Module: MSC MD (INTMMT) **Seminar 8**

Multimedia Application Design and Networking

Introduction

In this seminar we will integrate the media and principals learnt so far by designing networked and distributed multimedia applications. We will learn the different aspects that should be taken into consideration and the varied requirements that a fully fledged networked/distributed multimedia project requires. The relevant aspects will be accompanied with relevant examples, often from video-conferencing, which is very popular and widely used (many of you probably use Skype and know it), and also a high demand and complicated application that includes varied media, network requirements and considerations, while using relevant standards. When planning multimedia applications, especially ones that are networked or distributed, we have to consider many different aspects – software, hardware, infrastructure, communication, etc., and their availability, maturity, and cost.

1. User Interface

When looking at the user interface, considering that you have already experienced designing user interfaces for a multimedia presentation during the project so far, this is not entirely new to you. However, there are many more aspects to consider and we will try to touch on some of them here.
When planning a multimedia application, the first question that arises is what media will the application use. This will determine the complexity and the different requirements (e.g., hardware, software tools, network and bandwidth).
For each media, we should determine the quality characteristics (frequency, resolution, color, speed, etc.), and the needed user interface controls (e.g., zoom, pan, pause, play, volume, etc.). These will serve as the building blocks of the user interface. Then, we can consider the functionalities of the application (what the application itself needs to do). Example functionality can be a list of contacts in a video-conferencing application, and the “call”, “hangup”, “transmit video”, capture/record, and other options.

Furthermore, when we relate to the different media, we should consider the source of the data. How is it captured or entered into the system? What is the source format? What is the output format?

Integration of media

As part of UI design, it is important to consider the multimodal aspects and take advantage of the different modalities to provide the user with a Multimodal UI – using the best modality for each subtask. While using different modalities can improve the accessibility and usability, the different modality alternatives for the application should be considered from carefully to avoid awkward use of media.

Synchronization

When using multimedia, there are elements of synchronization that should be taken into consideration. A well-known one is lip-synch in video. However, synchronization can be between several separate media steams. For example, in an e-learning application, where a teacher video or audio is sent in addition to a presentation (e.g., PowerPoint), we want the video/audio of the speaker to be synchronized with the presented slides. This synchronization is probably less demanding than lip-synch. In any case, the synchronization required should be well specified, including its level/parameters (Steinmetz & Nahrstedt, 1995).

In addition, in group communication, there is also the requirement to synchronize streams coming from different locations. This is called Orchestration. For example, in e-learning, we would like to make sure a student would receive the question from the teacher, before receiving answers from other students, who may have received the question faster and earlier.

The user interface definition and requirements mentioned so far are somewhat subject to changes based on limitations imposed (such as budget, bandwidth, etc.) and is thus an iterative process.

P2P, Meeting Point, or Client Server

Another consideration is the way the application works. Is it a client-server, Peer-to-Peer (P2P) or group server-based?

While some applications may work entirely as P2P, most applications require some kind of server support as a meeting point or for other services. Common P2P applications require a server to find appropriate peers. Often, the server will include lists of content (files, songs, etc.) to enable finding a matching peer.

Many group multimedia applications require a content distribution server support and/or full server support. For example, online group gaming supports **content distribution** of a game and management of users. Group video-conferencing requires a server to distribute the video streams of the participants. In this scenario, each participant sends his/her video stream to the server, and the server distributes the video streams to each conference participant. Some participants might open a private video session, possibly without using the server distribution mechanism. Normally, a free/non-managed conference like this will not support audio well, because unlike video streams that have their own display area and can be shown in parallel, mixing the audio streams results in extremely poor quality.

When the conference has structure and a manager, like in a formal meeting or a teaching session, the manager can run the conference and give speaking rights to participants and thus enable appropriate audio support. Obviously, the user interface should support these management options. This type of management called **floor control** (Chen & Harper 2009, Garcia-Luna-Aceves et. al. 2005, Wikipedia 2010a).

2. Special Hardware

Rich multimedia applications often require special hardware. The obvious one is high quality graphic displays and speakers. However, when we want to be more interactive, we also need microphones, cameras, and other hardware devices.

Starting with these “simple” devices, there are many types and features to these devices. For example, let’s look at audio equipment. One can use a headset (personal), a regular microphone (unidirectional), or conference room microphones (omni-directional). The microphones can provide additional features, such as hands-free operation, wireless access, parabolic (to focus sound waves), noise and/or echo cancellation, etc (Wikipedia 2010b).

