
K. Katrinis, A. Wagner and B. Plattner

*Easy Teach & Learn: An Integrated
Synchronous Distance Learning Platform*

TIK-Report

Nr. 182, November 2003

K. Katrinis, A. Wagner and B. Plattner
Easy Teach & Learn
November 2003
Version 1
TIK-Report Nr. 182

Computer Engineering and Networks Laboratory,
Swiss Federal Institute of Technology (ETH) Zurich

Institut für Technische Informatik und Kommunikationsnetze,
Eidgenössische Technische Hochschule Zürich

Gloriastrasse 35, ETH-Zentrum, CH-8092 Zürich, Switzerland

Executive summary

In this proposal of the Easy Teach & Learn II project (ET&L II), we re-focus the ET&L project, adapting it to the real situation of today, also recognizing the fact that e-learning technology generally has advanced less than previously anticipated. While the broad goals remain as in ET&L, the distinctive objectives of ET&L II include the following:

- To develop a system for teaching and tutoring, which will be usable as a building block in a comprehensive e-learning environment. This will be achieved by providing simple interfaces to future Learning Management Systems or Learning Content Brokerage Systems.
- To create a system that adapts itself to the quality of service offered by the Internet connection (instead of expecting a specific QoS); sophisticated scalable video technology will be used to achieve this.
- To use a new approach to user interface design, making system handling easy for the students
- To carry out a live trial with students

ET&L II will use existing technology as much as possible: It will be a Java application using the Java Media Framework with an embedded Web browser for accessing information on the Web (thereby bridging the gap between synchronous and asynchronous self-learning tools); it will use the wavelet-based WaveVideo codec (developed at TIK) for encoding live video, but will resort to standard codecs for audio; and it will use CATV Internet and ADSL as the access medium for the students.

ET&L II will be a project in which specific research directions will be explored and the research results will be integrated into one system; thus, ET&L II is a symbiosis of research and software development, which should provide an easy road for the transfer to reality.

The project is scheduled to start early in 2002, with a duration of 24 months. It can be funded from the previous grants, i.e. there are no additional funds requested.

Table of Contents

1.	INTRODUCTION	5
1.1	WHY DISTANCE LEARNING?	5
1.2	DEFINING DISTANCE LEARNING AND ITS TYPES	5
2.	OBJECTIVES OF ET&L II	6
3.	REVIEW OF SIMILAR EXISTING APPROACHES AND SYSTEMS	9
3.1	SYNCHRONOUS TELETEACHING SYSTEMS	9
3.1.1	IRI-h	9
3.2	VIDEO CONFERENCING SYSTEMS	10
3.2.1	Microsoft Netmeeting	10
3.2.2	Click to Meet	10
4.	TECHNICAL APPROACH	11
4.1	INTRODUCTION	11
4.2	COMMUNICATION LAYER	12
4.3	MIDDLEWARE	13
4.3.1	Session Announcement (session lookup service)	13
4.3.2	Session Management	13
	A. Session Structure	14
	B. Session Establishment	15
	C. Session Join	16
	D. Session Termination/Leave	16
	E. Interaction Control	17
4.3.3	Video Codec	17
4.3.4	Audio Codec	18
4.3.5	Real-Time Media Handling	18
	A. Capturing – Processing - Presenting	18
	B. RTP Packetization	19
4.3.6	Application Adaptivity	23
4.4	APPLICATION LAYER	24
4.4.1	Application Graphical User Interface	24
	A. Student's GUI	25
	B. Lecturer's GUI	26
	C. Administrator's GUI	26
4.4.2	Teaching-Aid Tools	26
	A. Presentation tool	26
	B. Annotation tool	27
5.	DISSEMINATION	27
6.	REFERENCES	28

1. Introduction

1.1 Why Distance Learning?

One of the main criteria for estimating the grade of “development” of a society is its transport infrastructure, i.e. how well it provides means to make transportation of people between discrete spaces faster and more convenient. Through the emergence of new telecommunication services, this notion of development is taken further to include the degree, by which available tele-services can minimize the need for transportation.

One of the recent breakthroughs of tele-services is distance learning. Although it is not meant to completely replace conventional learning, it brings many advantages. Some of these are:

- Reduced travelling time both for students and teachers when attending/teaching a course.
- Support of the education of students who have difficulties in being physically present at the university or other educational institutions, for example, disabled students and full-time working students.
- Educational institutions or companies may reduce costs by transposing the ever-demanding need for class space and infrastructure to virtual classrooms.
- Due to the rapid advances in technology, rapid changes are caused in many areas of life. In order to keep the workforces up-to-date, life-long learning is becoming a need, but with the requirement of only minimally disrupting the production process.

