UNIT 8

MOBILE APPLICATION LANGUGES- XML, JAVA, J2ME AND JAVA CARD, MOBILE OPERATING SYSTEMS

UNIT 8 : SYLLABUS

MOBILE APPLICATION LANGUAGES – XML, JAVA, J2ME AND JAVACARD, MOBILE OPERATING SYSTEMS: Introduction, XML, JAVA, Java 2 Micro Edition (J2ME), JavaCard.
Operating System, PalmOS, Windows CE, Symbian OS, Linux for Mobile Devices.

7 Hours

8.1 Introduction

Software processes the data and information. Software is generated by a set of **(a)** statements, functions, service routines, threads, objects and classes written in a high-level language, **(b)** codes using a framework or **(c)** codes using a development tool.

In addition to phone which includes SMS, MMS, other additions to teleservices and various supplementary services, there are many mobile applications. A mobile application can be developed through any one of the following three approaches:

- ✓ Using a language like Java that complies and runs on diversified operating systems and hardware platforms.
- ✓ Using a language supported by the operating system which provides a development platform such that compiled program can run on the hardware supported by that OS.
- ✓ Using common framework and OS platform which supports different languages.

8.2 XML

- XML (extensible markup language) is a derivative of SGML (standard generalised markup language). Extensible means that the special instances of the tag-based languages.
- An XML- based language uses the extensible property of XML to define the standardized set of instances of the tags, attributes and their representation and behaviour and other characteristics. Example: SyncML, Funambol, SMIL.
- An XML or XML-based language not only encapsulates the data and the metadata but can represent a behaviour or set of actions.
- XML document can have a text with the tags. A tag in the document specifies the meaning of the text encapsulates within the start and corresponding end tag.
- XML document can represent a database with tags and a pair of start and end tags in the document specifies the start and end of a record in the database which hence can be used to retrieve the specific record.

- An XML document can make non-textual use of text. A tag can represent a command to process the data using the command name within the pair of start and end tags in the document text.
- A tag along with its attributes can specify the command , source files and data to process the command

Below is the example of an XML document which can be used as contacts in mobile smartphones.
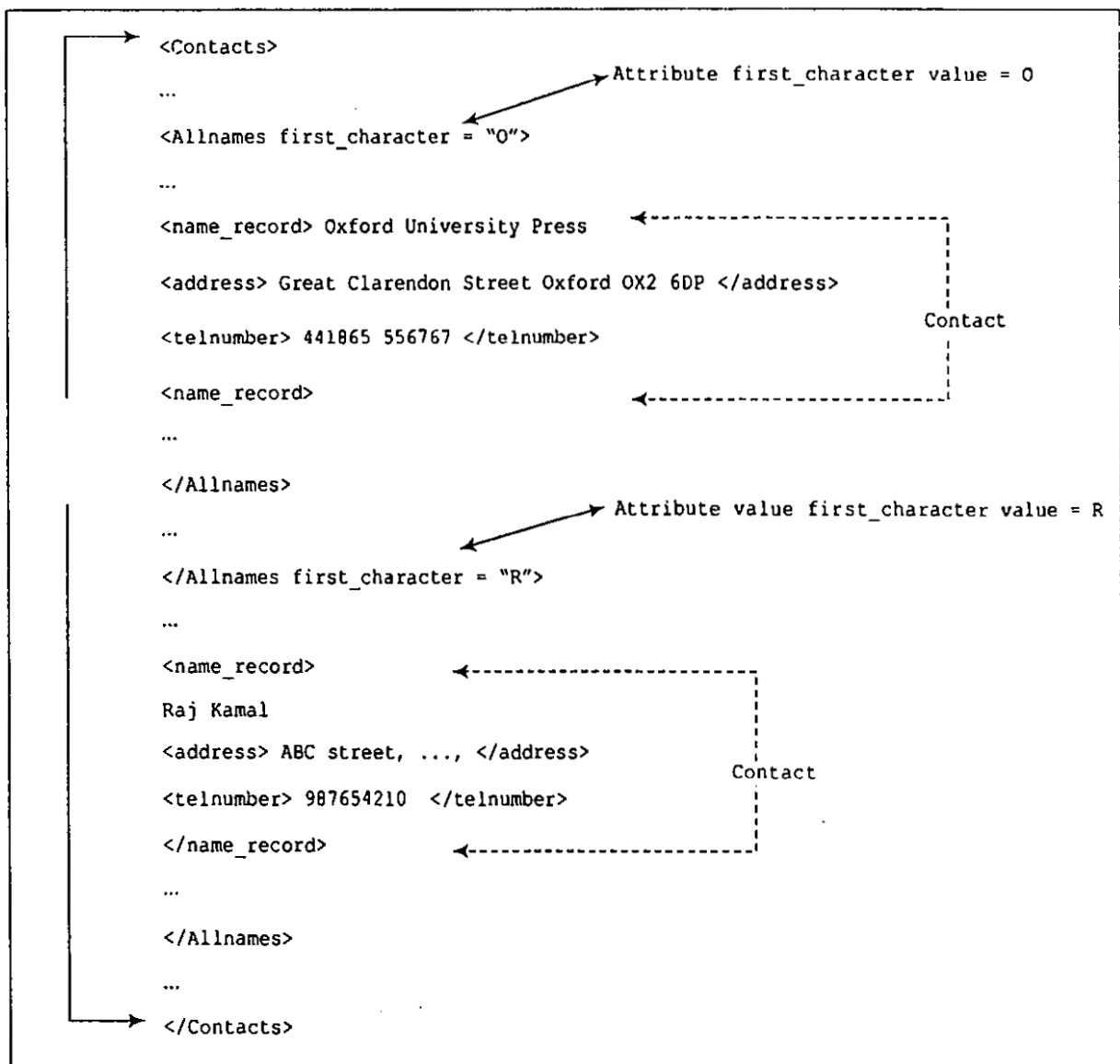
```
<Contacts>
...                                    → Attribute first_character value = O
<Allnames first_character = "O">
...
<name_record> Oxford University Press    ◄----------------------┐
<address> Great Clarendon Street Oxford OX2 6DP </address>      │
                                                        Contact  │
<telnumber> 441865 556767 </telnumber>                          │
<name_record>                            ◄----------------------┘
...
</Allnames>
...                                    → Attribute value first_character value = R
</Allnames first_character = "R">
...
<name_record>                            ◄----------------------┐
Raj Kamal                                                       │
<address> ABC street, ..., </address>                  Contact  │
<telnumber> 987654210  </telnumber>                             │
</name_record>                           ◄----------------------┘
...
</Allnames>
...
</Contacts>
```

Fig. 13.1 Example of XML document which can be used as Contacts in a mobile Smartphone

8.2.1 Document Type definition and Parsing of XML Documents

- An XML document is given by Document Type Definition (DTD) either internally or in a separate DTD file. The functions of DTD are as follows:
    - ✓ To enable validation of a document.
    - ✓ To specify which document structures can be used for authoring of the document.
    - ✓ To specify which structures a parser must handle.

PARSER:

- Parser is a software written in JAVA or any other language. Its functions are to validate and parse the tags, their attributes, and the text within each pair of start and end tags and to enable extraction of specified information either from the document file or using document and DTD files.
- A parser first validates an XML document and then handles the specified document structures. Validation means investigating about (i) whether the document contains a root element with the same name as of DTD (ii) whether it contains the header information for the version, encoding, and reference to other files (iii) whether it contains DTD with declaration of the mark-ups in the document or in a linked external DTD document.
- The XML document has *.xml* extension and the external DTD document has file extension *.dtd*.

8.2.2 Models of an XML Document and Parsing

There are two models of an XML document and correspondingly two types of parsers. They are as follows:

- ➢ **SAX Model of an XML Document**
    - ✓ SAX (Simple API for XML) model is one in which each set of elements within tags is independent and need not be considered as tree-like structure.
    - ✓ SAX model parser does serial access of the XML data in the API and is modelled as a stream parser, with an event-driven API.
    - ✓ An advantage of XML SAX parser is that the entire document need not be first parsed thoroughly and all the needed data need not be extracted.
    - ✓ SAX parsing and processing is fast.