Similarly, there are many types of display devices and cameras – for personal or group (projector), ones that detect motion, and many other features and options.

One has to learn the available hardware that exists and their space/setup limitations and cost considerations to make a calculated decision (or in the case of a designer, a well researched recommendation).

In addition, some multimedia applications require special and proprietary hardware. For example, remote surgery (also called Telesurgery, see Ottensmeyer 1995) combines elements of robotics. Other applications might need mobile devices.

3. Software Tools

Smart multimedia systems often require turning the data stream (audio, video, images, sensors) into meaningful input. This can include eye tracking, speech recognition, gesture recognition, parsing input from sensors, biometrics, etc. Most of these features can be achieved by applying special hardware or special software parsers (called tools here). These tools may be more advanced or less, but most have usage limitations that should be taken into consideration.

For example, let’s look at text to speech (TTS) and speech to text (STT) tools. You can try a demo of TTS avatar on Oddcast website (see tools), and explore with gender and other characteristics as well as different languages. Try using different words, terms, and names (also foreign). This technology works quite well as long as not required to pronounce foreign names or specific technical terms.

However, the other direction of speech-to-text (or dictation software) that requires speech recognition is more complicated and less reliable. Very often it requires training for the specific speaker and has higher error rate for people with accents or speech problems. While this is sometimes problematic for certain applications, others can still benefit from it. A sample use for speech recognition software is control commands, in which the user issues voice commands. When the vocabulary is constrained and well defined (e.g., does not include aliases) it can work much better. A sample application is phone directory services, in which the user's spoken text includes limited vocabulary such as states (Deng & Huang 2004). If you want to explore and try yourself, there are several products available for free download under "speech recognition" category in CNET or other software directories. Another application is Windows (both 7 and Vista) accessibility options. These enable controlling the computer using voice commands as well as dictating and editing text (Microsoft n.d.1). Demos are available at Microsoft (n.d.2).

Another sample tool is Optical Character Recognition (OCR), where images are converted to text. In this case, scanned printed text can be relatively reliably be converted to text, but hand written text is still a challenge (see Wikipedia 2010c for a quick overview and list of software).

Since there are several SDKs available online for the above technologies that application developers can take advantage of, in most cases, it makes sense to purchase/license such technology and integrate it in the application rather than developing it from start for each project.

4. Network Requirements

When using rich media, there is a need to transfer the media from the source to the destination. The source is the generator or keeper of the media and the destination is the consumer, or the storing area. This requires use of a communication channel of some sort, leading us to the topic of networking and the usage of networks and protocols.

To some of you, networks and protocols are nothing new so this section is aimed specifically at describing how we connect our servers with our clients to those of you without that background. The subject is treated fully in the CC (Computer Communication) module. In order to ensure practical universal exchange of any data over a network we need a set of standards and rules, known as protocols, to implement them. The standard for networking services on the Internet is the Internet Protocol suite. It is often just referred to as TCP/IP after two of the key protocols in the suite; the Transmission Control Protocol and the Internet Protocol.It uses packet switching ideas first developed in 1974 by Vint Cerf, Bob Kahn and Leonard Kleinrock and was finalized in 1979 so that by the time the WWW arrived in 1988 it was already the de facto standard for the Internet.

Since TCP/IP is a packet switched network, any information using the protocol must be split into suitable packets. This splitting and later recombining goes unnoticed by the higher protocol user. It is the same with HTTP, which to the web browser is the protocol used for communication but it is actually carried out by TCP/IP packets.

The Internet Protocol (IP) is used to provide the basic means of connection between two nodes on a network. Unfortunately it does not understand sequencing or error correction. It merely passes packages on the best route available to reach the receiver. The checking and assembling of the packets back into a meaningful message is left to the recipient.

TCP provides the means to ensure reliable connections between end nodes by doing all the necessary housekeeping. In doing so, it puts an additional overhead on the message which requires that more packets be sent by IP for a given message. For simple, mainly text based applications this is acceptable but for faster streams where losses can be accepted (such as the case of many multimedia applications) other protocols such as User Datagram Protocol (UDP) are used. You may wonder how message losses can be acceptable? A simple example is a video stream where the loss of a few bytes of information appears as a small disturbance to one frame of a video. Nobody would notice it. UDP is not good enough on its own for most real-time purposes where synchronization and data type identification may be needed. Therefore the Real Time Protocol runs on top of it. RTP has several flavors one of which is used for audio and video streaming particularly in regards to video conferencing. For the web we are mainly concerned with the HTTP protocol which will be dealt with in a different module.

