

This service incorporates the experiences gained in the domain of Interactive Voice Response (IVR) systems during the last three years. The work described in this section is part of the Quick Phone project at UBILAB. The Quick Phone research project started in 1992. Its aim is to investigate new technologies for building state-of-the-art telephone information services. So far, a prototype of a phone banking service has been built. Different aspects of the service are of interest and some of them can be classified as novel in comparison with existing ones. One important aspect is the user interface which makes use of an application independent, list-based Navigation technique. The user interface also supports both voice and key input. Voice recognition vocabularies, including the digits and a set of application independent navigation commands, were trained in the three languages German, French and Italian on a homogeneous distributed Swiss population. It has been shown that the voice input technology is ready for

building easy-to-use and efficient telephone voice interfaces. In order to apply IVR systems successfully in current and future banking environments additional aspects should be considered as well.

The first is to gain more flexibility by using Text-to-Speech (TTS) as the voice output technology instead of, or in addition to, playing pre-recorded phrases, words and numbers as in most current systems. The second is to build more secure systems by using speaker identification and verification techniques for applications with restricted access, such as account information or transaction services. Both technologies are to be investigated in the Quick Phone project. The goal, again, is to show the usability and reliability of the technology for operational services at UBS. As a first step toward TTS integration we launched a 13-week student project with the aim of investigating the state-of-the-art technology of the TTS systems currently available. Its results are described in this report. In Europe, and especially in Switzerland, IVR systems must support multiple languages. Many TTS systems on the market are designed for different languages and different hardware and software environments. All of them have their merits and draw backs in terms of speech quality, software and hardware requirements, and costs. Only a few of them, however, support multiple languages. For this project only systems supporting multiple languages were considered.

Rating System for the Evaluation of Name Pronunciation

The overall impression is that when pronouncing names none of the three systems provide satisfactory output. LHS even refused any names with letters specific to foreign languages such as “é”, “à”, etc. Therefore, none of the systems is suitable for applications where name pronunciation is crucial. It is important to note that fairly common abbreviations, such as “Frl.”, "Dr.", and “Prof.”, were not treated properly by all systems. Some were not recognised at all. Others resulted in unnecessary pauses, probably due to incorrect sentence terminations. To do justice to the systems evaluated, we must emphasise that they had to perform a task for which they were not designed. It is shown in the literature that systems with large lexicons of names are rarely used. Specialised name synthesisers are necessary, and such modules are currently being developed, at least for English. The methods used in Anglophone systems are, however, probably not directly usable for German name pronunciation systems. In English, all names are pronounced anglicized, at least to a certain degree, whereas in German, names are usually pronounced as they are in the original language. We believe that neither general purpose synthesisers with a large name dictionary nor specialised name synthesisers as stand-alone systems are suitable for real world applications such as automatic ordering systems. Our vision of a flexible and open solution consists of first determining the language were a name comes from, and then,

pronouncing it according to the rules of that language. This can be done by a "language detector" followed by a combination of a general purpose synthesiser for ordinary text and a name synthesiser for names for every language. The determination of the language of origin of names is still an interesting research challenge.

Can you convert written language in to spoken language if yes then how can you do this?

5.2 MULTICHANNEL AND MULTIMODAL USER INTERFACES

Multi-channel panning permits the operator to observe how a manipulated source signal will be heard by a listener at a reference point in a sound space. The panner user interface (UI) displays a separate visual element for each channel of source audio, thus the operator can see how each channel will be heard by a listener at a reference point in the sound space. Each visual element may depict the apparent point of origination of its corresponding source channel. Each visual element may depict the apparent width of origination of its corresponding source channel.

Each visual element may depict the amplitude gain of its corresponding source channel. An operator can choose a mix of attenuating or collapsing behavior when panning. Moreover, the visual elements depict the relative proportions of attenuating and collapsing behavior.

5.2.2 Multimodal Contents

Device and Language

In his theory of action, Norman structures the execution and evaluation lfs in terms of semantic and articulator distances that the user needs to cover in order to reach a particular goal (Norman 86). This user-centred approach pays little attention to the processing steps that occur within the computer system. Our Pipe-lines model makes these stages explicit (Nigay 1994). By so doing, we extend Norman's theory in a symmetric way within the computer system. Two relevant concepts emerge from this model: the notion of physical device and that of interaction language. Interestingly, these concepts cover the semantic and articulatory distances of Norman's theory. A physical device is an artifact of the system that acquires (input device) or delivers (output device) information. Examples include keyboard, loudspeaker, head-mounted display and GPS.

Although this notion of device is acceptable for an overall analysis of an interactive multimodal system, it is not satisfactory when one needs to characterise the system at a finer grain of Design space for multimodal interaction.

The design spaces of input devices such as that of Mackinlay. (Mackinlay et al., 1990) and of Foley (Foley et al., 1994) are frameworks that valuably refine a physical device. A review of these taxonomies is presented in (Nigay, 1996). An interaction language is a language used by the user or the system to exchange information. A language defines the set of all possible well-formed expressions, i.e., the conventional assembly of symbols that convey meaning. Examples includepseudo-natural language, direct manipulation language.

Interaction Modality and Multimodality

In the literature, interaction modality is discussed at multiple levels of abstraction from both the user and the system perspectives. At the lowest level, a modality may refer to a human sensory capability or to a computer physical device such as a microphone, a camera, or a screen. At a higher level of abstraction, a modality is viewed as a representational system, such as a pseudo-natural language that the user and the system might share. Whereas the device level is related to the human sensory capabilities, the representational level calls upon cognitive resources.

Clearly, the physical and the representational computer models are tightly coupled to the sensory and cognitive dimensions of human behaviour. For this reason, in (Nigay 95) we define a modality as the coupling of an interaction language with a physical device d: <d, L>. Examples of input modalities while using a Laurence NigayPDA (Zouinar et al., 2003) include: <microphone, pseudo natural language>, <camera, 3D gesture>, <stylus, direct manipulation> and <PDA, 3D gesture> (Embodied user interface (Harrison et al. 1998)). Within the vast world of possibilities for modalities, we distinguish two types of modalities: the active and passive modalities. For inputs, active modalities are used by the user to issue a command to the computer (e.g., a voice command or a gesture recognised by a camera). Passive modalities refer to information that is not explicitly expressed by the user, but automatically captured for enhancing the execution of a task. For example, in the “Put that there” seminal multimodal demonstrator of R. Bolt (Bolt 1980), eye tracking was used for detecting which object on screen the user is looking at.

Combination of Modalities

Although each modality can be used independently within a multimodal system, the availability of several modalities in a system naturally leads to the issue of their combined usage. The combined usage of multiple modalities opens a vastly augmented world of possibilities in user interface design. Several frameworks addressed the issue of relationships between modalities. In the seminal TYCOON framework (Martin 1997) six types of cooperation between modalities are defined:

(a) Equivalence involves the option of choosing between several modalities that can all equally well convey a particular chunk of information.

(b) Specialisation implies that specific kinds of information are always conveyed by the same modality.

(c) Redundancy indicates that the same piece of information is conveyed by several modalities.

(d) Complementarily denotes several modalities that convey complementary chunks of information.

(e) Transfer implies that a chunk of information processed by one modality is then treated by another modality.

(f) C on curren cy d es crib es th e ca se o f se ve ra l m od al it ie s co nv ey in g independent information in parallel.

5.3 SOFTWARE AND SYSTEM ARCHITECTURES FOR DELIVERING MULTIMODALITY

This document describes the architecture of the Multimodal Interaction (MMI) framework and the interfaces between its constituents. The MMI Working Group is aware that multimodal interfaces are an area of active research and those commercial implementations are only beginning to emerge. Therefore, we do not view our goal as standardising a hypothetical existing common practice, but rather providing a platform to facilitate innovation and technical development. Thus, the aim of this design is to provide a general and flexible framework providing interoperability among modality – specific components from different vendors – for example, speech recognition from one vendor and handwriting recognition from another. This framework places very few restrictions on the individual components or on their interactions with each other, but instead focuses on providing a general means for allowing them to communicate with each other, plus basic infrastructure for application control and platform services.

Our framework is motivated by several basic design goals:

(a) **Distribution**
The architecture should support both distributed and co-hosted implementations.

(b) **Encapsulation**
The architecture should make no assumptions about the internal implementation of components, which will be treated as black boxes.

(c) **Recursiveness**
The architecture should allow for nesting, so that an instance of the framework consisting of several components can be packaged up to appear as a single component to a higher-level instance of the architecture.

(d) **Extensibility**
The architecture should facilitate the integration of new modality components. For example, given an existing implementation with voice and graphics components, it should be possible to add a new component (for example, a biometric security component) without modifying the existing components.

(e) **Modularity**
The architecture should provide for the separation of data, control, and presentation.

Even though multimodal interfaces are not yet common, the software industry as a whole has considerable experience with architectures that can accomplish these goals. Since the 1980s, for example, distributed message-based systems have been common. They have been used for a wide range of tasks, including in particular high-end telephony systems. In this paradigm, the overall system is divided up into individual components which communicate by sending messages over the network. Since the messages are the only means of communication, the internals of components are hidden and the system may be deployed in a variety of topologies, either distributed or co-located. One specific instance of this type of system is the DARPA Hub Architecture, also known as the Galaxy Communicator Software Infrastructure. This is a distributed, message-based, hub-and-spoke infrastructure designed for constructing spoken dialogue systems. It wasdeveloped in the late 1990's and early 2000's under funding from DARPA. This infrastructure includes a program called the Hub, together with servers which provide functions such as speech recognition, natural language

pro cessing, and dial ogue management. The servers communicate with the Hub and with each other using key-value structures called frames.

Another recent architecture that is relevant to our concerns is the Model-View-Controller (MVC) paradigm. This is a well known design pattern for user interfaces in object oriented programming languages, and has been widely used with languages such as Java, Smalltalk, C, and C++. The design pattern proposes three main parts: a Data Model that represents the underlying logical structure of the data and associated integrity constraints, one or more Views which correspond to the objects that the user directly interacts with, and a Controller which sits between the data model and the views. The separation between data and user interface provides considerable flexibility in how the data is presented and how the user interacts with that data. While the MVC paradigm has been traditionally applied to graphical user interfaces, it lends itself to the broader context of multimodal interaction where the user is able to use a combination of visual, aural and tactile modalities.

5.3.1 Design versus Run-Time Considerations

At the design level, we assume that multimodal applications will take the form of multiple documents from different namespaces. In discussing the design of MMI systems, it is important to keep in mind the distinction between the design-time view (i.e., the markup) and the run-time view (the software that executes the mark-up). In many cases, the different namespaces and mark-up languages will correspond to different modalities, but we do not require this. A single language may cover multiple modalities and there may be multiple languages for a single modality.

At run-time, the MMI architecture features loosely coupled software constituents that may be either co-resident on a device or distributed across a network. In keeping with the loosely-coupled nature of the architecture, the constituents do not share context and communicate only by exchanging events. Though nothing in the MMI architecture requires that there be any particular correspondence between the design-time and run-time views, in many cases there will be a specific software component responsible for each different markup language (namespace).

5.3.2 Overview of Architecture

Here is a list of the Constituents of the MMI Multimodal Interface architecture Figure 5.1.

(a) The Interaction Manager, which coordinates the different modalities. It is the Controller in the MVC paradigm.

(b) The Data Component, which vides the common data model and represents the Model in the MVC paradigm.

(c) The Modality Components, which provide modality-specific interaction capabilities. They are the Views in the MVC paradigm.

(d) The Runtime Framework, which provides the basic infrastructure and enables communication among the other Constituents.

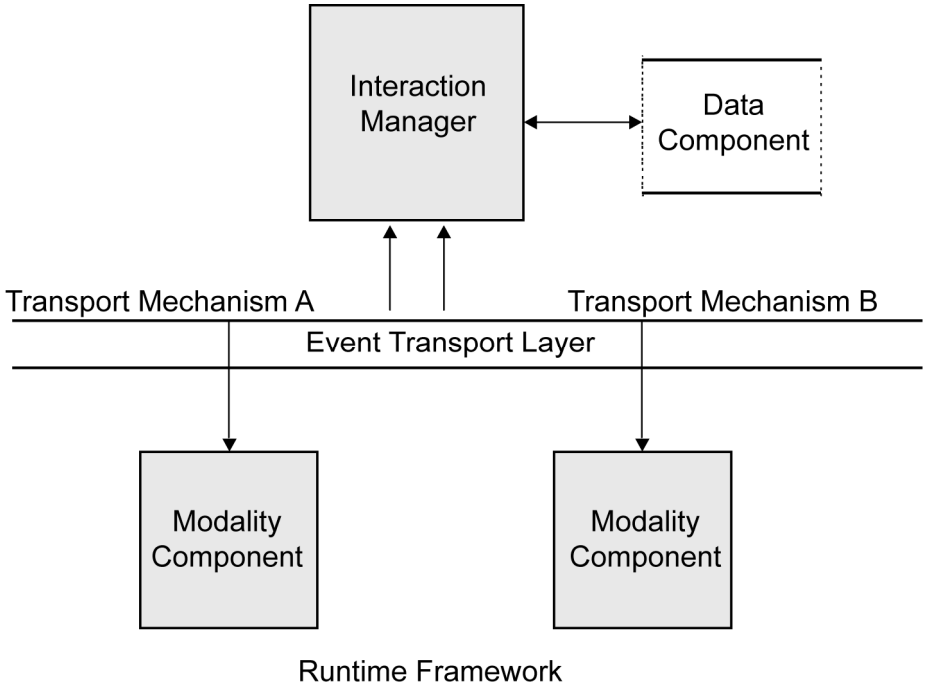


Figure 5.1: Overview of architecture

5.3.3 Internationalisation and Localisation

In computing, internationalisation and localisation means of adapting computer software to different languages and regional differences. The main idea behind the Internationalisation is to develop the process of designing a software application in such a way that the developed application can be adapted to

various languages and regions without engineering changes. On the other hand, the localisation may be defined as the process of adapting internationalised software for a specific region or language by adding locale-specific components and translating text.

Many of the companies, like Microsoft, IBM and Sun Microsystems, use the combination of internationalisation and localisation and term as "globalisation".

Scope

Focal points of internationalisation and localisation efforts include:

(a) **Language**

(i) Computer-encoded text

- Te xt processing differences, such as the concept of capitalisation which exists in some scripts and not in others, different text sorting rules, etc.
- Alphabets/scripts; most recent systems use the Unicode standard to solve many of the character encoding problems.
- Different systems of numerals.
- To provide the writing direction which is for example, left to right in German, right to left in Persian, Hebrew and Arabic.
- Spelling variants for different countries where the same language is spoken, e.g., localisation (en-US, en-CA, en-GBoed) vs. localisation (en-GB, en-AU).

(ii) Input

- Enablement of keyboard shortcuts on any keyboard layout.

(iii) Graphical representations of text (printed materials, online images containing text).

(iv) Spoken (Audio).

(v) Subtitling of film and video.

(b) **Culture**

(i) Images and colours: issues of comprehensibility and cultural appropriateness.

(ii) Names and titles.

(iii) Government assigned numbers (such as the Social Security number in the US, National Insurance number in the UK, and Resident registration number in South Korea) and passports.

(iv) Telephone numbers, addresses and international postal codes.

(v) Currency (symbols, positions of currency markers).

(vi) Weights and measures.

(vii) Paper sizes.

(c) **Writing Conventions**

(i) Date/time format, including use of different calendars.

(ii) Time zones (UTC in internationalised environments).

(iii) Formatting of numbers (decimal separator, digit grouping).

(d) Any other aspect of the product or service that is subject to regulatory compliance.

The distinction between internationalisation and localisation is subtle but important. Localisation is the addition of special features for use in a specific locale while Internationalisation is the adaptation of products for potential use virtually everywhere. The process of Internationalisation is done one time per product; on the other hand localisation is done once for each combination of product and locale. The processes are complementary, and must be combined to lead to the objective of a system that works globally. Subjects unique to localisation include:

(a) Local content

(b) Local customs

(c) Language translation

(d) National varieties of languages (see language localisation)

(e) Cultural values and social context

(f) Special support for certain languages, such as East Asian languages

(g) Symbols

(h) Order of sorting (Collation)

(i) Aesthetics.

How can we use the concept of localisation and internationalisation in Mobile phones?

5.4 THE EVOLVING DEFINITION OF MULTIMODALITY

The fourth part of the content representation are a number of multimodal conce pts which are bindi ngs of IDTs with correspondin g proposit ional formulations. The imagistic parts of these multimodal concepts are characterised by under specification, i.e. they contain more than one alternative interpretation and thus represent abstract concepts. It is this under specification which draws the distinction between the imagistic part of the multimodal concepts and the imagistic description of concrete spatial objects. For example, the property of being "longish" can be represented as an underspecified IDT in which one axis dominates the other one or two axes, as well as in terms of a logical formula (e.g. "longish(X)").

Such an underspecified IDT can be matched with any other IDT by means of a formal graph unification algorithm as described in. Currently, multimodalconcepts for dimensional adjectives (longish, round, tall, etc.), stereotyped object shapes (box, circle, etc.), and basic spatial relations (right-of, side-by-side, above, etc.) are predefined as part of long-term memory. Matching of IDT-structures is realised by the unification procedure which also determines the necessary geometrical transformations for one IDT.

The modality is defined as follows:

“Perception via one of the three perception-channels. You can distinguish the three modalities: visual, auditive, and tactile (physiology of senses).”

(a) **Visual**
Concerned with, used in seeing (comp. against optical)

(b) **Auditive**
Related to the sense of hearing (comp. against acoustical)

(c) **Tactile**
Experienced by the sense of touch

(d) **Haptic**
Most authors are using “tactile” and “haptic” as synonyms. However, in tactile as perception-modality is distinguished from haptic as output manner.

(e) **Optics/Optical**
Optics is the theory on light as well as the infrared and ultraviolet radiation. Thus the attribute “optical” refers to physical quantities and laws rather than to physiological ones. (comp. against visual)

(f) **Acoustics/Acoustical**

Acoustics is the theory on vibrations and oscillations in elastic mediums, especially of sound, its generation, spreading, and its reception. Thus, the attribute "acoustical" refers to physical rather than to physiological quantities and laws. (comp. against auditive)

The author defines only three modalities and associates them with three of the human senses. Although they will be the three ones considered in, there are some more senses as defined by physiology.

Table 5.2: Different Senses and their Corresponding Modalities

Sensory Perception	Sense Organ	Modality
Sense of sight	Eyes	Visual
Sense of hearing	Ears	Auditive
Sense of touch	Skin	Tactile
Sense of smell	Nose	Olfactory
Sense of taste	Tongue	Gustatory
Sense of balance	Organ of equilibrium	Vestibular

In our opinion, the sense of smell and the sense of taste are not very interesting for our concerns. However, the sense of balance seems to become more and more interesting with respect to virtual reality environments. Presently it is already used in flight simulators for example.

Whenever more than two of these modalities are involved, we will speak of multimodality. To be more precise, in some cases we will also use the term bimodal (or bimodality) to denote the usage of exactly two different modalities. In this sense, every human-computer interaction has to be considered as multimodal, because the user looks at the monitor, types in some commands or moves the mouse (or some other device) and clicks at certain positions, hears the reaction (beeps, key clicks, etc.) and so on.

Therefore, in our understanding of multimodality is restricted to those interactions which comprise more than one modality on either the input (i.e., perception) or the output (i.e., control) side of the loop and the use of more than one device on either side. Thus, the combination of, e.g., visual, auditive, and tactile feedback which is experienced by typing on a keyboard is explicitly excluded, whereas the combination of visual and auditive output produced by the monitor and a loudspeaker when an error occurred is a 'real' multimodal (or in this case - bimodal) event.

EXERCISE 5.3

1. Explain multimodal content.
2. Explain the evolving definition of multimodality.

SUMMARY

- Voice or speech recognition is the ability of a machine or program to receive and interpret dictation, or to understand and carry out spoken commands. The concept of transcription says that it is the process of taking data in one format and transferring it to another.

KEY TERMS

CS-CDPD	Circuit Switching Cellular Digital Packet Data
PCSI	Pacific Communication Systems Inc.
RF	Radio Frequency

SELF-TEST

1. Define the segment evaluation method.
2. What is Rhyme Test?
3. Write short notes on CLID and DMCT.
4. What is multichannel and multimodal user interface?
5. Differentiate between internationalisation and localisation.
6. Define the term VUI.

Topic 6 ▶ Mobile Agents and Peer to Peer Architecture for Mobile Applications

LEARNING OUTCOMES

By the end of this topic, you should be able to:

1. Define mobile agents;
2. Describe mobile agents for mobile computing;
3. Identify UML extensions for mobile agents;
4. Underline the applications for mobile agents and its implementation tools;
5. Recognise development problem with mobile agents; and
6. Describe mobile peer-to-peer computing.

▶ INTRODUCTION

In the previous topic, we have studied the Multichannel and Multimodal User Interfaces, Modeling Multichannel and Multimodal Applications with UML, Multimodal Content, Software and System Architectures for Delivering Multimodality, Internationalization and Localization and the evolving definition of Multimodality. Now, in this topic, we would focus on Mobile Agents and

Peerto- Peer Architectures for Mobile Applications. A mobile agent may be defined as the computer program which represents a user in a computer network and migrates from one computer to another computer and continues the execution on the destination computer.

6.1 MOBILE AGENT

Mobile agents are programs being sent across the network from the client to the server or vice-versa. An agent that can be executed after being transferred over the network will be called an agent host. A software agent is a common name and describes a software entity that computerises some of the regular or difficult tasks on behalf of human or other agents. Mobile agents can travel in network following their itinerary and carrying logic and data to perform a set of management tasks at each of the visited nodes in order to meet their designed objectives.

Mobile agents are a powerful software interaction model that let a program to be moved between hosts for remote execution. They are solutions for managing distributed networks. Mobile agents allow the transformation of current networks into remotely programmable platforms. The concept of remote programming using mobile agents is considered as an alternative to the traditional client-server programming based on the remote procedure call or the static distributed object paradigm (e.g., CORBA).

A software agent is recognised by a life-cycle model, a computational model, a security model, and a communication model. But a mobile agent is additionally identified by a basic agent model and navigation model. The primary goal of using mobile agents in management of telecommunication network is reducing network traffic by using load balancing and building scalable and reliable distributed network management system.

Some of the advantages of using agent technology in telecommunication networks are as follows:

(a) Deals with huge amount of data, which agents can search, collect and filter.

(b) Allows more intelligence to be used in managing a network, integration of different services to value added services and negotiation of quality of service.

(c) Develops higher level communication, and organising a network.

(d) Re-activeness, agents can react quickly to local events, such as the breakdown of a link.

(e) Robustness, agents can carry out their tasks at least to a degree, even if parts of the network are not reachable temporarily. This is important in mobile computing, where links are expensive and unstable.

(f) Distributes management code to the SNMP agents for reducing bandwidth in wireless network.

(g) Decentralises network management functions. Mobile agents can autonomously, proactively carry out administration tasks and reduce traffic need for management.

(h) Dynamically changes network policies. Mobile agents can change the rules underlying network management from time-to-time.

(i) Network monitoring, mobile agents are useful for supervision of SNMP variables and long-term controlling of network elements, especially in wireless network as the configuration might change over time.

In order to perform the above properties, agents must communicate to find their peers, to co-operate and negotiate in open environments. It is significant that agent systems build on an interface with a diversity of existing and upcoming standards at the underlying network systems level.

6.2 MOBILE AGENTS FOR MOBILE COMPUTING

We can say that a mobile agent is a process that can transport its state from one environment to another. Mobile agent has the ability to decide when and where to move. When the mobile agent decides to move, it saves its own state and transports this saved state to the new host and resumes the execution there from the saved state.

Mobile agents are mobile objects or programs that carry executable code and data within them. They have several features that help them achieve their goals or business functions such as negotiating and ordering.

6.2.1 Features of Mobile Agents

Some of the main features of Mobile Agents are:

(a) **Autonomy**

Autonomy means that the mobile agents have the freedom to move from one location to another according to their own choice. The mobile agents also decide when they want to move from one host to another host.

(b) **Adaptive Learning**
Adaptive learning means that the mobile agents adapt themselves in any environment.

(c) **Mobility**
Mobility implies that the mobile agents have the ability to move from one host to another host in the network.

SELF-CHECK 6.1

Fill in the blanks:

1. Mobile agents are programs being sent across the ..... from the client to the server or vice-versa.
2. .....can travel in network following their itinerary and carrying logic and data to perform a set of management tasks at each of the visited nodes in order to meet their designed objectives.
3. Mobile agents are solutions for managing..... .
4. ..... means that the mobile agents have the freedom to move from one location to another according to their own choice.
5. ..... implies that the mobile agents have the ability to move from one host to another host in the network.

6.3 UML EXTENSIONS FOR MOBILE AGENTS

Mobility is not only important as a programming concept. It is inherently introduced by the advent of mobile devices such as laptops, mobile phones, PDAs, etc. Therefore, it has to be appropriately represented in corresponding models. In this context, it is worth mentioning that the approaches previously described can be used when there are only two kinds of entities: mobile agents and static locations; they are not well suited for mobile computing modelling. In order to deal with it, a few UML extensions have been proposed. Kosiuczenko et al., have extended UML sequence diagrams to model objects that are mobile and can play the role of locations too. These diagrams generalise the concept of object lifeline of Use Case Maps, to model complex mobility patterns, providing also

the possibility to abstract away from irrelevant details. Alternatively, Baumeister et al., have proposed to extend UML class and activity diagrams to model mobile systems, assuming that mobile objects can migrate from one location to another, while these locations can be nested and mobile too.

There exist some preliminary proposals dealing with the lack of appropriate concepts and notations in the standard UML to model agent mobility. Klein has proposed some extensions to UML providing concepts for the modelling of:

(a) Mobility (strong);

(b) Remote execution (weak mobility); and

(c) Cloning.

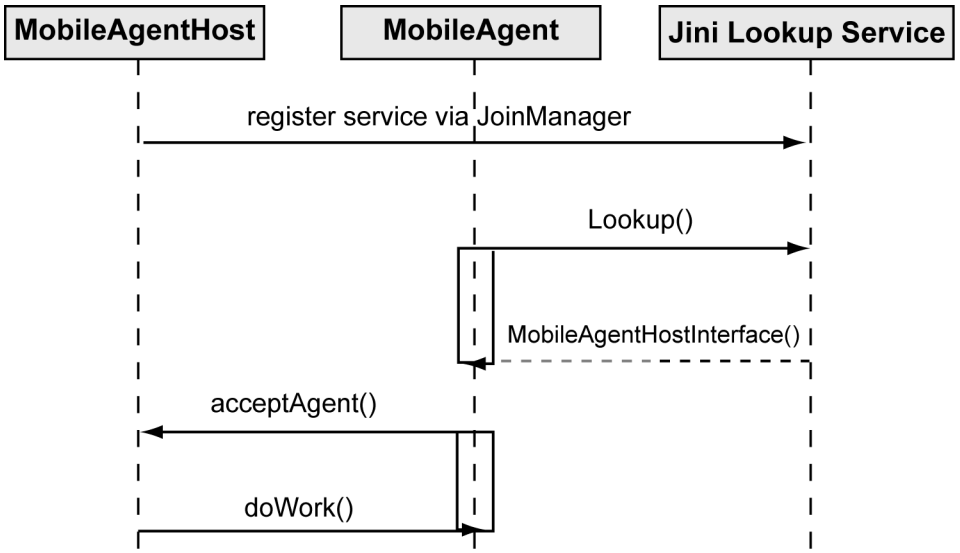


Figure 6.1: Sequence diagram for interactions between the MobileAgentHost, MobileAgent, and Jini lookup service
Source: http://www.javaworld.com/javaworld/jw-06-2002/images/jw-0628-jini4.jpg

Stereotyped classes are used to model mobile agents, while stereotyped components are used to model mobile-agent systems. Agents moving from one location to another are modelled, in sequence diagrams (Figure 6.1), by stereotyped messages (move, remote execution and clone) using the dependency relationship. Finally, stereotyped actions (move, remote execution and clone) are used to express, in state charts, whether an agent will also change its location when its state is changed.

(a) Muscutariu et al., have proposed an Architectural Description Language (ADL) for the design of mobile-agent systems. This ADL is defined as a simple UML profile, presenting a minimum set of concepts and operation

interfaces necessary for interoperability among heterogeneous mobile-agent systems. It proposes a graphical representation using deployment and component diagrams.

(b) Mouratidis et al., have introduced extensions to the UML deployment and activity diagrams to give answers to some questions that arise from the use of mobile agents. Their extended version of the deployment diagram allows developers to capture mobile agents (components), along with the platforms (nodes) they might visit – i.e. where the agents move.

6.3.1 Towards a Comprehensive Proposal for Mobile Agent Modelling

Although the UML extensions described above are useful and practical contributions, they yet represent incomplete approaches for the modelling of mobile-agent features and applications. In this sense, an attractive trend seems to be the definition of a unique UML extension integrating the features of these different approaches. Nevertheless, more work must take place in order to explore how UML can be used and extended to model other relevant abstractions in the mobile-agent paradigm, which are not captured with such approaches. Particularly, we argue that it is necessary to give answers through modelling to some questions that also arise during analysis and design phases of mobile-agent applications development, such as, which are the roles that an agent could play?; Which are the platforms or mobile-agent systems involved?; Which are the entities involved in the application and which are the structural relationships among these entities?

Essentially: Which are the types of agents involved?; Which are the types of resources manipulated and shared by these agents?; Whom an agent represents?; Which are the services and execution places provided by these mobile-agent systems?; How these mobile-agent systems are organised in regions in the context of the application?; Which are the types of bindings that an agent maintains to resources?

Fundamentally: Which are the places that a mobile agent visits?; Which are the activities performed by an agent? Which are the locations where these activities are performed? Which is the execution states pattern of an agent? What does an agent do during its life cycle?

Particularly: Is it proactive or reactive? What portion of a mobile agent, i.e., its code, data or execution state, needs to be moved? How the set of bindings to resources accessible by an agent, must be reorganised upon migration of the agent to a different location?

Which entities (agents, roles or systems) need to communicate between them? Wh y the se en ti tie s n ee d t o com munic at e? Wh en th ese en ti tie s nee d to communicate? Is such communication synchronous or asynchronous? Is it local or remote? Is it point-to-point, broadcast or multicast? Is it direct? Do the entities use an intermediation mechanism?

These issues have been identified from a bibliographical review of relevant analysis, design and implementation aspects of mobile agents, as well as, from own development experiences in the domain. In this context, we have defined a coherent set of views and models that organise and integrate the line of the contributions in the area, also incorporating new abstractions and mechanisms in order to develop a more comprehensive approach dealing with the issues formulated. This set represents a preliminary UML profile, named MAM-UML – it stands for Mobile- Agent Modelling with UML. The MAM-UML profile currently includes views to model organisational, life cycle, interaction and mobility aspects of mobile-agents applications. These views are described in the next sub-sections.

Organisational View

The organisational view describes the entities, and their structural relationships, involved in a mobile-agent application. This view addresses the static logical view of an application in the mobile agent paradigm.

Stereotyped classes are used to specify both the roles, which can be played by agents in the application, and the resources, which can be manipulated and shared by several agents. Stereotyped classes and packages are used to model the entities. Stereotyped active classes allow developers to specify two types of agents: stationary and mobile agents. Stereotyped packages allow developers to specify mobile-agent systems, execution places and regions. On the other hand, structural relationships among these abstractions are represented in a UML class diagram. It represents an analysis class model in the application domain. A stereotyped dependency relationship, named home, enables to specify the origin place of a mobile agent. In addition, a stereotyped association relationship (named acquaintance) enables to specify the existence of at least one interaction involving the entities concerned.

In order to describe properties of the new stereotypes, different tagged values are also defined, such as:

(a) **Identifier**
It can be applied to regions, mobile-agent systems, execution places, agents and resources. It specifies the identifier of an entity in order to be uniquely recognised.