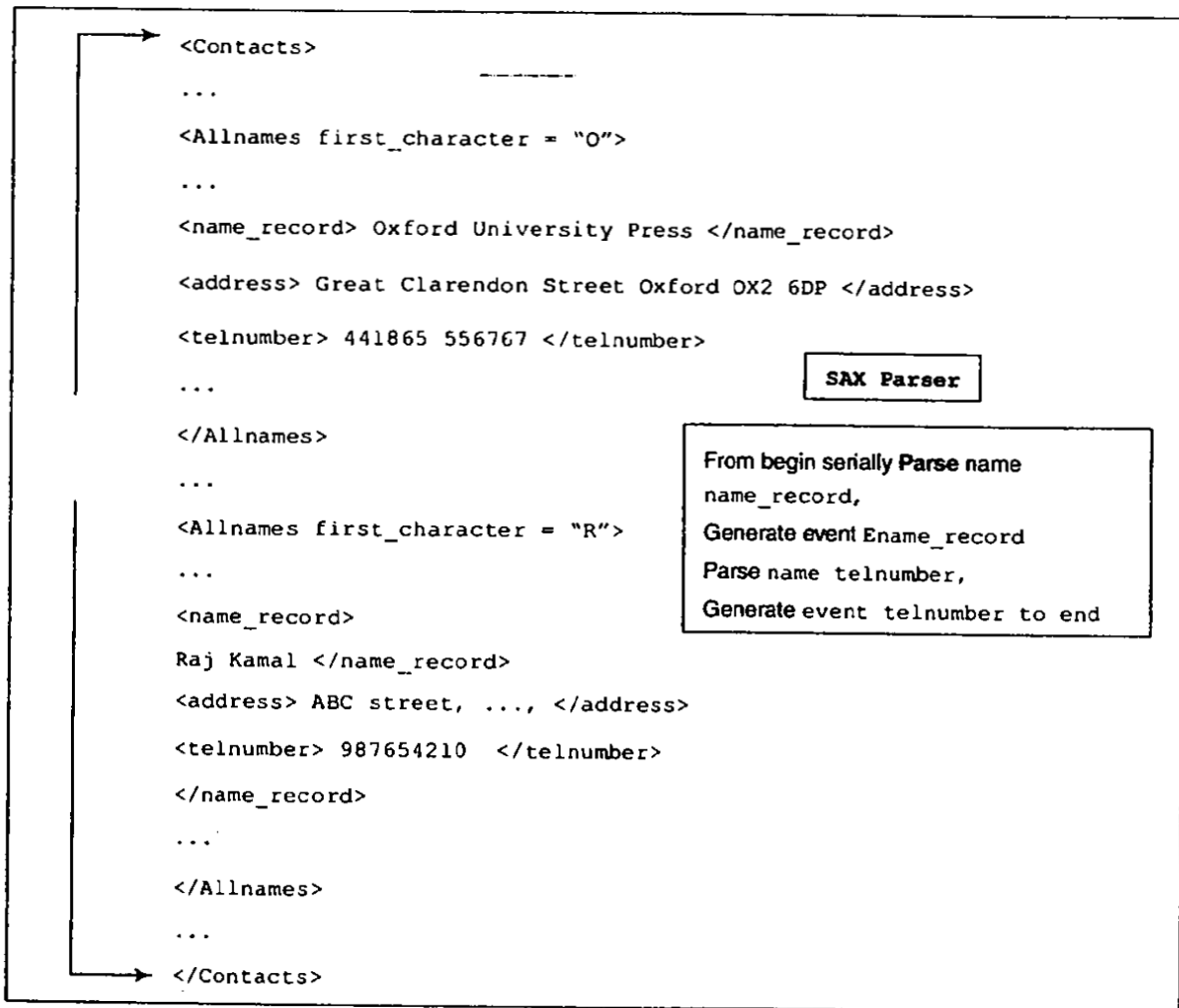   Below example shows SAX model of parsing of an XML document corresponding to Contacts.

```
<Contacts>
...
<Allnames first_character = "O">
...
<name_record> Oxford University Press </name_record>
<address> Great Clarendon Street Oxford OX2 6DP </address>
<telnumber> 441865 556767 </telnumber>
...
</Allnames>
...
<Allnames first_character = "R">
...
<name_record>
Raj Kamal </name_record>
<address> ABC street, ..., </address>
<telnumber> 987654210  </telnumber>
</name_record>
...
</Allnames>
...
</Contacts>
```

SAX Parser

From begin serially **Parse** name
name_record,
Generate event Ename_record
Parse name telnumber,
Generate event telnumber to end

Fig. 13.2 SAX parser of XML document

➢ **DOM Model of an XML Document**
  ✓ DOM (Document Object Model) is the one in which each set of elements is dependent and derives from a root element and whole document forms a tree-like structure.
  ✓ DOM model document is always hierarchically arranged.
  ✓ The whole document is parsed initially to create a hash table of keys and corresponding values for each key.
  ✓ An advantage of DOM parsing is that the structure is well-defined and the same parser can be used for parsing all XML documents and later the interpreter or processing program is able to extract the desired information by simply using the keys.
  ✓ Extracting and parsing a program for a DOM model document and parser is not complex.
    Below example shows SAX model of parsing of an XML document corresponding to Contacts.
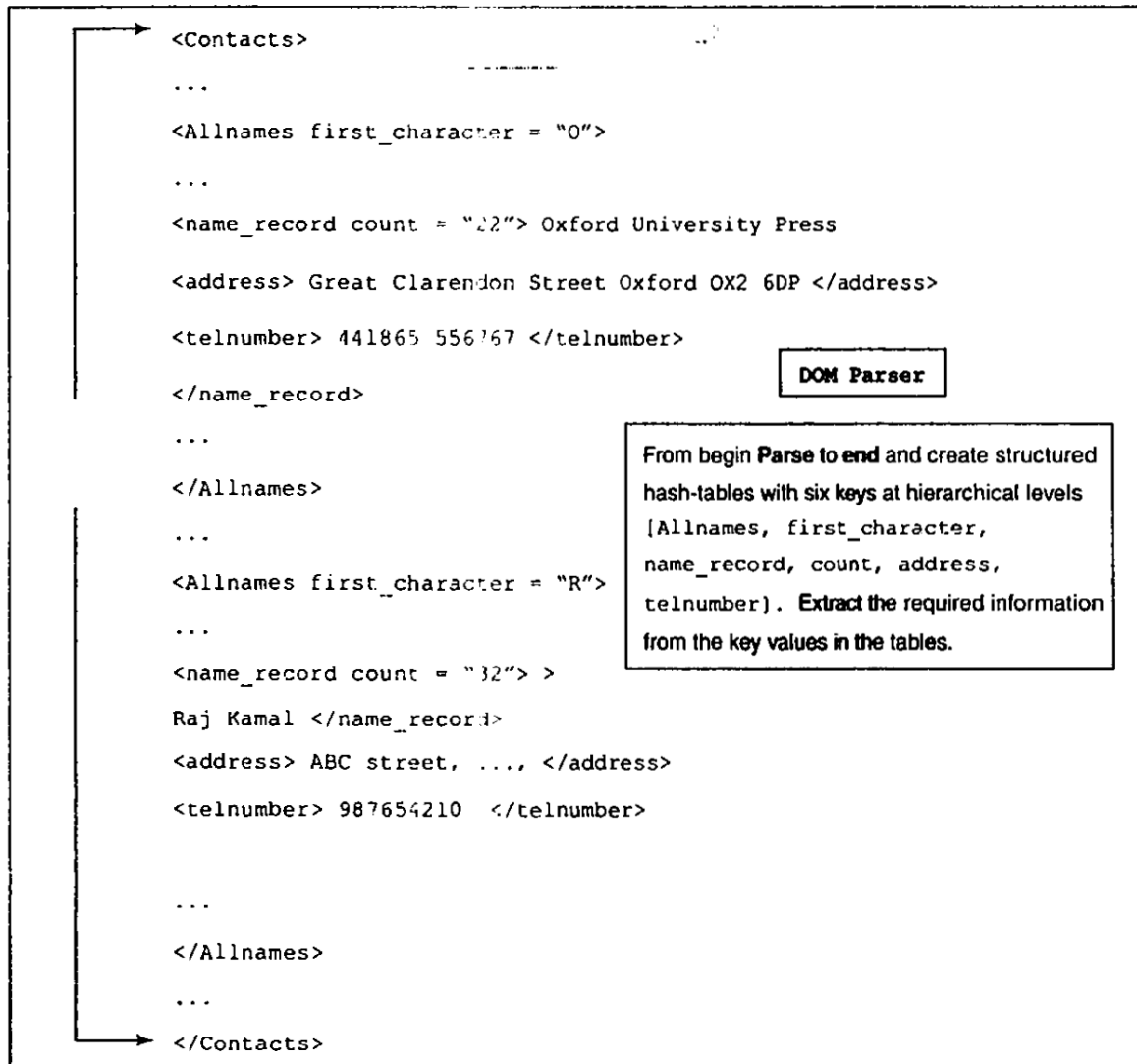
```
<Contacts>

...

<Allnames first_character = "O">

...

<name_record count = "22"> Oxford University Press

<address> Great Clarendon Street Oxford OX2 6DP </address>

<telnumber> 441865 556767 </telnumber>
                                              ┌─────────────┐
</name_record>                                │  DOM Parser │
                                              └─────────────┘
...                            ┌────────────────────────────────────────┐
                               │ From begin Parse to end and create structured
</Allnames>                    │ hash-tables with six keys at hierarchical levels
                               │ [Allnames, first_character,
...                            │ name_record, count, address,
                               │ telnumber]. Extract the required information
<Allnames first_character = "R"> │ from the key values in the tables.
...                            └────────────────────────────────────────┘
<name_record count = "32"> >

Raj Kamal </name_record>

<address> ABC street, ..., </address>

<telnumber> 987654210  </telnumber>


...

</Allnames>

...

</Contacts>
```

Fig. 13.3 DOM parser of XML document

8.2.3 Applications of Parsed Information and Data

An application uses the extracted output information and data from the parser after a further processing using a programming language. The parsed data is interpreted at the application. An application can also use the metadata in XML document to speed up and enrich searching for the resources, organising the information and managing the data. An application in addition to XML data can thus also use the metadata during the processing. The following are the examples of XML usage:

- ✓ As a language for preparing a textual document for platform-independent application data.
- ✓ As a language for representation behaviour of or action by a tag and its attributes.
- ✓ As an integrator of two diverse platforms for running an application.
- ✓ As a client application.
- ✓ Used by server to push the data to devices.
- ✓ Used by WAP protocol for presentation of data to client using an XML browser.

- ✓ Used internally for specifying the information in an application or framework.
- ✓ Used by HTML browser after translating XML information into HTML information using a technology called XSLT (Extensible style-sheet language transformation).
- ✓ XHTML-MP format of XML can be used as HTML web pages with portability and extensibility in mobile devices.