When defining a multimedia application, the designer should calculate the bandwidth requirements and consequently consider and decide on the appropriate communication medium and protocols.

When calculating the bandwidth, one has to take into consideration the raw bandwidth but also compression and scaling options (the latter is described in more detail in the next section). A simple calculation of the resulting bandwidth is the basis for the overall estimate. For example, the bandwidth required for an uncompressed video stream is resolution times the color depth times the frame rate. However, seldom is the video sent uncompressed; some compression scheme is always used, thus reducing significantly the bandwidth required, so the characteristics of the specific compression and the data can be used for estimation. Often, the compression scheme has variable bit rate, the compression ratio and resulting bandwidth changes depending on the characteristics of the specific media stream. In this case, or when the application does not transmit the data at a fixed rate, there are other aspects of the bandwidth to consider. This is expressed as requirements of Quality of Service (QoS). For example, except general throughput that is the first to consider, one needs to consider also required application parameters (Kurose & Ross 2008, ch.7) such as:

- Burstiness: the occurrences of abrupt bursts of submission.
- Loss: the loss of packets of information during their transfer.
- Delay: the time taken for the information to pass from one end to the other.
- Jitter: delay variance, when the delay is not constant.

The designer of the application should define the needed QoS requirements and consider if the selected communication infrastructure is capable of delivering this. For example, a delay of more than 400 ms is not well accepted in a telephony application.

Furthermore, some applications can highly benefit from multicasting. Multicasting is the transmission of one stream to multiple specific locations (unlike Unicasting that is transmission from one point to one other point, or broadcasting that is transmission to all). Real network-supported multicasting never submits more than one stream on one channel. The stream is split up to multiple streams only when there are several different channels (“directions”) the stream has to be sent to. When there is a distribution of streams from one location to several locations, having a communication infrastructure to support this can simplify the application and save resources significantly.

The Communication and Networks (CC) module delves more into communication considerations.

Storage

Many multimedia applications require storing some or all of the information manipulated. When designing the application, a thorough calculation of the storage requirements over time should be made. It should also take into account the location of the storage (especially in networked or distributed application, where there are a few locations with storage devices and perhaps servers). The overall decision of where to store the data and for how long is another consideration here.

5. Other Considerations

Scaling

When the optimal QoS requirements cannot be met because there are constraints (budget, availability of equipment or communication services) we should consider adapting the application, if possible. One way to adapt the application is by scaling it. Scaling is mostly mentioned in the context of image scaling or network scaling, but it can be applied to multimedia in general. A good definition:

“*Altering the spatial resolution of a single image to increase or reduce the size; or altering the temporal resolution of an image sequence to increase or decrease the rate of display. Techniques include decimation, interpolation, motion compensation, replication, resampling, and subsampling. Most scaling methods introduce artifacts*.” (Ace5.com, n.d.)

The idea is that in order to meet the constraints (e.g., storage or communication limitation), an application (and the media it uses) must be scaled. One main way to scale the bandwidth (or storage) is to compress the data. Compression can be lossless or lossy. The decision on the type of compression depends on the specific media and application. For example, an application that does face recognition will save different data than one that only stores images for viewing (the latter could be based on human vision). This was covered already in detail the seminar about images.

However, there are additional ways to scale (most are “lossy”):

- For Images – scaling is done by reducing the resolution or the color depth.
- For audio – reducing sampling rate or frequency.
- For video – reducing image resolution (size) and frame rate (and whatever is relevant to images and audio).

Scaling is almost always required in video applications, especially over the Internet. Most current video standards from ITU (H.26x) define the video sizes in Common Intermediate Format (CIF), which is 352x288 pixels. The standard sizes are SQCIF (128x96), QCIF (176x144), CIF (532x288), 4CIF (704x576) and 16CIF (1408x1152).

For example, Microsoft NetMeeting 3.0 (Videoconferencing software that includes a developer API) has three video window formats:

- Large – CIF
- Medium – QCIF
- Small – SQCIF

For a given bandwidth, using a larger video frame for the same video means slower and degraded quality of video. Choosing the right way to scale is the choice of the designer or the end user (if the latter is given a choice), as window size directly affects the video speed, and the codec type affects the quality as well.

While images and video can be scaled down significantly, there is less flexibility with audio. This was described in the section dealing with audio compression.

When the media cannot be scaled, for example, in medical applications, the choice can be to scale the application. This means that the application should support data transfer (transfer, and then view), instead of viewing in real-time.