(b) **Location**
This tagged value specifies the current location place of a mobile agent or resource.

(c) **Authority**
This tagged value can be associated to agents in order to specify whom they represent. It can be also associated to places, mobile-agent systems, and regions, in order to specify whom they accept as an officially legal authority, in the context of a particular application. It specifies persons or organisations.

(d) **State**
This tagged value specifies the current execution state of a mobile agent, e.g., it can be active, suspended, etc.

(e) **Clone**
This tagged value specifies whether an entity is a clone. It can be applied to mobile-agents and resources.

(f) **System Type**
It specifies the type of a mobile-agent system. This value indicates whether the system is complaint to a standard interoperability interface, such as those defined by OMG or FIPA.

(g) **Language**
This tagged value specifies the programming language (or languages) that are used by a mobile-agent system for the coding of mobile agents.

Figure 6.2 provides a class diagram, which models the organisational view of a simple mobile-agent application. In such application, personal assistants move between different places provided by two different mobile-agent systems, in the context of a region. These mobile agents act on behalf of some user and can play two different roles. Moreover, the diagram prescribes that these mobile agents at least interact, at auctioning places, with stationary employee agents.

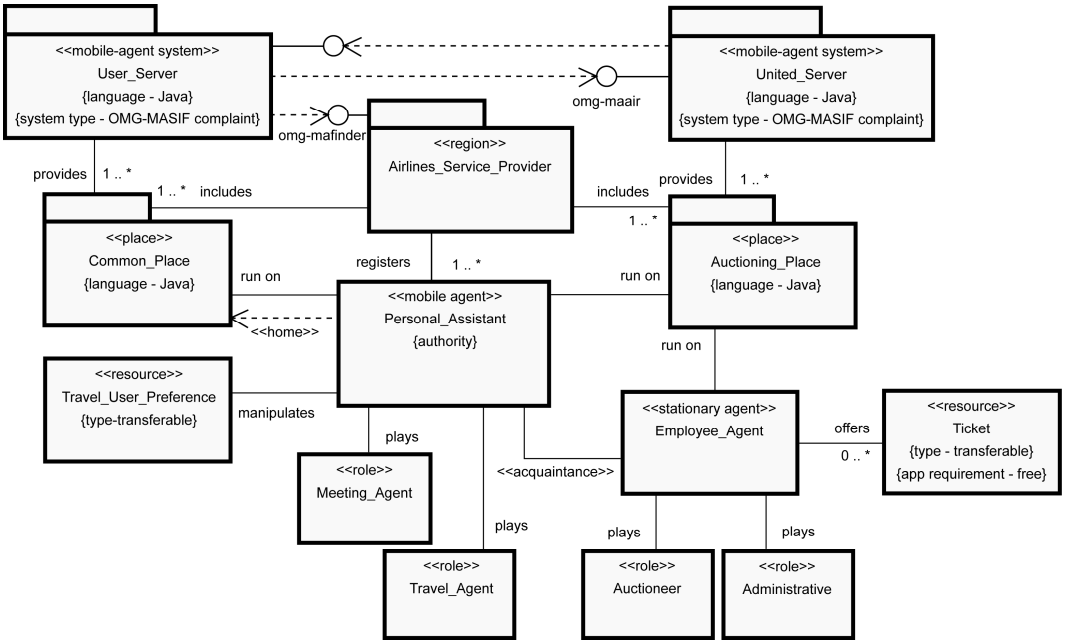


Figure 6.2: Organisational view of mobile agent application

Life-cycle View

The life-cycle view describes the itineraries, states activities of each kind of mobile agent, during its life cycle. This view includes three models that specify same: a model of itinerary, a model of execution states, and a model of activities.

Model of Itinerary: Mobile agents move from a location (place) to another one, in order to meet other agents or to access services and resources provided there, while they are acting on behalf of someone or something.

Interaction diagrams, both sequence and collaboration ones can be used to specify the itinerary followed by an agent, during its life cycle. The standard UML provides two stereotyped messages, become and copy, which can be used to model correspondingly, agent migration and cloning. Although these stereotypes are useful in early phases of development, when it is necessary to model precisely the type of mobility that an agent uses other concepts need to be defined. For this reason, we define new stereotyped messages using the dependency relationship: strong move, weak move, and remote cloning. On the other hand, pro-active or reactive mobility can be explicitly recognised, in interaction diagrams, depending on who sends such messages to the agent; the agent itself, in the case of pro-active mobility, or a different entity in reactive mobility. Additionally, constraints associated to the mobility messages enable to specify why and when an agent moves.

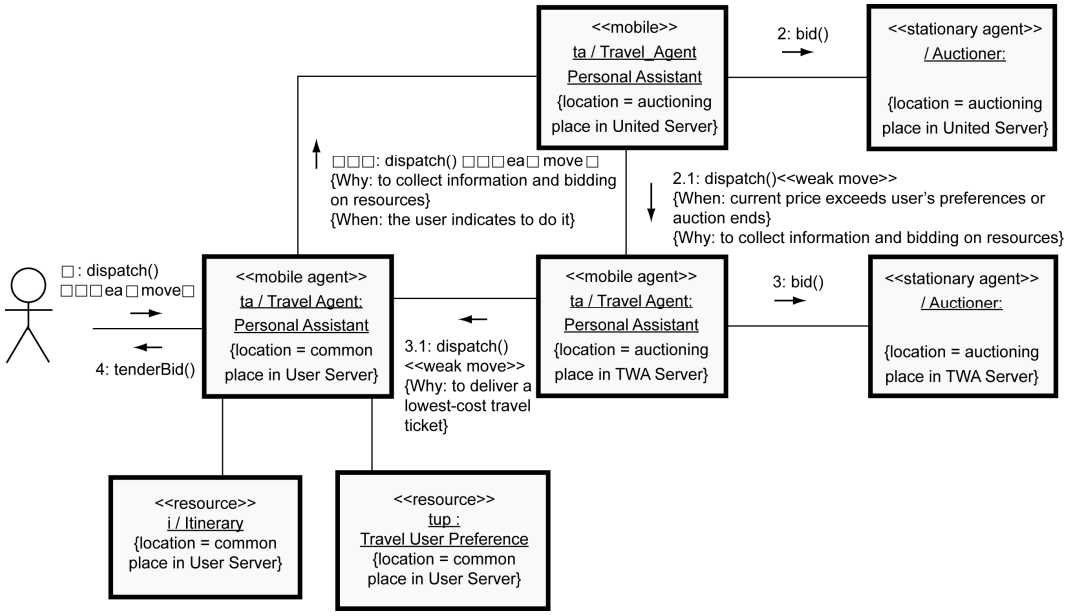


Figure 6.3: Modelling of agent itinerary

Figure 6.3 provides an interaction diagram that models an agent that moves between different places, collecting information and bidding on resources to automatically deliver a lowest-cost travel ticket to the user that it represents. Specifically, the diagram models an instance (named ta) of the class Personal Assistant, playing the Travel Agent role, migrating between auctioning places provided by different servers (mobile-agent systems). The mobile agent begins its itinerary reacting to a message of the user. Along this itinerary, the agent interacts at each auctioning place with anonymous Employee Agent instances playing the Auctioneer role.

ACTIVITY 6.1

Discuss the UML extension for Mobile Agents with you friends and teacher.

6.4 APPLICATIONS FOR MOBILE AGENTS AND TO MOBILE APPLICATION AND IMPLEMENTATION TOOLS

(a) **Parallel Computing**
Solving a single task on a single computer takes a lot of time. To, achieve this, mobile agents migrates to computers on networks which have the required recourses and use them to solve the parallel problem.

(b) **E-Commerce**
Mobile agents travels to different trading sites. A mobile agent can be programmed to bid an online action on behalf of the users.

(c) **Mobile Computing**
Wireless internet access is slow and expensive. Users can dispatch a mobile agent, which collect their queries and logoff and the result can be received at a latter time.

(d) **Information Gathering**
An agent accumulates knowledge during the process that allows it to make decision for the future purpose.

(e) **Secure Brokering**
Allows parties to meet on a trusted platform where collaboration can take place without worrying that the host will add one party on the other.

(f) **Information Dissemination**
Mobile agents can disseminate information such as news and automatic software updates.

SELF-CHECK 6.2

Fill in the blanks:

1. OMG stands for.....
2. Stereotyped classes are used to model mobile agents, while stereotyped components are used to model ..... systems.
3. Stereotyped packages are used to model both..... and .....
4. Architectural Description Language is for.....
5. Stereotyped classes and packages are used to model the .....
6. Identifier can be applied in .....
7. Secure broking allows. .....
8. Common Management Information Protocol is for .....

6.5 SOLVING MOBILE APPLICATION DEVELOPMENT PROBLEM WITH MOBILE AGENTS

A network management system performs the tasks of managing a network, undertaking its proper functionalities, maintenance, security control, gathering and archiving of data and fault management. Various applications, particularly those related to multimedia, require guaranteeing transmission of data with a certain degree of reliability and quality of service in a communication network.

(a) Common Management Information Protocol (CMIP) is protocol for telecommunication network.

(b) Current network management systems such as Simple Network Management Protocol (SNMP) is for data networks.

These models are typically designed according to a centralised model, which are characterised by lack of distribution, a low degree of flexibility, re-configurability, efficiency, scalability, and fault tolerance. They also require network administrator to make real-time decisions and find solutions for the series of problems in the network. These network managements deal only with data gathering and reporting methods, which in general involves substantial transmission of management data. This causes to consume a lot of bandwidth,

computational overhead, a considerable strain on the network at all the times and a reason for traffic jam at the manager host. These management activities are limited, and since they cannot do intelligent processing such as judgment, forecasting, decision making, analysing data, and make positive efforts to maintain quality of service. Therefore, all these problems recommend distribution of management intelligence by using mobile agent to ov erco me th e limitations of centralised management and meet today's requirements.

Mobile agents are computer programs, which are autonomous, proactive and reactive, and have ability to learn. They move from one node to another node and interact with each other, sharing information to better carry out their goals. Mobile agents spread intelligence across the network while they move in a network. The mobility of mobile agents allows them to be created, deployed, and terminated without disrupting the network configuration.

Network components (e.g., hosts, gateways, severs) in a centralised network management, have management agents, but these agents are different from mobile agents, and they carry out the network management tasks. Consequently various nodes in a centralised network may be managed with a single network management station, which causes traffic and congestion in a network. The fundamental problem in telecommunication network management is load balancing, which avoids overloading and traffic congestion in networks, even if many of the network's nodes are not used to their capacities at all.

Therefore, we need to use mobile agents for managing the network. Mechanism of the mobile agents is based on the load balancing method and by using this method, there will be no routers idle or overloaded and lets a network use all its capacity efficiently. Quality of service method proposes ways for managing the network resources in a well-organised approach. It lets for better use of the exciting network infrastructure, improves service to the end users and reduces the cost of providing these services. The main purpose of the quality of service is to dedicate bandwidth, control jitter and avoid latency, which are required by some of the real time applications, and for recovering loss qualities.

6.6 TECHNIQUES FOR AGENT BASED SOFTWARE

This topic introduces an agent-based software development method. It defines a limited number of components of an agent-based software system and shows the possibility of designing and implementing actual software. Starting points in developing agent-based software are the business rules and the basic agent-based concepts as defined in the paper. A notation for visualising the defined concepts is introduced. The notation of the unified modelling language UML is used. The

possibility to define rules for mapping agent model elements onto source code is shown using the JADE agent platform.

Several efforts have been made to develop agent-based software methodologies. However, most of the studies concentrate on specific areas of the agent software -agent models, reasoning logic and agent actions, agent communication, agent programming languages and frameworks. To gain wide acceptance of agent-based software in practice, methods covering the full software development cycle from analysis to implementation are required. Such methods can be developed on the basis of agreed agent software concepts, modelling techniques supporting these concepts, and finally agent software implementation frameworks supporting the same concepts. We are going to present a method covering all these stages. The aim of the paper is not to introduce a fully functional agent-based method but rather to show that it is possible to develop one.

SELF-CHECK 6.3

1. What do you mean by mobile agents?
2. Explain the techniques for agent based software.

6.6.1 Scope of Agents, Background and Related Work

In the present work, the word agent will be used in the meaning of an abstraction unit when designing and implementing software systems. Starting to design new software, the initial specification should describe what kinds of agents can be found in the system and how do they co-operate with each other in order to offer the needed functionality. When it comes to the design and implementation of the specified agents themselves, then traditional object-oriented programming languages must be used.

Agent-based approach is not applicable everywhere – it can be used only in the circumstances where software system will be built of autonomous units, each executed separately. The restriction is quite natural – like simple procedural approach is the best for designing simple software for calculating square root or like object-oriented approach is the best for designing complex software working as one unit.

Agents add value to the traditional software design by offering tools for handling the most general level of the problem domain. They represent the main building blocks of a distributed software system without describing the internal structure of the individual blocks.

According to Nwana, the concept of agent dates back to the early days of research into DAI in the 1970s, to Carl Hewitt's concurrent actor model (1977) where "...an actor is a computational agent which has a mail address and behaviour. Actors communicate by message passing and carry out their actions concurrently". Nwana states that the overuse of the word agent has masked the fact that, in reality, there is a truly heterogeneous body of research carried out in this manner. Nwana even states that the chance of agreeing on a consensus definition for the word agent is nil. He makes an effort to classify different types of existing software agents according to the abilities to learn, co-operate, act autonomously, and comes up with the following classification:

(a) Collaborative Agents;

(b) Interface Agents;

(c) Collaborative Learning Agents; and

(d) Smart Agents.

Nwana admits also that there exist other ways of agent classification. This paper is using the word agent in the meaning of collaborative agent according to the above classification. Collaborative agents emphasise autonomy and co-operation. In order to co-operate, agents need a communication language. Wagner refers to an informal definition of software agent by Genesereth and Ketchpel:

"An entity is a software agent if and only if it communicates correctly in an Agent Communication Language (ACL)".

Knowledge Query and Manipulation Language (KQML) is the most widely used ACL. Java Agent Development (JADE) framework [6] uses the ACL, specified by the Foundation for Intelligent Physical Agents (FIPA) that is very similar to KQML. FIPA is a non-profit association whose purpose is to promote the success of emerging agent-based applications, services, and equipment. FIPA has published a set of specifications for agent-based software development frameworks. Some of the normative specifications by FIPA are Agent Management, ACL, and Agent Software Integration.

Taveter and Tamm have introduced layered architecture of agent-based software where the software is considered as consisting of three layers: agent, object, and binary layer. Similar approach is also followed in this paper. We regard agents as the top-level abstraction units in software design while the agents themselves are implemented using object-oriented programming languages. We accept also objects in the top agent layer, but these are the objects that agents manipulate with and not the implementation level objects. We do not accept any object-to-object communication in the agent layer. The computers will finally execute only

the compiled binary code regardless of the high level languages used by humans. The first two levels are meant for humans and should make software developing easier, faster, and more reliable. Adding a new agent layer on top of the object layer will be justified only if it adds value compared to the object-oriented approach. The added value is achieved by scoping agents:

(a) To distributed systems consisting of separate autonomous software units.

(b) To general level where common sense is sufficient to model the software system without going into any technical details.

6.6.2 Concepts and Visual Notation

In the following paragraphs the main concepts of agent-based software model are outlined. We shall specify a limited list of components of agent-based software. There is evidently a need for more components (like actors, virtual knowledge base, state, transition, case studies, etc.) but this is out of the scope of this paper. We shall explain the meanings (brief semantics) of selected components and introduce the notation of the components on diagrams. After that we shall introduce mapping rules from the visual model onto actual software code. JADE framework will be used in the code examples. Note that there is a difference between the code, generated according to the mapping rules, and the final fully functional code. The generated code serves as a frame and that must be manually modified in order to become fully functional. Not every detail should be present in the model. Unfortunately, there are no strict rules about what to include in the visual model and what should be added only to the code. Similar statement can be found also in the UML Summary: "The UML, a visual modelling language, is not intended to be a visual programming language". Similar statement can be found also in the UML Summary, "The UML, a visual modelling language, is not intended to be a visual programming language".

An agent is an autonomous software unit that can exist independently of other similar units in the software system. An agent performs some functions for other agents or external actors. Agents communicate with each other via messages in an agent communication language. It is interesting to compare this definition with the FIPA agent definition, "An agent is the fundamental actor in a domain. It combines one or more service capabilities into a unified and integrated execution model which can include access to external software, human users and communication facilities". The main difference is that in our approach, we emphasise the software nature of an agent and the communication between agents. An agent is expressed on diagrams as a dashed rectangle. The rectangle can contain only the name of an agent because in our approach an agent does not have any internal structure. There are two reasons for using dashed rectangles for agent notation: (a) to distinguish an agent from an object and (b) an agent can

be in the role of a system boundary in used case diagrams and in UML this is denoted with dashed lined rectangle.

A message is a speech act that one agent performs in order to request or send information to other agent(s) in the format of an ACL. Note that UML defines message from a different viewpoint putting emphasis on the general-to-specific relationship: "A message instance is a communication between objects that conveys information with the expectation that action will ensure. A message is the description of a set of message instances of the same form". Messages are sent asynchronously; an agent can have multiple conversations at the same time and can receive different messages from different agents with no particular order. A message is depicted as a labelled arrow. The label contains the message description and the arrowhead direction defines the sender and the receiver. The message label string can have two different forms (both at the same time) – a free form description and a form that accords to the syntax of the ACL used in the system. The free form description should be used for discussing the model with people not familiar with the ACL. The language specific syntax should be used for easier migration from the model to the actual software code.

A behaviour is a sequence of agent's actions performed as a result of a specific event. Actions in our approach can be sending of messages, waiting for incoming messages, internal actions, and object manipulations. An event can be receiving a message or a specific return value of some periodical test procedure (timeout, end of day, etc.). Note that UML and FIPA do not define the term behaviour. The closest concept in UML is activation. Behaviour is depicted on sequence diagrams by grouping an agent's actions with a rectangle area on the agent's lifeline. A single diagram can express multiple behaviours of an agent.

An internal action is an activity or a group of sequential activities of an agent. Internal action is not related to any objects manipulated by the agent or to any messages sent or received.

An object is a passive component in the system that is manipulated directly by an agent. An agent's virtual knowledge base consists of the information tied to objects. Examples of objects in our approach are bill, schedule, switch, etc. An agent can manipulate an object in order to get or change some information or to perform some action with the object.

We define the communication between an agent and an object as a manipulation. An agent communicates with objects not in ACL but in the form corresponding to the object. For example, if an object is a row in a relational database table then the agent should use the SQL commands specific to the database for manipulation.

6.7 MOBILE PEER-TO-PEER COMPUTING (MOBI-DIK)

Let us observe that local search-and-discover arises in many application domains including social networks, transportation, mobile electronic commerce, emergency response, asset tracking and management, and mobile collaborative work. For example, in a large professional, political, or social gathering, the technology is useful to automatically facilitate a face-to-face meeting based on matching profiles; or for sending free SMS messages.

We are facing an unprecedented proliferation of mobile devices, many equipped with in regulated wireless technologies such as Bluetooth. This environment will enable a new class of local search-and-discover applications that are independent of an infrastructure or a database server.

In transportation, the Mobi-dik incorporated in navigational devices can be used to disseminate to other similarly-equipped vehicles information about relevant resources such as free parking slots, traffic jams and slowdowns, available taxicabs, and ride sharing opportunities. In mobile electronic commerce, Mobi-dik is useful to match buyers and sellers in a mall, or to trade data (e.g., music files) and knowledge. In emergency response, Mobi-dik can be used by first responders to support rescue efforts even when the fixed infrastructure is inoperative; it will match specific needs with expertise (e.g., burn victim and dermatologist), and help locate victims. In asset management, sensors mounted on neighboring smart artifacts (e.g., containers) can communicate and transitively relay alerts to remote check-points.

ACTIVITY 6.2

1. What do you mean by Peer to Peer Application for Mobile Computing?
2. Explain the UML extension for mobile agents.

SUMMARY

- The Definition of mobile agents say that a Mobile Agent, is a type of software agent, with has the feature learning, and most importantly, mobility. In other world we can say that a mobile agent is a process that can transport its state from one environment to another, with its data intact, and be capable of performing appropriately in the new environment. Mobile agents decide when and where to move. A mobile agent is a specific form of mobile code. However, in contrast to the Remote evaluation and Code on demand programming paradigms, mobile agents are active in that they can choose to migrate between computers at any time during their execution. This makes them a powerful tool for implementing distributed applications in a computer network.

KEY TERMS

LMP Link Manager Protocol

Loc Location

LF Low Frequency

LLC Logical link Control

MIB Management Information Base

SELF-TEST

1. Define Peer-to-peer architecture in detail.
2. What are the techniques for the agent based software?
3. Write short notes on the visual notation.
4. Define the extensions regarding mobile agents.

Topic 7 ▶ Wireless Connectivity and Mobile Applications

LEARNING OUTCOMES

By the end of this topic, you should be able to:

1. Describe wireless connectivity;
2. Underline the concept of quality of service;
3. State the working of SMS; and
4. Elaborate the concept of mobile IP.

▶ INTRODUCTION

In the previous topic, we have studied Mobile Agents and Peer-to-Peer Architectures for Mobile Applications, Mobile Agents for Mobile Computing, UML Extensions for Mobile Agents, Applications of Mobile Agents to Mobile Applications and Implementation Tools, Solving Mobile Application Development Problems with Mobile Agents, Techniques for Agent-based Software and Peer-to-Peer Applications for Mobile Computing. Now, in this topic, we would focus on Wireless Connectivity and Mobile Applications.

7.1 WIRELESS CONNECTIVITY

Wireless connectivity is a telecommunication in which electromagnetic waves carry the signal over part or the entire communication path. Some monitoring devices, such as intrusion alarms, employ acoustic waves at frequencies above the range of human hearing; these are also sometimes classified as wireless. Wireless technology is rapidly growing and playing an increasing role in the lives of people throughout the world. In addition, ever-larger number of people are relying on the technology directly or indirectly. Some of the common examples of wireless equipment in use today include:

(a) Cellular phones and pagers – provide connectivity for portable and mobile applications, both personal and business.

(b) Global Positioning System (GPS) – allows drivers of cars and trucks, captains of boats and ships, and pilots of aircraft to ascertain their location anywhere on earth.

(c) Cordless computer peripherals – the cordless mouse is a common example;

(d) keyboards and printers can also be linked to a computer via wireless.

(e) Cordless telephone sets – are limited-range devices, not to be confused with cell phones.

(f) Home-entertainment-system control boxes – the VCR control and the TV channel control are the most common examples; some hi-fi sound systems and FM broadcast receivers also use this technology.

7.2 QUALITY OF SERVICE (QOS)

Internet Tool Survey describes the quality of service provided by the Internet. In the present day internet environment, the internet offers a point-to-point delivery service, which is based on the "best effort" delivery model. In this model, data will be delivered to its destination as soon as possible, but with no commitment as to bandwidth or latency. We use the protocols like TCP that provide the highest guarantee of reliable delivery of data. This is adequate for traditional data applications like FTP and Telnet, but inadequate for applications requiring timeliness. For example, distributed multimedia applications need to communicate in real-time and are sensitive to the quality of service they receive from the network.

For these applications to perform adequately and be widely used, QoS must be quantified and managed, and the Internet must be modified to support real-time QoS and controlled end-to-end delays. The notion of QoS must be extended from the communication layer up through the intervening architectural layers to the application level.

There are a number of definitions of QoS at the communication level. The notion originated to describe technical characteristics of mainly non-time-dependent data transmission. Emerging networks such as ATM can provide QoS guarantees on bandwidth and delay for the transfer of Continuous Media (CM) data.

(a) A number of QoS parameters describe the speed and reliability of data tra nsmiss ion, e.g., throughput, transit delay, and e rror rate. The International Telecommunication Union (ITU), Standard Information technology, Open distributed processing, Reference Model, refers to QoS as "A set of quality requirements on the collective behavior of one or more objects".

(b) Five service classes are defined in terms of QoS parameters. Class 0 refers to best effort service in which no specific traffic parameters and no absolute guarantees are provided. The ATM Lexicon defines QoS as "A term which refers to the set of ATM performance parameters that characterise the traffic over a given virtual connection." QoS parameters apply mostly to lower level protocol layers, and were not meant to be directly observable or verifiable by the application. These parameters include cell loss ratio, cell error rate, cell misinsertion rate, cell delay variation, cell transfer delay, and average cell transfer delay.

(c) Recently the Internet Engineering Task Force has begun to address QoS issues for ATM., Native ATM Support for ST2+, states "As the demand for networked real time services grows, so does the need for shared networks to provide deterministic delivery services. Such deterministic delivery services demand that both the source application and the network infrastructure have capabilities to request, setup, and enforce the delivery of the data. Collectively these services are referred to as bandwidth reservation and Quality of Service (QoS)." IP over ATM.

A paper presented on the topic provides a more general definition of QoS for applications that must communicate in real-time.

"The set of those quantitative and qualitative characteristics of a distributed multimedia system, which are necessary in order to achieve the required functionality of an application".

Several research groups are investigating QoS support for the WWW, in particular, researchers at BBN, the Distributed Systems Technology Centre, and Washington University. These and other QoS-based projects are summarised below:

7.2.1 Generalised QoS Processing Model

Building QoS into a system involves:

(a) Mapping of QoS requirements to resources;

(b) QoS mechanisms which realise desired QoS behaviour;

(c) The construction of a generalised QoS framework; and

(d) QoS specification which captures application QoS requirements.

7.2.2 QoS Specification

QoS specification is different at each system layer and is used to configure QoS mechanisms at each layer. An application's QoS requirements are conveyed in terms of high-level parameters that specify what the user requires. Possible system layers are:

(a) Protocols – transport, network

(b) Operating system – scheduling, resource management, real-time support

(c) Network

(d) Middleware

(e) Distributed platforms – CPU, memory/buffers, devices

(f) Application

QoS specification encompasses requirements for:

(a) **Performance**
The performance characteristics are needed to establish resource commitments.

(b) **Synchronisation**
Synchronisation characterises the degree of synchronisation required between related services, events, or information flows.

(c) **Level of Service**
Specifies the degree of resource commitment required to maintain performance guarantees.

(d) **Cost of Service**
The cost of service describe the price a user is willing to incur to obtain a level of service.

(e) **QoS Management**
The degree of QoS adaptation that can be tolerated and scaling actions to be taken in the event the contracted QoS cannot be met.

QoS requirements are assessed to determine if they can possibly be met. If, for example, the level of service requested cannot be provided, the user can be asked if a certain level of degradation is acceptable before proceeding further.

7.2.3 QoS Mapping

QoS requirements are used to derive resource requirements for entities such as computation, communication, and storage. QoS parameters may be oriented towards:

(a) **Format**
Transfer rate, data format, compression schema, image resolution.

(b) **Performance**
Sequential versus parallel processing, delays, data rate.

(c) **Cost**
Platform rates, copyright fees, connection and data transmission rates.

(d) **Synchronisation**
Loosely versus tightly coupled, synchronous versus asynchronous.

(e) **User**
Subjective quality of images, sound, response time.

Each QoS parameter can be viewed as a typed variable with bounded values, and the values are subject to negotiation between the system layers.

7.2.4 QoS Enforcement

To ensure that the contracted QoS is sustained, it must monitor QoS parameters and reallocate resources in response to system anomalies. To provide and sustain QoS, resource management must be QoS-driven. In allocating resources, the resource management system must not only consider resource availability and resource control policies, but also an application's QoS requirements measured in terms of the QoS parameters Prior to allocating resources, the system layers negotiate to determine if they can collectively ensure that the required QoS

parameters can be consistently satisfied. Negotiation involves dynamic adaptation and the transmission and translation of QoS parameters between the layers as the layers enter into different types of agreements, e.g., guaranteed, best-effort, or predictive. If negotiation ends in agreement, the application is launched. After resources are allocated, QoS mechanisms at each layer guarantee the contracted QoS, and the resource manager guarantees the sustained availability of the allocated resources. This requires monitoring resource availability and its dynamic characteristics, e.g., measuring processing workload and network traffic, to detect deviations in the QoS parameters. When there is a change of state, i.e., degradation in the QoS, and the resource manager cannot make resource adjustments to compensate (e.g., reschedule shared resources to satisfy allocations or switch to an optimised implementation of an object/ service), then the application is notified, e.g., application handlers are called. The application can either adapt to the new level of QoS or scale to a reduced level of service.

7.2.5 Extending QoS to the Internet

RSVP, the emerging standard for QoS negotiation over IP, is a network control protocol for establishing and maintaining internet integrated service reservations that allow Internet applications to obtain both best-effort and real-time QoS for their data flows. Hosts and routers use RSVP to deliver QoS requests to all nodes along the path of the data stream, typically resulting in a reservation of bandwidth for that particular data flow. RSVP is designed for use over both IPv4 and IPv6, the next generation Internet protocol. IPv6 offers a choice of QoS levels beyond the single "best effort" delivery service offered by IPv4. With these added QoS capabilities, still in the experimental stage of development, IPv6 will provide a better range of support for real-time data traffic. The latest RSVP functional specification can be found at CNRI (postscript) or USC/ISI (text) and includes a discussion of RSVP's design and operation. The tutorial QoS over the Internet. The RSVP protocol provides motivation for RSVP and details about QoS support.

SELF-CHECK 7.1

1. Wireless connectivity is a telecommunication in which ..... carry the signal over part or the entire communication path.
2. GPS means .....
3. "A set of quality requirements on the collective behavior of one or more objects" is the definition of .....
4. Class 0 of QoS parameters refers to .....
5. QoS specification is ..... at each system layer and is used to configure QoS mechanisms at each layer.
6. To provide and sustain QoS, resource management must be.....
7. RSVP is designed for use over both .....and ....., the next generation Internet protocol.

ACTIVITY 7.1

The rapid expansion of wireless technology in the marketplace has resulted in a plethora of radio systems being released for use over the last few years. Consequently, many people with little or no practical experience with this type of telecommunications have been thrust into the wireless arena. The amount of information that needs to be assimilated by the current wireless workforce is daunting. Give your views on the above statement.

7.3 SURVEY OF WIRELESS NETWORKING TECHNOLOGIES

The acceptance of WSNs by the industrial automation community is impeded by open issues, such as security guarantees and provision of Quality of Service (QoS). Wireless Sensor Networks (WSNs) are gradually adopted in the industrial world due to their advantages over wired networks. In addition to saving cabling costs, WSNs widen the realm of environments feasible for monitoring. They thus, add sensing and acting capabilities to objects in the physical world. They allow for communication among these objects or with services in the future Internet. Figure 7.1 to examine both of these perspectives, we select and survey relevant WSN technologies dedicated to industrial automation.