8.2.4 XML-based Standards and Formats for Applications

- ✓ XML is an extensible language. The tags and attributes, and markup language format can be standardized for applications.
- ✓ XForms is a form of in XML format which specifies a data processing model for XML data and UIs for the XML data.
- ✓ XForms controls are *<textarea>, <input>,* and *<secret>* and a string data type is attached with these fields. Other XForms controls are *<selectOne>, <choices>, <item>,* and *<itemset>.*The attributes of XForms inout control are *ref* and *xml:lang.xml:lang ="en",accessKey, navIndex* and *class.* The elements of XForms are *<alert>, <caption>, <hint>* and *<extension>.* There are client-side as well as server-side validations before processing of the XForms.
- ✓ XML user Interface (XUI) is a framework in XML and Java for mobile and mobile handheld device applications.
- ✓ WBXML (WAP binary markup language) is a format in which WAP presents XML document as binary numbers.
- ✓ VoiceXML (VXML) 2.0 is a version of an XML-based language for interactive talking between a human being and a computer.
- ✓ SSML (Speech synthesis markup language) is used in TTS (text to speech converter). It is for synthesizing the speech by interpretation of a text by synthesizer. It creates audio book.
- ✓ Speech recognition grammar specification (SRGS) is used in SSML. It lets speech recognizer to define the pattern of sentences.
- ✓ Call control extensible markup language (CCXML) is a standard for an XML- based languagewhich can be used independently or with VXML for telephony support. CCXML commands the browser to handle the calls of the voice channel.

8.3 JAVA

- • Java is a language which compiles into byte codes and the code runs on virtual machine. Therefore these can run on any operating system and hardware platform on which JVM is installed.
- • The JVM provides execution engine and native interfaces to run the codes on the given OS and hardware. Java is the most popular language used for mobile computing. Java 2 standard Edition (J2SE) is standard Edition.
- • Java is used to program the application threads, applets, servlets and aglets.
  - ✓ Application threads – these are the programs of the application, each one of which is assigned a priority and runs concurrently.

- ✓ Servlets – these are the programs running on a web server. The life cycle of a servlet consists of the sequential step – init( ), service( ) and destroy( ).
- ✓ Applets – these are the programs running on a client. The life cycle of a applet consists of the sequential step – init( ), start( ), stop( ) and destroy( ).
- ✓ Aglet – it is the mobile agents which run on host. The life cycle of an aglet consists of the sequential steps – creation, dispatch, activation, deactivation and disposition.
- ✓ A class is a construct in Java or C++, which is used to define a common set of variables, fields, and methods and whole instances give the objects.
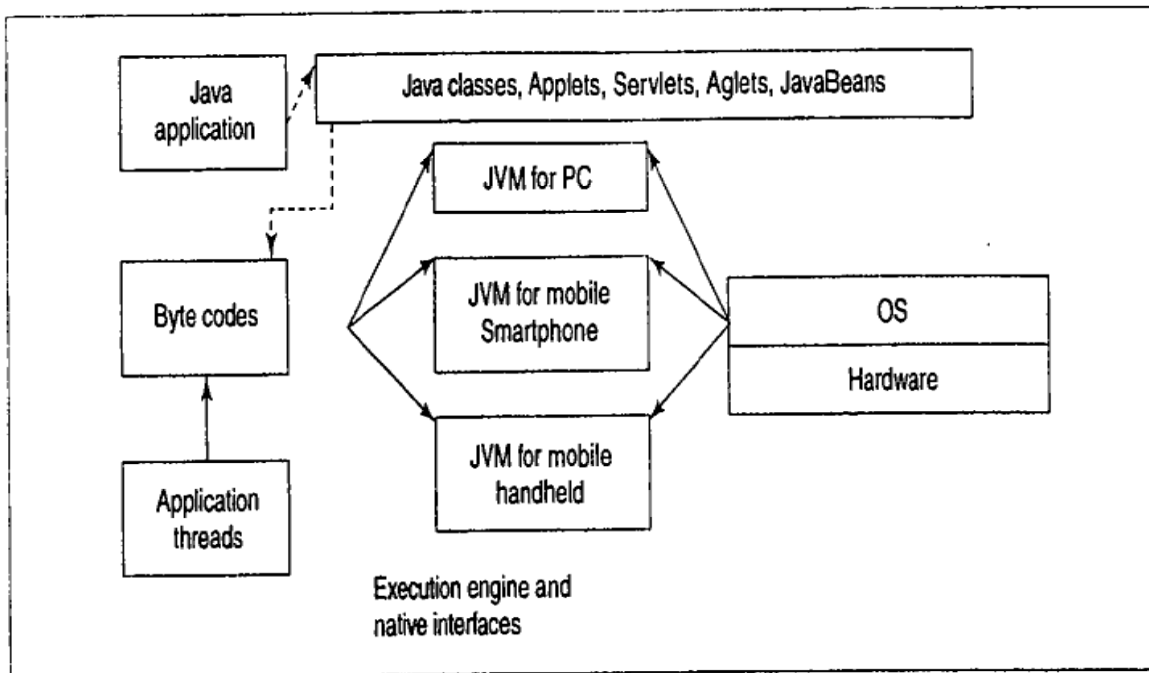
Fig. 13.4 Java application, classes, and application threads compiled as byte bytes and JVMs for platform-independent execution

- ➢ **Characteristic features**
    - ✓ Java is *object-oriented language.* Each class is a set of codes that has number of members – fields, methods, etc., so that objects can be created as instances of class.
    - ✓ *Platform-independence* is the most important characteristics of Java, which means that the program code compiled in Java are independent of the CPU and the OS used in the system.
    - ✓ *Robustness* is one of the characteristics of Java. All references to memory, freeing of the memory, de-allocation, and validation of object types and arrays indices are done internally at compile and run time.
    - ✓ *Standard APIs* enforcement is another characteristics of Java. APIs help Java program to connect an application to a program, database, distributed object, or server developed on other platforms.

> **Classes and Beans**

Java has packages consisting of the classes and interfaces. The packages help in fast development of the code. Some of the widely used packages are:

- java.lang – package for fundamental classes of Java
- java.io – package for input, output and file access.
- java.math – package for classes for the mathematical methods.
- java.awt – package for Java foundation classes for creating GUIs.
- java.swing – package for Java swing classes for creating lightweight GUIs.
- java.net– package for network related classes like TCP/IP, UDP, HTTP and FTP.
- java.security – package for classes in Java security framework.
- java.sound – package for audio processing.
- java.io.Serializable –it has serializable interfaces.

> **Java 2 Enterprises Edition (J2EE)**

- The Java Enterprise edition (J2EE) is used for web- and enterprise-server-based programming of the applications. It is used for distribution across the multiple computing tiers.

- Enterprise Java bean (EJB) is component at a server in an enterprise application. It is managed such that it leads to a modular design. EJBs enable persistency of the application components. An application server manages the security, scalability, transactions and concurrent execution of the applications deploying the EJBs.

- J2EE supports the API specifications for the following:
  ✓ XML
  ✓ Web services
  ✓ Email
  ✓ RMI (Remote method Invocation)
  ✓ JMS (Java messaging service)
  ✓ JDBC (Java database Connectivity)
  ✓ JTA (Java transaction API)
  ✓ Servlets
  ✓ JNDI (Java Naming and Directory Interface)

Below fig shows an N-tier architecture in which a client device connects to the enterprise databases.
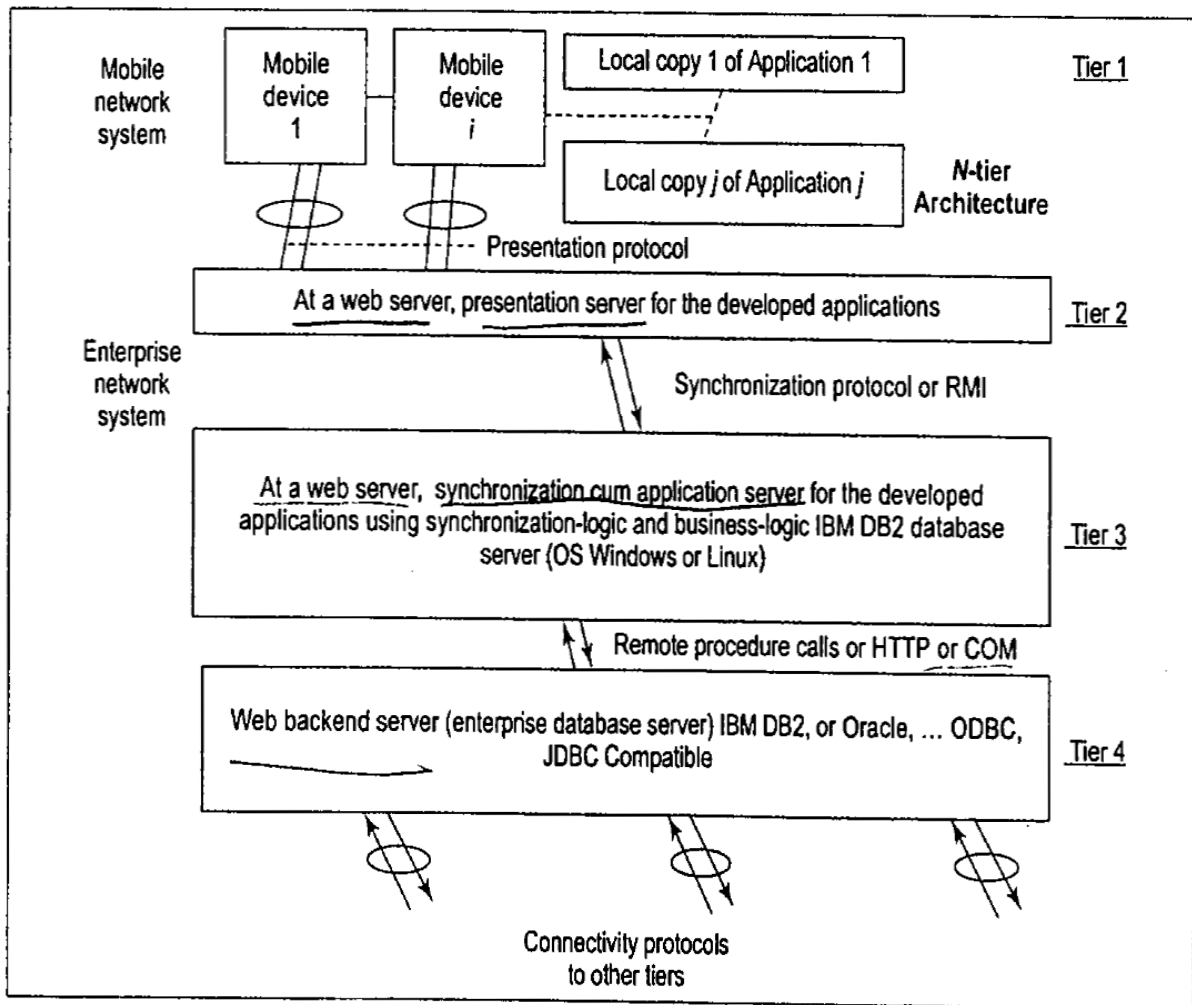
Fig. 13.5 *N*-tier architecture in which a client-device connects to the enterprise databases

- A container has EJB as Session Beans and Entity Beans. It provides transactions, security, scalability, pooling of resources, and concurrent execution of the application. Resources are pooled by transactions between Entity Beans and backend servers for the resource at the enterprise tier. Entity Beans connect to enterprise tier which can have a database server. Transactions maintain persistency of data at enterprise server.

