Petri Vuorimaa

Software

- Introduction
- Browser based software architecture
- Distributed software
- Servers
- Network
- Terminals

Networked multimedia

• Network multimedia can be defined as multimedia applications and services, which are distribute into a network
• For example, www services, digital television, and WAP

User interface

Middleware

Communications

Petri Vuorimaa

Networked multimedia

Causes of multimedia

• Multimedia has continuous media elements, which have to be synchronized
• Media is usually transferred as streams
• Inside the stream, the samples (audio sample, video frame, etc.) are in order
• Media streams can be synchronized
  + Internal synchronization: isochronous
  + External synchronization: synchronous

Buffering of media streams

• It is impossible to synchronize media streams exactly
• Data transfer delay fluctuates, which causes jitter
• Human can detect even small fluctuation
• Fluctuation can be reduced with buffering
• Usually, buffering is required in several stages

Petri Vuorimaa

Programming

• There are two ways to implement distributed multimedia system:
  + client / server
  + distributed software
• Browser software architecture is typical example of client / server architecture
• Distributed software means usually distributing a object oriented software into a network

Browser based architecture

• Internet applications are based on client / server architecture
• WWW browser is a typical client program
• WWW server is a typical server
• There can also be other servers (database, video on demand, video conference, etc.)
• In addition infrastructure servers are needed (proxies, directories, etc.)
WWW architecture

RealSystem
- RealSystem is a typical example of client / server architecture
- The system can stream different media (audio, video, animations, etc.)
- The system is composed of development tools, intermediate servers, and client programs
- The basic version of the client software is free, but other software products are commercial

RealSystem architecture

Distributed software
- Object oriented software can easily be distributed to several computers
- Objects have to find each other somehow
  - directory service
- The method calls and replies of the objects have to be forwarded to the right computer
  - Object Request Broker (ORB)
- Commercial solutions are, e.g., OMG Corba, Microsoft DCOM & .NET, and Java RMI & Jini

Corba architecture

Components
- Client and object implementation talk via ORB
- The interfaces are defined with IDL (Interface Description Language)
- Normally, the calls are made to the static stub, which forwards the call via skeleton to the object
- The implementation can also be dynamic
- Object register with the help of the adapter
Example: lip synchronization

- In movies, the speech and lip movements have to be synchronized
- Let’s implement lip synchronization as distributed application
- Example is from the book:
  + G. Blair & J-B Stefani, Open Distributed Processing and Multimedia, Addison-Wesley, 1998

Objects

- Stream interface
- Signal interface
- Media stream
- Signal
- Reactive object

Video and audio synchronization

- Video camera
- Video stream
- Video window
- Microphone
- Audio stream
- Speaker

Software architecture implementation

- The software architecture can be implemented also on the system component level
- Multimedia affects both servers, network, and terminals
- Efficient implementation requires additional features and even full redesign of some components

Servers

- The servers can be classified into transaction (www, database) and streaming servers (video)
- Scalability is most important feature of transaction servers
  + if necessary, the task can be divided for several servers (replication)
- Most important feature of streaming servers is real-time support
  + in practice, this means real-time scheduling

Disk scheduling

- The hard disk usage of video-on-demand servers have to be designed carefully
- Usually, the disk seek time and space consumption is optimize
  + thus data retrieval is based on fairness
- Video-on-demand server has to keep buffers full
  + real-time scheduling algorithms
  + relative location of files
Network has to support multimedia transfer:
+ Quality of Service (QoS)
+ real-time media streams
+ scalability
+ Resources have to be reserved for multimedia traffic
  + ATM QoS
  + IP Integrated Services
  + IP Differentiated Services

Network (cont.)
• Network protocols have to support real-time media streams
  + e.g., IP Real-Time Protocol (RTP), Real-Time Control Protocol (RTCP), and Real-Time Streaming Protocol (RTSP)
• In addition, the network has to support several simultaneous users
  + e.g., IP Multicast

Terminals
• The biggest problem of terminals is limited resources:
  + processing power, memory, and communications
• Current operating systems support primarily graphical user interface
• Support for multimedia is more limited
• Real-time operating system support better multimedia

Multimedia processing steps
• Receiving of packets
• Network drivers
• IP/UDP protocol
• Real-time transfer protocols
• Codec
• Player
• Windowing system
• Audio, video, etc. drivers

User interaction
• The terminal also has to track the user
  + keyboard
  + mouse
  + etc.
• The devices create interrupts
• The interrupts can easily jam with the network, etc. interrupts
Terminal software architecture

- The terminal software architecture can be implemented with several ways
  - + operating system + drivers + windowing
  - + browser + plug-in players
  - + Java
  - + XML browser

Operating system

Browser

- Java
- JavaScript
- Stylesheets
- HTML
- ActiveX
- Plug-Ins
- RealPlayer
- VRML

Java

- Application
  - JMF
  - AWT
  - IO
  - NET
  - Application programming interfaces
  - Java interpreter

Java Media Framework

- JMF allows use of multimedia in Java applications
- Real-time network protocols
- Multiplexing
- Codecs
- Players
- Effects
- Capture
- Control

Multimedia Home Platform

- One example of Java environment is the Multimedia Home Platform (MHP) of digital television
- The environment is composed of Java, JMF, TV, net, etc. APIs
- Applications are called Xlets
- The Xlets are transmitted via broadcast network
- Data is transferred through so called data- and object carousels
Operating systems

- In current software architectures, operating system has very important role
- Most of the media processing is done by calling system software
- Application acts as coordinator
- Data copying between application and operating system causes problems
- Operating systems are poor resource managers

Real-time operating system

- Small size: extra features have been removed
- Interrupt processing: jamming prevented
- Real-time scheduling: time of task processing
- Memory management: shared memory
- Message forwarding: communication between different levels is fast
- Resource reservation: task will not be accepted, if resources are not available
Microkernel

- Operating system becomes more compact, if all “unnecessary” features are removed
  + windowing, share file systems, etc.
- One solution is use so called Microkernel
  + the kernel of the operating system is as small as possible and real-time
  + other features are implemented as user processes
  + the required system can be composed from different components

Embedded Linux

- Linux is an interesting alternative also in embedded devices
- Unnecessary features (e.g., X-Windows) can be removed so that memory consumption is reduced
- Implementation of real-time features in kernel is very difficult, but not always necessary
- For example, in set-top box hardware takes care of video and audio processing

Multimedia services (GO-MM)

- Services
- Dynamic Quality of Service
- Communication Protocols
- User Interface
- Real-Time Operating System

Mobile multimedia software architecture

Main results

- Operating system
  + Embedded Linux + Kaffe Java Virtual Machine
- Communication protocols
  + Integration of the different multimedia protocols
- User Interface
  + X-Smiles XML-browser
- Multimedia services

X-Smiles browser history

- Development of X-Smiles was started as student software project in 1998
- The development has been continued in research projects GO and XML Devices
- Released as open source software in 2001
  + www.x-smiles.org

X-Smiles features

- XSL style sheets
- SMIL multimedia
- SVG graphics
- XForms forms
- XHTML and CSS style sheets
- User interfaces for different terminals