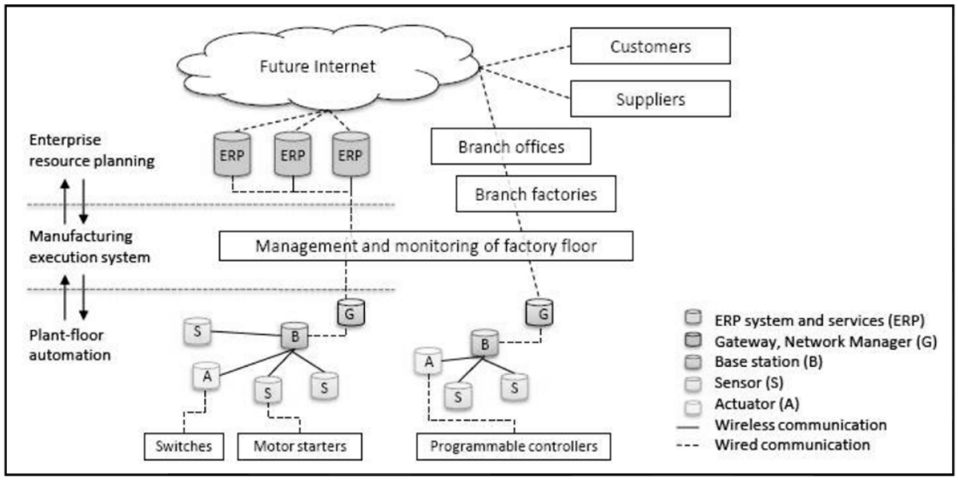


Figure 7.1: From the sensors to the customers

We determine QoS requirements and carry out a threat analysis, which act as basis of our evaluation of the current State-of-the-art. According to the results of this evaluation, we identify and discuss open research issues. Selected Wireless Sensor Networks Standards: State-of-the-art Wireless communication in industrial automation is mostly based on standardised technologies, such as the IEEE 802.11 and IEEE 802.15 standard families also designated as Wireless Local Area Networks (WLAN) and Wireless Personal Area Networks (WPAN). Both of these standard families were conceived for application purposes different than industrial automation. In fact, the IEEE 802.11-based standards offer high data rates in the order of tens of Mbit/s and ranges up to tens/hundreds of meters, while the IEEE 802.15-based standards only supports data rates of hundreds of kbit/s to several Mbits/s with ranges from a few meters up to hundreds of meters. However, to provide greater data rate and range, IEEE 802.11 technology consumes a greater energy budget that can limit the benefits obtained by wireless communications. Indeed, the sensor nodes are either powered by cables or batteries. In the former case, the advantages provided by wireless communication are partially negated, whereas in the latter case, the scarce energy resource has to be cheaply consumed in order to avoid frequent human interventions to recharge the batteries. Energy is thus, a major concern in both previous cases and we therefore, focus on the IEEE 802.15-based standards, and particularly on the IEEE 802.15.1 and IEEE 802.15.4 standard, within the scope of this work.

7.3.1 IEEE 802.15.1-based Standards

The IEEE 802.15.1 standard, also known as Bluetooth, can be classified to fall between the IEEE 802.11 and IEEE 802.15.4 standards in terms of energy consumption and data rates. With medium data rates and lower energy consumption than the IEEE 802.11 standard, IEEE 802.15.1 offers an interesting compromise between energy consumption and data rate, and is therefore, particularly suited for high-end applications requiring high data rates as well as applications with strong real-time requirements such as factory automation.

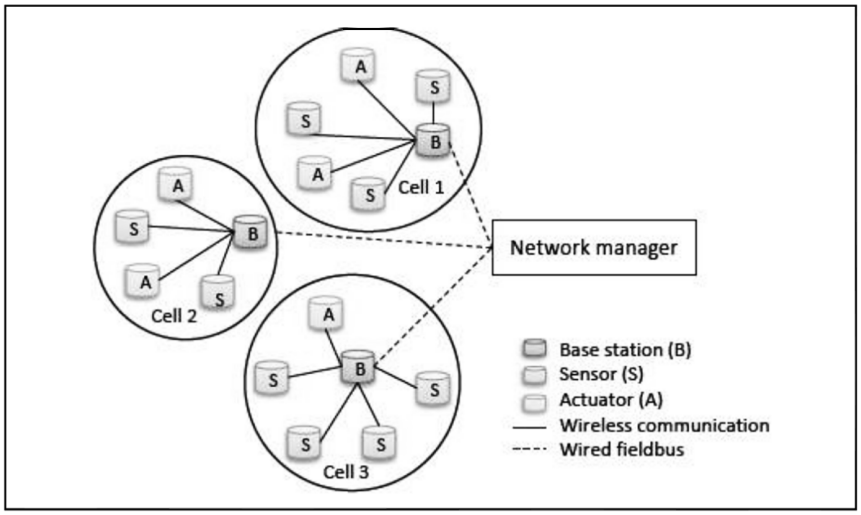


Figure 7.2: WISA network elements

The Wireless Interface for Sensor and Actuators (WISA) has been selected as a representative 802.15.1-based specification for further discussion. Wireless Interface for Sensor and Actuators (WISA) released by ABB and presented in the proprietary Wireless Interface for Sensors and Actuators (WISA) specification is based on the IEEE 802.15.1 physical layer and targets factory automation WSNs with packet error rate less than 1??9 and cycle time of 2msNetwork Elements WISA networks can be deployed in cellular topology with up to three cells (Figure 7.2). Each cell uses a different transmission frequency, and is composed of a base station and up to 120 end devices including sensors and/or actuators organised in a star topology. The end devices communicate wirelessly via standard Bluetooth transceivers, while the base station is equipped with a specific transceiver, which is able to receive up to four channels in parallel. Additionally, the base stations exchange information with the network manager via wired fieldbus such as DeviceNet and Modbus.

EXERCISE 7.1

1. What are the advantages of wireless network over wired network?
2. What is threat analysis?
3. What are IEEE 802.15.1-based Standards?
4. Explain wireless Networking Technologies.
5. Briefly explain the working of QoS Mapping?

7.4 MOBILE IP

Mobile IP is most often found in wireless WAN environments where users need to carry their mobile devices across multiple LANs with different IP addresses.

Mobile IP is a standard that allows users with mobile devices whose IP addresses are associated with one network to stay connected, when moving to a network with a different IP address. When a user leaves the network with which his device is associated (home network) and enters the domain of a foreign network, the foreign network uses the Mobile IP protocol to inform the home network of a care-of address to which all packets for the user's device should be sent. A common analogy to explain Mobile IP is when someone moves his residence from one location to another say person moves from Shah Alam to Kuala Lumpur, the person drops off new mailing address to Kuala Lumpur post office. Kuala Lumpur post office notifies Shah Alam post office of new mailing address. When Shah Alam post office receives mail for the person, it knows that the mail needs to be forwarded to person's Kuala Lumpur address.

7.4.1 How Mobile IP Works

Before explaining how Mobile IP works, it is useful to become familiar with the terminology used in the rest of this topic. We will use these terms extensively in our subsequent description of Mobile IP operation and in describing the exchange of messages between the mobile nodes and the other key entities within the network.

Mobile IP was designed by the "IP Routing for Wireless Mobile Hosts" working group (mobile IP WG) of the Internet Engineering Task Force (IETF) and published as a proposed standard in November 1996.

Mobile Node: A node running the Mobile IP protocol stacks which moves between different IP subnets. This node is assigned a (permanent) IP address which defines where all its packets should be sent. When other nodes send packets to the mobile node, they only specify this home IP address in the packet, regardless of where the mobile node is physically located. This is illustrated in Figure 7.3 where a mobile node moves between different IP Subnets.

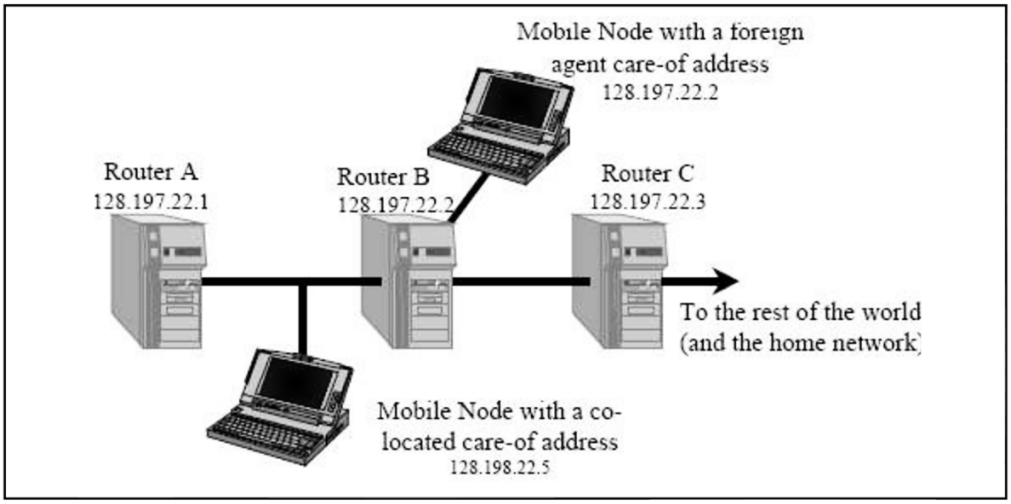


Figure 7.3: A mobile node with foreign agent care of address

Home Network: The subnet which corresponds to the home address of the mobile node as well as that of the home agent. It is considered the mobile node's "home" point of attachment (Figure 7.4).

Home Agent: A router on the home network that is responsible for intercepting packets destined for the mobile node when the mobile node is attached to a foreign network. The home agent is responsible for forwarding these packets to the mobile node (Figure 7.4).

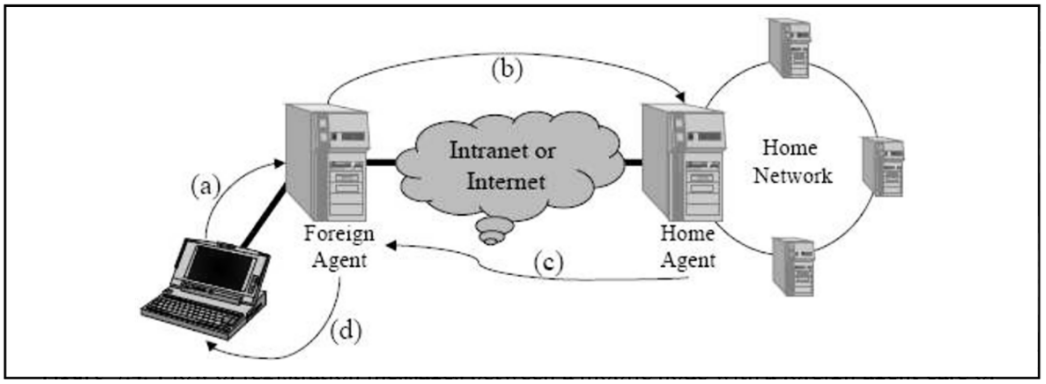


Figure 7.4: Flow of registration messages between a mobile node with a foreign agent care of address and its home agents

Foreign Network: A network, other than the mobile node's home network, that a mobile node attaches itself to.

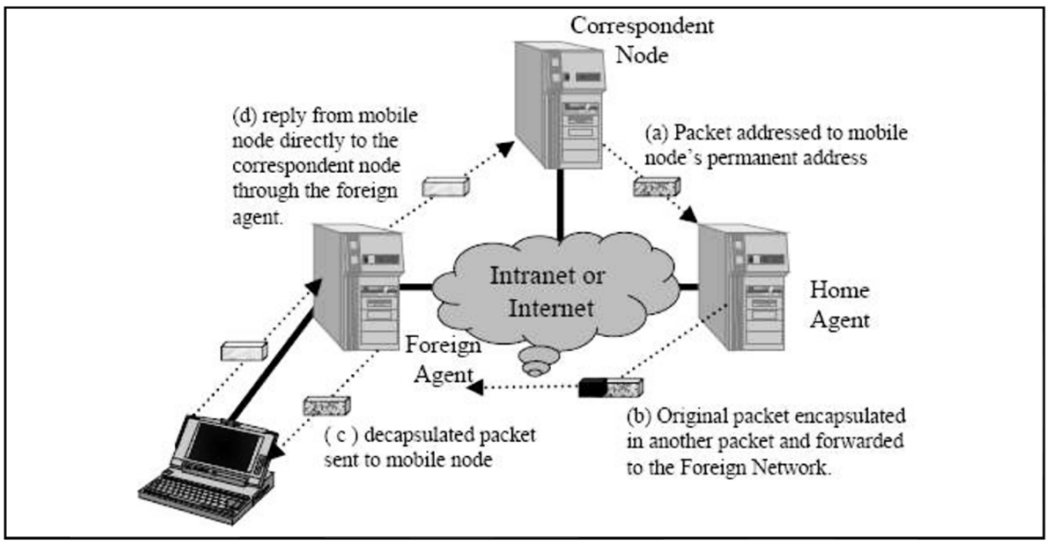


Figure 7.5: Communication between a correspondent node and a mobile node on a foreign network

Foreign Agent: A router on the foreign network configured for Mobile IP. When the mobile node has a foreign agent care-of address all packets are relayed through this node. When using a collocated care-of address, the mobile node may still use a foreign agent for its default router or for registration with the foreign network.

Care-of Address: The address that the mobile node uses for communication when it is away from its home network. This address can either be a foreign agent care- of address, when the mobile node uses the foreign agent's IP address as its care-of address or a collocated care-of address, where the network interface of the mobile node is temporarily assigned an IP number on the foreign network.

Correspondent Node: Any host which is communicating with the mobile node. This node could be located on the home network, foreign network, or any other place which is able to route packets to the mobile node's home network.

Tunnelling: The process of encapsulating an IP packet within another IP packet for the purpose of routing it to a location other than the one specified in the original destination field. Specifically, when a packet is received by the home agent, it encapsulates the original packet inside a new packet, placing the mobile node's care-of address in the new destination address field before forwarding it to the appropriate router. The path that is followed by this new packet is called the tunnel.

From the above descriptions, it should be clear that only three entities have to be modified in order to support Mobile IP over the Internet: the mobile node, the home agent, and the foreign agent. When a collocated care-of address is used by the mobile node, a foreign agent is often not even required. The communication between home agent & foreign agent is shown in Figure 7.5.

Home agents and foreign agents periodically broadcast their willingness to act as Mobile IP routers through agent advertisements. If a mobile node needs to immediately know the address of a potential agent without waiting for the next advertisement, it can broadcast an agent solicitation message. The mobile node uses these advertisements to determine if it has moved to a new location and where it can attach to a network if it has. If it is on the home network, no changes take place and no modifications to the IP protocol operation are required for communications. If the node determines that it is on a foreign network, it obtains a care-of address from the foreign agent (for a foreign agent care-of address), or through another protocol such as DHCP (for a collocated care-of address).

When using a foreign agent care-of address, the mobile node registers the IP address of the foreign agent with its home agent. The foreign agent is responsible for unmarshalling the tunnelled packets sent to it by the mobile node's home agent and relaying these to the mobile node. It is also responsible for relaying packets from the mobile node to correspondent nodes and to the home agent. Alternatively, the mobile node can be directly connected to the foreign network itself and therefore directly communicate with the home agent (in the figure, this machine is given IP address 128.197.22.5). The foreign agent type of care-of

address is preferable in IPv4 due to its limited address space. Once a node has obtained a new care-of-address, it registers this address with its home agent. Registration with the home agent is performed for the following reasons: a node has moved to a new foreign network and therefore, needs to register a new care-of-address; a node needs to deregister an old care of address with its home agent when it returns to the home network, or when a node needs to re-register with the home agent because its previous registration is about to expire. Understand that many care-of-addresses can be registered for a mobile node at once. This allows for potential situations such as a node quickly alternating between two adjacent wireless cells, and multiple points of attachment are preferable to rapid deregistration and registration between cells.

The mobile node submits a registration request to the foreign agent.

(a) The care-of-address which the mobile node requests will be determined by the foreign agent's agent advertisement messages. The foreign agent will maintain information such as the link-layer address and the IP source address that the mobile node is using before it resends the registration request with its own address as the IP source address to the home agent. The foreign agent does some validity checks on the registration request and then relays the packet to the home agent. The foreign agent makes sure that parameters such as the registration lifetime, and the tunnelling method are supported. It also verifies security authentification information.

(b) The home agent receives the registration request, checks the options of requested service and authentification from the foreign agent. If the request is considered to be valid and serviceable, the agent updates its bindings to record the new care-of-address for the mobile node. The home agent then forms an authenticated registration reply to send back to the foreign agent informing it of a successful or unsuccessful registration.

(c) Finally the foreign agent receives the reply, and sends a new authenticated message back to the mobile node. If all these replies indicate a successful registration, the mobile node will begin to receive its packets tunnelled from the home agent through the foreign agent.

All registration messages include strong authentification to eliminate the possibility of Denial of Service attacks to the nodes on the network. Denial of Service can occur if a rogue machine sends bogus registration requests to a home agent which results in a mobile node's packets being rerouted to the wrong place. Mobile IP keeps this from occurring by requiring all registration messages contain a 16-bit message digest. This is created by using a well-known cryptographic algorithm to create a short unique sequence of bits from the packet's data, a random number to prevent Replay attacks, and a secret key which is shared between the destination and source of the message. The secret

key is a 16-bit number known only between two parties. A discussion on how to distribute keys is beyond the scope of this article, but is a topic which has been considered heavily by the IETF.

In order for the home agent to get packets to the mobile node when it is located away from the home network, the home agent uses encapsulation to forward the mobile node's packets across the network. When a correspondent node wants to send a packet to a mobile node, it simply uses its permanent home address to send the packet to the home network.

(a) The home agent intercepts this packet and performs an encapsulation which consists of creating a new packet which has the entire original packet as its payload. This new packet has the mobile node's care-of address as its destination so it can be sent to the foreign network.

(b) The foreign agent accepts this packet since it is identified as the destination, and de capsulates the original packet to forward it to the mobile node.

(c) Finally, if the mobile node wants to communicate with the corresponding node, it can route its packets directly to that node through the foreign agent, using its home address as the source IP address of the packet.

(d) Note that this sequence of routing events results in what is called "triangle routing". Many of the packets from a corresponding node may have to travel sub-optimal paths in order to be routed through the home network. This is the small price to be paid for the advantage of the corresponding node not needing any sort of Mobile IP extensions.

There are three types of encapsulation which a home agent may use for Mobile IP: IP-in-IP encapsulation, minimal encapsulation, and generic routing encapsulation. IP-in-IP encapsulation creates a new packet with the payload section of the packet containing the original packet. This adds 20 bytes of overhead for the tunnelling, but has the advantage that the resulting packet is exactly like any other IP packet and can handle intermediate network issues such as fragmentation.

Minimal encapsulation takes the approach that since much of the information in an IP-in-IP encapsulated packet is redundant, it only adds the new header information which is different from what was included in the original IP header. Consequently, minimal encapsulation essentially just changes the source and destination addresses and includes the original addresses elsewhere in the packet. This requires an only eight to twelve bytes of overhead to perform the tunnelling, but is less robust since the packet's Time to Live (TTL) timer is not protected in this scheme, and the packet cannot survive fragmentation across the

network. These two side-effects limit the usefulness of using minimal encapsulation for the purposes of tunnelling.

Finally, Generic Routing Encapsulation was designed such that any protocol can be used inside or outside the encapsulated packet. It suffers from even greater amount of overhead than IP-in-IP, but unlike the others can support any protocol. It has explicit protection support for situations such as recursive encapsulation. To recap, in this section we described the basic operation of Mobile IP. We showed how agent advertisements and solicitation allow the mobile node to determine its point of attachment on the network. We then described how the registration process uses authenticated messages to inform the foreign agent and the home agent of its new IP address. This is a necessary step for equipping the home agent with enough data so it can tunnel packets to the mobile node's new location.

SELF-CHECK 7.2

I. Choose one of the following options:

1. Mobile IP is most often found in wireless WAN environments where users need to carry their mobile devices across multiple LANs with ..... IP addresses.

   (a) Different (b) Same

2. The mibile IP exchange of messages between the mobile nodes and the other key entities ..... the network.

   (a) Outside (b) Within

3. The node running the Mobile IP protocol is assigned a .....IP address which defines where all its packets should be sent.

   (a) Temporary (b) Permanant

4. The process of encapsulating an IP packet within another IP packet for the purpose of routing it to a location other than the one specified in the original destination field is the work of ..... .

   (a) Tunnelling (b) Foreign agent

5. The address that the mobile node uses for communication when it is away from its home networkis..... .

   (a) Correspondant address (b) Care- off address

6. If a mobile node needs to immediately know the address of a potential agent without waiting for the next advertisement, it can broadcast an ..... .

   (a) agent solicitation message (b) mobile IP over the internet

7. Denial of Service can occur if a rogue machine sends bogus registration requests to a home agent which results in a mobile node's packets being .....

   (a) Rerouted to wrong place (b) Routed to another place.

II. Answer the following questions

1. What do you mean by Quality of Service?
2. Explain the concept of mobile IP.

ACTIVITY 7.2

Is mobile IP necessary for connecting one mobile phone to another? If yes, then explain its working.

7.5 S M S

SMS is a universal data service and is supported on GSM, TDMA, and CDMA networks. The SMS may be defined as the transmission of alphanumeric messages to and from mobile phones to external systems like e-mail and pagers.

SMS is the delivery of alphanumeric messages to mobile phones over wireless networks. The length of the message can be no longer than 160 characters. In Europe, two-way SMS messaging has been popular for some time and is slowly gaining popularity in North America as some of the major wireless networks (like AT&T) are beginning to support it. An SMS message can originate from an

external system such as e-mail or mobile device and is routed through the network, via the Short Messaging Service Center (SMSC), to its destination. A distinguishing feature of SMS is its ability to deliver messages any time, regardless of whether data or voice calls are in progress.

7.5.1 Benefits

SMS first appeared in Europe in 1991 as part of the Global System for Mobile Communications (GSM) Phase 1 standard. SMS was made available in North America recently, and was first adopted on digital networks built by early wireless carriers such as BellSouth Mobility, Nextel, and AT&T. SMS is supported on digital wireless networks based on GSM, Code Division Multiple Access (CDMA), and Time Division Multiple Access (TDMA). SMS has a number of benefits, which include:

(a) Guaranteed delivery of notifications and alerts to single or multiple users.

(b) Increased user productivity through instant delivery of notifications and alerts.

(c) Another service and source of revenue for service providers.

(d) Low cost and reliable communication mechanism for information delivery.

(e) Integration with Internet-based applications.

(f) Very possible replacement of existing two-way paging.

7.5.2 SMS Architecture Description

The following are descriptions of the SMS architecture elements:

(a) **MS (Mobile Station)**
A wireless terminal that is capable of receiving and sending alphanumeric messages.

(b) **SME (Short Message Entity)**
Which can be a device like a mobile phone, or an application like e-mail that is capable of receiving and sending alphanumeric messages.

(c) **SMSC (Short Message Service Center)**
Responsible for storing and forwarding messages to and from the mobile station.

(d) **STP (Single Transfer Point)**
Which allows for interconnections over Signaling System 7 (SS7) links and multiple network elements. For more information on SS7 links, see the Resources section.

(e) **HLR**
A database in the network that holds information like subscriber and service profile, as well as subscriber. Routing information for the subscriber is also stored in the HLR, which is requested by the SMSC.

(f) **MSC (Mobile Switching Service Center)**
The job of the mobile switching service center is to switch connections between mobile stations, or between mobile stations and the fixed network.

(g) **BS (Base Station)**
Which relays information to and from the mobile station to the MSC. The BS consists of controllers and transceiver stations, also known as "cells."

7.5.3 SMS Applications

(a) **Consumer Applications**
Examples of SMS consumer applications include:

(i) **Peer-to-peer messaging** – This is the most common type of use of SMS. A message like, "Hello, how are you?" or "Meet you at 8:00 PM for dinner" is usually exchanged between two mobile users. This is a quick, efficient, and inexpensive method of communication.

(ii) **Information services** – Information services include stock quotes, weather forecasts, and news updates. A simple SMS application would require a user to type in "ST" for stock quotes or "WEA" for weather forecast. Upon submitting the request, the user receives the appropriate information in the form of an SMS message.

(iii) **Advertising** – SMS can be used to send targeted alerts to a user. The user would sign up to receive special alerts informing the user of upcoming events. Additionally, businesses can use SMS as a form of low-cost advertising.

(b) **Commercial/Enterprise Applications**
Examples of SMS commercial/enterprise applications:

(i) **Customer Service** – SMS can be used as a customer service tool, thereby avoiding expensive person-to-person voice calls to customer service centers. This is an efficient and inexpensive method of providing account status and other pertinent information.

(ii) **Job dispatch** – SMS can be used in job dispatching applications to communicate information between office-based and mobile staff. Sending an address to a message courier in the field is one example of

a job dispatch application. The dispatch application can be integrated with other applications, such as vehicle positioning applications.

SELF-CHECK 7.3

1. Explain the concept of SMS.
2. What do you know about SMS Applications?

SUMMARY

- QoS is a method of providing better service for selected traffic types over various types of packet-switched networks. The network medium used could be any one of several types of technology ranging from Ethernet to Wireless and Frame Relay networks. QoS provid es a method for determining which traffic should be given priority on a network segment.

- The definition of mobile IP says that this is the underlying technology for support of various mobile data and wireless networking applications. For example, GPRS depends on mobile IP to enable the relay of messages to a GPRS phone via the SGSN from the GGSN without the sending needing to know the serving node IP address.

KEY TERMS

GR	GPRS Register
GSN	GPRS Support Node
GGSN	Gateway GSN HBR High Bit Rate
MIB	Management Information Base

SELF-TEST

1. Define the concept of wireless connectivity.
2. What do you mean by Mobile IP and how mobile IP works?
3. When do we need SMS and how it is related to the wireless connectivity?

Topic 8 ▶ Synchronisation and Replication of mobile Data

LEARNING OUTCOMES

By the end of this topic, you should be able to:

1. Describe the concept of replication and synchronisation;
2. Define Sync ML; and
3. Outline the concept of WebDAV.

▶ INTRODUCTION

In the previous topic, we have studied Wireless Connectivity and Mobile Applications, Quality of Service, Survey of Wireless Networking Technologies, Mobile IP and SMS. Now, in this topic, we would focus on Synchronization and Replication of Mobile Data.

8.1 TAXONOMY OF REPLICATION AND SYNCHRONISATION

The taxonomy of replication and synchronisation defines web synchronisation for merge replication. It lets you replicate data by using the HTTPS protocol, and is useful for the following scenarios:

(a) Web synchronisation for merge replication is used for synchronising data from mobile users over the Internet.

(b) It is also used for synchronising data between Microsoft SQL Server databases across a corporate firewall.

The Adventure Works' IT department has configured each portable computer with SQL Server and has enabled merge replication to use Web synchronisation. The Merge Agent on each portable computer has an Internet URL that points to

the replication components that are installed on a computer that is running Microsoft Internet Information Services (IIS). These components synchronise the Subscriber with the Publisher. Each representative can now connect through any available Internet connection without using a remote dial-up connection, and can upload and download the appropriate data. The Internet connection uses Secure Sockets Layer (SSL); therefore, a Virtual Private Network (VPN) is not required.

8.1.1 Overview of How Web Synchronisation Works

When Web synchronisation is used, updates at the Subscriber are packaged and sent as an XML message to the computer that is running IIS by using the HTTPS protocol. The computer that is running IIS then sends the commands to the Publisher in a binary format, typically by using TCP/IP. Updates at the Publisher are sent to the computer that is running IIS and then packaged as an XML message for delivery to the Subscriber.

The following illustration shows some of the components that are involved in Web synchronisation for merge replication.

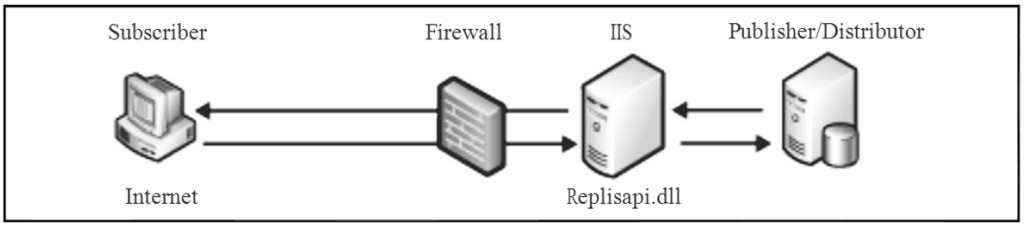


Figure 8.1: Web synchronisation for merge replication
Source: https://myserver.mydomain.com

Web synchronisation is an option only for pull subscriptions; therefore, a Merge Agent will always run on the Subscriber. This Merge Agent can be the standard Merge Agent, the Merge Agent ActiveX control, or an application that provides synchronisation through Replication Management Objects (RMO). To specify the location of the computer that is running IIS, use the Internet Url parameter for the Merge Agent. The web synchronisation for merge replication is shown in Figure 8.1.

The SQL Server Replication Listener (Replisapi.dll) is configured on the computer that is running IIS and is responsible for handling messages that are sent to the server from the Publisher and Subscribers. Each node in the topology handles the XML data stream by using the Merge Replication Reconciler (Replrec.dll).

SQL Server 2005 or a later version is required for all computers that participate in Web synchronisation.

8.1.2 Synchronisation Process

The following steps occur during synchronisation:

(a) The Merge Agent is started at the Subscriber. The agent does the following:
   (i) Makes an SQL connection to the subscription database.
   (ii) Extracts any changes from the database.
   (iii) Makes an HTTPS request to the computer that is running IIS.
   (iv) Uploads data changes as an XML message.

(b) The SQL Server Replication Listener and Merge Replication Reconciler that are hosted on the computer that is running IIS do the following:
   (i) Respond to the HTTPS request.
   (ii) Make an SQL connection to the publication database.
   (iii) Apply the upload changes to the publication database.
   (iv) Extract the download changes for the Subscriber.
   (v) Send an HTTPS response back to the Merge Agent.

(c) The Merge Agent at the Subscriber then accepts the HTTPS response and applies the download changes to the subscription database.

8.2 DATA REPLICATION AND SYNCHRONISATION FOR MOBILE APPLICATION

Data replication involves copying of data from one source location to another target location. Data replication is a technique used in distributed environment to increase the data availability and consistency and to improve the system performance. These environments are characterised by a fixed infrastructure where the user uses the fixed machine that have sufficient resources' and are permanently connected to the network. Replication has been used to improve performance and reduce service time.

Two techniques are used in the replication one of them is static replication and other is dynamic replication. In passive replication, clients communicate with a primary replica that propagates the update message to other replicas. In active replication, clients communicate by multicast with all replicas.

The replication scheme is represented by three plans namely a placement plan, a localisation plan and a consistency plan. These plans are provided by the replica planner, the localisation manager, and the consistency manager, respectively.

The placement plan indicates the placement of each replication nodes. The localisation plans indicates for each user a set of nodes and links where the user can reach each replica. The consistency plan indicates for each replica a set of nodes and links where an update can be propagated from this replica to other replicas. The replication scheme is then stored in the history. The placement plan is immediately projected into the environment. The localisation and consistency plans are used at the read/write time.

SELF-CHECK 8.1

I. Fill in the blanks

1. The taxonomy of replication and synchronisation, lets you to replicate data by using the ..... protocol.
2. SSL stands for ..... .
3. Replication has been used to improve ..... and reduce .....
4. Two techniques used in replication are ..... and .....

II. Answer the following

1. What do you mean by Replication and Synchronisation?
2. Explain Synchronisation Process in detail.

ACTIVITY 8.1

Is the concept of replication useful for mobiles? Discuss this statement with your friends.

8.3 SYNC ML

The Sync ML may be termed as Synchronisation Markup Language. The Sync ML may be defined as an XML based protocols family, aimed at providing remote synchronisation of mobile devices. The Sync ML is a technology which is very frequently used in the present day's mobile computing environment. It is used in the devices like, mobile phones or PDAs.

Synchronisation Markup Language (Sync ML), is an open source standard which is now known as Open Mobile Alliance (OMA) for digital devices to inter-operate regardless of manufacturer or brand. For example, SyncML might be used between a cell phone and a computer, or a Personal Digital Assistant (PDA) and a network. The advantage of SyncML is that it is platform-independent, making it a flexible solution for connecting digital devices of all kinds.

The power of synchronisation is that it allows updates on one device to be automatically and instantly reflected in all other devices without manual updating. To a single user at home, this means that synching a PDA to the computer will update the calendar or task list automatically. But what happens if a company wants to update inventory to a field of 200 internationally placed sales representatives.

8.3.1 Features of Sync ML

The features of Syn ML are:

(a) Basic authentication support.

(b) It is used to Synchronise with a P800 (Data book, Calendar, Jotter).

(c) Slow Sync/two way Sync decision, based on “Last” Sync events “anchors”.

(d) The Sync ML handles and generates Add, Delete, Replace commands.

(e) The Sync ML handles v Cards, v Calendars, v Notes.

(f) The P800 doesn’t send the drawings with the Jotter notes.

These features lack certain aspects of Syn ML, they are:

(a) Better error handling

(b) Sync ML compliance!/field testing

(c) For some reason, “Add” from server is refused by the phone (500).