8.4 Java 2 Micro Edition (J2ME)

- J2ME is a micro edition of J2SE which provides for configuring the run time environment.
- J2ME is a set of Java APIs which require small memory while developing Java applications. It is also a platform for development of mobile games.
- Windows mobile devices do not support J2ME or Java based Virtual machine. Java platform binary implementation and virtual implementation are done by another source not from Windows.

> **Profiles**
> - ✓ A profile means a standardized agreed-upon subset and interpretation of a specification. A profile may also mean a specification for a set of configuration settings and other data which are used in the APIs for a device, user, or group of devices.
> - ✓ Mobile information device profile (MIDP) is the profile for mobile devices with small screen option for GUIs, wireless connectivity, and greater than 128kB flash memory.
> - ✓ A MIDlet is a J2ME application (similar to applet) for embedded devices which runs with MIDP. MIDlet are programmed to run games and phone applications. Also they are compiled once and are platform-independent. Below table shows the MIDP source packages and sets of Java class libraries.

Source package	Sets of Java class libraries
java.lang	Standard java types and classes for string, Integer, Math, Thread, Security, and Exception.
java.io	Standard java types and classes for input and output streams.
java.microedition.lcdui	LCDUI for mobile devices with no internet connectivity.
java.util	For classes such as Timers, Calendars, dates, hashtables, vectors and others.
java.microedition.rms	A record management system(RMS) API to retrieve and save data and limited querying capability.
java.microedition.pim	Personal Information management API, access the device's address book.
java.microedition.pki	Secure connections authenticate APIs.
java.microedition.messaging	Wireless messaging APIs used when sending SMS and MMS messages.

> - ✓ MIDP enables inter-MIDlet communication which in turn allows querying of device capabilities required to be done when a server service is to discover the device services. Development tools are used to develop MIDP applications. Example : NetBeans.
> - ✓ Information Module Profile (IMP) is for embedded devices. An IMlet is an application created from IMP APIs. IMlet is inherited from MIDlet.

> **Configurations**
> - ✓ A configuration is a subset of profile.
> - ✓ CLDC (connected limited device configuration) is a configuration for limited connected devices. CLDC defines a base set of APIs and VM for the resource-constrained mobile phones or handheld computers. CLDC configuration is a subset of MIDP used for developing Java applications.

✓ CDC (Connected device Configuration) is a configuration for connected devices. CDC is a Java framework for developing an application that can be shared in networked devices. CDC provides a J2ME framework for applications which run on wirelessly connected devices and APIs for HTTP.

Below table gives source profiles, packages for the configurations of CLDC and CDC and the required Virtual Machine.

Features	CLDC	CDC
Profiles and source packages	A configuration for the MIDP, which does not provide for the applet, awt, beans, math, net, rmi, security, sql, and text packages in the java.lang.	A configuration for the foundation and Personal profiles, TV profiles etc which includes class inherited from a limited number of classes at net, security, io, reflect, security.cert, text, text.resources, util, jar, and zip packages.
Virtual machine	KVM	CVM (Coherent Virtual machine) for multi-protocol and multi-threading support.

8.5 JavaCard

- JavaCard (Java for card) is a micro-edition for Java for the cards. JavaCard is a limited-memory sized edition for cards, labels, tokens, and similar devices which have limited memory as well as processing capability.
- JavaCard 2.2.2 provides interoperability for cards and APIs for highly memory-efficient applications. It has multiple communication interfaces for inter card-host contact/contactless communication.
- JavaCard framework (javacard.framework) provides the library functions, Card interfaces, PIN and APIs for Card applet.JCardSystem is a class in JavaCard. It has a method called TransientArray( ). The method creates a transient array and it persists till power is down.
- Card profile is a set of limited class libraries. JavaCard profile is smartcard profile which has a separate virtual machine called card VM. Card VM has an instruction set for a subset of Java language. It installs applets and libraries into JavaCard-based devices.
- CardVM is a virtual machine which has no char, double precision, or single precision and no floating point mathematical operations support. It has limited exception handling, no object clones, no String class libraries, no automatic garbage collection, no SecurityManager class libraries and no multi-threading.

- The card applet creates card-specific byte code on compilation. Java applet has lifecycle starting from init( ),start( ),stop( ) and destroy( ). Objects created in card applet are persistent.
- JCRE (JavaCard runtime environment) interprets the card byte codes and implements them using JavaCard Virtual Machine. It does not support inter- communication between different card applets and provides runtime support to the various different services.

8.5 Operating System

An operating system (OS) is the master control program in a system that manages all software and hardware resources. It controls, allocates, frees, and modifies the memory by increasing or decreasing it. An OS has utility programs like file manager and configuration of OS.

➢ **Process, Task, Thread, ISR and IST**
   ✓ A *process* is a program unit which runs when scheduled to do so by OS and each state of which is controlled by OS. A process can be in many states − created, active, running, suspended, pending for specified time interval, and pending for want of specific communication from another process through the OS. It call a function but cannot call another process directly.
   ✓ A *task* is a application process which runs according to its schedule set by the OS and each state of which is controlled by OS.
   ✓ A *thread* is an application process unit which runs as scheduled by the OS, each state of which is controlled by OS, and which runs as light-weight process. Light-weight means it does not depend on the certain system resources.
   ✓ A *interrupt service routine (ISR)* is a program unit (functions, method or subroutine) which runs when a hardware or software event occurs and running of which can be masked and can be prioritized by assigning a priority. Example for hardware event are division by zero, finishing of DMA transfer, data abort etc. Software event can be exception or illegal operation code provided to CPU.
   ✓ An *Interrupt service thread (IST)* is a special type of ISR or ISR unit which initiates and runs on an event and which can be prioritised by assigning a priority. The type of IST depends on the specific OS. Example: For Windows CE IST, FIFO is the best example.
   ✓ *Page* is a unit memory which can load from a program stored in a hard drive or from any other storage device to the programming memory, RAM, before execution the program. It is a contiguous memory address block.
   ✓ A *page table* is used for address mapping. The pages of memory are spread over the memory-address space leading to fragmentation of codes and data in physical memory space. A page table provides the mapping of fragmented physical memory pages with the pages of the virtual addresses which are the memory addresses assumed by the programmer.

➤ **Mobile Operating System**

✓ Mobile OS is an OS which enables running of application tasks taking into account of various hardware constraints like CPU speed, memory, battery file, display and size of input device.

✓ An application consists of application task. The OS provides the functions used for scheduling the multiple tasks in a system.

✓ Mobile Os also provides the configurable libraries for the GUI in the device. It provides for multi-channel and multi-model user interfaces.

8.6 PalmOS

PalmOS is one of the most popular OS for handheld device. It offers high performance. Below table shows the highlight features of PalmOS.

Table 14.1 Highlights of PalmOS

Property	Description
Strength	Efficient running
OS basic functions	• Single process (no multi-processing and multi-threading)
	• Compiled for a specific set of hardware, performance very finely tuned
	• Memory space partitioned into program memory and multiple storage heaps for data and applications
	• A file is in format of a database
	• IP-based network connectivity and WiFi (in later version only)
	• Integration to cellular GSM/CDMA phone
Memory	• OS memory requirement is ~16 MB
	• 256 MB internal flash memory
	• Memory address space—256 MB in a card

(Contd)

GUIs	APIs for the buttons, menus, scroll bar, dialogs, forms, and tables using HTML markup language
User interface (GUI) display resolution	Generally wide screen—160 × 160 pixels with optimized layout of desktop programs displayed on screen and 256 colour touch screen (higher resolution support depends on new versions)
Software	• Simple APIs compared to Windows CE desktop for Windows and Mac both and other essential software • Simple APIs compared to Windows CE • PIM and e-mail and Internet , address book, data book for task-to-do and organizationing, memo pad, SMTP (simple mail transfer protocol) e-mail download, offline creation and sending of POP3 (post office protocol 3) e-mail, Internet browsing functions using Blazer (a browser, for handhelds), Windows organizer, and PDA (personal digital assistant) • GUI development support on C/C++, Java platforms using Palm SDK and for Java application using J2ME • Query development support—Palm query applications (PQA) written using HTML and are ported at Palm device • Support to client-side application and GUI development support on C/C++ platform using Palm SDK and for Java application using J2ME • Support to multimedia applications such as playing music (Palm Tungsten) wireless communications
Desktop program examples	SMS, Address, Card-Info, HotSync, To-Do-List, SMS, Security, Date Book/Calendar, Calc, Welcome, and Clock
Ports support	Serial and infrared ports for communication with mobile phones and external modems and for synchronizing a PC personal area computer using HotSync after resolving the conflicts in different versions of files during data exchange
HotSync synchronization	HotSync synchronizes a personal area computer PC through a serial port or infrared port (HotSync)
Personal area computerPC to handheld connection and file transfers	IrDA or serial device mounted on a cradle connects to computerPCs through IR or serial port. (A cradle is an attachment on which the handheld device can rest near a PC and connects to the PC via a USB or infrared.) A device is assumed as a new flash drive of a PC—the computer and HotSync facilitates drag and drop of files from device to PC and vice versa
Cards support	MMC (multimedia card), SD (secure digital) memory card, and SDIO (secure digital input/output) memory card