Standards

In order for different devices and systems to work together, there is a need for interoperability. Thus, a number of standard organizations are providing standards, such as ITU, W3C and many more (for example, the video standards mentioned above). These organizations usually organize committees in which scientists and representatives from the different vendors meet to develop standards. As a consequence, there are many standards. Thus, one has to learn and decide which standard is best and most appropriate for a specific system.

6. The Future

Before rushing to design and develop applications, one should have an open eye on the emerging technologies – new user interface paradigms, hardware, software, networks, and concepts. For example, these can include Natural User Interface (NUI) that suggests a more intuitive interface should be used, one that is derived from natural actions of the users rather than from artificial actions and commands that the user has to learn. Another new paradigm is Organic User Interfaces (OUI), which includes biometric sensors, skin displays, brain interfaces, etc. New technologies to support these concepts are emerging including flexible screens, 3D displays, retinal displays and other biological sensors (Yonck 2010). Together with the increase in computing power, the development of smaller devices for input and display, and new and future interfaces using mobile and screenless/keyboardless devices with immersive devices (Mims 2010), we should keep our eyes open to new developments and take advantage of them when developing new systems.

Links and References

Ace5.com (n.d.), Glossary, Ace5.com [online] available at: http://www.dvdmadeeasy.com/glossary/s.html (accessed September 23rd, 2010).

Chen L. & Harper M.P. (2009), Multimodal floor control shift detection, Proceedings of the 2009 international conference on Multimodal interfaces, pp.15-22, http://portal.acm.org/citation.cfm?doid=1647314.1647320.

Deng L. & Huang X.(2004), Challenges in adopting speech recognition, CACM V47(1):69-75, http://doi.acm.org/10.1145/962081.962108.

Garcia-Luna-Aceves, J. J. ; Mantey, P. E.; Potireddy, S. (2005), Floor control alternatives for distributed videoconferencing over IP networks. First International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom 2005); 2005 December 19-21; San Jose, CA.

ITU http://www.itu.int/rec/T-REC-H/en

Kurose J. & Ross K. (2008), Computer Networking: A Top-Down Approach Featuring the Internet: 4th edition, Addison Wesley.

Microsoft (n.d.1) Microsoft Accessibility [Online] Available from: http://www.microsoft.com/enable/default.aspx (accessed September 23rd, 2010).

Microsoft (n.d.2) Demos of Accessibility in Windows Vista [Online] Available from: http://www.microsoft.com/enable/demos/windowsvista/default.aspx (accessed September 23rd, 2010).

Mims C. (2010), The Future of Interfaces is Mobile, Screen-less and Invisible, Mim’s Bits, MIT Technology Review [ONLINE] Available from:

http://www.technologyreview.com/blog/mimssbits/25623/?nlid=3390 (accessed September 23rd, 2010).

NetMeeting (n.d.) http://technet.microsoft.com/en-us/library/cc749977(printer).aspx

NetMeeting 3.0 Resource Kit (n.d) http://technet.microsoft.com/en-us/library/cc749977.aspx

Ottensmeyer M. (1995), Cooperative Tele-Surgery http://web.mit.edu/hmsl/www/markott/cooptelesurg.html

Steinmetz R. & Nahrstedt K. (1995), Multimedia: computing, communications, and applications, Prentice Hall, ISBN:0-13-324435-0.

Wikipedia (2010a), Floor Control, [ONLINE] available at: http://en.wikipedia.org/wiki/Floor_control (accessed September 23rd, 2010).

Wikipedia (2010b) Microphone, [ONLINE] available at: http://en.wikipedia.org/wiki/Microphone (accessed September 23rd, 2010).

Wikipedia (2010c) Optical Character Recognition, [ONLINE] available at: http://en.wikipedia.org/wiki/Optical_character_recognition (accessed September 23rd, 2010).

Yonck R. (2010) The age of the Interface, The Futurist [ONLINE] Available from: http://intelligent-future.com/wp/articles/Interface.pdf (accessed September 23rd, 2010).

Tools

Alsagoff Z. A. (2008), A Free Learning Tool for Every Learning Problem? [ONLINE] Available from: http://zaidlearn.blogspot.com/2008/04/free-learning-tool-for-every-learning.html (accessed September 23rd, 2010).

Oddcast Text-to-Speech avatar demo http://www.oddcast.com/home/demos/tts/tts_example.php

Pittpatt Face detection, recognition and mining (from video) http://demo.pittpatt.com/