(d) Transaction oriented processing

(e) WBXML (binary) support, only plain XML now.

(f) Emails sync.

(g) Content type capabilities enforcing: not sending to the remote device fields it doesn't support.

(h) A command line client, so that you can easily synchronise your .vcf files.

SELF-CHECK 8.2

I. True and False

1. The Sync ML may be termed as Synchronisation Markup Language.
2. The Sync ML does not come under the XML based protocol family.
3. The Sync MLh and les and generates Add, Delete, Replace commands.
4. 4.The Sync ML does not handles v Cards, v Calendars, v Notes.

II. Answer the following

1. What do you mean by Sync ML?
2. Explain the concept of replication and synchronisation in the case of mobile agents.

8.4 WEB DAV

Web-based Distributed Authoring and Versioning (Web DAV) is a set of methods based on the Hypertext Transfer Protocol (HTTP) that facilitates collaboration between users in editing and managing documents and files stored on World Wide Web servers. Web DAV was defined in RFC 4918 by a working group of the Internet Engineering Task Force (IETF).

The Web DAV protocol makes the Web a readable and writable medium, in line with Tim Berners-Lee's original vision. It provides a framework for users to create, change and move documents on a server (typically a web server or "web share"). The most important features of the Web DAV protocol include:

(a) Locking ("overwrite prevention") properties (creation, removal, and querying of information about author, modified date et cetera);

(b) Name space management (ability to copy and move Web pages within a server's namespace); and

(c) Collections (creation, removal, and listing of resources).

As of 2009, many modern operating systems provide built-in support for Web DAV.

Web DAV began in 1996 when Jim Whitehead worked with the World Wide Web Consortium (W3C) to host two meetings to discuss the problem of distributed authoring on the World Wide Web with interested people. Tim Berners-Lee's original vision of the Web envisaged a medium for both reading and writing. In fact, Berners-Lee's first web browser, called Worldwide Web, was able to both view and edit web pages; but, as the Web grew, it became, for most users, a read-only medium. Whitehead and other like-minded people wanted to fix that limitation.

The W3C meeting decided that the best way to proceed was to form an IETF working group, because the new effort would lead to extensions to HTTP, which was being standardised at the IETF.

As work began on the protocol, it became clear that handling both distributed authoring and versioning would involve too much work and that the tasks would have to be separated. The Web DAV group focused on distributed authoring, and left versioning for the future. Versioning was added later by the Delta-V extension- see the Extensions section below.

The protocol consists of a set of new methods and headers for use in HTTP. The added methods include:

(a) **PROPFIND** – used to retrieve properties, stored as XML, from a resource. It is also overloaded to allow one to retrieve the collection structure (a.k.a. directory hierarchy) of a remote system.

(b) **PROPPATCH** – used to change and delete multiple properties on a resource in a single atomic act.

(c) **MKCOL** – used to create collections (a.k.a. a directory).

(d) **COPY** – used to copy a resource from one URI to another.

(e) **MOVE** – used to move a resource from one URI to another.

(f) **LOCK** – used to put a lock on a resource. Web DAV supports both shared and exclusive locks.

(g) **UNLOCK** – to remove a lock from a resource.

ACTIVITY 8.2

In your own word explain in which manner the concept of Web DAV is useful in internet.

8.4.1 Implementations

Linux

Linux users can mount Web DAV shares using davfs2 or fused which mount them as coda or FUSE file systems. KDE has native Web DAV support as part of kio_http. This enables Conqueror and every other KDE application to interact directly with WebDAV servers. Nautilus also has WebDAV support built in. The cadaver command-line client, which provides an FTP-like command set, is included in many Linux distributions.

Mac OS X

Mac OS X version 10.0 and following support Web DAV shares natively as a type of file system. The system can mount Web DAV-enabled server directories to the file system using the traditional BSD mounting mechanism or, more comfortably, through the 'Connect to Server' dialog found in the Finder. Mac OS X version 10.1.1 introduced support for HTTP Digest Access authentication. Mac OS X 10.4 (Tiger) extended WebDAV interoperability to include support for the https scheme, proxies, and additional methods of authentication.

The Finder presents a Web DAV share as an external disk, allowing users to interact with Web DAV just as they would with any other file system. Apple'si Disk uses Web DAV for file access

Microsoft Windows

Microsoft introduced Web DAV client support in Microsoft Windows 98 with a feature called "Web folders". This client consisted of an OLE object which could be accessed by any OLE software, and was installed as an extension to Windows Explorer (the desktop/file manager) and was later included in Windows 2000. In

Windows XP, Microsoft added the "WebDAV mini-redirector" which is preferred by default over the old Web folders client. This newer client works as a system service at the network-redirector level (immediately above the file-system), allowing Web DAV shares to be assigned to a drive letter and used by any software. The redirector also allows Web DAV shares to be addressed via UNC paths (e.g., http://host/path/ is converted to \\host\path\) for compatibility with Windows file system APIs. However, some versions of the redirector are known to have some limitations in authentication support. This includes the fact that recent versions disable basic authentication for HTTP connections for security purposes. Some suggested workarounds for problems in some versions include

In addition, Web DAV over HTTPS works only if a computer has KB892211-version files or newer installed. Otherwise Windows displays "The folder you entered does not appear to be valid. Please choose another" when adding a network resource. NOTE: 892211 has been superseded by KB907306.

Windows Vista includes only the Web DAV redirector, but if you install a version of Office, Internet Explorer, OLE-DB or "Microsoft Update for Web Folders" you will get the original "Web folders" client. The update will only work on the 32bit version of XP/Vista.

SELF-CHECK 8.3

1. Explain the concept of Web DAV.
2. Explain how Web Synchromisation works?

SUMMARY

- Sync ML is used for synchronisation of mobile devices it is an XML based protocols family. It is built-in more and more mobile devices, such as mobile phones or PDAs. Web-based Distributed Authoring and Versioning (Web DAV) is a set of methods based on the Hypertext Transfer Protocol (HTTP) that facilitates collaboration between users in editing and managing documents and files stored on World Wide Web servers.

KEY TERMS

DAB Digital Audio Broadcasting

FA Foreign Agents

FEC Forward Error Correction

SELF-TEST

1. How synchronisation is related to the replication of the data?
2. Define the synchronisation process in detail.
3. What is the web based distributed authoring?

Topic 9 ▶ Mobility and Location-based Services

LEARNING OUTCOMES

By the end of this topic, you should be able to:

1. Describe data acquisition of location information;
2. Underline the concept of geographic information systems;
3. Elaborate on location based service applied working;
4. State representation of location with UML;
5. Describe security and privacy of location information; and
6. Underline the concept of internationalisation and localisation.

▶ INTRODUCTION

In the previous topic, we have studied Synchronization and Replication of Mobile Data, Taxonomy of Replication Synchronization, Data Replication and Synchronization for Mobile Applications, SyncML, Mobile Agents, Replication, and Synchronization and Using UML to Represent Data Replication and Synchronization Schemes. Now, in this topic, we would focus on Mobility and Location-Based Services.

Location-based Services

Location-based services determine the location of the user by using one of the several technologies for determining position, then use the location and other informations to provide personalized applications and services. Location-based services answer three questions: Where am I? What's around me? How do I get there? Traffic advisories, navigation help including maps, directions and roadside assistance are natural location-based services. Other services can combine present location with information about personal preferences to help users find food, lodging, and entertainment to fit their tastes and pocketbooks.

9.1 DATA ACQUISITION OF LOCATION INFORMATION

Data acquisition systems, may be defined as the products and/or processes used to collect information to document or analyse some phenomenon. In the simplest form, a technician logging the temperature of an oven on a piece of paper is performing data acquisition. As technology has progressed, this type of process has been simplified and made more accurate, versatile, and reliable through electronic equipment. The equipments range from simple recorders to sophisticated computer systems. Data acquisition products serve as a focal point in a system, tying together a wide variety of products, such as sensors that indicate temperature, flow, level, or pressure. Some common data acquisition terms are given below:

(a) **Analog-to-Digital Converter (ADC)**
The analog-to-digital converter is the heart of most data acquisition systems. It is an electronic device that converts analog signals to an equivalent digital form.

(b) **Digital-to-Analog Converter (D/A)**
The working of Digital-to-Analog Converter is just opposite to the Analog to Digital converter. This is used to convert the Digital Signal into the Analog signal.

(c) **Digital Input/Output (DIO)**
Digital I/O are discrete signals which are either one of two states. These states may be on/off, high/low, 1/0, etc. Digital I/O is also referred to as binary I/O.

(d) **Differential Input**
Differential inputs have a unique high and unique low connection for each channel. Data acquisition devices have either single-ended or differential inputs, many devices support both configurations. It refers to the way a signal is wired to a data acquisition device.

(e) **General Purpose Interface Bus (GPIB)**
It is also called IEEE 488 in reference to defining ANSI/IEEE standards. Synonymous with HPIB (for Hewlett-Packard), the standard bus used for controlling electronic instruments with a computer.

(f) **Resolution**
Resolution can be expressed in bits, in proportions, or in percent of full scale. For example, a system has 12-bit resolution, one part in 4,096 resolutions, and 0.0244 percent of full scale.

(g) **RS232**
RS232 is the most common serial communication, however, it is somewhat limited in that it only supports communication to one device connected to the bus at a time and it only supports transmission distances up to 50 feet. It is a standard for serial communications found in many data acquisition systems.

(h) **RS485**
RS485 is not as popular as RS232, however, it is more flexible in that it supports communication to more than one device on the bus at a time and supports transmission distances of approximately 5,000 feet.

(i) **Sample Rate**
The sample rate defines the speed at which a data acquisition system collects data. The speed is normally expressed in samples per second. For multi-channel data acquisition devices the sample rate is typically given as the speed of the analog-to-digital converter (A/D). To obtain individual channel sample rate, you need to divide the speed of the A/D by the number of channels being sampled.

(j) **Single-ended Input (SE)**
In single-ended wiring, each analog input has a unique high connection but all channels share a common ground connection. Data acquisition devices have either single-ended or differential inputs. Many support both configurations.

SELF-CHECK 9.1

Choose one of the options below:

1. ..... systems, may be defined as the products and/or processes used to collect in formation to document or analyse some phenomenon.
2. ..... refers to the way a signal is wired to a data acquisition device.
3. The standard bus used for controlling electronic instruments with a computer is ..... .
4. ..... only supports communication to one device connected tothe bus at a time and it only supports transmission distances up to 50 feet.
5. The ..... defines the speed at which a data acquisition system collects data.

Options:

1. RS232,
2. Differential inputs,
3. Sample rate,
4. RS485,
5. Digital I/O,
6. General Purpose Interface,
7. Data acquisition

9.1.1 Wireless Data Acquisition Systems

These systems consist of one or more wireless transmitters sending data back to a wireless receiver connected to a remote computer. Wireless transmitters are available for ambient temperature and relative humidity, thermocouples, RTDs, pulse output sensors, 4 to 20 mA transmitters and voltage output transducers. Receivers can be connected to the USB or Ethernet port on the PC. Wireless data acquisition systems can eliminate costly and time consuming field wiring of process sensors.

9.1.2 Serial Communication Data Acquisition Systems

There are several different communication standards; RS232 is the most common but only supports transmission distances up to 50 feet. RS485 is superior and supports transmission distances to 5,000 feet. Serial communication data acquisition systems are a good choice when the measurement needs to be made at a location which is distant from the computer.

9.1.3 USB Data Acquisition Systems

USB offers several advantages over conventional serial and parallel connections, including higher bandwidth (up to 12 M bits/s) and the ability to provide power to the peripheral device. USB is ideal for data acquisition applications. Since USB connections supply power, only one cable is required to link the data acquisition device to the PC, which most likely has at least one USB port. The Universal Serial Bus (USB) is a new standard for connecting PCs to peripheral devices such as printers, monitors, modems and data acquisition devices.

9.1.4 Data Acquisition Plug-in Boards

Boards offered are primarily for IBM PC and compatible computers. Features provided by the cards can vary due to number and type of inputs (voltage, thermocouple, on/off), outputs, speed and other functions provided. Each board installed in the computer is addressed at a unique Input/output map location. The I/O map in the computer provides the address locations the processor uses to gain access to the specific device as required by its program. Computer data acquisition boards plug directly into the computer bus. Advantages of using boards are speed (because they are connected directly to the bus) and cost (because the overhead of packaging and power is provided by the computer).

ACTIVITY 9.1

You must have heard about the concept of location based billing. It is the ability to have a preferential billing. Perform a search on this concept and identify how this application works?

9.2 GEOGRAPHIC INFORMATION SYSTEMS

Geographic data is an important aspect of any location system. Geographic Information Systems (GIS) provide the tools to provision and administer base map data such as manmade structures (streets, buildings) and terrains (mountains, rivers). GIS is also used to manage point-of-interest data such as location of gas stations, restaurants, nightclubs, etc. Finally, GIS information also includes information about the radio frequency characteristics of the mobile network. This allows the system to determine the serving cell site of the user.

9.3 BUILDING INFORMATION MODELLING

The idea behind a building information model is that of a single repository. Every item is described only once. Both graphical documents, drawings and non-graphical documents, specifications, schedules, and other data are included. Changes are made to each item in only one place. Thus, every user of the repository can be certain that what they are seeing is exactly the same information that every other project participant sees. This one change to the way project documentation is handled can greatly reduce the number of communication problems that slow down projects and increase costs. However, a

building information model does not necessarily ensure bi-directional connectivity among all manifestations of the model. In some building-information-model systems, information is extracted, rather than linked; for example, production drawings. Modifications made to these extracted documents are not reflected in the model; it must be separately updated. The building information model also raises, but does not directly address, additional questions: Who owns the data in the model? Who is responsible for updating it? What about access and security?

9.4 LOCATION BASED SERVICE APPLIED WORKING

Location Based Service (LBS) is the application that users use to locate its own position with the help of some basic components like mobile devices, mobile communication network, and service provider like the Global Positioning Service (GPS), data and yellow pages for the service station.

This is one of the most popular services based on a different navigation technologies provided by the mobile communication network. Its receives the location coordinates from the ground based mobile station and sends it to the mobile phone user and the communication center that can be used for various location based services. It consists of basic electronic instruments like mobile phone, smart phone, laptop and other Personal Digital Devices (PDA) for accessing information. User can obtain information in form of voice, text, graphs, picture, etc. Devices can be for different purposes; specific and multi purpose.

LBS is designed to provide valuable information to the users based on location or position. So it gives answer to some of the most common answered question when one is travelling such that where am I? What is around me? Where is the nearest hotel, hospital? What is the most convenient route?

Application: The popularity of location- based service makes it one of most essential and useful asset in almost all industries. However, its market is divided into various categories including navigation, emergency assistance, tracking, advertising, billing, management, games and leisure.

Types: Location based service enables us to find the geographical location of a mobile device with its user and provides various services related to the location information. For example, you are in a town and want to know about a nearest hotel. Here you can use your GPS attached mobile phone that can calculate the nearest one from its current database and send it through message.

SELF-CHECK 9.2

1. What do you mean by GIS service in mobile?
2. Explain the Location/Building Information modelling.

9.5 UTILISING LOCATION BASED SERVICES WITH MOBILE APPLICATIONS

The main component in the nomadic computing architecture is the Mobility Manager which interacts with all of the other ancillary servers to make adaptability decisions based on changes in the computing environment. Information about changes comes from the Location Manager and the Network QOS Manager. The Mobility Manager is only responsible for deciding how applications should be adapted and not for the adaptation process itself (this is performed by an appropriate server which is invoked by the Mobility Manager). The types of adaptability include manipulating communication owes and object interfaces, reconfiguring applications by migrating or using replicas of some application objects, or replacing applications by other equivalent applications. The whole architecture and functionality of the servers have been designed as an extension to concepts in Open Distributed Computing. However, its prototype is being built in the CORBA environment which is not fully aligned with the RM-ODP framework and therefore, we had to enhance CORBA's functionality in the following aspects:

(a) An extension of the CORBA RPC model was necessary to allow exiles specification of RPC semantics and QOS with the semantics and QOS implemented using intermediate agents responsible for dealing with disconnections and relocation of communicating objects; and

(b) An extension to the CORBA's type management, trading and relationship services was needed in order to incorporate descriptions of object mobility prowlers, object states and relationships between object types and also between objects.

ACTIVITY 9.2

How can we use location based services in mobile phones?

9.6 REPRESENTING LOCATION WITH UML

Web services are based on open XML standards and define a standardised mechanism to describe, locate and communicate with online applications. In the last decade, many businesses and organisations have used different approaches to interact with others taking the advantages and infrastructure of the Web. Web services are emerging to provide a systematic and extensible framework for application-to-application interactions, built on existing Web protocols. Now, the services are one of the most important issues in the scope of Web Information Systems (WIS) development. There are several middleware platforms, such as JAVA or .NET that allow implementing Web services and facilitate the service-oriented applications development. However, the lack of a solid methodological base for the development of Web services, as well as service-oriented applications give rise to the need of methods or modelling techniques that can guarantee the quality in the development of this kind of applications.

In the last years, a large amount of modeling techniques and methodologies for the development of WIS and service-oriented WIS has appeared. MIDAS is a model-driven methodological framework for agile development of WIS that proposes to use standards in the development process. It is based on XML and proposes to model the whole system in UML.

MIDAS defines Platform Independent Models (PIMs) and Platform Specific Models (PSMs) according to three aspects: content, hypertext and behavior, and also proposes some mapping rules between the different models. As MIDAS proposes to use UML notation to model the whole system, both the PIMs and PSMs will be represented in this notation.

MIDAS proposes to model the behavior of a WIS following the service oriented approach. We define a service as a specific functionality that the WIS offers to the user. A service is a conceptual definition which will exist just at PIM level. A service could be represented, at PSM level, by one Web service or by the composition of several Web services. So, a service can be basic for example to validate an email address; or a composite as to locate the best prices on ticket airlines that probably involve many basic services. Thus, much basic services as composite services utilise "service description" to offer their functionality so that they can be located and utilised through the Web.

A web service is a software component, independent from platform and implementation that can be described using a service description language, published to a registry of services, discovered through a standard mechanism,

invoked through a declared API (Applications Programming Interface) and composed with other services.

Web Services Description Language (WSDL) is the language to describe Web services proposed by the World Wide Web Consortium (W3C). WSDL describes three fundamental properties of the Web service: what a service does (the operation that the service provides), how a service is acceded (details of the data formats and protocols necessary to access the service's operation) and where a service is located (details of protocols-specific network address, such as URL). In this work, a UML extension for Web service representation based on WSDL is proposed. This extension has been carried out with a double purpose: on one hand, to give a UML notation that allows representing a Web service and, on the other hand, to facilitate the automatic generation of WSDL description of a Web service from an UML diagram. Some other related works with Web service modelling and automatic WSDL code generation have appeared during the last years. However, these proposals have some limitations with respect to our goals. The extension proposal in is not complete, since it does not allow operations and parameters modelling, neither relation between these components and others like input or output messages. Since one of our goals is to make easy the automatic generation of Web services description in WSDL from a UML diagram, it will be necessary to define modeling guidelines that allow representing all the needed issues for Web services description maintaining the main benefit of modelling that is the reality abstraction. The XMLSPY5 case tool allows automatic generating of WSDL documents, but starting from its own graphical notation instead of from an UML diagram

9.6.1 Design Guidelines for the UML Extension

To choose the necessary stereotypes to represent in UML all of WSDL components and the relationships between them, the following design guidelines are defined:

(a) TYPES and SCHEMA components have been considered stereotyped compositions with <<Type Schema>> and represent the relation between a DEFINITION component and the data type definitions.

(b) DEFINITION components have been considered as stereotyped class because they are explicitly defined in WSDL and constitute the root component that groups all the used elements.

(c) MESSAGE, PART, PORT TYPE, OPERATION, BINDING, PORT, SERVICE and IMPORT components have been considered stereotyped classes because they represent important components and are explicitly defined in WSDL.

(d) MESSAGE component will be related to the PART component that it uses by means of a composition.

(e) The relationship between a PART component and an ELEMENT component will be represented by means of an association stereotyped with `<<Part-Type>>` if the PART component uses the ELEMENT as a type, and by means of an association stereotyped with `<<Part Elements>>` if the PART component uses the ELEMENT as an element.

(f) The relation between the OPERATION component and the MESSAGE component will be represented by means of an association stereotyped with `<<Input>>`, `<<Output>>` o `<<Fault>>`, depending on the type of message that it associates, that is an input message, an output message or a fault message.

(g) MESSAGE, PORTTYPE, BINDING, SERVICE e IMPORT components will be related to the DEFINITION component by means of a composition.

(h) PORTTYPE component will be related to the OPERATION component that it uses by means of an aggregation.

(i) BINDING component will be associated to the PORT TYPE component that it describes.

(j) SERVICE component will be related to the PORT components that it provides by means of a composition.

(k) The PORT component will be associated to the BINDING component that it uses.

(l) As already said, WSDL uses XML Schema for data type definitions that will be used for message sending. For this reason UML extensions are used to represent XML Schemas.

9.7 SECURITY AND PRIVACY OF LOCATION INFORMATION

Location Based Service (LBS) is primarily based on user's location to provide other value added services by means of a wireless device functioning through common cellular network or radio stations. Till now it has left a wide-ranging impact on society and business as a whole. Tracking and monitoring individuals, children and thieves and its uses in the law enforcement by police really has good implication on the society. In case of business as a vehicle tracking device or asset tracking component, LBS technology acts as a catalyst in the growth of industries especially telecommunication and transportation. However, as the system deals with confidential personal information like location, personal mobile number, concerning address, it becomes vital for the operator to offer

adequate security for maintaining user's privacy. The following paragraph will focus on various privacy issues associated with LBS.

LBS is one of the most exciting development in the telecom industry and with the innovation of other positioning services having excellent ability to locate the position of the user have both pros and cons. On one hand, it has a large implication on both society and economy in terms of a good communication network as a tracking device but on the other hand, the information of users can be misused thereby raising the issues of security both from the point of view of personal privacy and national security. As LBS is present and used all around the world with interoperability, it becomes necessary to deal with all privacy and security issues. Here in the following paragraph, we will focus on some of the most vital issues in using LBS.

9.7.1 Issues

When is the person permitted to monitor someone by using LBS? Is the concerned person's consent necessary? What about individual right and personal autonomy? What kinds of evidences are required to monitor a person? These are some of the questions that need to be answered before using LBS. Monitoring a person can have psychological effect on the person being monitored. In case of monitoring criminals or suspects by police or security agencies, the question of individual freedom came, as enforcing someone's freedom is not at all ethical when the person is only suspected of committing the crime.

(a) **Control (Legal)**
Commonly GPS and other LBS devices are used to control and offer various types of services to the user. Personally it controls one's own direction of moving in guiding the right way. In case of child tracking, parents have the exclusive right to look after their children, as it is not possible for the young ones to make their own decision. So it is their legal right to monitor their children thereby reflecting a sense of caring. In case of law enforcement, special laws provide legal rights to police or security departments to keep an eye on criminals or suspects.

(b) **Trust (Social)**
In social life, trust is the most essential part in human relationship. However, the use of LBS is being practiced in low trust conditions. Monitoring someone with the help of tracking system really affects personal relationship but as far as tracki ng criminals by cops or t racking children by parents are concerned, it is for the welfare of the individual & society.

(c) **Privacy (Ethical)**

As a human being, everyone has the right to privacy or being free from intrusion or disturbances in one's personal life. But in case of LBS or any other telecommunication technologies dealing with transformation of various kinds of information, it becomes essential to provide adequate security to these kinds of data for not being misused by any unauthorised person. Tracking and monitoring someone without his/her consent is purely unethical so needs high level of security. But again as in case of law and order where tracking devices are used to monitor criminals, it becomes essential for the sake of society as a whole. Here, social security is counted higher than Individual safety and security.

(d) **Security (Technological)**

Again for maintaining privacy, security system should be strong. Every technology has both positive and negative impact on human life and LBS also have shortcomings by locating accurate information data or even easily giving access to unauthorised person. On one hand LBS enhances both national and personal security but creates problem as regards the privacy of individual by not providing a foolproof security system to that highly sensitive information is stored in its database. For obtaining security, one needs to do a little compromise on his/her privacy but to what extent is a question.

However, on the whole privacy and security issues of LBS, there are chiefly four points, viz., control, trust, privacy and security as legal, social, ethical and technological aspects. But all four are mutually exclusive as control decreases trust, trust enhances privacy, which needs security, and security again increases control.

SELF-CHECK 9.3

Using location based services, how can we solve the security problem in this service area.

9.8 INTERNATIONALISATION AND LOCALISATION

In computing, internationalisation and localisation means adapting computer software to different languages and regional differences. Internationalisation is the process of designing a software application so that it can be adapted to various languages and regions without engineering changes. Localisation is the process of adapting internationalised software for a specific region or language by adding locale-specific components and translating text.

The terms are frequently abbreviated to the numeronyms i18n (where 18 stands for the number of letters between the first i and last n in internationalisation, a usage coined at DEC in the 1970s or 80s) and L10n respectively, due to the length of the words. The capital L in L10n helps to distinguish it from the lowercase i in i18n.

Some companies, like Microsoft, IBM and Sun Microsystems, use the term "globalisation" for the combination of internationalisation and localisation

9.8.1 Scope

Focal points of internationalisation and localisation efforts include:

(a) Language

(i) Computer-encoded text.

(ii) Alphabets/scripts; Most recent systems use the Unicode standard to solve many of the character encoding problems.

(iii) Different systems of numerals.

(iv) Writing direction which is e.g., left to right in German, right to left in Persian, Hebrew and Arabic.

(v) Spelling variants for different countries where the same language is spoken, e.g., localisation (en-US, en-CA, en-GB-oed) vs. localisation (en-GB, en-AU).

(vi) Text processing differences, such as the concept of capitalisation which exists in some scripts and not in others, different text sorting rules, etc.

(b) **Input**

(i) Enablement of keyboard shortcuts on any keyboard layout.

(ii) Graphical representations of text (printed materials, online images containing text).

(iii) Spoken (Audio).

(iv) Subtitling of film and video.

(c) **Culture**

(i) Images and colors: issues of comprehensibility and cultural appropriateness.

(ii) Names and titles.

(iii) Government assigned numbers (such as the Social Security number in the US, National Insurance number in the UK, and Resident registration number in South Korea) and passports.

(iv) Telephone numbers, addresses and international postal codes.

(v) Currency (symbols, positions of currency markers).

(vi) Weights and measures.

(vii) Paper sizes.

(d) **Writing Conventions**

(i) Date/time format, including use of different calendars.

(ii) Time zones (UTC in internationalised environments).

(iii) Formatting of numbers (decimal separator, digit grouping).

(iv) Any other aspect of the product or service that is subject to regulatory compliance.

The distinction between internationalisation and localisation is subtle but important. Internationalisation is the adaptation of products for potential use virtually everywhere, while localisation is the addition of special features for use in a specific locale. Internationalisation is done once per product, while localisation is done once for each combination of product and locale. The processes are complementary, and must be combined to lead to the objective of a system that works globally. Subjects unique to localisation include:

(a) Language translation,

(b) National varieties of languages,

(c) Special support for certain languages, such as, East Asian languages,

(d) Local customs,

(e) Local content,

(f) Symbols ,

(g) Order of sorting (Collation),

(h) Aesthetics,

(i) Cultural values and social context.

SELF-CHECK 9.4

1. Explain the Security and Privacy Issues of Location Information;
2. Explain the Concept of Localisation and Internationalisation.

SUMMARY

- Location based Service (LBS) is the application that users use to locate its own position with the help of some basic components like mobile devices, mobile communication network, service provider like the Global Positioning Service (GPS), data and yellow pages for the service station.

- GIS is a system of hardware and software used for storage, retrieval, mapping, and analysis of geographic data. Practitioners also regard the total GIS as including the operating personnel and the data that go into the system. Spatial features are stored in a coordinate system (latitude/ longitude, state plane, UTM, etc.), which references a particular place on the earth. Descriptive attributes in tabular form are associated with spatial features. GIS can be used for scientific investigations, resource management, and development planning.

- In computing, internationalisation and localisation means of adapting computer software to different languages and regional differences. Internationalisation is the process of designing a software application so that it can be adapted to various languages and regions without engineering changes. Localisation is the process of adapting internationalised software for a specific region or language by adding locale-specific components and translating text.

KEY TERMS

IMT International Mobile Telecommunications

IP Internet Protocol

IR Infra Red

PSK Phase Shift Keying

SELF-TEST

1. Write the short notes on the ADC,D/A,DIO,GPIB and RS485.
2. What do you mean by the localization information modelling?
3. What is the privacy processes used in the localisation of the information?

Topic 10 ▶ Active Transaction and Mobile Security

LEARNING OUTCOMES

By the end of this topic, you should be able to:

1. Describe the concept of security in wireless network;
2. Identify the security aspect in ad hoc network;
3. Outline location information services; and
4. Contrast between security and privacy.

▶ INTRODUCTION

In the previous topic, we have studied Mobility and Location-based Services, Data Acquisition of Location Information, Location Information Modeling, Location-based Services Applied, utilizing Location-based Services with Mobile Applications, Representing Location with UML, Security and Privacy of Location Information, Localization and Internationalization and Latest Developments in Location-based Efforts. Now, in this topic, we would focus on active transaction.

10.1 ACTIVE COMPUTING AND WIRELESS INFRASTRUCTURE

An active transaction includes all those behaviours exhibited by the system that are started autonomously by the application without the immediate and synchronous invocation of the software by the user, for example, Active mobile applications.

The initiation of active transactions with the user involves one of the two following conditions:

(a) Some condition observable by the mobile device or the application running on it occurs and there is a mobile application running on the mobile device that can recognise this condition. This application can then initiate a series of interactions with the user. For example, a simple reminder of an appointment that allows the user to push a snooze or a cancel button. The information resides in the PIM that notifies the user.

(b) The condition that is to trigger the initiation of the interaction with the user. The part or all of the application that interacts with the user.

A Wireless Local Area Network (WLAN) links two or more devices using some wireless distribution method (typically spread-spectrum or OFDM radio), and usually providing a connection through an access point to the wider internet. This gives users the mobility to move around within a local coverage area and still be connected to the network.

Large wireless network projects are being put up in many major cities. New York City, for instance, has begun a pilot program to cover all five boroughs of the city with wireless Internet access. Wireless LANs have become popular in the home due to ease of installation, and the increasing popularity of laptop computers. Public businesses such as coffee shops and malls have begun to offer wireless access to their customers; sometimes for free.

10.1.1 Architecture

(a) **Stations**

All components that can connect into a wireless medium in a network are referred to as stations. All stations are equipped with Wireless Network Interface Cards (WNICs). Wireless stations fall into one of two categories: access points, and clients.

Access Points (APs), normally routers, are base stations for the wireless network. They transmit and receive radio frequencies for wireless enabled devices to communicate with. Wireless clients can be mobile devices such as laptops, personal digital assistants, IP phones, or fixed devices such as desktops and workstations that are equipped with a wireless network interface.

(b) **Basic Service Set**

The Basic Service Set (BSS) is a set of all stations that can communicate with each other. There are two types of BSS: Independent BSS (also referred to as

IBSS), and infrastructure BSS. Every BSS has an identification (ID) called the BSSID, which is the MAC address of the access point servicing the BSS. An infrastructure can communicate with other stations not in the same basic service set by communicating through access points.

(i) An Independent BSS (IBSS) is an ad-hoc network that contains no access points, which means they cannot connect to any other basic service set.

(ii) An Extended Service Set (ESS) is a set of connected BSS. Access points in an ESS are connected by a distribution system. Each ESS has an ID called the SSID which is a 32-byte (maximum) character string.

(c) **Distribution System**

A Distribution System (DS) connects access points in an extended service set. The concept of a DS can be used to increase network coverage through roaming between cells.