(Contd)

Third party common applications support No support to multitasking	Examples—games, travel and flight planner, calculators, graphic drawings, and preparing slide shows • Instead of multi-tasking, PalmOS provides for running a sub-application from within an application • Not an ideal platform for running multimedia applications because due to PalmOS is not for designing real-time systems
No adaptability	• Does not offer much expandability • Inability to adapt to different sorts of hardware may also be considered a limitation for this operating system

Below fig shows the architectural layers of PalmOS.
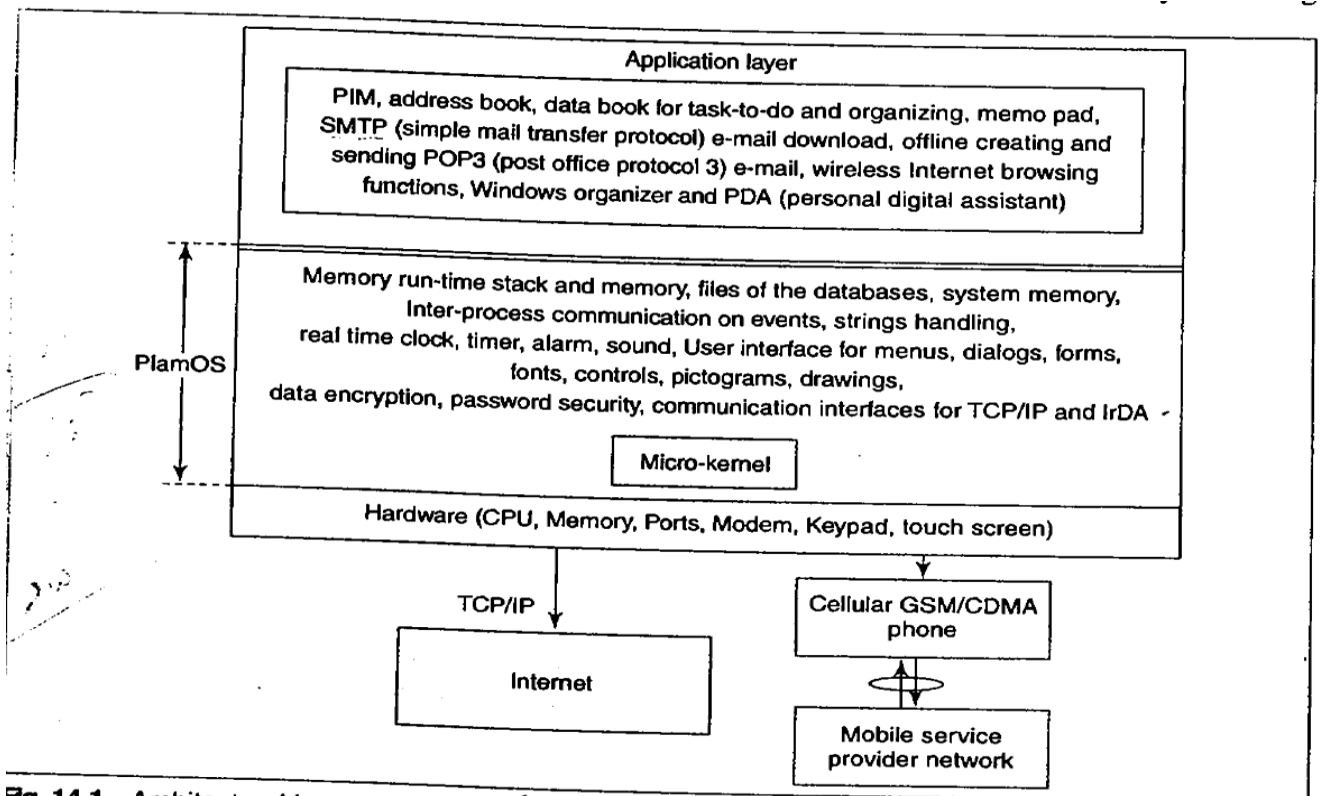
Fig. 14.1 Architectural layers of PalmOS with OS layer in between the applications and hardware

The lowest level in OS is a Kernel. This layer directly interfaces the assembler, firmware and hardware. PalmOS has a micro-kernel.

➢ **Memory Management**

PalmOS assumes that there is a 256MB memory card. The card can have RAM, ROM, and flash memories. A memory has logical hexadecimal addresses from 00000_h to $3FFFF_h$. Ram is used for stacks of processes and for the global variables in the running processes of the applications. Below fig shows the functions of the memory manager of the PalmOS.
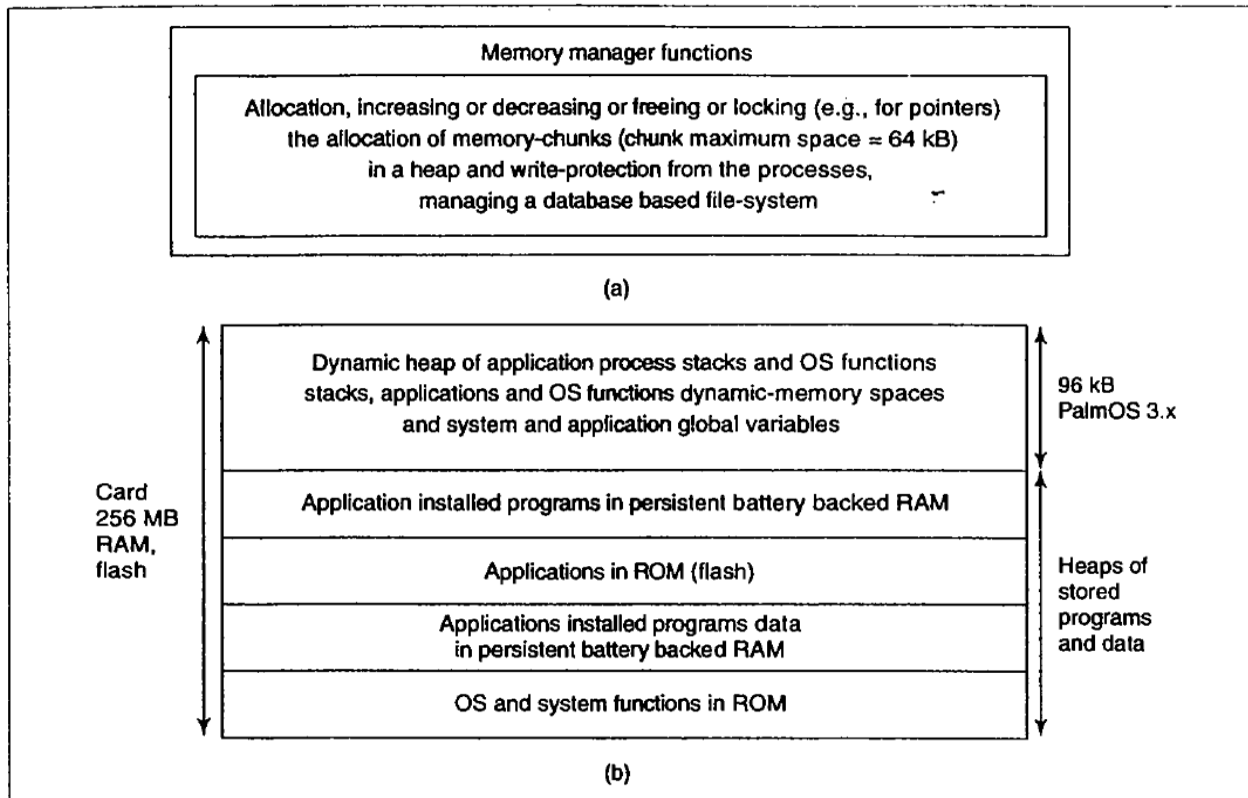

Fig. 14.2 (a) PalmOS memory manager functions, (b) PalmOS memory partitioning

➢ **File Management**
   ✓ PalmOS file manager manages each file as a database which has multiple records and information fields. Each record can have following attributes – protected record, deleted record, locked record and updated record.
   ✓ The info fields of each record have record ID and record attributes. Info fields about the file have – (i) name, (ii) file attributes, (iii) version of application database, (iv) modification number and access counter, and (v) file local ID.
   ✓ The file local ID is a number used to identify the file locally when an application is running. A local file sorting table uses the file local ID to sort the file in the required order.

- ➢ **Communication APIs**
  - ✓ PalmOS provides communication and network APIs for *serial, IrDA, and TCP/IP* communication.
  - ✓ Serial Communication uses a cradle. A Serial Manager (SM) provides interface to the device on cradle. Connection management (CM) protocol, modem manager (MM), or serial link protocol (SLP) interacts with SM to transmit the data to PC. A device receives the data from the other end through serial manager and SLP, MM or CM. SLP has on top of it a packet assembly and disassembly protocol (PADP).
  - ✓ IrDA asynchronous serial or synchronous serial communication uses exchange manager as session layer and IrDA library functions at lower level. Exchange manager enables data interchange directly without HotSync.

- ➢ **Network Library Functions**
  - ✓ PalmOS uses TCP/IP network library functions to send stack to a net protocol stack (NPS) and provides a socket. PalmOS uses HTTP/ HTTPS net library for Internet connectivity.