In the case of inter-institutional tele-teaching between universities of developed and developing countries [1], the contributing aspects to distance learning are:

- To overcome the lack of qualified teachers in developing countries and the future plethora of qualified teachers in developed countries.
- To supply the access to world leaders in a specific field of knowledge (“gurus”), without having to move them in space.
- To give to students coming from very poor countries the opportunity to be educated, although no university or expensive infrastructure exists in their home countries.

1.2 Defining Distance Learning and its Types

Distance learning is simply termed any type of education, where the students are separated in space from the lecturer and, in some cases, also in time [2]. The lecturer uses elements of new technology as a medium to deliver educational content to the students. The student then uses the same media to send its feedback to the lecturer. With the advance of technology, several media have been used to deliver content: The traditional postal system, audio/voice technologies (audio/video cassettes, telephone, audio conferences) computer technologies (content in CD-ROMs, e-mail, websites) and video conferencing.

As mentioned, in a distance learning system the students might also be separated in time from the tutor. Actually, this is the criterion that classifies systems supporting distance learning in two discrete kinds: synchronous and asynchronous. In a synchronous distance

learning system the tutor is only separated from the students in space, i.e. the students attend the class at the time it is given. Typical synchronous delivery systems are video conferencing systems, audio conferences and the telephone. On the other hand, in asynchronous distance learning systems, the tutor is separated from the students both in time and space. That is, the students view stored educational material, with obvious limitations as far as interaction is concerned. Typical delivery systems of this type are websites, audio/video cassettes, postal mail and cd-roms. Both types of systems are being used in higher education; however, it appears that the asynchronous type has received more attention in the past few years, with the obvious result that systems supporting synchronous distance learning are lagging behind, and urgently need more effort.

2. Objectives of ET&L II

Many research groups, standardization committees ([6], [7]) and companies have been performing research in distance learning. As with every emerging technology in informatics, a stabilization period is required until the field becomes acceptably standardized. During the past years the formerly mentioned research carriers were working in an uncoordinated way, trying to “define the problem” and delivering stand-alone tools, which were hard to interface with each other and/or were using not reusable content. Nowadays things have changed considerably: research on distant education has focused on several distinct areas - educational content (especially to create reusable learning objects (RLOs) and means to manage them), e-learning personalisation (Learning Management Systems (LMS)), content look-up, retrieval and commercial management (brokerage systems) and delivery media (e.g., such as proposed in the first ET&L proposal [3]).

ET&L II will develop a platform for synchronous delivery of educational content. As such it uses common ideas from video conferencing systems, which, however, are properly adjusted to satisfy the needs of tutors and students. Taking advantage of the advances in network technology and multimedia communication, it will be used to effectively deliver online university lectures. In the rest of this section we discuss in non-technical terms the functional parts of the ET&L II platform, its services and the key aspects that make it unique.

The system to be developed is not only aimed at students working at workstations inside the university campus, as is the case with many of the existing distance learning systems. On the contrary, we are aiming at supporting students using various popular platforms, such as PCs with standard operating systems, and different types of network access technologies, i.e. we address cross-platform issues and network heterogeneity. The first feature is important so as not to constrain the participation to on-line classes only to students, whose computers use a specific operating system. To support network heterogeneity is crucial in order to allow access for students from outside the university’s network: this will be achieved by designing the ET&L II system such that it will work with readily available, commercial high-speed Internet access technologies; specifically, the targets are ADSL and CATV-based Internet services, which are affordable and offer sufficient resources for distance learning applications.

Ease of use is another design target of the project: every available course will be announced in a common website, along with session details (time schedule, name of lecturer etc.) and links to download needed software. (The term “session” denotes the online course: a session consists of the conference’s participants and its duration equals the duration of the online lecture). The student will be able to view the available sessions and then join his/her lecture by just clicking on a link, assuming, of course, that the required software has

previously been installed. This click will automatically open the delivery application on the student's side and send to the student all the necessary information in order to enter the course ("session-join" in technical terms) and enable him to interact with the other participants. After this phase is successfully completed, the student will have "entered" the virtual classroom: he will be able to watch and hear the actual speaker of the session.

Apart from providing ease of use, a web-based session announcement environment has the potential for interfacing well with existing or future Learning Management Systems. An LMS will surely provide an HTML-formatted output that can be subsequently used to automatically produce the announcement site.