10.1.2 Types of Wireless LANs

Peer-to-Peer or Ad-hoc Wireless LAN

An ad-hoc network is a network where stations communicate only peer to peer (P2P). There is no base and no one gives permission to talk. This is accomplished using the Independent Basic Service Set (IBSS).

A peer-to-peer (P2P) network allows wireless devices to directly communicate with each other. Wireless devices within range of each other can discover and communicate directly without involving central access points. This method is typically used by two computers so that they can connect to each other to form a network.

If a signal strength meter is used in this situation, it may not read the strength accurately and can be misleading, because it registers the strength of the strongest signal, which may be the closest computer.

Hidden Node Problem

Devices A and C are both communicating with B, but are unaware of each other IEEE 802.11 defines the physical layer (PHY) and MAC (Media Access Control) layers based on CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance). The 802.11 specification includes provisions designed to minimise collisions, because two mobile units may both be in range of a common access point, but out of range of each other.

The 802.11 has two basic modes of operation: Ad hoc mode enables peer-to-peer transmission between mobile units. Infrastructure mode in which mobile units communicate through an access point that serves as a bridge to a wired network infrastructure is the more common wireless LAN application the one being covered. Since wireless communication uses a more open medium for communication in comparison to wired LANs, the 802.11 designers also included shared-key encryption mechanisms: Wired Equivalent Privacy (WEP), Wi-Fi Protected Access (WPA, WPA2), to secure wireless computer networks.

Bridge

A bridge can be used to connect networks, typically of different types. A wireless Ethernet bridge allows the connection of devices on a wired Ethernet network to a wireless network. The bridge acts as the connection point to the Wireless LAN.

Wireless Distribution System

A Wireless Distribution System is a system that enables the wireless interconnection of access points in an IEEE 802.11 network. It allows a wireless network to be expanded using multiple access points without the need for a wired backbone to link them, as is traditionally required. The notable advantage of WDS over other solutions is that it preserves the MAC addresses of client packets across links between access points.

An access point can be either a main, relay or remote base station. A main base station is typically connected to the wired Ethernet. A relay base station relays data between remote base stations, wireless clients or other relay stations to either a main or another relay base station. A remote base station accepts connections from wireless clients and passes them to relay or main stations. Connections between "clients" are made using MAC addresses rather than by specifying IP assignments.

All base stations in a Wireless Distribution System must be configured to use the same radio channel, and share WEP keys or WPA keys if they are used. They can be configured to different service set identifiers. WDS also requires that every base station be configured to forward to others in the system.

WDS may also be referred to as repeater mode because it appears to bridge and accept wireless clients at the same time (unlike traditional bridging). It should be noted; however, that throughput in this method is halved for all clients connected wirelessly.

When it is difficult to connect all of the access points in a network by wires, it is also possible to put up access points as repeaters.

10.1.3 Roaming

Roaming between Wireless Local Area Networks

There are two forms of wireless LAN roaming; they are: Internal Roaming and External Roaming.

(a) **Internal Roaming**
A Mobile Station roaming from one access point to another often interrupts the flow of data between the Mobile Station and an application connected to the network. The Mobile Station (MS) moves from one Access Point (AP) to another AP within a home network because the signal strength is too weak. An authentication server (RADIUS) assumes the re-authentication of MS via 802.1x (e.g., with PEAP). The billing of QOS is in the home network. The Mobile Station, for instance, periodically monitors the presence of alternative access points (ones that will provide a better connection). At some point, based upon proprietary mechanisms, the Mobile Station decides to re-associate with an access point having a stronger wireless signal. The Mobile Station, however, may lose a connection with an access point before associating with another access point. In order to provide reliable connections with applications, the Mobile Station must generally include software that provides session persistence.

(b) **External Roaming**
The MS (client) moves into a WLAN of another Wireless Internet Service Provider (WISP) and takes their services (Hotspot). The user can independently use his home network along with foreign network, if this is open for visitors. There must be special authentication and billing systems for mobile services.

10.2 PRACTICAL CONSIDERATIONS OF BUILDING ACTIVE SYSTEMS

For a given cost and level of technology, considerations of weight, power, size and will exact a penalty in computational resources such as processor speed, memory size, and disk capacity. Connectivity is highly variable in performance and reliability. These elements rely on a finite energy source. These constrains are not artifacts of current technology, but are intrinsic to mobility. Together they require us to rethink approaches and information access.

10.3 MOBILE SECURITY IN WIRELESS NETWORKS

Today's environment is an internet environment and a large range of systems come in to the wide range of networks and for a wide variety of applications drives need to support security solutions that meet the requirements of a wide variety of customers.

Much attention has been focused recently on the security aspects of existing Wi-Fi (IEEE 802.11) Wireless LAN systems.

10.3.1 Traditional Security Methods

Wireless security can be divided into two parts i.e., Authentication and Encryption. Authentication mechanisms can be used to identify a wireless client to an access point and vice-versa, while encryption mechanisms allow encrypting the original data. The encryption is a process to convert data in to the coded form so that it is not possible to intercept and decode data. For many years, MAC access control lists have been used for authentication, and 802.11 WEP has been used for encryption.

(a) **Authentication**
The access points are required to support MAC authentication of wireless clients, this show that traffic through only authorised MAC addresses will be allowed through the access point. WE use the Access Point to determine that particular MAC address is valid by checking it against either a RADIUS server external to the access point or against a database within the non-volatile storage of the access point. This is called the weak authentication mechanism we can say this because of it can be gathered, and because authentication is unilateral. There are two reasons for which it can be gathered. There exists software which is used to change the MAC address of some 802.11 cards. Second, authentication is tied to the hardware that a person is using and not to the identity of the user. Therefore, it could be possible to steal a legitimate user's PC and gain illegal access to a network. The concept of unilateral authentication shows that the access points are used to authenticate the user, but the user doesn't authenticate the access point. This technique would allow hacker to capture the unsuspecting user's credentials to gain access to other network resources.

(b) **Encryption**
The VPN software is used for encryption in WAP. Many wireless administrators elect to forgo WEP altogether and use VPN software for encryption. This option is preferable for public wireless hotspot providers that

are trying to attract as many users as possible by keeping client configuration as simple as possible. Hotspot customers use VPN software to connect to their company's network. The VPN option is also preferable to many enterprise administrators because VPN solutions offer the best commercially available encryption strength. VPN software uses advanced encryption mechanisms, such as AES, so that decryption is virtually impossible.

(c) **Wi-Fi Protected Access**

The Wi-Fi Protected Access is an intermediate solution that can be applied to existing WLAN hardware; the Wi-Fi user uses the Wi-Fi Protected Access (WPA). WPA is a standards-based, interoperable security enhancement. The WPA strongly increases the level of data protection and access control for existing and future wireless LAN systems. Designed to run on existing hardware as a software upgrade, Wi-Fi Protected Access is derived from, and will be forward compatible with the upcoming IEEE 802.11i standard. When the WAP is properly installed, then it will provide an assurance to the wireless LAN users that their data will remain protected and that only authorised network users can access the network. Wi-Fi Protected Access was created with several goals in mind: Available immediately.

(i) A strong, interoperable security replacement for WEP.

(ii) Applicable for both home and large enterprise users.

To meet these goals, 802.11 authentication and encryption were improved using parts of the 802.11i standard draft.

10.3.2 Terms

To understand 802.1x security mechanism, some terminology must be known.

Figure 10.1 shows the location role of each one of these terms in the authentication process.

(a) **Supplicant**

These are the end user systems seeking access to the network.

(b) **Authenticator**

These are the Access Points that control access to the network.

(c) **Authentication Server (RADIUS Server)**

Authenticates the end user, negotiates key material with the end user, and controls access to the network via the authenticator.

(d) **EAP**

The EAP stands for Extensible Authentication Protocol. This is a secure protocol used for negotiating other security protocols.

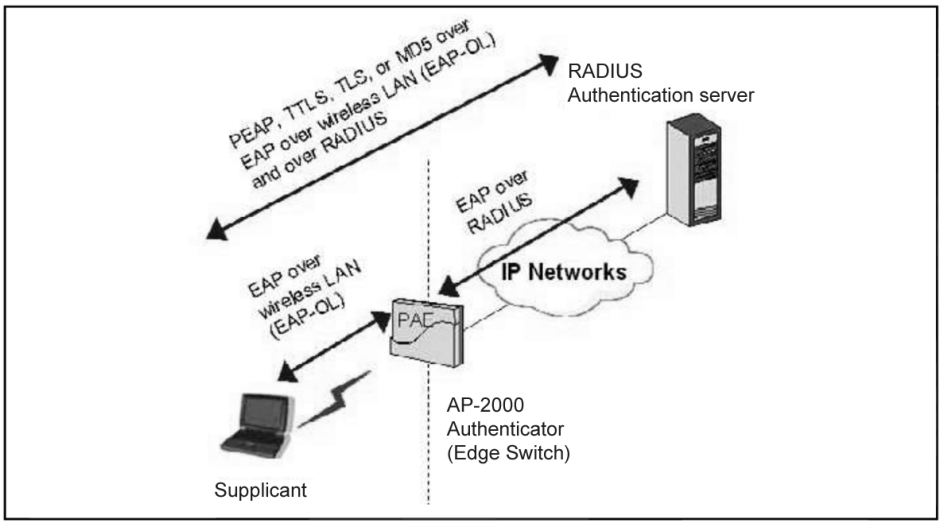


Figure 10.1: EAP and 802.1x

(e) **EAPOL**
EAPOL stands for EAP over LAN: That defines the version of EAP that is used over wireless networks.

(f) **PAE – Port Access Entity**
PAEs are similar to toggle switches. When the switch is open, no traffic is allowed to pass except for 802.1x traffic. After authentication is successful, the switch closes and user data is allowed to pass.

EXERCISE 10.1

1. Explain the security issues in Wireless Network.
2. What are the active transaction techniques?

10.4 SECURITY AND AD HOC NETWORKING TECHNOLOGIES

The ad hoc network may be defined as the network in which the communicating nodes do not necessarily rely on a fixed infrastructure. This sets new challenges for the necessary security architecture they apply. The ad hoc network is designed for specific environment and may have to operate with full availability even in difficult conditions. Security solutions applied in more traditional networks may not directly be suitable for protecting them.

Ad hoc networks can be formed, merged together or partitioned into separate networks on the fly, without necessarily relying on a fixed infrastructure to manage the operation. Nodes of ad hoc networks are often mobile, which also implicates that they apply wireless communication to maintain the connectivity. In such a case, the networks are called as mobile ad hoc networks (MANET). Mobility is not, however, a requirement for nodes in ad hoc networks. In ad hoc networks, there may exist static and wired nodes, which may make use of services offered by fixed infrastructure.

Ad hoc networks may be very different from each other, depending on the area of application: For instance in a computer science classroom an ad hoc network could be formed between students', PDAs and the workstation of the teacher. In another scenario, a group of soldiers is operating in a hostile environment, trying to keep their presence and mission totally unknown from the viewpoint of the enemy. The soldiers in the group work carry wearable communication devices that are able to eavesdrop the communication between enemy units. Besides, it enables to shut down hostile devices, divert the hostile traffic arbitrarily or impersonate themselves as the hostile parties. As can obviously be seen, these two scenarios of adhoc Networking is very different from each other in many ways.

The security aspects that are applicable on the other networking technologies are not fully applicable in ad hoc networks. While the basic security requirements such as confidentiality and authenticity remain, the ad hoc networking approach somewhat restricts the set of feasible security mechanisms to be used, as the level of security and on the other hand, performances are always somewhat related to each other. The performance of nodes in ad hoc networks is critical, since the amount of available power for excessive calculation and radio transmission are constrained. In addition, the available bandwidth and radio frequencies may be heavily restricted and may vary rapidly. Finally, as the amount of available memory and CPU power is typically small, the implementation of strong protection for ad hoc networks is non-trivial.

10.4.1 Physical Security

In ad hoc networks especially mobile nodes are typically significantly more susceptible to physical attacks than wired nodes in traditional networks. However, the significance of the physical security in the overall protection of the network is highly dependent on the ad hoc networking approach and the environment in which the nodes operate. For instance, in ad hoc networks that consist of independent nodes and work in a hostile battlefield, the physical security of single nodes may be severely threatened. Therefore, in such scenarios, the protection of nodes cannot rely on physical security. On the contrary, in the

classroom example scenario, the physical security of a node is an important issue to the owner of the node. This could perhaps be for privacy reasons, but the breaking of the physical security does not affect the security of the system as such.

10.4.2 Security of Network Operations

The security of ad hoc networks can be based on protection in the link or network layer. In some ad-hoc solutions, the link layer offers strong security services for protecting confidentiality and authenticity. In such case, all of the security requirements need not be addressed in the network or upper layers. For instance, in some wireless LANs, link layer encryption is applied. However in most cases, the security services are implemented in higher layers, for instance in network layer, since many ad hoc networks apply IP-based routing and recommend or suggest the use of IPSec.

Most MANET routing protocols seem to handle the rapid changes to the networking environment rather well. As the routing protocol is responsible for specifying and maintaining the necessary routing fabric for the nodes, the protocol must be protected from any attack against confidentiality, authenticity, integrity, non-repudiation and availability. If confidentiality of the routing information is th reat ene d, t he adv ersa ry coul d b e ab le to i den tify or lo cate no des by eavesdropping the routing traffic they send and forward. Without the protection of location, identity and communication, the users of the ad hoc network are very vulnerable to all kinds of attacks. On the other hand, if availability of the network is broken, the users may not be able to carry out their mission at all, as the communication links are broken or compromised.

Authenticity and integrity of routing information are often handled in parallel, if public key cryptosystems are in use. This is because digital signatures are applied for both confirming the origin of the data and its integrity. Without any integrity protection, the attacker is able to destroy messages, manipulate packet headers or even generate false traffic so that the actions cannot be distinguished from hardware or network failures. Authenticity of the routing data is essential so that nodes can confirm the source of new or changed routing information. If authenticity is not guaranteed, the adversary could perform impersonation attacks, divert traffic to arbitrary destinations or even scramble the routing fabric so that connectivity is severely broken in the ad hoc network. In worst case, the attacker can perform his actions and leave the network without being regarded as a malicious party. Non-repudiation is somewhat related to authenticity. Routing traffic must leave traces so that any party sending routing information cannot later deny of having propagated the data to other parts of the network.

10.4.3 Network Management Data has Similar Security Requirements as the Routing Traffic

The management information must be protected from disclosure, if it can contain vulnerable information such as status data that the nodes collect. The protection of management traffic against tampering and impersonation attacks is perhaps even more important. For example, if the status information the nodes send to the management system is not authenticated or protected against integrity attacks, a malicious node could capture the valid information and send invalid status data instead. This may lead to wrong assumptions about the condition of the nodes within the management system and lead to the use of invalid configuration data, as a reaction to the observed changes to statuses of nodes. Obviously, the impersonation attacks against the exchanged configuration information may have severe and unpredictable consequences – especially if the adversary can at the same time control the sending of status information from the nodes. Moreover, as in ad hoc networks, the manual configuration of nodes may be impossible, the configuration data may have to be exchanged dynamically and on-demand, thus making the management operations even more vulnerable to the discussed attacks. In the worst case the adversary can arbitrarily configure any node and thus, control the management system, which may interpret the observed inconsistencies as "natural" failures, not malicious actions generated by an active attacker.

EXERCISE 10.2

1. Explain the concept of Active Computing and Wireless Infrastructure.
2. What do you understand by the Security of Network operations?

10.4.4 Service Aspects

Ad hoc networks may apply either hierarchical or flat infrastructure both in logical and physical layers independently. As in some flat ad hoc networks the connectivity is maintained directly by the nodes themselves, the network cannot rely on any kind of centralised services. In such networks, the necessary services such as the routing of packets and key management have to be distributed so that all nodes have responsibility in providing the service. As there are no dedicated server nodes, any node may be able to provide the necessary service to another. Moreover, if a tolerable amount of nodes in the ad hoc network crash or leave the

network, this does not break the availability of the services. Finally, the protection of services against denial of service is in theory impossible. In ad hoc networks, redundancies in the communication channels can increase the possibility that each node can receive proper routing information. Such approaches do, however, produce more overhead both in computation resources and network traffic. The redundancies in the communication paths, however, may reduce the denial of service threat and allow the system to detect malicious nodes from performing malicious actions more easily than in service provisioning approaches that rely on single paths between the source and destination.

Availability is a central issue in ad hoc networks that must operate in dynamic and unpredictable conditions. The network nodes may be idle or even be shut down once for a while. Thus, the ad hoc network cannot make any assumptions about availability of specific nodes at any given time. For commercial applications using ad hoc networks, availability is often the most important issue from the viewpoint of the clients. The routing protocol must guarantee the robustness of the routing fabric so that the connectivity of the network is maintained even when threatened by rapid changes in topology or attackers. Similarly, in the higher layers, the services must be able to rely on that the lower layers maintain the packet-forwarding services at any time. Finally, many ad hoc networking protocols are applied in conditions where the topology must scale up and down efficiently, e.g. due to network partitions or merges. The scalability requirements also directly affect the scalability requirements targeted to various security services such as key management. In networks where the area of application restricts the possible size of the network, assumptions can be made about the scalability requirements of the security services as well.

10.4.5 Security of Key Management

As in any distributed system, in ad hoc networks, the security is based on the use of a proper key management system. As ad hoc networks significantly vary from each other in many respects, an environment-specific and efficient key management system is needed. To be able to protect nodes against eavesdropping (for example) by using encryption, the nodes must have made a mutual agreement on a shared secret or exchanged public keys. For very rapidly changing ad hoc networks, the exchange of encryption keys may have to be addressed on-demand. Thus, without assumptions about a priori negotiated secrets. In less dynamic environments like in the classroom, the keys may be mutually agreed proactively or even configured manually (if encryption is even needed). If public-key cryptography is applied, the whole protection mechanism relies on the security of the private key. Consequently, as the physical security of nodes may be poor, private keys have to be stored in the nodes confidentially, for instance encrypted with a system key. For dynamic ad hoc networks, this is not a wanted feature and

thus, the security of the private key must be guaranteed with proper hardware protection (smart cards) or by distributing the key in parts to several nodes. Hardware protection is, however, never alone an adequate solution for preventing attacks as such. In ad hoc networks, a centralised approach in key management may not be an available option, as there may not exist any centralised resources. Moreover, centralised approaches are vulnerable as single point of failures. The mechanical replication of the private keys or other information is an inadequate protection approach. For example, the private keys of the nodes simply have then a multiple possibility to be compromised. Thus, for any cryptosystem in use, a distributed approach in key management is needed.

10.4.6 Access Control

The access control is an applicable concept also within ad hoc networking, as there usually exist a need for controlling the access to the network and to the services it provides. Moreover, as the networking approach may allow or require the forming of groups in for instance network layer, several access control mechanisms working in parallel may be needed. In the network layer, the routing protocol must guarantee that no authorised nodes are allowed to join the network or a packet forwarding group such as the clusters in the hierarchical routing approach. For example in the battlefield example of the introduction the routing protocol, the ad hoc network applies must control. This is to ensure that no hostile node can join and leave the group undetectable from the viewpoint of the other nodes in the group. In application level, the access control mechanism must guarantee that unauthorised parties cannot have accesses to services, for instance, the vital key management service.

SELF-CHECK 10.1

Write short notes on Routing area, Roaming number and TMSI.

10.5 SECURITY THREATS IN AD HOC NETWORKS

Security Threats, against ad hoc networks can be divided into two groups ? Passive attacks typically involve only eavesdropping of data. Active attacks involve actions performed by adversaries, for instance, the replication, modification and deletion of exchanged data. External attacks are typically active attacks that are targeted, e.g., to cause congestion, propagate incorrect routing

information, prevent services from working properly or shut down them completely. External attacks can typically be prevented by using standard security mechanisms such as firewalls, encryption and so on. Internal attacks are typically more severe attacks, since malicious insider nodes already belong to the network as an authorised party and are thus, protected with the security mechanisms the network and its services offer. Thus, such malicious insiders who may even operate in a group may use the standard security means to actually protect their attacks. These kind of malicious parties are called compromised nodes, as their actions compromise the security of the whole ad hoc network.

10.5.1 Denial of Service

The denial of service threat either produced by an unintentional failure or malicious action forms a severe security risk in any distributed system. The consequences of such attacks, however, depend on the area of application of the ad hoc network. In the classroom example any of the nodes, either the teacher's centralised device or the students' handheld gadgets, can crash or be shut down without completely destroying anything ? the class can continue their work normally by using other tools. Distributed denial of service attack is a more severe threat: if the attackers have enough computing power and bandwidth to operate with, smaller ad hoc networks can be crashed or congested rather easily. The denial of service attack has many forms – the classical way is to flood any centralised resource so that it no longer operates correctly or crashes, but in ad hoc networks, this may not be an applicable approach due to the distribution of responsibility. There are however more serious threats to ad hoc networks: As discussed in, e.g., compromised nodes may be able to reconfigure the routing protocol or any part of it so that they send routing information very frequently, thus causing congestion or very rarely, thus preventing nodes to gain new information about the changed topology of the network. In the worst case, the adversary is able to change routing protocol to operate arbitrarily or perhaps even in the (invalid) way the attacker wants. If the compromised nodes and the changes to the routing protocol are not detected, the consequences are severe, as from the viewpoint of the nodes the network may seem to operate normally. This kind of invalid operation of the network initiated by malicious nodes is called a byzantine failure.

10.5.2 Impersonation

Impersonation attacks form a serious security risk in all levels of ad hoc networking. If proper authentication of parties is not supported, compromised nodes may in network layer be able to, e.g., join the network undetectably or

send false routing information masqueraded as some other, trusted node. Within network management, the attacker could gain access to the configuration system as a super user. In service level, a malicious party could have its public key certified even without proper credentials. Thus, impersonation attacks concern all critical operations in ad hoc networks. In the classroom example, however, the impersonation attack is not probable or even feasible. If a malicious student impersonates himself as the teacher's device, he may be able to access or destroy data that is stored in students' or teacher's devices or exchanged between them. The benefit of the attack is small – it will most likely be noticed very quickly and the information he can manipulate or have access to is not that crucial to make the attack worthwhile. In the other example the implications of successful impersonation is much more severe (again): a hostile node controlled by the enemy may be able to join the ad hoc network undetectably and cause permanent damage to other nodes or services. A malicious party may be able to masquerade itself as any of the friendly nodes and give false orders or status information to other nodes. Impersonation threats are mitigated by applying strong authentication mechanisms in contexts where a party has to be able to trust the origin of data it has received or stored. Most often this means in every layer, the application of digital signature or keyed fingerprints over routing messages, configuration or status information or exchanged payload data of the services in use. Digital signatures implemented with public-key cryptography are as such a problematic issue within ad hoc networks, as they require an efficient and secure key management service and relatively much computation power. Thus in many cases, lighter solutions like the use of keyed hash functions or a priori negotiated and certified keys and session identifiers are needed. They do not, however, remove the demand for secure key management or proper confidentiality protection mechanisms.

10.6 LOCATION INFORMATION, SECURITY AND PRIVACY

Location information – In this topic, we will discus about the Mobility Management in GSM and UMTS networks including those functions related to GPRS.

One of the important issues related to GSM and UMTS network is the Mobility Management. The aims of mobility management are as follows:

(a) To track where the subscribers are so that calls can be sent to them.

(b) To allow contact to the subscribers to be established.

(c) To record what the subscriber has paid for and provide those, and only those services.

10.6.1 Location Update

Whenever the mobile user moves from one area to another, the location update procedure allow informing about network. The responsibilities to inform about the current location lies with the mobile which should detect the Location Area Code (see below) of any new cell that it moves in to the network. If this is different from the mobile's previous cell, then the mobile should contact the network and perform a location update. The mobile sends a message (location update request) to the network containing the mobiles previous location and TMSI.

(a) Location Area (Location Area Code)
In order to optimise the signaling, a set of cells are grouped together and are called a location area. A unique number which is called a location area code is assigned to each of the type of groups within the current mobile network. The location area code is broadcast by each base station known as a BTS in GSM or a Node-B in UMTS at regular intervals.

(b) Routing Area
Routing area is generally used by mobiles which are GPRS attached. A routing area is a subdivision of a location area. The bursty nature of packet traffic means that more paging messages are expected per mobile and so it is worth knowing the location of the mobile more accurately than it would be with circuit switched traffic. A change from routing area to routing area (called a Routing Area Update) is done in an almost identical way to a change from location area to location area.

(c) Roaming Number Allocation
When a call is routed to a mobile network subscriber, that call is initially routed toward the subscriber's home network using the subscriber's own number, just like a normal phone call. When the call reaches the subscriber's home network, and arrives in the Gateway MSC, the issue becomes different. It is perfectly possible for the subscriber to be roaming in a different country. For that reason, a number belonging to the Visited MSC where the subscriber is currently located is allocated on a temporary basis (maximum for a few seconds).

(d) TMSI
The TMSI is defined as Temporary Mobile Subscriber Identity. This is the identity most commonly sent between the mobile and the network. The TMSI is a randomly allocated number which is given to the mobile the first time that it is switched on. The TMSI is local to a location area so must be updated every time; the mobile does a location update procedure.

(e) Security and Privacy
Security and privacy are two different things. But these are closely related technologies. There are however, important differences that need to be

understood in order to design new systems that address both. Privacy is about informational self-determination – the ability to decide what information about you goes where. Security provides the ability to be confident that those decisions are respected. For example, suppose we talk about GSM then privacy issue includes voice privacy – can someone listen to my call? There is a privacy goal, which is to allow me to say no, and on the other hand, there is a security issue that uses a security technology, encryption, that allows me to enforce it. In this example, the goals of security and privacy are the same. But there are other times when they may be orthogonal, and there are also times when they are in conflict.

10.7 SECURITY – THE UNSOLVED PROBLEM FOR MOBILE AGENTS

A mobile agent is a computer program that migrates from one computer to another autonomously and continues its execution on the destination computer.

The mobile agent's security has two aspects, viz., one of protecting the host machine and the other of protecting the mobile agents. There are different types of threats caused by mobile agents. These threats are harmful for user in such a way that thread distract the files in the internal/external data storage, they are responsible for destruction of hardware, destruction of current execution environment.

Mobile agent and host are vulnerable to a number of threats:

Agent to Platform: The agent's to platform category represents the set of threats in which the agents' exploit security weakness of an agent platform or launch attacks against an agent's platform. The threats include masquerading, denial of services and unauthorised access.

(a) Masquerading
The masquerading agents may act as unauthorised agents to gain access to services and recourses. We can say that when an unauthorised agent poses the identity of another agent, then it is said to be masquerading.

(b) Denial of Services
Thee mobile agents can launch denial of service attacks by consuming or corrupting the agent platform's computing recourses. The denial of services attacks can be launched to exploit system vulnerabilities. Program testing, configuration management and design review reduces the risk of introducing malicious code into an organisations computer system.

(c) **Unauthorised Access**
The access control mechanism is used to prevent unauthorised users or processes from accessing services and recourses for which they have not given permission and privileges. The access control mechanism requires the platform or the agent to first authenticate a mobile agents identity before it is instantiated on the platform.

10.7.1 Security Services

(a) **Confidentiality**
The aspect of confidentiality implies that authorised users can only read and decode the data and the sensitive data must be secure.

(b) **Authentication**
An agent server must authenticate itself to the agent and an agent must authenticate it to the host.

(c) **Integrity**
The integrity provides assurance that traffic is not altered during the transmission.

(d) **Authorisation**
Host enforces strict access control to its recourses.

(e) **Auditing**
Auditing keeps track of the system and if an agent misbehaves, it will be logged.

(f) **Access control**
Access control implies that only authorised users will be able to access recourses.

EXERCISE 10.3

1. Explain Security Issues as regards Mobile Agents.
2. How does mobile security works?

10.8 DISTINGUISH BETWEEN PRIVACY AND SECURITY

There is confusion between the term Privacy and Security. Some people equate these things but these are two different aspects that are closely related to each other. In other words, we can say that security is a necessary tool to build privacy. Privacy is a basic human right and is recognised as a necessity by the European Union (Privacy Directive) and the OECD (Principles of Fair Information Practice). In the M-commerce environment, privacy is an important issue to gain widespread adoption by consumers around the world.

Suppose we talk about GSM, then privacy issue includes voice privacy – can someone listen to my call? There is a privacy goal, which is to allow me to say no, and on the other hand, there is a security issue that uses a security technology, encryption, that allows me to enforce it. In this example, the goals of security and privacy are the same.

There is a security goal of authenticating a handset. In some instances, this may be done by RF fingerprinting, which is not a privacy issue. We securely authenticate that the handset is the one that is linked to an account, thus, ensuring that the right person is billed. Here, security and privacy are orthogonal.

SSL is often confused with privacy. SSL offers "privacy" against eavesdroppers, but this is better called confidentiality. The well-publicised break-ins at CD Now and other retailers using SSL show that privacy requires more than SSL. It requires minimising the amount of information that is transmitted and stored.

In the area of location information, allowing disclosure of location information out on an continuous basis creates a number of privacy issues, but occasionally, when calling for emergency services, it is useful to reveal. It is important to design these system so that the phone's owner is in control and happy, and to ensure that the security measures in place support the owner's decision effectively.

Caller presentation is an example of a place where security and privacy can conflict. I may want privacy, in not letting anyone else see my number; while the person called may want the security of thinking they know who is calling. In this situation, in most countries, a balance has been struck in favour of informational self- determination, allowing the caller to choose if caller information is available.

SELF-CHECK 10.2

Are the concepts of Security and Privacy different? Justify your answer by giving suitable reason.

10.9 MODELLING SECURITY WITH UML

The Unified Modelling Language (UML) is a very frequently used language. UML specification still lacks formal, explicit support for access control. Security has become a very important issue in the development of software applications. The access control policies, along with other security requirements, must be an integral part of the software development process, to ensure that the proper level of security in an application is attained. Access control is defined as Limiting access to information system resources only to authorised users, programs, processes or other systems.

The software development process consists of a systematic series of tasks to create a software system: requirements capture, analysis, design, coding and testing. The scope of these researches at the design stage of the process, concentrate on modelling of access control. To analyse the issues of modelling security, one must understand the most common security schemes that are used to conceptualise access control policies:

(a) Mandatory Access Control (MAV)

(b) Discretionary Access Control (DAC)

(c) Role-Based Access Control (RBAC)

Mandatory Access Control (MAC) is well-suited to applications where the protection of information is paramount (i.e., releasing such information would have dire national security). In MAC, each object is labelled with a mastication level (e.g., top secret, secret, confidential, and unclassified) that represents the sensitivity of their information. Each subject has a clearance level. Security is enforced by ensuring that a subject's clearance level always dominates an object's classification level.

DAC targets applications that are collaborative and dynamic. In DAC, permissions are defined between subjects and objects, but a subject can be granted the permission to delegate a subset of its own permissions to another user.

RBAC groups' permissions into independent units called roles, which represent the role that a user assumes in an organisation. Roles, rather than permissions, are assigned to users (subjects) when they initiate an interactive session with the software system. The set of privileges granted to a user is defined by the set of permissions assigned to its corresponding role.

Security schemes such as MAC, DAC, and RBAC, specify the basic semantics for access control, but they do not provide a visual language to represent this information. UML, the dominant software and system modelling approach, while an obvious candidate to provide security, lacks explicit support for access control. Furthermore, security is a crosscutting concern that pervades the entire application, which makes it difficult for software practitioners to adequately integrate security into software. As a result, when designers wish to incorporate security concerns into an application using UML, the resulting model is very likely to have the security tangled and scattered throughout the entire design. Designers can select specific features and combine them (according to rules and limits) in order to create a security aspect-modelling infrastructure that is suitable for their requirements. Since security features comprise a small subset of the information of an access control schema, they should be easier to understand by designers. Furthermore, they assist in tracking security requirements from models to code, reducing scattering of access-control definitions across the application, and providing a collective view of the security policy.