- ➢ **Application Development Method**
  - ✓ Corresponding to event, there is an event- handler. Application development means defining additional events and coding for the corresponding handlers. An application can be assumed to be divisible into sub-applications a_m to a_{n-1} along with the existing event handlers a_0 to a_{m-1}. Assume that a_i runs on the events e_i and an event e_i is polled at a sleep interval of t_{i-1} in an infinite while loop. Below fig shows how an application program runs on PalmOS platform as number of event-driven sub-applications.
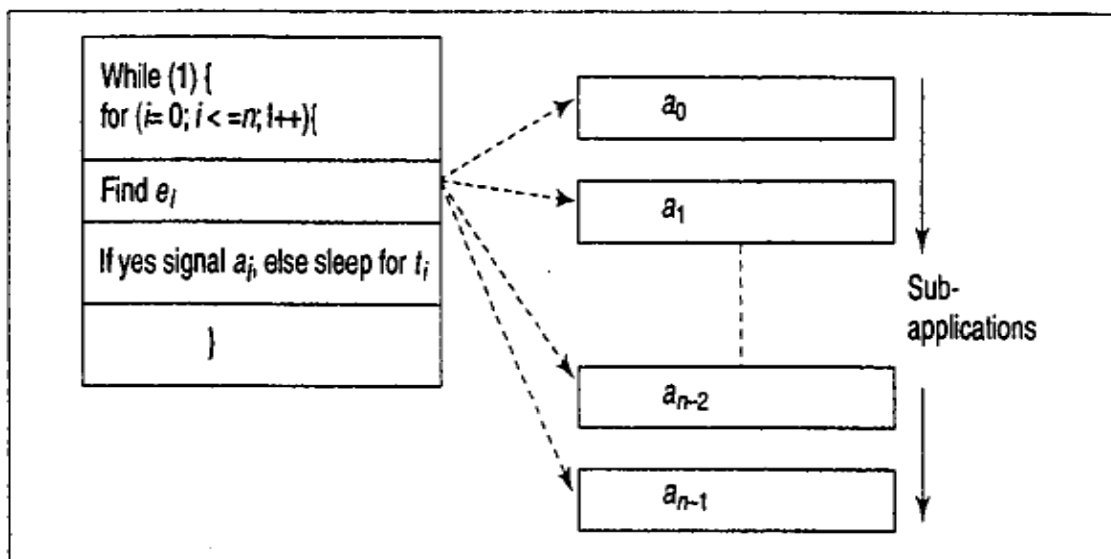

Fig. 14.3 Application program running on PalmOS platform as event-driven number of sub-applications

- ➢ **Application Development Packages**
  - ✓ PalmOS supports development packages – Palm SDK (software development kit) and CDK (conduit development kit). A conduit is a path. A CDK conduit of Palm provides a two-way path for data exchanges and synchronisation between a desktop PC application and the applications running on a device.

8.7 Windows CE

Windows CE is an operating system from Microsoft. It is a component-based, embedded, real-time operating system with the deterministic interrupt latency.

Below table gives the Highlights of Windows CE.

Table 14.2 Highlights of Windows CE

Property	Description
OS	• A 32-bit OS • Customized for each specific hardware and processor in order to fine-tune the performance • Compatible with a variety of processor architectures • Compiled for a specific set of hardware, its performance is very finely tuned • Kernel consisting of source code and hardware abstraction sublayer • Shared source and source code access • Modular/componentized to provide the foundation of several classes of devices and supports addition of features of other components for Windows, DCOM, and COM • Windows CE is real time OS, and thus supports multi-tasking. Windows CE 6.0 provisions for each simultaneously running process to have 2 GB of virtual memory space • Memory requirement is large but scales to the requirement of the device peripherals • Memory space partitioned • Data is in formats of a database or object file • File automatically compresses when stored and decompresses when loaded • Visual C/C++ platform integrates use of web • .NET XML parsing (trimmed version)

(Contd)

Windows Mobile 5.0	• Windows CE 5.0 with a set of specific applications and GUIs/VUIs and for a specific set of processors which are deployed in (i) Smartphone, (ii) handheld PocketPC which features the *digitizer* in the human computer interface (HCI), and (iii) portable media player • PDA with Microsoft Smartphone phone device, touch screen, touchpad, or directional pad
Memory	Minimum footprint of Windows CE is 350 kB (Section 14.3.2) Windows Mobile 5.x—all user data in persistent (flash) memory and RAM to be used only for running applications
GUIs and VUIs	GUI development support using markup language as well as C/C++ language and embedded complex APIs provided in Windows CE. It gives the user a PC-like feel and Windows-like GUIs (window resizing not provided.), VUIs (in PocketPC and automotive PC), buttons, shortcut icons, menus, scroll bar, dialogs, forms, and tables
User interface (GUI) display resolution	High resolution colour/display, touch screen, and stylus keypad with Windows layout of desktop programs displayed on coloured touch screen, built-in microphone for voice recording
Strength	Efficient running of most programs and support for multi-tasking real-time and multimedia applications
Software	Desktop for Windows and other essential software, PIM, Contacts, Task-to-do, Smartphone, and multimedia applications such as playing music. (Refer Fig. 14.4 for application layer software)
Desktop program, tool bar, start menu	• Today calendar, owner, number of messages not read, tasks, and present hour subject. Button and tool bar for task start menu [today calendar, contacts, Internet explorer, messages, phone, pocket MSN, album, MSN messenger, camera, programs, settings, and help], phone mode indicator (on/off), signal strength status, speaker status (on/off), and time • Two context-sensitive soft buttons at the bottom of the screen, which can be mapped to hardware buttons on any specific device
Ports support	USB and infrared port support for communication of a device with mobile phones and for synchronizing a PC using ActiveSync after resolving the conflicts due to different versions of object files during data exchange Bluetooth and/or TCP/IP. WiFi or Ethernet LAN interface
ActiveSync synchronization	Synchronization of mobile device data with PC using a USB, Bluetooth, and PC infrared port

PC to handheld connection and file transfers	USB 2.0 in Windows CE 5.0 PocketPC conform as the USB mass storage class, the storage on device can be accessed, and drag and drop menu can be used from any USB port of PC, which considers the handheld device just another flash drive. A cradle connects to PC
Memory support	External memory stick (strip) (e.g., 2 GB)
Multi-tasking real-time OS feature	Windows CE is real-time OS allowing multi-tasking on handheld devices with defined real-time constraints for each task and a deterministic latency (period within which it will definitely run) Kernel supports 32 simultaneous or concurrent running of—(i) the processes and (ii) threads of a process
Threading feature	Thread is basic unit of computation. A process can have any number of threads. Which can also run simultaneously. A device based on CE 6.0 can run a larger number of complex applications and can run as many as 32,000 simultaneous processes
Inter-process communication	Threads are synchronized by synchronization objects, for example, a *mutex* (semaphore) wait and critical section or wait object for a *signal*
Third party common applications support	Very large support (about 20000) for games, applications, mobile e-commerce, and stock-trading
No support to simultaneous multi-modality	Cooperative running of multi-threading does not support simultaneous multi-modal user interfaces (data by multiple modes, for example, text as well as speech)
Poor adaptability	Adapts to different sorts of hardware limits mainly because of two reasons—(i) compiled for a specific set of hardware for very fine-tuned Windows CE performance, (ii) large parts of OS offered in the form of source code first and then adjusted to the hardware by the manufacturer

ARCHITECTURE:

✓ Architecture of Windows CE consists of Application layer which supports the following APIs—Bluetooth, WiFi, Word mobile with embedded graphics, Excel mobile, Powerpoint mobile, set of hardware application buttons, email download etc.

✓ Next to application layer is Windows CE which consists of virtual memory managers, process manager, thread, and interrupt handlers, and user interfaces. This layer supports priority allocation to processes, interrupt service threads, handles real-time latency etc.

✓ The Windows CE layer next to the above is kernel source code. Hardware abstraction layer is layer added for customising Windows CE for device hardware.