10.9.1 Modelling Security Aspects

To model a security policy, designers must identify three key components: subjects, objects, and permissions. Subjects are the entities that require access to the system. The system contains a set of objects that are the entities that require protection against subjects. For the proposed approach, class methods (operations) are the objects in the system that requires protection. Permissions determine which operations can access each subject in the system.

SUMMARY

- Active Computing is a small business that provides research and development in information and software systems for the Department of Defence and other government customers. With more than 20 years of experience in innovative solutions for command and control, simulation, and analysis, the principal and available subcontractors are uniquely qualified to solve the toughest decision-making problems faced by the military today.

- With recent performance advancements in computer and wireless communications technologies, advanced mobile wireless computing is expected to see increasingly widespread use and application, much of which will involve the use of the Internet Protocol (IP) suite. The vision of mobile ad hoc networking is to support robust and efficient operation in mobile wireless networks by incorporating routing functionality into mobile nodes.

KEY TERMS

VCC Visitor Country Code

VLF Visitor Low Frequency

WAE Wireless Application Environment

WAP Wireless Application Protocol

SELF-TEST

1. What are the active transaction techniques?
2. How does Mobile security work?
3. What is ad hoc networking technologies?
4. Write short notes on: routing area, roaming numbers, TMSI.
5. Differentiate between privacy and security.

Topic 11 ▶ Mobile Development Process

LEARNING OUTCOMES

By the end of this topic, you should be able to:

1. Identify the dimensions of mobility;
2. Outline the wisdom methodology to mobile development;
3. Identify the UML based development cycle for mobile application; and
4. Recognise the architecture pattern for mobile application.

▶ INTRODUCTION

In the previous topic, we have studied Security in Wireless Networks, Security and Ad Hoc Networking Technologies, Location Information, Security, and Privacy, Security. The Unsolved Problem for Mobile Agents, Distinguishing Privacy and Security and Modeling Security with UML. Now, in this topic, we would focus on the mobile development process. An application developed for the mobile should not be thought as a replacement for the web. A mobile application needs to be designed keeping the mobile user in mind, that has very little time and desires to accomplish what ever he/she wants with a minimum number of clicks.

11.1 DIMENSIONS OF MOBILITY

(a) Spatial Dimension

The definition of mobility says that the no one can understand mobility without referring the location because mobility is the movement of an entity from one location to another. Mobility provides the interaction between the people who are geographically independent. However, the spatial mobility does not only describe mobility of people, it also includes the mobility of objects, symbols and space itself. Objects that include artificially designed objects exhibit mobility because they are carried by mobile humans to mediate certain activities which satisfy particular needs, intensions, motives and needs. These objects possess symbols that are manipulated during their use; hence object mobility implies the mobility of symbols.

The spatial mobility of humans, objects and services are interrelated and intertwined. For example, the individual who carries a PDA, laptop computers or a mobile phone certainly does so with an intension to use the services provided by the artifact to perform an activity aimed at satisfying a motive. During this activity he or she captures, processes or transmits information using the artifacts symbol properties and here, the mobility of individuals the objects and symbols occurs simultaneously. If he or she interacts with others using the device, then spatial mobility of particular information service occurs. Therefore, the interaction and computing information services provided by the portable ICTs. Together with the fundamental mobility of humans and objects, achieve spatial mobility in terms of human independence of geographical constraints.

The spatial and temporal dimensions of mobility extend computing and allow, in principle, anytime and anywhere access to information, communication, and services.

(b) Temporal Dimension

The concept of spatial mobility is meaningful when the phrase of information service follows it. The concept of temporal mobility is meaningful when it is linked to the information service supporting a particular human activity. Technology development has always acted as an automated human operation. Automation is a new time saving technique that is being used for doing work through technology. Zuboff gives an idea of automation that was closely related to the static ICTs yet it incorporated temporal efficiency. The temporal efficiency in static ICT is only achieved when the user of the technology is attached to the technology and hence also static. For example, PDAs and Laptop PCs can automate human

operations as they are carried around; in short enhanced temporal efficiency is achievable with portable ICTs. Portable ICTs are time saving and also afford mobile interaction. Interaction through fixed telephones and exchange of paper based mails are asynchronous and characterized by time delays. Also the fixed telephone interaction confines the users to a specific location; otherwise he or she has to set up the voicemail to allow callers to save their message for a later reading. Reading of paper based mails and voice mail messages constitute asynchronous interaction. The Portable ICTs ensure a synchronous interaction because user can interact with others anytime and anywhere. Spatial and temporal mobility have together aroused interest incontemporary Computer Supported Cooperative Work (CSCW) research. Before interaction with portable ICTs, CSCW researchers concern themselves with communication, collaboration and coordination issues of organizational actors and actions and support for these attributes through static computing.

(c) Contextual Dimension

The third dimension of mobility is the contextual dimension that provides the extended perspectives of mobility.

11.1.1 Applying the Wisdom Methodology to Mobile Development

WISDOM methodology is an important concept related to the 3rd generation mobile computing environment. WISDOM is a 3rd Generation (3G mobile) solution to the needs of deaf people for interaction and for visual information. WISDOM exploits the pervasive nature of the third generation mobile network to offer to deaf people, participation, independence and the means to make their own contribution. Deaf needs for visual interaction at a distance, challenge the telecommunications infrastructure in terms of quality of service and content and the specific needs of groups such as the elderly require new interfaces. By trials with Bluetooth enabled systems for video capture, compression, text entry and sign recognition, WISDOM provides solutions for person-to-person live communication and information retrieval in sign language. By involving users and purchasers, network operators and manufacturers, WISDOM ensures a pathway to exploitation and real benefits for the end users. WISDOM advances services for the whole community. DESCRIPTION OF WORK Visual Telecommunications is possible in the third generation mobile networks. Deaf people are the ideal users of this development for their sign language communication. WISDOM provides service in sign language in a mobile environment. To achieve this, partners must:

(a) Specify, implement and validate a mobile video terminal which allows communication in sign language, and text.

(b) Extend the system of sign language recognition developed by Aachen Technical University to create an interface with the sign language server.

(c) Set up a sign language server which provides information to deaf users 24 hours, 7 days a week.

(d) Establish network quality of service for mobile video communication.

(e) Integrate recently approved text conversation standards and feed back experience to standardization bodies.

(f) Set up and run sign language remote interpretation and voice call relay services.

(g) Carry out lab and field trials

(h) By consultation, ensure a deaf user's and purchaser's viewpoint for exploitation.

Milestones

J2ME is the java platform for mobile phones that can be used to develop small applications and games for mobile phones. In this article we will see how to develop a small java game using J2ME; we will be going through the minimum things that you need to know for developing a game in J2ME.

In this article we will walk through the process of developing a simple java game for mobile phones.

We shall be covering the following topics:

(a) The tools you need

(b) Things you need to know

(c) About the game

(d) Painting

(e) Handling key events

(f) Adapting to touch screen phones

(g) Further Improvements.

Developing games and applications in J2ME is slightly different from developing games and applications in J2SE, the main difference is because J2ME is designed to run on mobile phones with very less memory and processing capability while

J2SE is designed to run on workstations with gigabytes of memory and disk space.

The APIs and libraries that come with J2ME are very limited and more specific to the functionality of mobile phones.

In this section we shall see the tools that you would need to develop a game/ application in J2ME.

Net beans IDE (Recommended)

Net beans is a JAVA IDE (Integrated Development Environment) developed by SUN in which you can develop and debug java applications. Net beans comes with a mobility pack add-on which allows you to develop test and debug mobile applications using J2ME.

To develop a J2ME game, you need to be familiar with below classes:

(a) Midlet

(b) Game Canvas

(c) Graphics

(d) Display

SELF-CHECK 11.1

Do the issues of mobility affect the dimensions of mobility?

11.1.2 UML based Development Cycle for Mobile Application

The Unified Modeling Language (UML) provides various features which are attractive for the specification and design of complex electronic systems. The UML can be defined and act as a common modeling language for both hardware and software, it enables the easy visualization of complex systems, and it is a widely recognized standard with strong penetration into a number of engineering disciplines. But, there has not been strong EDA tool support for UML. Axilica has developed Falcon ML that is being used as tool which enables the behavioral synthesis of UML models. The Falcon ML is a UML model with C++ actions. Falcon ML can generate Verilog, VHDL and System C realizations

of the model. Here we describe the use of UML for the design and implementation of an H.264 video decoder core through the Falcon ML design flow to an ASIC technology.

UML is much frequently used in the software engineering and systems engineering fields, however it is not yet widely used in hardware. A UML-based hardware design flow is more beneficial as compared to the conventional design techniques. UML enables description of both hardware and software at a high level of abstraction, while also provides representations for lower-level features where needed. UML can readily describe concurrency and interactions between threads. In addition, UML profiles and stereotypes allow the definition of a UML subset appropriate for behavioral synthesis.

There has long been interest in the application of UML to hardware synthesis problems. Indeed, some of the early work on state charts (which later became UML state machines) was aimed at hardware synthesis. There is also considerable interest in the use of UML class diagrams for the description of hardware. In particular, there is interest in the use of class diagrams together with ESL languages, primarily System. In this context UML ports and structured classifiers can be extended to provide a direct representation for Systems, while state machine diagrams with Systems actions can be used to describe object behavior. The transformation to a hardware model is quite direct, and can be accomplished using a template-based system. This interest has led to the development of a UML profile for System on a Chip (SoC) design, which is intended as a standard UML- based representation for ESL designs. UML can also be used in refinement flows, as in the 2 Robert Thomson1, Scott Moyers1, David Mulvaney2 and Vassilios Chouliaras2 where an abstract model is refined into an architectural model which incorporates both structure and timing information. In contrast to the above techniques, Axilica Falcon ML incorporates a behavioral synthesis engine which directly converts UML into synthesizable hardware, reducing the need for specialized UML profiles and annotations, and aiding the reuse and retarget ability of the UML model. Falcon ML enables the realization of UML-based designs via a conventional hardware synthesis back-end flow, resulting in either FPGA or ASIC hardware.

Falcon ML Design Flow

The Falcon ML design flow is illustrated in the figure 11.1. As shown in the figure, input to FalconML is an XMI file containing the UML model and C++ actions? In the next step Falcon ML performs a behavioural synthesis, and producing Verilog/ VHDL which can then be entered into a conventional simulator or synthesis flow. FalconML also produces a transaction-level SystemC implementation of the design, which can be used for rapid functional simulation.

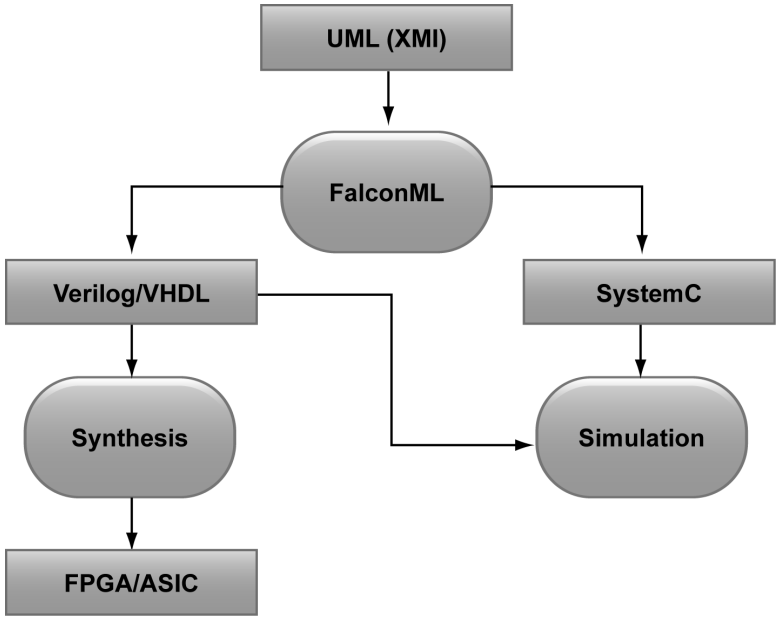


Figure 11.1: FalconML design flow

The input to Falcon ML conforms to a UML profile which includes several structural and behavioral diagrams, specifically: class diagrams, object/ deployment diagrams, state machine diagrams, and activity diagrams. The models make use of a C++ subset as the action language. The action language is used to define methods, actions and activities.

The class diagrams can define classes, class features and relationships between classes. Classes can contain attributes and operations. Attributes with a non-zero multiplicity are typically mapped to a memory in hardware. Multiple threaded control can be denoted through the use of active classes. Operations can be marked as "signal", in which case the operation can be executed in parallel with the calling code. Class diagrams can include inter-class relationships such as association, composition or specialization. The elements defined in a class diagram can be explicitly instantiated in an object or deployment diagram.

State machine and activity diagrams provide a visual representation for object behavior. They can describe parallel execution through the use of composite states or the fork/join constructs. Transitions can be triggered by call events or change events, and the transitions and states can include C++ actions.

The FalconML UML profile defines a number of stereotypes which control the hardware realization of the design. In particular, stereotypes are used in the specification of signal-level I/O ports

The H.264 Decoder Model

The H.264 decoder model was based upon the JM H.264/AVC reference software. The JM reference code was stripped down to the baseline profile and that C code was then converted to C++ and incorporated into the UML model.

The initial version of the decoder is single-threaded. The "Caller" class defines a signal-based interface to the outside world. The "Decoder" class contains the top-level H.264 algorithm. The compressed input data stream is stored in the "Bit Stream Buffer" class, while the decompressed frames are stored in the "Image" class. The other classes define parts of the H.264 algorithm. The "Caller" class defines an I/O interface which allows external hardware to write 32-bit data and commands into a virtual address space a selection of outputs are used as flags, to inform external hardware of the current Parallel Version.

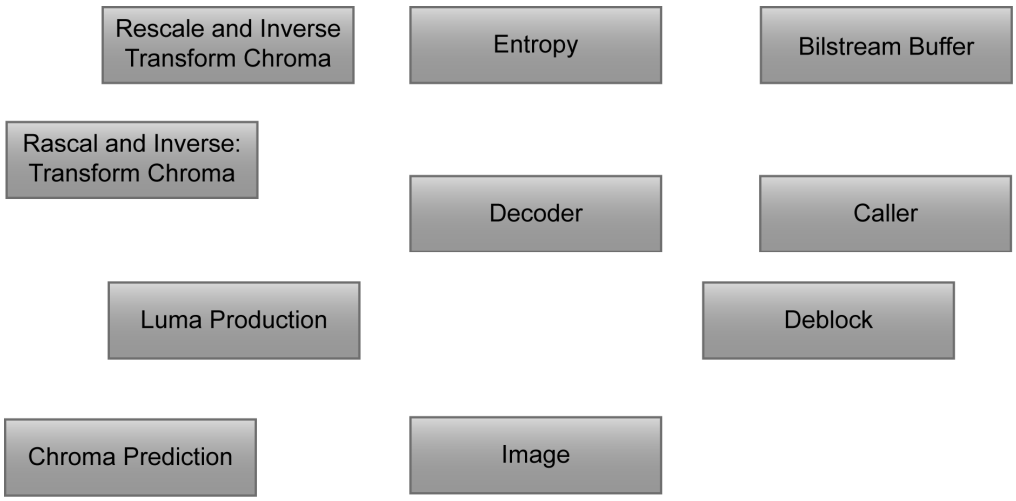


Figure 11.2: Shows the class diagram for the H.264 model

An improved parallel version of the H.264 decoder was also developed. The parallel version was modified to incorporate high-level pipelining and parallelisation of the decoding process, as follows:

Load Bitstream	Decode Bitstream	Deblock	Output Frame				
		Load Bitstream	Decode Bitstream	Deblock	Output Frame		
				Load Bitstream	Decode Bitstream	Deblock	Output Frame

These optimizations were achieved by implementing state machines in multiple classes and by using the "signal" stereotype to mark operations as asynchronously executed. Call events were used for communication between the two state machines. The machines are shown in Figure 11.3 amd 11.4.

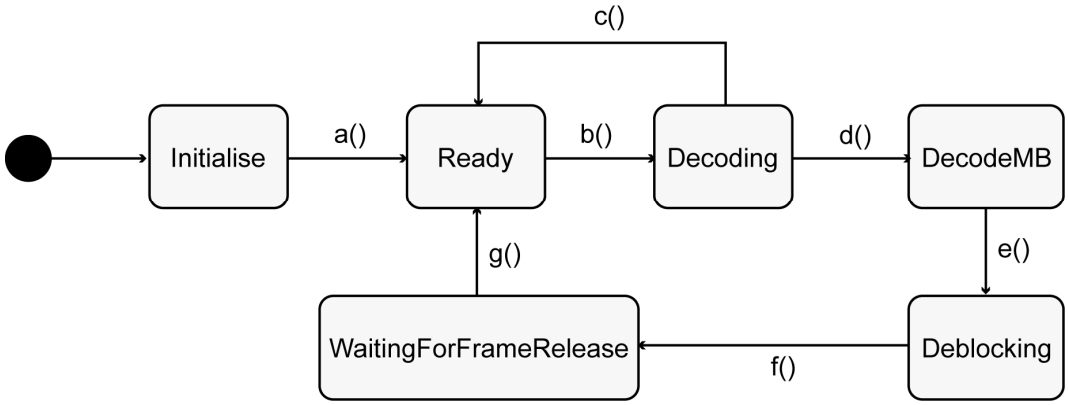


Figure 11.3: Decoder State Machine

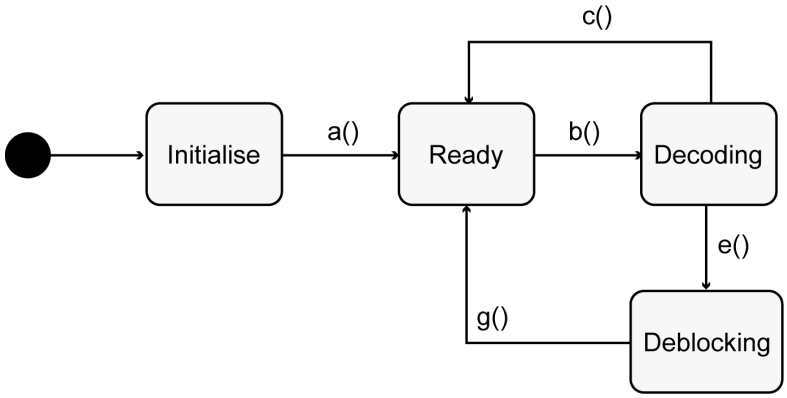


Figure 11.4: Caller state machine

EXERCISE 11.1

1. Explain the wisdom methodology to mobile development.
2. Explain the UML based development cycle for mobile application.

11.2 ARCHITECTURE, DESIGN AND TECHNOLOGY SELECTION FOR MOBILE APPLICATIONS

11.2.1 Practical Concerns with Architecture

Traditional network architecture is static by nature. Network designers know in advance which computer hardware and software will participate in a specific computing and the network infrastructure is built in accordance with this concept. For example, in the Client/Server network architecture, specific server resources are designated to be served to clients on request, while other software units are indigenous on the clients. In dynamic network architecture we do not decide in advance the specific hardware and software that will participate in the solution. Because of the multiplicity of software and hardware available on the network that could participate in the solution, it is advantageous to defer the decision until the software or hardware is actually required. The idea is to have the solution itself seek and recover on the network the components and resources it requires. Should the selected components and resources degrade or fail during execution, the solution can replace them or/and continue to operate.

Mobile Computing Architecture

A working space has to be organized, if we want to achieve a specific mission quickly. A Java Space follows this idea. In the Replicated-Worker pattern on Java Spaces, also known as the Master-Worker pattern, a master process creates a collection of tasks that need to be run. As shown in Figure 11.5. Workers take tasks from the collection and run them, then hand over the computed result to the master. A space is a natural channel for passing messages between master and workers, due to the decoupled programming style it encourages. Typically, there are many workers, and they are identical; hence the term replicated. This pattern neatly provides load balancing, whereby each worker contributes whatever resources it can afford. The worker on a faster machine will execute more tasks than the worker on a slower or otherwise heavily loaded machine; and as long as the granularity of the tasks is sufficiently fine, no worker will hold up the computation.

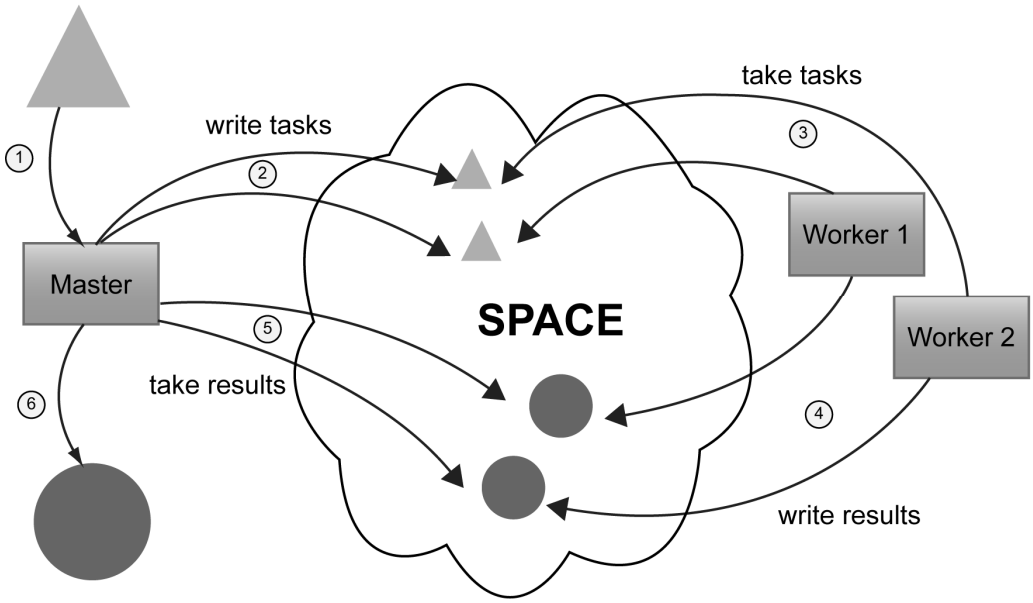


Figure 11.5: Master-washer pattern

The open source Java framework Compute Farm grew out of an implementation in Java Spaces of this pattern. But our approach of this pattern differs from Compute Farm because the tasks do not contain the required computation. The computation is on a mobile agent apart from the space. In the space, there are the parts of the file and the scheduler, which is the main component of the computation. Based on the Replicated-Worker pattern, our mobile computing architecture goes further than the Compute Farm framework. The adaptability and the replay of a computation case is the heart of our architecture. In fact, our mobile computing architecture consists of four main components. All the components are on various nodes of a network. The link between all the components is the Space. Each computer containing a component must reach the computer containing the Space. The Computing Master initializes the computation case and finishes it. It creates the parts of the source file and collects the files generated by the Computing Worker. The Computing Agent is the mobile agent that contains the code of the computation case. An essential part of the computation is the Scheduler, which schedules tasks for Computing Worker to facilitate multitasking. When a component takes the Scheduler, the Scheduler is not available anymore. The component must rewrite it to make it available again. That will avoid conflicts and two executions of the same task. The strategy of choices of tasks takes part in the semantic of the scheduler. The Scheduler which we will use from now on attributes the tasks to the Computing Worker in the order of task arrival. Another implementation of the Scheduler is completely possible: priorities into the tasks for instance.

Finally, the Computing Worker take tasks from the Scheduler and with the Computing Agent they can work with the parts of the file on the Space. Moreover, the result of a same task is identical whatever the Computing Worker.

11.2.2 Architectural Pattern for Mobile Application

The increasing deployment of mobile technologies across industry sectors is creating fertile ground for organizations to exploit new revenue streams generated from applications that exploit the mobile ecosystem. M-Commerce has been recognized as a key driving force of next generation computing and industry analysts such as IDC have predicted revenue growth arising from m-commerce to far exceed US$27 billion by the end of the decade. The phenomenon of mobility has driven an evolution and revolution of technologies and new ways of working, exploiting the practice of anytime, anyplace, anyhow computing whilst gaining critical mass as a research discipline and paradigm – the study of which has provided a grounding for conceptual and theoretical perspectives that allow framing and discussion of mobility.

Mobile commerce has created a wide range of business opportunities – the spectrum of which includes the transformation of mobile network operators to go beyond the provision of core voice services to the delivery of value added services and service aggregation that provide comprehensive consumer packages, the emergence of business models to develop and deploy mobile applications (such as the Apple 'App' Store, the O2 Litmus programmed, and the Sony Ericsson Developer World – most of which are based on shared revenue distribution through sales through the respective hosting channels. Furthermore, open programmers and platforms such as Android (Google) are likely to contribute to the enlargement of the developer community that seeds the growth of mobile applications. With the supply chain of mobile computing and commerce, a range of opportunities exist for contributing parties to collaboration to provide value- added solutions based on new relationship types, rules and ecosystems further enhancing the composite capability and reach of service providers into new market areas. Beyond service providers, the ease of access, penetration, and diffusion of mobile technologies has enabled individual sectors and organizations therein to apply the concepts and practice of mobility to create innovative domain-specific m-commerce applications that leverage the specific nuances associated with those domains resulting in value-added solutions for end-users and new revenue-generating opportunities for business. Examples of these include mobile location-based tourism, travel and navigation systems, m-ticketing and booking applications. Beyond the core commercial imperative of increased revenue generation, the application of mobile computing and principles of m-commerce also underpin the development of private and public sector mobile applications that aim to reduce operating costs, enhance efficiency

and provide better platforms for engaging the end-user population. Basole presents an Enterprise Mobility Continuum that frames mobile solutions from point-specific solutions to those that diffuses across entire organizations to create stakeholder value. Across and throughout the complex m-commerce ecosystem the question arises regarding the construction of m-commerce solutions and how these are best approached. The increasing attention to business underpinned by mobility, mobile services, mobile applications, and technologies has become a major driver for the development of m-commerce systems. This growing trend has become a focus for a significant number of organizations. This paper proposes that in developing m-Commerce Systems organizations need to establish enterprise architecture for m-commerce. The rationale for this is rooted in the need to develop a holistic and integrated view of strategic direction relating to m-commerce which will enable a coordinated and controlled approach that reduces complexity and yields effective systems based on the structured integration of services, practices and technology resources. In doing so, the potentially complex universe of discourse associated with mobility is harnessed to produce an organizational asset to drive the development of m-commerce. The next section proposes a framework to establish such enterprise architecture for mobile commerce. Firstly, an enterprise architecture framework for mobile commerce is presented followed by a brief view of associated issues concerned with method.

Enterprise Architecture (EA) Framework for Mobile Commerce

An EA (Enterprise Architecture) framework provides the basis or template for the creation and establishment of enterprise architecture. Zachman is credited with developing the discipline of enterprise architectures as a concern for both researchers and practitioners. An enterprise architecture framework is essentially a meta-construct used to define the scope of the associated architecture and how the areas of the architecture relate to each other. Architecture can be considered analogous to a blueprint or plan of a building structure, where different perspectives may exist and each perspective contains structures that demonstrate inter-relationships based upon some predefined constraint and yield a solid foundation and approach upon which the building is constructed.

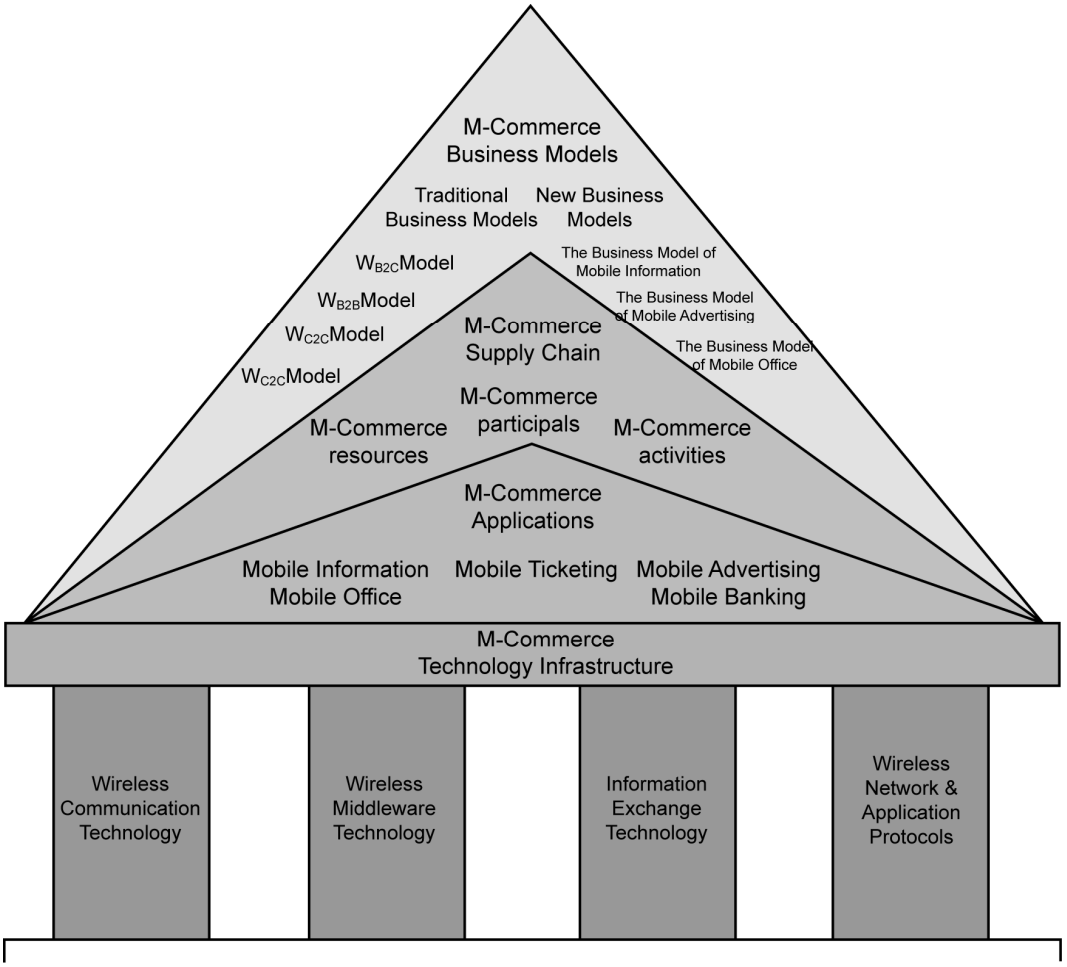


Figure 11.6: An enterprise architecture framework for mobile commerce

Generally speaking, the EA framework defines the scope of the resulting architecture, which typically includes a business view, information integration, application-level views, and technology infrastructures. Definitions of each view may include more refined constructs and relationships at a lower level of granularity. The application of an EA approach is considered relevant and appropriate since the ecosystem within which the development of m-commerce solutions occurs comprises a set of interrelated perspectives based upon the integration of mobile devices, technologies, and business processes. Therefore, an EA framework for mobile commerce can be considered to address, at least, the scope of architecture covering the business level/view, the application level and the technology infrastructure level. In each level of our proposal, core components have been identified. These are: a business model of m-commerce (business level), supply chain of m-commerce (supply chain level), m-commerce applications (application level) and technology infrastructure for m-commerce

(technology level). As shown in Figure 11.6. Our research has shown that integration of core EA approaches with m-commerce is relatively sparse, whereas literature in both contributing areas is substantial. Furthermore, Leist & Zellner state that "as information systems grown in complexity and scope the need for a comprehensive and consistent approach in modeling these systems becomes of paramount importance." Basole recognizes that businesses are "just beginning to recognize the importance and potentially transformative impact of enterprise mobility." Given this our approach to applying EA principles to m-commerce appears well- grounded. A proposed Enterprise Architecture Framework for mobile commerce is shown in Figure 11.5.

First Level: M-Commerce Business Models

Figure 11.5 shows the first (topmost) level in our Enterprise Architecture framework for m-commerce relating to business models. This level is identified as the first level in the EA framework and provides a description of the roles and relationships of an organization, its customer, partners and suppliers and stakeholders, as well as the flows of goods, information and money between these parties and the main benefits for those involved. The stakeholder transactional models are based upon those presented by Coursaris & Hassanein and are divided into four models:

(a) Wireless Business-to-Consumer (WB2C) model;

(b) A Wireless Consumer-to-Business (WC2B) model;

(c) A Wireless Consumer-to-Consumer (WC2C) model; and

(d) Wireless Consumer-to-Self (WC2) model.

These models mainly describe business activities between the contributing parties, and the specific nature of these. The abstraction of these models provides a basis for higher level understanding of the spectrum of stakeholders and relationships at the topmost level of the architecture. Other business models of mobile commerce attempt to address the complexities associated with the m-commerce ecosystem comprising different participants (such as mobile consumers, network operators, service providers, application developers, content providers and technology providers) and encompass the types of services and sources of profits. In addition, models describe operations and processes relating to mobile growth, value-added benefits, revenue models and return on investment, and the transfer of benefits across stakeholders for mutual gain. During the development of our framework we reviewed models broadly associated with the business models for mobile information, mobile advertising and for mobile office work. Typical characteristics of these were associated with relationships between content providers, network operators and service

providers for direct gain through revenue opportunities from mobile consumers and shared distribution amongst stakeholders and participants. Interestingly, one specific finding was that, "mobile consumers represent the only sustainable revenue source for participants". From the characteristics identified, it is seen that business models for mobile commerce are helpful in enabling organizations to increase the possibilities for increasing revenue and profit, and enhancing competitiveness. Taking Vodafone Group Plc as a typical example rather than complex business activity can be observed where the business acts as wireless network operator providing information transmission services for its customers. Whilst also acting as a content or service providers of specific service content for its customers such as music downloading, online gaming, e-mail and location-based information. In this and other similar cases the business models appear not to be mutually exclusive (e.g. single and independent WB2C, WC2B, WC2C, WC business models). Thus, encapsulating the diverse stakeholder groups and associated mobile information needs can be seen as providing competitive advantage and market diversification.

Second Level: The Supply Chain of Mobile Commerce

Figure 11.5 shows the second level in the Enterprise Architecture framework for mobile commerce and is concerned with the supply chain. The materialization of business models for m-commerce depends on a complex chain of business relationships between participants of the supply chain The supply chain for m-commerce is therefore identified as the second and supporting level in the enterprise architecture framework. This supply chain for m-commerce can be seen as rooted in mobile telecom markets, within which a variety of participants possess resources, perform activities, and are in relationships that are established or evolving, delivering an end-to-end service. However, given the increasing diversity of applications, services and associated information delivery, this supply chain is being extended to include a spectrum of market areas which, when aggregated provide value-added services to the mobile user. The core elements of the m-commerce supply chain include mobile commerce participants, mobile commerce resources and mobile commerce activities. Mobile commerce participants are a major driving force behind the m-commerce supply chain. Mobile commerce participants fundamentally include mobile consumers, wireless network providers, content providers, service providers, application developers and technology providers - all of whom engage in a business-oriented relationship and, in the specific case of m-commerce, one that is focused on commercial gain for service delivery participants. For example, the shared/ distributed revenue approach entails the wireless network providers delivering part of that revenue to other participants in the supply chain - such as content providers and other service providers.

Third Level: Mobile Commerce Applications

The third level of the framework is concerned with mobile commerce applications. Because an organization uses mobile applications to support and deliver its business models through the supply chain, mobile applications are seen as the tangible end-user vehicles that mechanize and enable the m-commerce transaction. Mobile applications are therefore identified at the third level in the Enterprise Architecture framework for m-commerce. These applications can be broadly categorized as communication applications, information applications, entertainment applications and commerce applications. Five main application types were identified as mobile.

Ticketing, mobile advertising, mobile information, mobile banking and mobile office applications. These were found to support the key business models and imperatives (revenue generation or cost reduction) and were constructed using the supply chain to deliver an aggregated service. The constantly developing landscape of mobile technologies, and more specifically application capability, raises a proposition of re-aligning business models and supply chains to fully leverage the potential of that change – thus, suggesting a commercially synergistic relationship (in this case, the applications and technology forcing a re-evaluation of the business models and supply chain).

Fourth Level

The Technology Infrastructure for Mobile Commerce The fourth level of the framework is concerned with technology infrastructure for m-commerce. This includes wireless communication technology, wireless middle ware technology, information exchange technology, wireless network and application protocols and mobile security technology. Essentially, these are the core technological components and infrastructures that enable mobile users in their environments. These technologies support upper levels of the framework (m-commerce business models m-commerce supply chain, and m-commerce applications).

SELF-CHECK 11.2

Is the wisdom methodology useful for 3rd generation mobile computing environment?

EXERCISE 11.2

What do you understand by Enterprise Architecture Framework for mobile commerce and architecture pattern for mobile applications?

SUMMARY

- Location Based Services (LBS) are gaining much importance for all kind of human activities ranging from tourist navigation to support of rescue teams in disaster management. LBS have been developed as an independent from GIS stream for wireless services, but nowadays about 80% of all supplied data is spatially related, i.e. they are presented in a form of 2D map (raster of vector). LBS is becoming a special kind of GIS.

KEY TERMS

MS Mobile Satellite

UDP User Data gram Protocol

WML Wireless Mark-up Language

SELF-TEST

1. How spatial dimensions differ from mobility dimensions?
2. What is the UML development cyclic process?
3. Define the caller state machine with all its process.
4. Explain the enterprise architecture for the mobile commerce.

Topic 12 ▶ Mobile Applications

LEARNING OUTCOMES

By the end of this topic, you should be able to:

1. Define voice user interface hurdles;
2. Describe hurdles with multimodal applications;
3. Identify the problems with location-based applications;
4. Underline the concept of validating before development;
5. Recognise the importance of stress testing and scalability issues; and
6. Describe location based functionality.

▶ INTRODUCTION

In the previous topic, we have studied Back to the Dimensions of Mobility, Applying the Wisdom Methodology to Mobile Development, UML-Based Development Cycle for Mobile Applications, Practical Concerns with Architectures and Architectural Patterns for Mobile Applications. Now, in this topic, we would focus on the hurdles in mobile application development. The rising penetration of smart phones enables new mobile services and business models. A huge number of different operating systems, available functionalities, open questions regarding revenue sources and streams, legal issues, as well as a lack of knowledge in designing mobile user experiences call for a holistic product development process in this domain.

12.1 VOICE USER INTERFACE

A voice portal can be defined as "speech enabled access to web based information". In other words, a voice portal provides telephone users with a natural language interface to access and retrieve Web content. An Internet browser can be frequently used in the present day computer environment and provide Web access from a computer but not from a telephone.

Natural speech modality is used when communicating with other people. This makes it easier for a user to learn the operation of voice activated services. As an output modality, speech has several advantages. First, auditory input does not interfere with visual tasks, such as driving a car. Second, it allows for easy incorporation of sound-based media, such as radiobroadcasts, music, and voicemail messages. Third, advances in TTS (Text To Speech) technology mean text information can be transferred easily to the user. Natural speech also has an advantage as an input modality, allowing a hands-free and eyes-free use. With proper design, voice commands can be created that are easy for a user to remember. These commands do not have to compete for screen space. In addition, unlike keyboard-based macros (e.g., ctrl-F7), voice commands can be inherently mnemonic ("call United Airlines"), obviating the necessity for hint cards. Speech can be used to create an interface that is easy to use and requires a minimum user attention. One of the most important technologies for a voice portal to function is a good VUI (Voice User Interface). The use of VUI technology is the field of interaction between human voice and the system. In the present day, computer environment a company's customer care services, have turned to Interactive Voice Response (IVR) systems to provide telephonic customer self-service, that are being used to minimise the load on call-centre staff, and cut overall service costs. There are some problems faced by the users. The main consideration is that how well these systems perform and, ultimately, whether customers leave the system satisfied or frustrated-depends in large part on the user interface. Many IVR applications use Touch-Tone interfaces-known as Dual-Tone Multi-Frequency (DTMF) – in which customers' choices are limited to making selections from a menu. In the complex transaction services effectiveness of DTMF systems decrease. In fact, IVR and speech recognition consultancy, Enterprise Integration Group (EIG), reports that customer utilisation rates of available DTMF systems in financial services, where transactions are primarily numeric, are as high as 90 percent; in contrast, customers' use of insurers' DTMF systems is less than 40 percent. The Automated Speech Recognition (ASR) is a technique which drives today's Voice User Interface (VUI) systems. It allows the customers to break the 'menu barrier' and perform more complex transactions over the phone. "In many cases the increase in self-service when moving from DTMF to speech can be dramatic," said EIG's President Rex

Stringham. The best VUI systems have the quality to understand a naturally spoken dialog regardless of the speaker.

Those nuances are good for human beings; they allow us to recognise each other by voice. For computers, however, they make the process much more difficult. That's why a handheld or pocket computer still needs a stylus, and why the 'voice dialling' offered by some cell-phone companies still seems high-tech.

We have definitely headed toward multi-modal applications. ASR vendors are working to make sure that VUI evolves free staff from dealing with voice-related channels; it is better to have them supporting the various modes of service that are just beginning to emerge. Good VUI systems have multiple fallback strategies for speech recognition failure, such as asking callers to spell names or breaking a question into parts. The interface could revert to DTMF, especially if ASR was added to an existing DTMF-enabled system. Still, transferring an unintelligible customer to a human agent is frequently an early fallback option in order to minimise customer frustration. Facilitating this evolution is XML's telephonic flavour, Voice XML. As its name implies, Voice XML is a mark-up language specification that defines the key elements of voice-enabled transactions, which also allows repurposing of this data across any number of platforms and systems.

12.1.1 Speech Recognition Challenges

The first speech-recognition system in the early 1950s was developed in the Bell Labs. The technology was highly accurate at determining single digit spoken numbers. By early 1970s natural language speech understanding was demonstrated by Terry Winograd's SHRDLU system, a robot that understood commands such as "move the red block on top of the small green one." By the 1990s speech recognition was being applied to the internet. In 1996 California State University at Northridge was able to demonstrate an experimental voice controlled web interface. By 2000 many voice driven portals were being developed. Until recently, a number of issues made it almost impossible to develop a speech-recognition engine that would recognise fluent and natural speech. The basic challenges faced by voice application developers included the following:

(a) **Variability of Speech Patterns**
Different people speak the same language differently and even speak the same word in many different ways. Interpreting speech variability has led to the development of complex pattern analysis. Understanding natural pauses; speaking rates and changes in volumes has been complex and difficult.

(b) **Processing Power**
In the mid-1980s, a new technique known as Hidden Markov's Models improved the ability to recognise word relationships. This computation

intensive technology eventually led to powerful speech recognition applications. Achieving real time voice recognition found in voice portals requires processing power that is not commercially viable until recently.

(c) **Extracting Meaning**
Very few speech recognition applications are able to accurately determine the meaning of words. The quality of speech interpretation depends on the ability of the speech recognition engine to properly choose the best match for spoken sounds from its list of expected text phrases. A more advanced process was required to extract meaning from those words. Because of the many possible ways that people speak and the many words that are used to communicate the same concept, a full understanding of human factors was critical to properly interpret the meaning.

(d) **Background Noise**
Before mobile phone users often access voice portals, background has been difficult for voice recognition developers to filter out. The development of better microphones has helped, but issues such as wind, murmurs and music have made it a challenge to properly isolate voice from noise.

(e) **Continuous Speech Recognition**
Designing systems that are powerful enough to understand and respond to continuous speech requires a large amount of processing power that was not available at reasonable cost in the past. When a person speaks at a natural rate, it has been difficult to distinguish which sounds were associated with specific words. For example, the phrase "to recognise speech" could have been misunderstood to be "to wreck a nice beech."

Users do not naturally speak with pauses between words. As a result processing phrases in real time as they are naturally spoken has been a major challenge. Many of the solutions to problem associated to speech recognition are still being fine tuned. Powerful computers have provided the processing power to overcome many of the limitations of the past.

However the most common speech-recognition systems of today are still very different from the way in which people naturally interact.

12.1.2 Hurdles of Speech user Interfaces

Good design is to a great extent about originality and about knowing the particular assets of the used stuff or media, which comprises competently trading with the restrictions of the intended object. Knowing about the restrictions and how to evade them is mainly significant in the design of speech user interfaces. These restrictions symbolize the reasons of errors and usability problems verify

the key confronts that design mainly solve, and their perceptive offers influence for improved designs. Thus, as one establishment for our structure for speech user interface design, we produced organization of speech user interface hurdles. Relying on an inclusive appraisal of suitable literature and specialist knowledge, and expanding similar nomenclatures from other works, we recognized six wide grouping of restrictions: speech recognition, spoken language, environment, human cognition, user, and hardware. Table 12.1 shows the constraint classes, their definition, and the precise restrictions that we have acknowledged in each type. The subsequent subsections converse those three of the six restrictions categories in more intensity that confine the most exclusive properties of speech user interfaces: recognition, spoken language, and human cognition. The other three categories – environment, user, and hardware – are well-understood from other classes of boundaries, and a complete conversation of all categories is beyond the scope of this topic.

Table 12.1: Taxonomy of Limitations of Speech user Interfaces

Limitation Category	Definition	Specific Limitations
Speech recognition	Limitations of (current) speech recognition technology	Errors Finite vocabulary grammar acoustic model spontaneous
Spoken Language	Limitations arising from characteristics of spoken language	Public Natural turn-taking protocol Anthropomorphism Limited expressive power Noise
Environment	Disturbances from the user's environment	Multiple people speaking Interruptions Sequential and slow
Human Cognition	Properties of the human cognitive system	Working memory capacity Low persistence
User	Differences between and preferences among users	Competition with verbal processing Task knowledge Expert/novice Speech competence Channel
Hardware	Properties of the hardware used to implement a speech user interface	Microphones Computing platform

Source:
http://pages.cs.brandeis.edu/~cs216/CS216_docs/Human%20Factors%20in%20IVR%20-%20Chapter%201.pdf

12.1.3 Limitations of Speech Recognition

At the stage of acknowledgment expertise, four categories of recognition restrictions are illustrious: finite expressions, language model, acoustic model, and finally, recognition errors. The first three occur from the essential components of speech recognizers. Expanding the corresponding rows from Table 12.1, lists specific design problems that arise from these problems. As the problems of speech recognition are usually well-known, the focus in this section is not on the boundaries themselves, but on how boundaries are connected to important design troubles in speech user interface design. The subsequent paragraphs converse a few particular design tribulations for each restraint of speech recognition technology.

(a) Recognition Errors

As automatic speech recognition is not ideal (and probably never will be), any speech interface has to supervise recognition faults. Diminishing the error rate is mainly an manufacturing problem. The many communications between recognition parameters make it a non-trivial difficulty that needs speech recognition proficiency. But still with most favorable setup of the recognizer recognition faults remain. Graceful revival from those errors is hard for numerous reasons, together with that it is frequently hard to sense errors, that errors can be frequent or even surge into error spirals (Karat, Halverson, Horn and Karat, 1999), and that hyper talk worsens recognition problems. Hypertalk refers to speech that is over-enunciated and spoken more slowly and loudly, in an effort to defeat communication problems in human dialog. While hypertalk is useful in recovering communication troubles during human-to-human discussion, hypertalk alsodegrades speech recognition performance, rather than helping the recognizer. The last design problem defined in Table 12.2 refers to the perplexity and disturbance that counterintuitive acknowledgment errors can cause in consumers. Speech recognizers do not function like human ears, so words that may "sound" similar to the recognizer sometimes sound very unlike to the human ear.

(b) Finite Vocabulary and Grammar

Most speech systems recognize only language from a pre-defined, finite vocabulary. All speech input is drawn to (a sequence of) words inside that vocabulary. Words that are not comprised by the finite vocabulary, known as out-of-vocabulary words, unavoidably escorts to recognition errors. Connected to the problem of which words to contain in the vocabulary is the trouble of grammar exposure. As the grammar specifies which succession of words is tolerable to the recognizer, the grammar must cover the a variety of ways users may prepare their responses to a certain prompt.

What makes the problem of vocabulary and grammar coverage difficult is the fact that growing vocabulary size and grammar difficulty must be unbiased against making routine recognition more hard by adding confusability between words and word sequences. Usually, a small vocabulary and tight grammar help to accomplish high recognition correctness. The next trouble mentioned in Table 12.2 takes place in the situation of statistical grammars. Statistical grammars are engaged to identify reactions to open-ended prompts, like in AT&T's famous "How may I assist you?" investigate system (Gorin, Parker, Sachs and Wilpon, 1996). Any disparity among definite user input and the data that are used to guide the statistical language model degrades recognition appearance. At a very fundamental level, a recognizer taught for transcription will perform rather poorly on recognizing telephone dialogs. But mismatches can be quite subtle, for instance, changes in word choice owing to regional pronunciations, or shifts over time in how customers portray their cause for calling, in reply to a prompt like "How may I help you?".

(c) **Acoustic Models**

Acoustic models of speech identifiers have only a restricted potential to model inconsistency intrinsic in speech. As a result, any supplementary inconsistency in the acoustic signal makes automatic acknowledgment more complicated. Inconsistency in the acoustic signal occurs from many sources: voices of diverse users, local or foreign pronunciations, fast and slow speakers, co-articulation of words, confusable words, as well as hyper talk, conversed previously in the situation of identification errors.

Table 12.2: Design Problems Arising from Specific Limitations of Speech Recognition

Limitation	Design Problems
Errors	• Minimize errors • How to detect errors • Alleviate (avoid) repeated errors • Cascading errors in correction • Hyper talk exacerbates errors • Errors are often not intuitive o users
Finite Vocabulary	• Out-of-vocabulary words cause recognition errors • Trade-off coverage with confusability and speed

Grammar	• What people say is often difficult to predict • Mismatch between training data and user input
Acoustic Models	• Multiple speakers and accents • Fast and slow speakers • Co-articulation • Confusable words • Acoustic variability in "spontaneous" speech • Poor quality of audio input • Distorted speech due to barge- in or background speech • Speech (endpoint) detection

12.1.3 Limitations of Spoken Language

Outlines and actions educated in human discussion "instinctively" take over to verbal communication with computers and other computerized systems. Fresh research confirms that users even attribute human qualities to interactive media, like speech interfaces (Reeves and Nass, 1996), which is usually termed to as anthropomorphism. Certain qualities of spoken language must thus be measured in speech interface design. We have recognized the subsequent traits of verbal language as being pertinent to speech interface design: that it's unprompted, its public nature, the policies of turn-taking in human discussion, anthropomorphism (recounting the propensity of consumers to allocate human traits to non-human speech systems), and the restricted communicative power of language. Whereas spontaneity, turn-taking, and anthropomorphism can be leveraged usefully in speech user interface design, these qualities often results in usability tribulations, unless alleviated all the way through cautious design. Table 12.3 displays particular design troubles that occur from these five restrictions of spoken language. The following part argues in more aspect how the traits of spoken language results in design tribulations.

Table 12.3: Limitations (Characteristics) of Spoken Language

Spoken Language Characteristic	Specific Design Problem
"Spontaneous" character	Chatty behavior leads to inefficient communication, not suitable for command and control, data entry, dictation. Some users are surprised by open-ended prompts, and confused about how to respond.
Public	Speech can be heard by others: no privacy, and possible disturbance.
Turn-taking	Users abide by turn-taking protocol of human conversation.
Anthropomorphism	Degree of interface personification ("persona"). Complex nuances of human conversation are not conducive to machines. Raised user expectations lead to backlash when disappointed.
Limited expressive power	Need to resolve ambiguities to determine meaning. Difficult to refer to objects in physical environment. Difficult to describe locations or spatial manipulation precisely.

The "unprompted" character of spoken language marks itself in assets like redundancy and disfluencies. Though very usual of human discussion, these actions are not appropriate to goal-oriented, competent message, as needed when performing goal-oriented jobs such as device control, data entry, or transcription. Disfluencies make unprompted speech more complicated to identify than verbal reactions from a directed dialog.

In spite of the "unprompted" trait of human discussion, some consumers are astonished if automated systems imitate the unprompted form of dialog in indefinite prompts, like "How may I help you?" Usually, this design problem is solved by supplying callers with examples when they do not react to this type of prompt.

12.2 PROBLEMS ON BUILDING LOCATION BASED SERVICES

Here we discuss about the location based services and the problems that many users face when using these services. In the present time there is competition in the telecommunication industry and different companies provide different services to our customers. In this age of significant telecommunications competition, mobile network operators continuously seek new and innovative ways to create differentiation and increase profits. One of the best ways to achieve this is through the delivery of highly personalised services. One of the most powerful ways to personalise mobile services is based on location. We will discuss Location Based Services (LBS), but we will first discuss the basis of LBS – location technology.

12.2.1 Positioning

One of the most common location based services is positioning. The modern mobile companies provide the Global positioning System for this purpose. There are some other meanings of positioning in addition of a GPS. These other technologies are network based positioning which typically rely on various means of triangulation of the signal from cell sites serving a mobile phone.

12.2.2 Geographic Information Systems

One important aspect of any location service is the geographic data. Geographic Information Systems (GIS) provide the tools to provision and administer base map data such as manmade structures (streets, buildings) and terrain (mountains, rivers). GIS is designed in such a way that it can show point-of-interest data such as location of gas stations, restaurants, nightclubs, etc. Finally, GIS information also includes information about the radio frequency characteristics of the mobile network.

12.2.3 Location Management Function

In mobile phones there must be a location management function to process positioning and GIS data on behalf of LBS applications. The location management function is defined as a gateway and mediator between positioning equipment and LBS infrastructure.

12.2.4 Services

(a) **Location-Based Information**
In case, the mobile user wants to know any location at any point of time and if the mobile is WAP enabled, then by using this service the user can get the information about any place at any time. The LBS application would interact with other location technology components to determine the user's location and will provide a list of restaurants within a certain proximity to the mobile user. A diagram showing location based service in mobile phones is given below Figure 12.1.

(b) **Emergency Services**
When we dial 9-1-1 from our mobile phones then we will see that it is a location based service that pinpoints your location and relays it to the appropriate authorities. The FCC has mandated that by October of 2001, all wireless carriers in the United States must provide a certain degree of accuracy in pinpointing the location of mobile users who dial 9-1-1.

(c) **Tracking**
Tracking is also an application of mobile commerce services. A mobile user can opt for tracking and gets information that he desires, such as notification of a sale on men's suits at a store close to the user's current proximity.

There are some problems faced by the users when they use location-based services. These problems are:

(a) No coverage indoors or underground.

(b) Drains battery very rapidly.

(c) Available only with the costly phones.

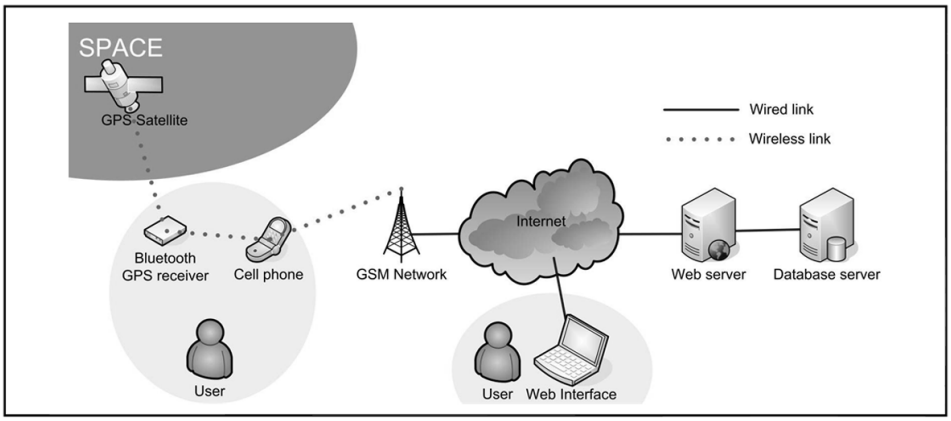


Figure 12.1: Location based service in mobile phones

SELF-CHECK 12.1

1. Explain the voice user interface hurdles.
2. Explain the problems with building location based services.

12.3 USAGE OF POWER

In this topic, we present a measurement study of the energy consumption characteristics of three widespread mobile networking technologies: 3G, GSM, and Wi-Fi. We find that 3G and GSM incur high tail energy overhead because of lingering in high power states after completing a transfer. Based on these measurements, we develop a model for the energy consumed by network activity for each technology. How are the energy consumption characteristics of network activity over 3G, GSM, and Wi-Fi on mobile phones compared with each other? How can we reduce the energy consumed by common applications using each of these three technologies? To investigate these questions, we first conduct a detailed measurement study to quantify the energy consumed by data transfers across 3G, GSM, and Wi-Fi. We find that energy consumption is intimately related to the characteristics of the workload and not just the total transfer size, e.g., a few hundred bytes transferred intermittently on 3G can consume more energy than transferring a megabyte in one shot. Above is a summary of the key findings of our measurement study, which remain consistent across three different cities, diurnal variation, mobility patterns, and devices.

(a) In 3G, a large fraction (nearly 60%) of the energy, referred to as the tail energy, is wasted in high-power states after the completion of a transfer. In comparison, the ramp energy spent in switching to this high-power state before the transfer is small. Tail and ramp energies are constants that amortise over larger transfer sizes or frequent successive transfers.

(b) In GSM, although a similar trend exists, the time spent in the high-power state after the transfer, or the tail time, is much smaller compared to 3G (6 vs. 12 secs). Furthermore, the lower data rate of GSM implies that more energy is spent in the actual transfer of data.

(c) In Wi-Fi, the association overhead is comparable to the tail energy of 3G, but the data transfer itself is significantly more efficient than 3G for all transfer sizes. Based on these findings, we develop a simple model of energy consumption of network activity for each of the three technologies. We utilise these models to identify opportunities for reducing the energy consumption of network activity induced by common mobile applications.

To this end, we design Tail Ender, an energy efficient protocol for scheduling data transfers. Tail Ender considers two classes of applications: (1) delay-tolerant applications such as email and RSS feeds, (2) applications such as web search and web browsing that can benefit from aggressive pre-fetching.

For delay-tolerant applications on 3G and GSM, Tail Ender schedules outgoing transfers so as to minimise the overall time spent in high energy states after completing transfers, while respecting user specified delay-tolerance deadlines.

12.3.1 Energy Measurement Methodology

3G and GSM

Our 3G and GSM measurements quantify the:

(a) Ramp energy: energy required to switch to the high-power state;

(b) Transmission energy; and

(c) Tail energy: energy spent in high-power state after the completion of the transfer.

We conduct measurements for data transfers of different sizes (1 to 1000 KB) with varying intervals (1 to 20 seconds) between successive transfers. We measure energy consumption by running NEP in the background while making data transfers. For each configuration of (x, t), where x 2 [1K, 1000K] and t 2 [1, 20] seconds, the data transfers proceed as follows: The phone initiates an x KB upload/download by issuing an http-request to a remote server. After the upload/download is completed, the phone waits for t seconds and then issues the next http request. This process is repeated 20 times for each data size. Between data transfer experiments for different intervals, the phone remains idle for 60 seconds. The energy spent during this period is subtracted from the measurements as idle energy.

Wi-Fi

Our Wi-Fi measurements quantify the energy:

(a) To scan and associate to an access point; and

(b) To transfer data. We conduct two sets of measurements.

In the first set of measurements, for each data transfer, we first scan for Wi-Fi access points, associate with an available AP and then make the transfer. In the second set of measurements, we only make one scan and association for the

entire set of data transfers to isolate the transfer energies. In addition, all three networks, 3G, GSM and Wi-Fi, incur a maintenance energy, which is the energy used to keep the interface up. We estimate the maintenance energy per second by measuring the total energy consumed to keep the interface up for a time period.

Accounting for Idle Power

For all measurements, we configure the phone in the lowest power mode and turn off the display and all unused network interfaces. The energy profiler itself consumes a small amount of energy, which we include in the idle power measurement. We measure idle energy by letting the energy profiler run in the background with no other application activity. The average idle power is less than 0.05W and running the energy profiler at a sampling frequency of 0.25 seconds increases the power to 0.1 W.

3G Measurements

Figure 12.2 shows the average energy consumption for a typical 50KB download over 3G. We find that the Tail energy is more than 60% of the total energy. The Ramp energy is significantly small compared to the tail energy, and is only 14% of the total energy. 3G also incurs maintenance energy to keep the interface on, and is between 1-2 Joules/minute.

Figure 12.2 shows the average energy consumed for download when the time between successive transfers is varied. We ignore the idle energy consumed when waiting to download the next packet. Consider the data points for downloading 100 KB data. The energy increases from 5 Joules to 13 Joules as the time between successive downloads increases from 1 second to 12 seconds. When the time between successive downloads is greater than 12.5 seconds, the energy consumed for 100 KB transfers plateaus at 15 Joules. When the device waits less than the tail- time to send the next packet, each data transfer does not incur the total Tail energy penalty, reducing the average energy per transfer.

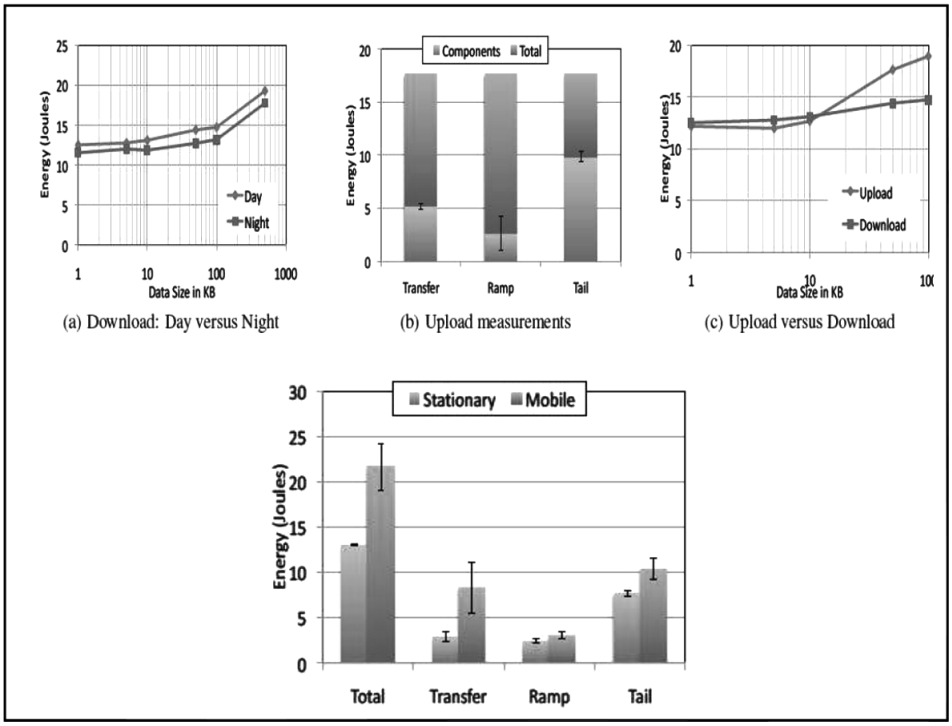


Figure 12.2: 3G measurement

This observation suggests that the Tail energy can be amortised using multiple transfers, but only if the transfers occur within tail-time of each other. This observation is crucial to the design of Tail Ender, a protocol that reduces the energy consumed by network applications running on mobile phones. G SM Measurements We conducted a set of measurements using the two Nokia phones equipped with GSM.

Figure 12.2 shows the average energy consumption in GSM networks as a proportion of the Tail energy, Ramp energy and transfer energy for a 50K download. Unlike in 3G, the Tail energy only accounts for 30% of the transfer energy. However, similar to 3G, the Ramp energy in GSM is small compared to the Tail energy and the transfer energy. We also observed that the tail-time is 6 seconds and GSM incurs small maintenance energy between 2-3 J/minute. Due to the small tail-time in GSM (unlike 3G), data sizes dominate energy consumption rather than the inter-transfer times. Figure 12.2 shows the average energy consumed when varying the time between successive transfers. The average energy does not vary with increasing inter-transfer interval. For example, for data transfers of size 100 KB, the average energy consumption is between 19 to 21 Joules even as the time between successive transfers varies. In comparison, Figure 12.2 shows that the average energy consumption varies

significantly in 3G with varying inter-transfer interval, until the inter-transfer interval grows to more than the tail-time.