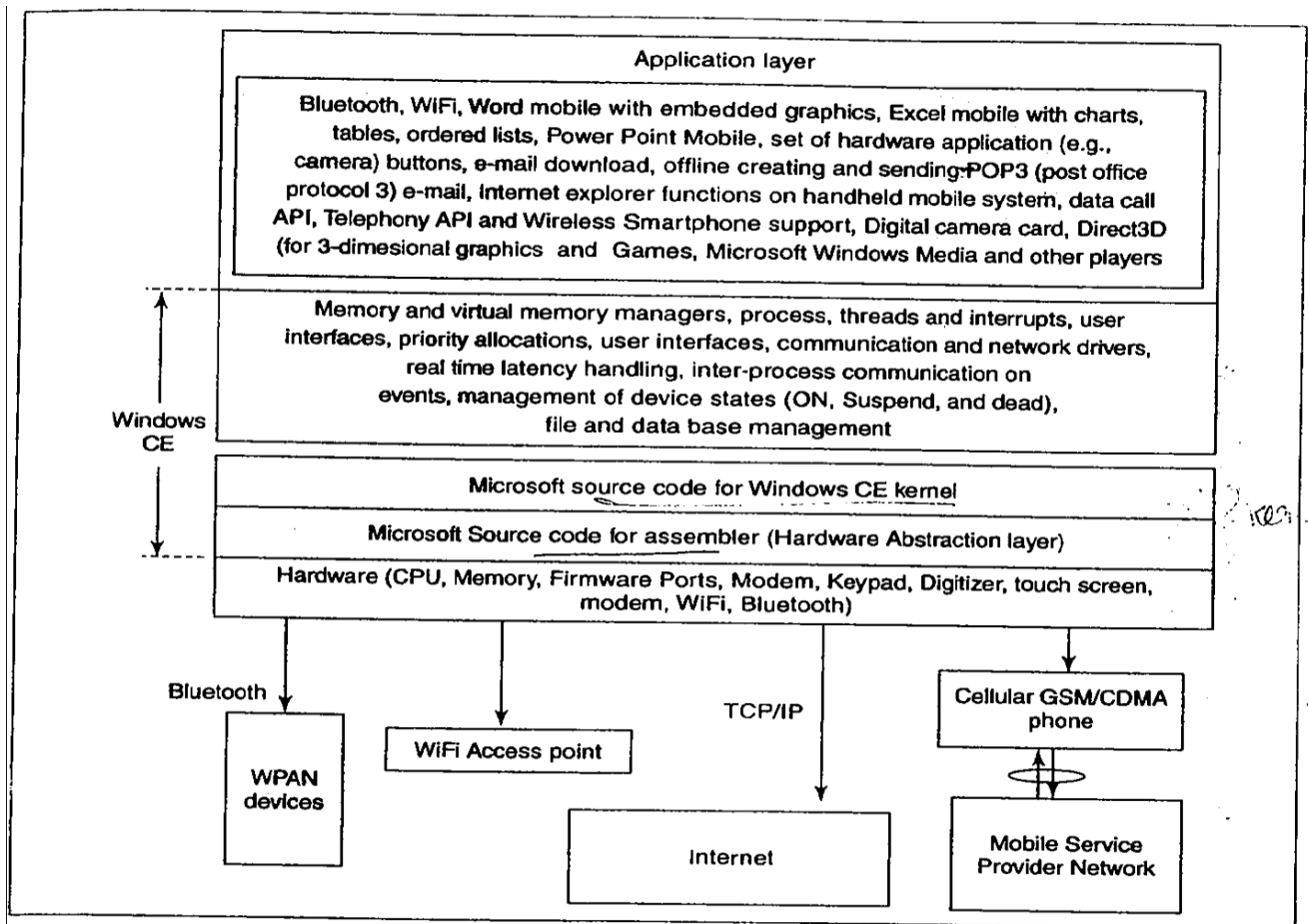Below shows the architectural layers of Windows CE.

Fig. 14.4 Architectural layers of Windows CE with OS layer in between the applications and hardware through the kernel hardware abstraction layers

Windows CE can be in three states – (i) ON with the clock frequency lowered in idle state, (ii) suspend with power to unused system units and port peripherals disconnected, memory data persistent, CPU idle till next interrupt, and clock running, and (iii) dead with power disconnected.

➤ **Windows Mobile**
   ✓ Handheld PC using Windows OS is also known as PocketPC. Windows mobile is a suite of basic applications for handheld devices along with the compact OS.
   ✓ It enables PIM, Internet explorer, Windows Media Player, various audio and video formats, Voice over IP, pocket Excel, Pocket word, Microsoft Outlook etc.
   ✓ The Windows Mobile Microsoft .NET platform is an open platform.

> **Memory and File management**
>> ✓ Windows CE assumes that there is 4GB virtual memory. A system can have RAM,ROM and flash memories.
>> ✓ A memory has logical hexadecimal addresses from 000000000_h to $FFFFFFFFF_{h.}$
>> ✓ A ROM image is the footprint of OS, Data and the applications at the permanently installed ROM. When the OS is configured and customized for an embedded application, then OS footprint is the ROM image. The Customisation reduces the OS footprint.
>> ✓ There is a dynamic heap of application process stacks, TCP/IP stack, OS functions stack, application and OS function dynamic-memory spaces, Bluetooth stack ,Wi-Fi stack etc in the *program memory.*
>> ✓ OS and system functions are saved in the ROM.
>> ✓ Windows CE memory manager has four different types of access mechanism as follows:
>>> • Simple—There is no use of paging or cache.
>>> • Cache—Cache is used but paging is not used. Processor cache is used along with the RAM/ROM/flash memory.
>>> • Page—Paging is used but not cache.
>>> • Sophisticated—both cache and paging are employed.
>> ✓ Windows CE assigns contagious pages in Virtual address space to one of the 32 memory slots. Memory manager allocates distinct slots to a distinct process thus reducing fragmentation of pages.
>> ✓ Windows CE file manager manages data as a database or object file. A file system has a root directory with which the file folders associate in a tree-like structure. A file system divides all file folders into volumes. Each directory can de further divided into subdirectories and file folders. Each subdirectory can be further divided into files and subdirectories till the leaf node which has a single file folder.
>
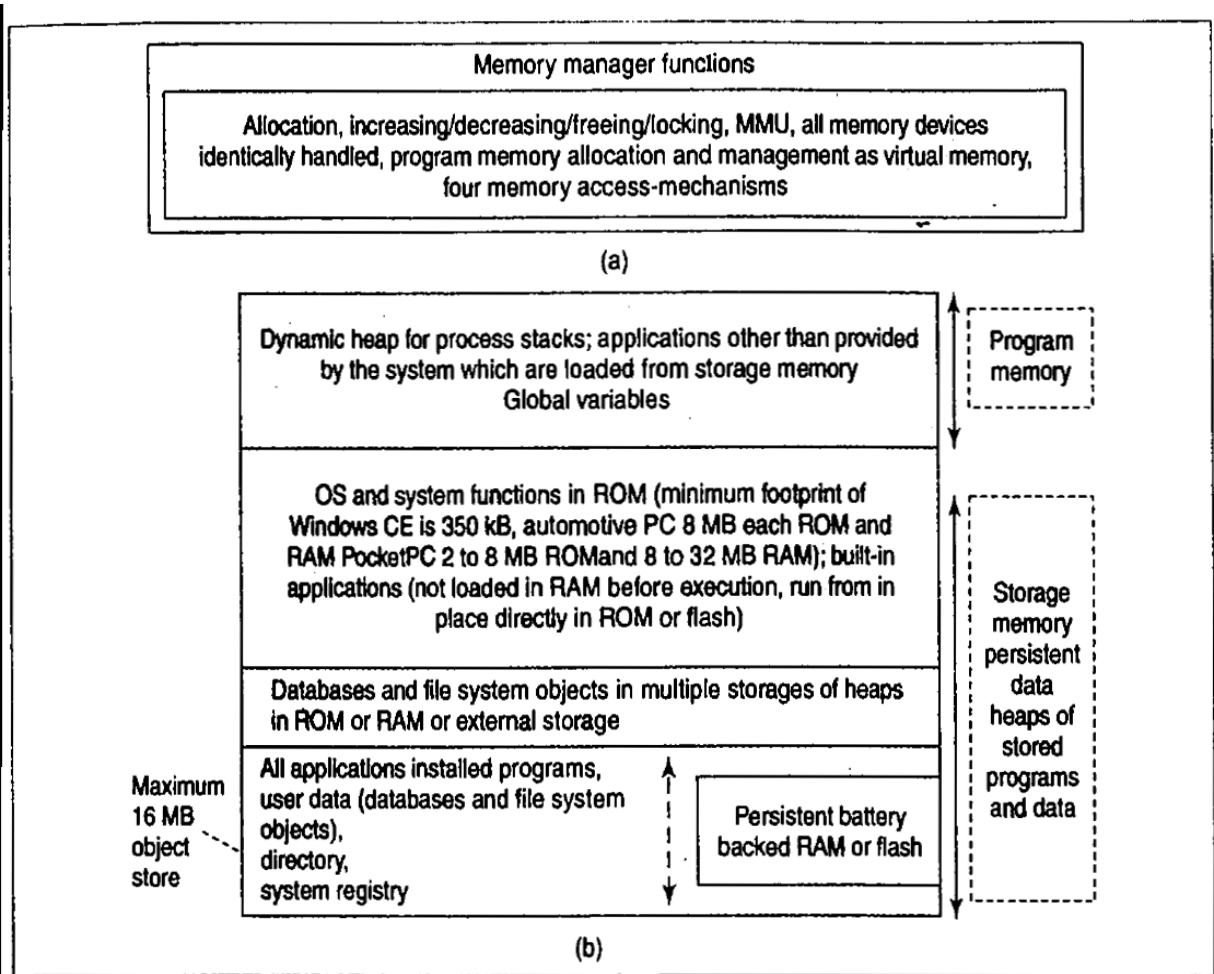> Below fig shows the windows CE memory manager functions and memory partitioning.

Fig. 14.5 (a) Windows CE memory manager functions, (b) Windows CE memory partitioning

➢ **Communication, Network, Device, and Peripheral Drivers**

✓ Windows CE provides communication and network APIs for *serial, IrDA, TCP/IP, Bluetooth stack, and Wi-Fi stack.*

✓ Windows CE integrates Microsoft Smartphone software which enables the application of Windows CE device as cellular GSM.CDMA phone.

✓ Serial communication uses a cradle. A serial manager(SM) provides interface to the device on cradle. Connection management is by using the serial link interface protocol (SLIP) and point-to-point (PPP) protocol.

✓ IrDA asynchronous serial or synchronous serial communication uses ActiveSync.

✓ Windows CE uses TCP/IP network library functions to transmit or receive the stacks and application requests and response on or from the network.

✓ Windows CE provides device driver and peripheral driver functions for low-level drivers at the kernel. USB connectivity is provided for the peripherals.

➢ **Application Development**

✓ Windows CE considers threads of process to be a fundamental unit of execution and providing access of CPU. Application are developed by coding for the threads. It supports 256 priority levels for assignment of threads.

✓ Corresponding to event, there is an event handler. An event sends interrupt signal which is checked for source and is identified whether it is hardware event, software exception, user-action-based event etc.

✓ Event handler is called Interrupt service thread (IST) in windows CE device. An IST has 8 priority levels. Highest priority levels means *time critical* priority and the lowest priority level means *idle priority.*

✓ An external interrupt is notified to the OS and for each request, there is an ISR assigned by the OS on notice.

✓ An application development needs to define processes and each process has threads.

Below fig shows how an application program runs on Windows CE platform as event-driven number of ISTs and application threads.
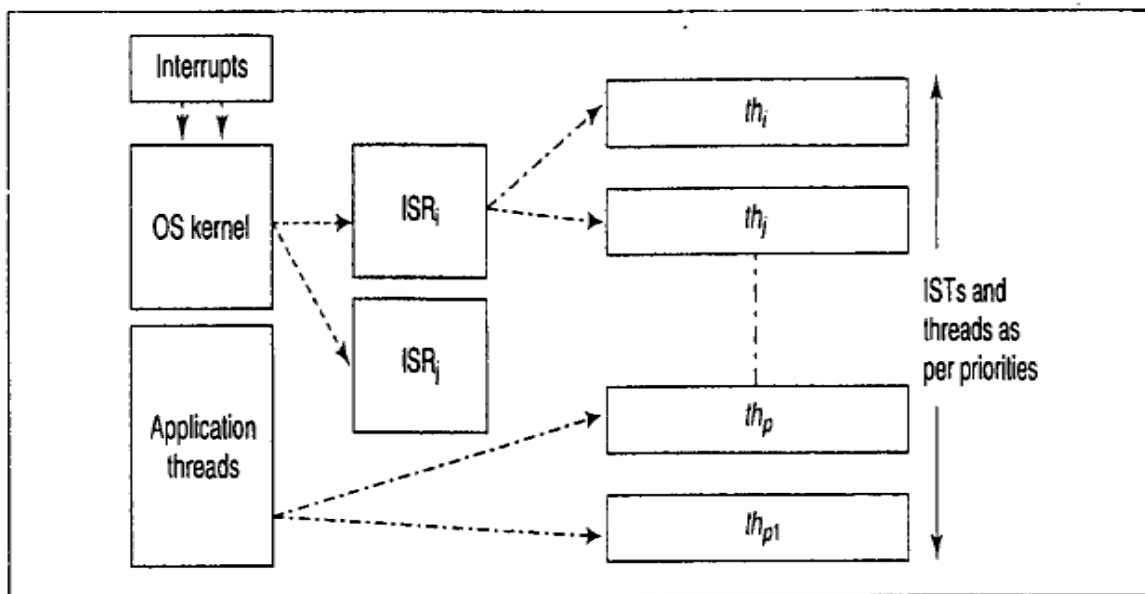

Fig. 14.6 Application program running on Windows CE platform as event driven number of ISTs (*i, j, ..*) and application threads (*p, q, ...*)

8.8 Symbian OS

Symbian is one of the most popular Oss for handheld Smartphones and mobile handhelds with phone and multi-modal communication. *Multi-modal* means usage of different modes—text, image, video, or audio. Symbian OS supports application development in C/C++ as well as Java and many communication protocols.

Below table shows the highlights of Symbian OS.

Property	Description
Strength OS	• Efficient running of application programs • Compiled for a system with specific set of hardware, its performance is very finely tuned for running exclusively on ARM processors • Low boot time • OS supports multi-processing or multi-tasking and multithreading • C/C++ OS functions, supports Java and JavaPhone • Internet connectivity for Web browsing, IP-based network connectivity, and WiFi (in later version only) • JavaPhone and integration to cellular GSM/CDMA phone.
Memory	Large storage memory which includes 80 MB of built-in memory in a multimedia card (MMC)
GUIs/VUIs	APIs for the buttons, menus, advanced voice features such as a hands-free speakerphone, and conference calling capability
User interface (GUI) display resolution	Generally small screen 160 × 160 pixels with optimized layout of desktop programs displayed on screen and 256 colour touch screen (higher resolution support depends on new versions)
Software	Desktop both for Lotus and Windows programs, other essential software, push-to-talk, high-end security enhancement features, graphics support including support for 3D rendering, simple APIs as compared to Windows CE, PIM, JavaPhone, JVM, MIDP (mobile information device profile), contacts, SyncML, Office, address book, spreadsheet, calendar, agenda, word processor, text-to-speech converter, browsing, messaging (SMS, MMS, email, and IMAP4), WAP push Microsoft Office formats (MS Office 97 onwards) and slide shows, email download, offline creation and sending of POP3 (post office protocol 3) email, and Internet browsing functions, GUI development support on C/C++ and Java platform, Java application using J2ME, multimedia applications such as playing music (Palm Tungsten), wireless communications Support for WLAN, Adobe Reader for accessing PDF files, Symantec Client Security 3.0 and the Fujitsu mProcess Business Process Mobilizer, corporate solutions such as IBM WebSphere Everyplace Access, BlackBerry Connect, Oracle Collaboration Suite, and secure mobile connections via VPN Client
Ports support	Serial, USB, infrared ports, telephony, and Bluetooth for communication with mobile phones and external modems

(Contd)

(*Contd*)

SyncML and other synchronization software	SyncML PC synchronization feature, synchronizes and chains to a PC in the vicinity. Also supports synchronization software developed on C/C++ platforms
Multi-threading support	Supports multi-threading, MMC (multimedia card), SD (secure digital) memory card, and SDIO (secure digital input/output) memory card
Third-party common applications support	Large number of examples—games, travel and flight planner, enterprise solutions, calculators, graphic drawings, and preparing slide shows.
Support to multi-tasking	• Multi-tasking Symbian allows running of multiple processes of an application and pre-emption of low priority process by a higher priority one • Platform for running multimedia applications due to Symbian is not for designing real-time systems 3

The below fig gives the architectural layers of Symbian OS. The lowest level layer in OS is a kernel which directly interfaces the assembler, firmware and hardware.
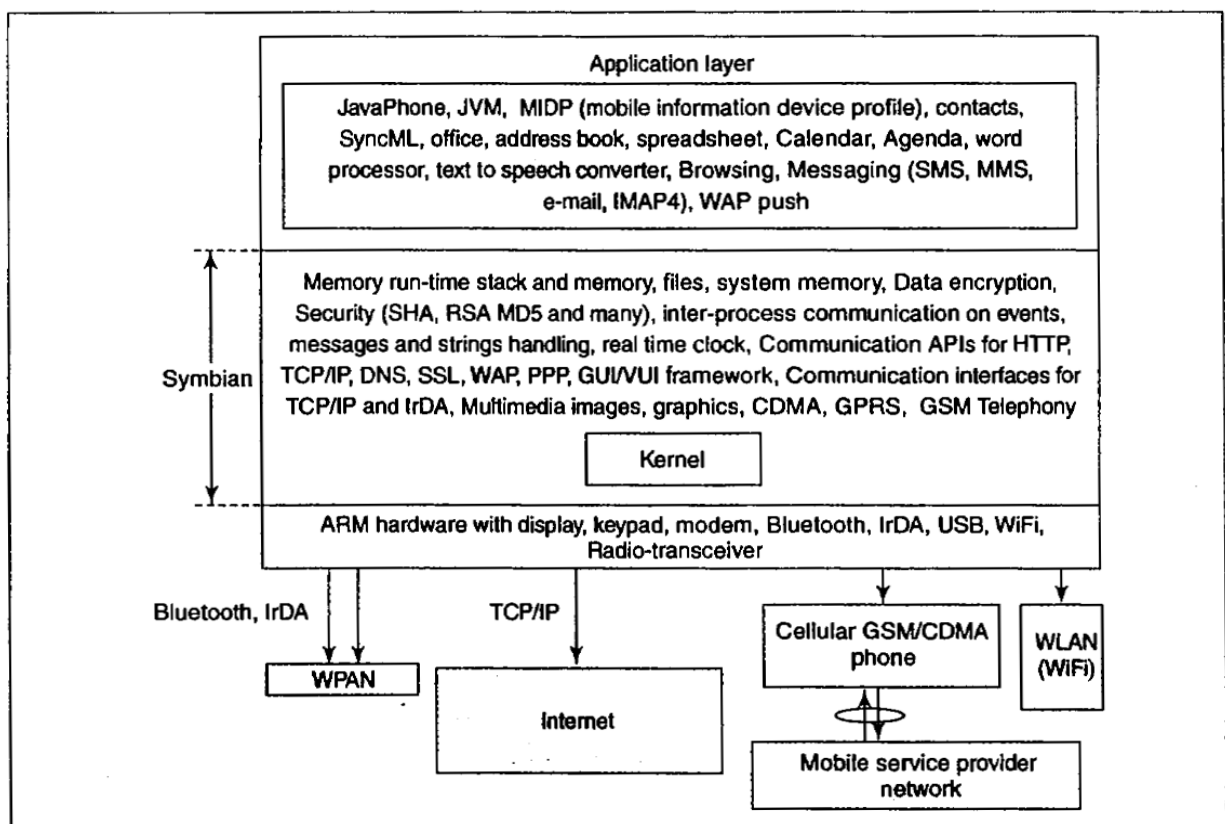

Fig. 14.7 Architectural layers of Symbian with OS layer in between the applications and hardware

> ## Communication and network APIs
>
> Symbian provides communication using WAP, WiFi, and network APIs for serial Bluetooth, TrDA, and TCP/IP, communication APIs for HTTP, TCP/IP, DNS,SSL, WAP, PPP, GUI/VUI framework, and communication interfaces for TCP/IP etc.

8.9 Linux for Mobile Devices

Linux is one of the type of OS which is best suited for different sorts of hardware and software applications. It enables the user to customize their device to suite their specific needs. It is also considered more secure OS. Its main features are as listed below:

- A 240 × 320 TFT display with 262,000 colours
- USB 2.0 PC networking for fast 'drag-and-drop' data transfer
- Built-in FM radio
- Support for Motorola's iRadio service
- Support for Bluetooth
- 1.3-megapixel camera for video capture and playback
- MMS (multimedia messaging service) enabled
- Opera web browser
- *Airplane mode* for safely listening to music when aboard an aircraft
- PIM (personal information manager) with picture caller ID