12.3.2 Limitations and Future Work

Tail Ender is naturally suited to be implemented in the operating system, exposing a simple API to applications. Applications only need to provide a delay-tolerance limit for each item sent. Today, commodity phones such as the iPhone already requests the user to specify a delay-tolerance limit for certain applications in order to improve battery life. Implementing Tail Ender in the kernel and refining the API to make it easily usable by users or application developers is left for future work. Our results are based on email, RSS feeds and web search traces collected from real desktop or laptop users. For a more realistic evaluation of Tail Ender's energy savings however, we need the application usage patterns of users on mobile devices. Usage patterns on mobile devices provide two benefits. First, it helps quantify the energy benefits in the presence of cross-application optimisation. For example, if a user multi-tasks between sending an email and searching the web, then the transmissions for the two activities can be scheduled together to reduce energy consumption. Second, the usage patterns provide us the fraction of time each application is used by a mobile user. This will help quantify the average per day energy savings for a given usage pattern. As part of future work, we seek to collect traces of mobile usage patterns that can inform cross-application opportunities and better quantify the energy benefits of Tail Ender for mobile users.

12.3.3 Conclusion

Energy on mobile phones is a precious resource. As phones with multiple wireless technologies such as 3G, GSM, and Wi-Fi become common, it is important to understand their relative energy consumption characteristics. To end this, we conducted a detailed measurement study and found significant tail energy overhead in 3G and GSM. We developed a measurement-driven model of energy consumption of network activity for each technology. Informed by the model, we developed Tail Ender, a protocol that minimises energy usage while meeting delay- tolerance deadlines specified by users. For applications that can benefit from pre- fetching, Tail Ender aggressively pre-fetches data, including potentially useless data, and yet reduces the overall energy consumed. We evaluate Tail Ender for three case study applications-email, news feeds, and web search based on real user logs and find significant savings in energy in each case. Experiments conducted on the mobile phone shows that Tail Ender can download 60% more news feed updates and download search results for more than 50% of web queries, compared to using the default policy. Our model

driven simulation shows that Tail Ender can reduce energy by 35% for email applications, 52% for news feeds and 40% for web search.

12.4 TESTING MOBILE APPLICATIONS

12.4.1 Validating the Mobile Use Cases before Development

What was a Use Case in 1987? According to the OOPSLA'87 paper, "A use case is a special sequence of transactions, performed by a user and a system in a dialogue." This is pretty similar to our current (informal) definition. We developed a separate model for describing a system from an outside perspective and called it a use-case model. This provided a black-box view of the system – the system's internal structure would be of no interest in this model. Some people have misunderstood the term outside, mistaking it for a synonym for user interface. Instead, the use-case model represents the functional requirements of the system.

At this time the use-case model also included entity (domain) objects, so we could show how use cases could <<access>> entities. Use cases and entities were class- like; they had operations and data. The other relationship depicted in the use- case model was <<built-on>> which was described as "an extended form of inheritance relation. Multiple inheritances are common." In fact, the built-on relationship was a combination of the generalisation relationship and the <<extend>> relationship. After the use cases were specified, they were also designed and tested.

You create as many processes [today we would say activities] as there are use cases. The conceptual model of the use cases is translated seamlessly into a new model showing how each use case is implemented by means of the identified blocks [today a block would be a subsystem, class, or component]. Each use case is tested separately to safeguard that the system meets the requirements of the user. Please, note that the use cases constitute the key aspect through the entire [set of] development activities. Sequence diagrams were used to show interactions among the blocks/components. This was no surprise, since sequence diagrams had shown their value in practice for almost twenty years prior to that time.

What was a Use Case by 1992? As you can see, use cases had assumed much of their current shape (syntax and semantics) by this time. Between 1987 and 1992, they evolved and matured, as about twenty customers put the Objectory Process to practical use for new product development. These customers were involved in many different kinds of systems: management information; defence (pilot, counter measure, C3I); and telecom (POTS, mobile). Use cases took on a new relationship: "inheritance" (now replaced by "generalisation"). We avoided what is now called the <<include>> dependency, which we thought would damage the modelling by inviting functional decomposition.

To increase clarity, we made it an important issue to distinguish between a use case (as a class-like thing), an instance of a use case, and a description of a use case. The depth of the use-case model was in its use-case descriptions. Each use-case description contained the following:

(a) A brief description of the purpose of a use case;

(b) A flow of control;

(c) Base flows and alternative flows;

(d) Sub-flows (reusable in many places within the same use-case description); and

(e) Preconditions and post conditions.

The use case was more than just a requirement technique; it was like the hub of a wheel:

Use cases were traceable to analysis, design, implementation, and test. For each use case in the model, we created collaboration (a view of participating classes) in analysis and design. Each use case resulted in a set of test cases. Use cases were important in designing user interfaces and structuring the user manual. Use cases also moved into the space of business modelling, since they could perfectly capture business processes.

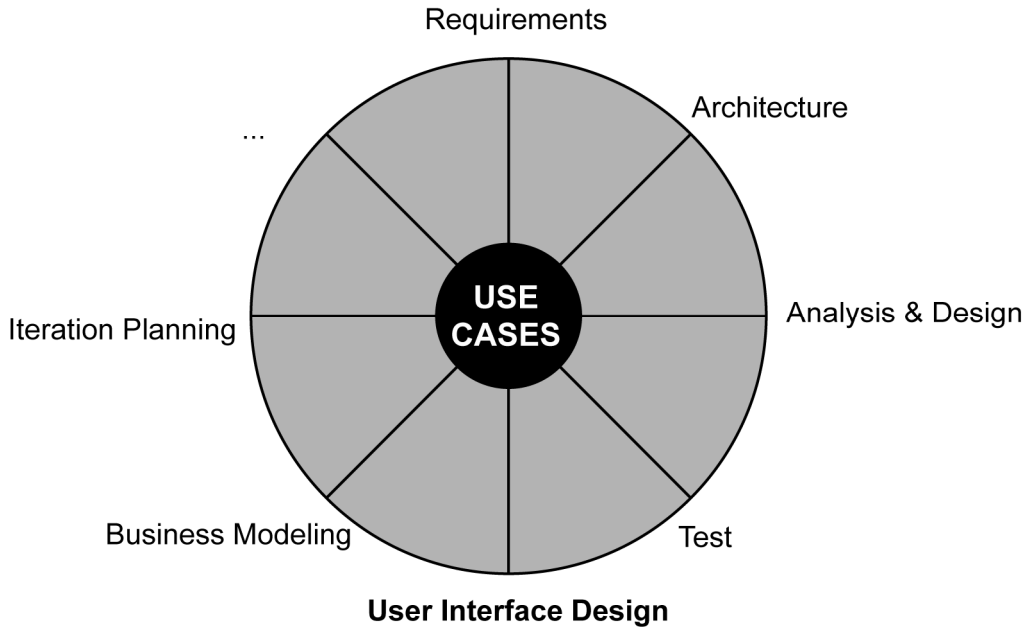


Figure 12.3: Use cases were like the hub of a wheel

We coined the term use-case driven development for our approach. First, we identified all use cases and specified each one of them in requirements. Then we analysed and designed each use case in respectively and finally tested each.

Today: A Lot has Happened Since Then

The adoption rate of use cases has surprised us. They were embraced almost immediately by all methodologists and basically adopted worldwide. Other important techniques such as component-based design and object-oriented modelling were much more controversial and needed a much longer adoption time. Probably this is because use cases are basically a simple and obvious idea; they work well with objects and object thinking. Using use cases is not just a technique for managing requirements; it binds together all the activities within a project, whether a mini-project such as a single iteration or a major project resulting in a new product release.

How Many Use Cases is Enough?

The current definition of use cases basically goes back to 1994. To help teams strike a balance between defining too many use cases or too few, I added a requirement that a use case must give a "measurable value" to a "particular actor." As a rule of thumb, it has been suggested that a large system supporting one business process should have no more than twenty use cases. We realised

that giving any such number could lead people to take undesirable actions to get the "right" number. If they had fewer than twenty use cases, they might split some of them to get up to twenty. Or, if they had more than twenty use cases, they might combine use cases to bring down the count. But that is the wrong approach. We have seen good use-case models for commercial systems with as few as five use cases, and some with as many as forty use cases. However, we have also seen use- case models with as many as 700 use cases! Obviously, these were unsound models. The twenty use cases we suggested should all be concrete (real) use cases and not generalisations or extension/inclusion fragments.

Use Cases and the Unified Modelling Language

Use cases have become part of the Unified Modelling Language (UML). And because the UML is precisely defined, use cases and associated concepts (such as use-case instance [UCI]) are also now precisely defined, thanks to the UML's powerful classifier concept. Although the old definition that "a use case is a sequence of actions," is still compatible with the user's perspective, the definition based on classifiers is what methodologists, process engineers, and tool builders need for clarity.

Note, however, that although the UML effort resulted in a much more precise definition of use cases, it did not do much to evolve them. Roughly speaking, we only changed the "uses" relationship to a generalisation, and we added an <<include>> relationship. The "uses" relationship in the Objectory Process was previously called "inheritance" and was never intended to be used for <<include>> fragments. In the past, we didn't allow developers to model these fragments, but used another technique involving text objects instead. We shall discuss these text objects later.

Since the company became part of Rational, we have been very happy with the way our Rational Unified Process, Â or RUPÂ team has correctly implemented use cases and improved their practical use. Although they have made no really dramatic changes, they have provided much better explanations of use cases, based on the experience of thousands of customers and our own experts. In particular, a new book, Use Case Modelling, by Kurt Bittner and Ian Spence, is now on the shelves. This is THE book on use cases. We strongly recommend that everyone involved in software engineering and requirements development read it. Also, work by Jim Conallen and by Peter Eeles, Kelli Houston, and Wojtek Kozaczynski on user experience design with use cases is a great improvement on our earlier work in this area, and is very much in line with the original use-case concept.

Now may be the time to take steps to grow (clarify and extend) the idea of use cases. But first, a word of warning about formalising uses cases.

12.4.2 Effect of Dimensions of Mobility on Software Testing

Overview

Ideally, we would like a fresh test suite every time we test. Although the rationale for testing with an operational profile is compelling, realising its promise presents significant practical problems. Since critical failures are often a result of high load in combination with unusual input sequences, fresh test suites that can vary load and achieve stress are highly effective. The article presents a case study where these problems were solved and discusses application to testing mobile systems.

A Compelling Approach

Software testing presents two fundamentally hard problems: choosing tests and evaluating test results. Choosing tests is hard because there is an astronomical number of possible test inputs and sequences, yet only a few of those will succeed in revealing a failure. Evaluation requires generating an expected result for each test input and comparing this expected result to the actual output of a test run.

Even with rigorous software engineering, it is a near certainty that undiscovered bugs are present, especially in complex distributed systems. Testing can reveal bugs, but is expensive. We must therefore marshal available testing resources to achieve the greatest possible improvement in reliability of a system under test. But how?

Suppose we have achieved adequate component testing and demonstrated minimal end-to-end stability. Now what? How can we best spend scarce testing resources? The logic of testing with an operational profile is compelling. "Testing driven by an operational profile is very efficient because it identifies failures (and hence the faults causing them), on the average, in the order of how often they occur. This approach rapidly reduces failure intensity as test proceeds, and the faults that cause frequent failures are found and removed first. Users will also detect failures on average in order of their frequency, if they have not already been found in test." When we want to maximise the reliability of a fielded system (and thereby maximally enhance the user/customer experience) and have no better information about how to find bugs, any other allocation of testing

resources is sub-optimal. Testing in accordance with an operational profile is therefore a key part of my test automation strategy.

This article summarises how our automated model-based testing approach has evolved over the last ten years and how it continues to evolve to meet the challenges of 21st century information technology.

SELF-CHECK 12.2

How are the dimensions of mobility useful on software testing?

12.4.3 Stress Testing and Scalability Issues

Stress testing is an important part of software engineering. Stress testing is a form of testing that is used to determine the stability of a given system or entity. It involves testing beyond normal operational capacity, often to a breaking point, in order to observe the results. Stress testing may have a more specific meaning in certain industries, such as fatigue testing for materials.

The concept of stress testing is used in software testing. A stress test refers to tests that put a greater emphasis on robustness, availability, and error handling under a heavy load, rather than on what would be considered correct behaviour under normal circumstances. The goal of such tests may be to ensure the software does not crash in conditions of insufficient computational resources (such as memory or disk space), unusually high concurrency, or denial of service attacks.

Examples:

- A web server may be stress tested using scripts, bots, and various denials of service tools to observe the performance of a web site during peak loads.

Stress testing may be contrasted with load testing:

- System stress testing, also known as stress testing, is loading the concurrent users over and beyond the level that the system can handle, so it breaks at the weakest link within the entire system.
- Load testing examines the entire environment and database, while measuring the response time, whereas stress testing focuses on identified transactions, pushing to a level so as to break transactions or systems.
- During stress testing, if transactions are selectively stressed, the database may not experience much load, but the transactions are heavily stressed. On the other hand, during load testing the database experiences a heavy load, while some transactions may not be stressed.

In telecommunications and software engineering, scalability is a desirable property of a system, a network, or a process, which indicates its ability to either handle growing amounts of work in a graceful manner or to be readily enlarged. For example, it can refer to the capability of a system to increase total throughput under an increased load when resources (typically hardware) are added. An analogous meaning is implied when the word is used in a commercial context, where scalability of a company implies that the underlying business model offers the potential for economic growth within the company.

It is a highly significant issue in electronics systems, database, routers, and networking. A system, whose performance improves after adding hardware, proportionally to the capacity added, is said to be a scalable system. Scalability, as a property of systems, is generally difficult to define and in any particular case it is necessary to define the specific requirements for scalability on those dimensions which are deemed important. An algorithm, design, networking protocol, program, or other system is said to scale if it is suitably efficient and practical when applied to large situations (e.g., a large input data set or large number of participating nodes in the case of a distributed system). If the design fails when the quantity increases then it does not scale.

The concept of scalability applies to technology as well as business settings. The base concept is consistent – The ability for a business or technology to accept increased volume without impacting the contribution margin (= revenue – variable costs). For example, a given piece of equipment may have capacity from 1-1000 users, and beyond 1000 users, additional equipment is needed or performance will decline (variable costs will increase and reduce contribution margin).

Measures

Scalability can be measured in various dimensions, such as:

(a) **Load Scalability**
The ability for a distributed system to easily expand and contract its resource pool to accommodate heavier or lighter loads. Alternatively, the ease with which a system or component can be modified, added, or removed, to accommodate changing load.

(b) **Geographic Scalability**
The ability to maintain performance, usefulness, or usability regardless of expansion from concentration in a local area to a more distributed geographic pattern.

(c) **Administrative Scalability**
The ability for an increasing number of organisations to easily share a single distributed system.

(d) **Functional Scalability**
The ability to enhance the system by adding new functionality at minimal effort.

Scale Vertically vs. Horizontally

Methods of adding more resources for a particular application fall into two broad categories.

Scale Vertically (Scale-up)

To scale vertically (or scale up) means to add resources to a single node in a system, typically involving the addition of CPUs or memory to a single computer. Such vertical scaling of existing systems also enables them to use virtualisation technology more effectively, as it provides more resources for the hosted set of operating system and application modules to share. Taking

advantage of such resources are called "scaling up", such as expanding the number of Apache daemon processes currently running.

Scale Horizontally (Scale-out)

To scale horizontally (or scale out) means to add more nodes to a system, such as adding a new computer to a distributed software application. An example might be scaling out from one web server system to three.

As computer prices drop and performance continues to increase, low cost "commodity" systems can be used for high performance computing applications such as seismic analysis and biotechnology workloads that could in the past only be handled by supercomputers. Hundreds of small computers may be configured in a cluster to obtain aggregate computing power which often exceeds that of single traditional RISC processor based scientific computers. This model has further been fuelled by the availability of high performance interconnects such as Martinet and Infini Band technologies. It has also led to demand for features such as remote maintenance and batch processing management previously not available for "commodity" systems.

The scale-out model has created an increased demand for shared data storage with very high I/O performance, especially where processing of large amounts of data is required, such as in seismic analysis. This has fuelled the development of new storage technologies such as object storage devices.

12.4.4 Testing Location based Functionality

GPS-enabled, location-based services hold out considerable promises as a valuable revenue generator for wireless carriers. Their extension beyond the realms of the emergency services and into the commercial arena looks set to open up a significant opportunity for wireless carriers to provide their subscribers with value-added applications and services utilising location-based functionality.

Consequently, they are becoming a powerful motivator in the development of new mobile handsets that incorporate high accuracy Assisted GPS (A-GPS) technology. But, as always, time-to-market and cost-to-market are critical.

Reducing development time is fundamental to ensuring rapid deployment of new mobile handsets with location features. Needless to say, however, these conflicts with the testing complexity that results from a constantly evolving LBS feature set and the requirement for compliance testing before a new mobile handset can be launched.

Out of the lab, into the field until recently, handset testing has taken place mostly in the laboratory. But the rapidly emerging development and conformance test requirements for A-GPS based mobile handsets, as well as time-to-market pressures and the need for field trials has created a pressing need among OEMs for in-field testing of mobile handsets.

The solution must provide GSM/(E) GPRS network emulation, a well-proven te chno log y platf orm , an d s oftw are en viro nme nts for dev elo pme nt a nd conformance testing. Typical of such test platforms is Aero flex's fully integrated A-GPS test solution. It is based on its well-proven 6103 AIME and 6103 AIME/CT mobile handset test systems.

The integrated 12 channel NLC"L1"AGPS Constellation Simulator' from Navigation Laboratories is widely used in global avionics, defence, and space applications. The platform emulates the entire 24 satellite GPS constellation and provides high performance modelling of all associated GPS satellite and vehicle effects, to provide a complete environment for developing and testing A-GPS devices, and this functionality is included in Aero flex's A-GPS solution.

The systems support MS-Based and MS-Assisted positioning methods, and provide analysis of the pseudo-range and latitude/longitude location response for the unit under test.

Support for control and user plane techniques using RRLP allows simulation of the standard A-GPS call flow location requests for Control Plane, and for User Plane the capability to support TCP/IP interface and SUPL (Secure User Plane). RF minimum Performance testing of the GPS receiver is also a key test requirement. Aero flex is currently participating at 3GPP GERAN meetings as the minimum performance test specifications are being developed.

The standard real time logging and decoding of the protocol signalling messages between mobile and simulated network has been enhanced to support the control and user plane messaging. A complete decode of the A-GPS message content is provided in an exclusive application window.

Test Environment

The A-GPS functionality should be a fully scale able upgrade that can be added to both the basic test platform such as the 6103 AIME and 6103 AIME/CT systems. This allows the OEM to support the development lifecycle of A-GPS devices from initial R&D through to conformance test.

Development: R&D requirements must focus on state machine debugging, protocol stack design and regression testing. In addition, a scripting environment and powerful API such as that found in the AIME platform provide complete testing flexibility.

Scripts can easily be created to run development test simulations for specific A-GPS elements of a mobile handset's operation through to complex simulations that emulate a field environment. Example test scripts include access to each Layer of the protocol structure including the RRLP messaging and the ability to define any RRLP protocol message such as the configuration of the Assistance Data content.

An intuitive user interface such as the 6103 AIME's can provide a number of features designed to support A-GPS test requirements such as real time logging, decoding of detailed protocol signalling messages between the mobile handset and network, A-GPS results analysis of the device under test and the generation of log files to allow off-line protocol analysis.

It also includes the colour coding of information to improve readability, the ability to synchronies traffic (Layer 2, Layer 3 including RRLP protocol frames) and an exposed COM based API allowing example scripts to be written in any COM compliant language.

Conformance: The 6103 AIME/CT + Navigation Laboratories NLC-L1-AGPS GPS Constellation Simulator, platform 69 at the Global Certification Forum (GCF) and PCS Type Certification Review Board (PTCRB), can easily be upgraded from a 6103 AIME system. This upgrade enables the user to execute validated 3GPP conformance test cases. Conformance testing is required during the latter stages of the development lifecycle of a GSM device in order to verify that the implementation is fully compliant with the 3GPP standards prior to formal certification at an independent test laboratory.

Within the conformance environment, the automated report generator and analysis software produces detailed results analysis defined in accordance with the associated conformance standards.

Test Cases

A-GPS conformance testing requires a fully compliant test case as listed in 3GPP TS51.010 section 70. Test cases are currently being validated and will be submitted within the next month to GCF and PTCRB. These test cases can be run in any combination or in a campaign of GSM, (E) GPRS, AMR, DTM, DARP and GAN test cases for use in terminal regression test plans.

The test cases are fully automated ensuring that the network emulator hardware and the GPS simulator are configured to the user's requirements. In the development environment, test cases are user modifiable allowing parameters to be changed for testing beyond the test standards. In the conformance environment, test cases are not modifiable to ensure that every test run complies with the appropriate test standard.

The GPS Simulator

The GPS simulator provides a feature rich emulation environment that supports the complete modelling of the complex effects required for verification and evaluation of all characteristics of GPS equipment performance. It combines and integrates an RF signal generator, with scenario development and modelling software within a Windows operating system. The flexible but intuitive user interface ensures that the software meets the needs of operators of varying levels of experience.

The simulator is supplied with pre-written GPS scenarios, developed to 3GPP minimum performance standards. The scenarios include GPS Accuracy (moving and stationary), GPS Sensitivity, GPS Dynamic Range and GPS Multipath.

The Scenario Generation operating mode (such as the one optionally supplied with Aero flex's the development solution) provides an operator interface that allows modification of the configuration, environmental and dynamic vehicle modelling in addition to the GPS system functions. The 3GPP GPS referenced scenarios are file based and can be deterministically executed. The graphical user interface provides simple to use pull-down menu bars and buttons within the modelling and scenario generation software to allow the operator to modify or develop their own scenarios to test above and beyond the conformance standards.

EXERCISE 12.2

1. Explain validating the mobile use cases before development.
2. Explain stress testing and scalability issues.

SELF-CHECK 12.3

How are we test Location Based Services.

12.5 A CASE STUDY

12.5.1 Requirement Driving Architecture

The importance of mobile device grew up within the last 20 years. People use their mobile devices not only as phone sets but also as powerful data communication and entertainment tools. Each major mobile device manufacture has its own proprietary operating system such as Symbian OS, RIM BlackBerry, Windows Mobile, iPhone OS, Palm OS, Android, and so on. At present, most of the manufactures have adopted the MIDP2 to support cross platform Java running environment and to improve interoperability of mobile software developed in J2ME. There are many J2ME games which have been well accepted and greatly enjoyed by mobile users, especially the youth.

However, because of the heterogeneous mobile devices, even for the MIDP2 ready mobile devices, running mobile software on multi-platform is still a big challenge in mobile computing. To extend widely used mobile devices beyond their cellular phone capabilities, we can utilise mobile devices to serve for an extensive application needs with installation of various mobile application software. For example, an adaptive mobile learning application system integrated with location- based service can offer location based learning. It needs mobile learners to provide their geographical information, such as GPS coordinates by running application software on their mobile devices. Similarly, a truck driver can use his mobile device running a logging application to collect the driving information including GPS data for the purpose of compliance with the driving safety regulation. To develop and implement this type of mobile application software, there are five design principles for mobile computing, multi-platform adaptation, little resource usage, little human/device interaction, small data communication bandwidth use, and no additional hardware should be followed. Among the Principles, multi-platform adaptation is most critical and challenging. In the present market, widely available MIDP2 ready mobile devices support the J2ME APIs defined in JSR 118, which has improved J2ME application's interoperability to a certain extent. In most mobile applications, however, the requirement of J2ME APIs is usually far beyond MIDP2 specification. For instance, location API is defined in JSR 179 for writing mobile location-based applications; Bluetooth API is defined in JSR 82 for integrating into a Bluetooth environment; and PDA optional packages API is defined in JSR 75 for accessing PIM data and local file systems. Here we present a mobile computing architecture for multi-platform adaptation. The architecture is a practical and effective way to handle the multi-platform issue in mobile computing. The development of mobile application system has proved the usability of this architecture. The implementation of the architecture with a

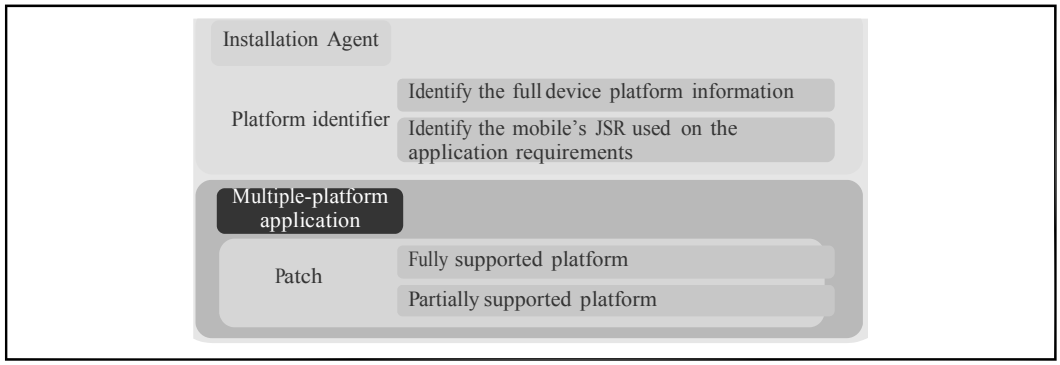
mobile application shows a substantial improvement in mobile platform adaptation for mobile computing. With our architecture and solutions, the mobile application software can be run on over 90% of mobile devices with supported features in the current Canadian market. In the following section, we present the mobile computing architecture for multiple-platform adaptation and the solutions. In the fourth section, we show a mobile application required location data as a scenario to demonstrate multi-platform adaptation.

12.5.2 Detailed Design

Multi-platform Adaptation Architecture

Based on our study and development experiences, we propose mobile computing architecture for multi-platform adaptation. This architecture consists of an installation agent and three adaptable mobile application computing solutions to accommodate different mobile platforms as shown in Figure 1. To implement the multi-platform adaptation architecture requires MIDP2 and CLDC1.1ready mobile device.

With the multi-platform adaptation architecture, a mobile user needs to go to a URL with the mobile web-browser and to download the installation agent, a J2ME program that can run on any MIDP2 and CLDC1.1 ready mobile device. After installing and running the installation agent program, the mobile device's platform information will be automatically sent to the installation server. In the server, the mobile device's platform will be assessed with its platform information based on the running environment requirements of the mobile application software. The assessment will identify the mobile device's platform as a fully supported platform or a partially supported platform. According to the assessment results, necessary patch programs and matching mobile application software will be provided for the particular mobile device's platform. Then the mobile user will be prompted to download the patch programs and mobile application software to achieve the multi-platform adaptation in mobile computing.



12.5.3 The Implementation

Mobile Platform Identification

In the multi-platform adaptation architecture, firstly, the installation agent is a J2ME program that can run on any MIDP2 and CLDC 1.1 ready mobile device without other Java APIs defined by optional Java Specification Requests (JSR). It detects a mobile device's operating system and other platform information and returns the detected information automatically to the installation server. Secondly, a platform identifier on the server will assess the information to identify the mobile device's platform and to find out the deficiency of the mobile device's Java Running Environment for running the mobile application software. The Installation agent workflow and its pseudo code are shown in Figure 12.4.

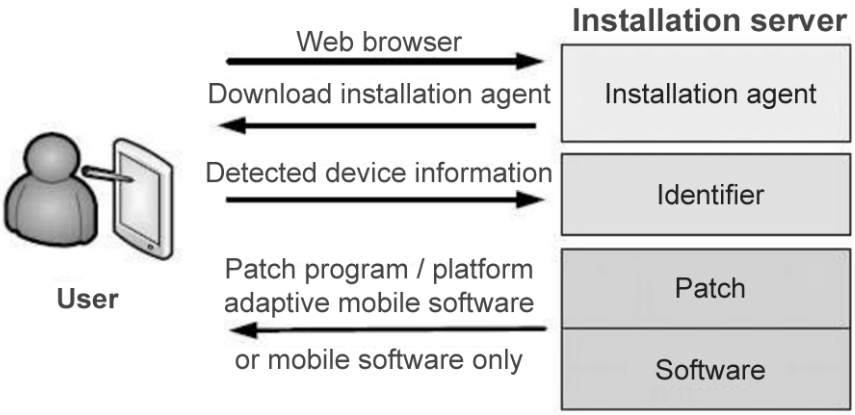


Figure 12.4

Mobile Platform Adaptation

The platform identifier on the server assesses the identified mobile platform and compares with the running environment requirement of the mobile application software. Then, the platform identifier categorises the mobile platform as fully supported platform or partially supported platform. A fully supported platform is fully compatible with the running environment required by the mobile application software. A partially supported platform is lack of some APIs defined by the JSRs required by the mobile application software in addition to MIDP2. According to identified platform, the platform identifier provides the necessary patch to the mobile platform and prepares adapted mobile application software to achieve the mobile platform adaptation. The detail solutions are shown in Table 12.1.

Table 12.1

Platform Category	Platform Specification	Solutions
A. **Fully supported platform**	MIDP2/CLDC1.1 Full required JSR APIs supported	(1) No patch needed (2) Generic mobile application software
B. **Partially supported platform**	MIDP2/CLDC1.1 Partial required JSR APIs supported	(1) Patch with customized running environment (2) Generic mobile application software

For category A, because the identified mobile platform has the fully supported running environment for the mobile application software, the platform identifier will only make generic mobile application software available to be downloaded and installed on the mobile device.

Generally speaking, a series of patch programs have to be developed in C/C++ language to deal with local device and compiled against the popular mobile operating systems in the market. Besides, a series of mobile application software have to be developed in related to each patch running environment. According to the identified absent APIs, the platform identifier will make the exact patch program(s) and software available to be downloaded and installed on the mobile device. The patch program(s) will upgrade the mobile running environment to fully support the platform adapted mobile application software. However, this will need a large amount of available patch programs and mobile application software to create multiple combinations for the platform adaptation. The amount and the combinations will increase exponentially with the number of additional JSRs. Therefore, the reasonable solution for category B is proposed as follows. Because the identified mobile platform does not have the application required APIs except MIDP2, a patch program has to be developed to upgrade the mobile platform in order to fully support the generic mobile application software. The platform identifier will make the patch program and generic

mobile application software available to be downloaded and installed on the mobile device. To implement the multi-platform adaptation architecture, except to develop the architecture required installation agent and platform identifier, the solutions required development include one patch APIs for each mobile operating system to upgrade MIDP2 running environment and only one platform adapted mobile application software.

SUMMARY

- In its most generic sense a voice portal can be defined as "speech enabled access to Web based information". In other words, a voice portal provides telephone users with a natural language interface to access and retrieve Web content. An Internet browser can provide Web access from a computer but not from a telephone. A voice portal is a way to do that. Now in the present time there is competition in the telecommunication industry and different companies provide different services to our customers. In this age of significant telecommunications competition, mobile network operators continuously seek new and innovative ways to create differentiation and increase profits. The best way to achieve this is through delivery of highly personalised services.

KEY TERMS

PAL Phase Alternating Line

PC Personal Computer

MT Mobile Terminal

NAV Net Allocation Vector

SELF-TEST

1. What are the services for the next generation networks?

2. Define in short about the: GIS, Location management function and location based information.

3. Write short notes on the GPS simulator.

4. Differentiate between load scalability and geographic scalability.

MODULE FEEDBACK

MAKLUM BALAS MODUL

If you have any comment or feedback, you are welcome to:

1. E-mail your comment or feedback to modulefeedback@oum.edu.my

OR

2. Fill in the Print Module online evaluation form available on myVLE.

Thank you.

Centre for Instructional Design and Technology
(*Pusat Reka Bentuk Pengajaran dan Teknologi*)

Tel No.: 03-27732578

Fax No.: 03-26978702