Synchronous communication between the tutor and students will be provided in a manner similar to video conferencing. Every student will be able to interact with the classroom by sending a video and an audio signal to the other participants. The video signal will be encoded and decoded using the WaveVideo ([26], [27], [28]) format, developed by the Communication Systems Group at TIK. The encoding method used by WaveVideo is based on wavelet theory ([29], [30]) and produces a signal that allows effective media scaling and high error tolerance, making it suitable for network channels with low and variable quality. Only one participant will be "active" in the session at every instance of time. The decision of who is the active user is made by the lecturer (assisted by an interaction control protocol, discussed in the next paragraph). In case the available network resources allow for it, we may consider adding a second video signal at every receiver side (video of the real classroom). Moreover, a presentation and an annotation tool will aid the teaching process. With the presentation tool, the lecturer will be able to present a slide show, whereas in the annotation window several texts could be input or schemes drawn to aid the understanding of the educational content being in focus.

Which participant gains write access to the annotation tool, as well as who becomes the active speaker (called also floor holder) of the main session are both decided by the lecturer¹. We also refer to this procedure as interaction control or floor control (the floor is the focus of the conference). The interaction control scheme strictly follows the procedures of students attending a normal lecture, i.e. the general focus of interest is set to the lecturer. A student that intends to ask a question signals his/her intention and waits until the lecturer grants the requested attention, in which case the student will become the active speaker. Then he/she asks a question and releases the floor control again. Besides the "official" interaction with the lecturer, students may whisper with their virtual desk-neighbors without disturbing the main flow. This is accomplished by building sub-sessions within the main session. Note that the teaching-aid tools' scope is the main session and their content should not be propagated in the scope of sub-sessions. This will allow for co-operation in groups: for example, a student might use the annotation tool to explain to a "classmate" of his a point that student missed.

Through the concept of sub-sessions, ET&L II realizes "tutoring", going beyond traditional tele-teaching, where the lecturer delivers the same content to a non-uniform group of students. Tutoring is a personalized form of education, which takes into consideration the background, the educational strengths and weaknesses and the individual personality of every student. By establishing sub-sessions on demand, the tutor can personally instruct the student,

¹ We haven't decided yet, whether the one that receives the token to write to the whiteboard, will become also the active speaker. It may prove useful for tutoring to maintain different tokens for these services, especially in one-on-one tutoring, where the lecturer (active speaker) instructs and the student (active owner of write access right to the whiteboard) implements.

cover his specific needs and assign “personalized” tasks. Additionally, sub-sessions introduce facilities for group work into distance learning platforms: the tutor could group the students into sub-sessions and assign them different tasks.

Summarizing, the innovative objectives of ET&L II, regarding functionality provided by educational content delivery platforms, comprise of the following:

- **Ease of use**
- **Cross-Platform functionality (write once - run anywhere)**
- **Network heterogeneity**
- **Tutoring capability**
- **Facilities for group work**

The project will provide the opportunity to address the following research topics:

- **Application Level Quality of Service:** to cope with varying network performance during the session, we will monitor connection statistics and try to adapt the quality of the video signal individual students receive to available network resources.
- **Social interaction protocol:** as already explained, the student needs to ask the lecturer for the session focus in advance to address a question or opinion to the other participants. We will research on whether this mode of interaction is technologically feasible and educationally beneficial.
- **Scalable video content streaming via RTP:** We will integrate the WaveVideo (scalable video format used to deliver video content) into the Real-Time Transport Protocol (RTP), the Internet standard to support media streaming, and investigate its capability to provide an adaptive video service in real-life situations.

The user-related features mentioned above, as well as the research results, will be evaluated during a “test and trial” phase. We are considering to use and evaluate the new platform in a theory lecture of a “PPS-Praktikum” course in the curriculum of the Department of Information Technology and Electrical Engineering at ETH.

3. Review of similar existing approaches and systems

3.1 Synchronous teleteaching systems

3.1.1 IRI-h

IRI-h (Interactive Remote Instruction-heterogeneous, [37], [38]) is distance education system, which is currently being designed and implemented in an R&D project at the Old Dominion University in Norfolk, Virginia, USA. The system will work on a number of heterogeneous platforms and within heterogeneous network environments. Cross-platform nature is achieved through the use of the Java language. The implementation is successfully tested on multiple platforms including PCs running the Microsoft Windows operating system (NT, 98, 2000) and Unix machines running Solaris. The existing prototype is by now running over the university's LAN and its extension to cater heterogeneous network environments (ex. To support home users) is considered in future steps.

The main functional features of the current prototype are:

- Audio/video communication between participants. It allows a maximum of three video windows to be present in the shared view at one instant of time. The respective number for audio is ten.
- Application sharing: every participant is capable of sharing every application running in his workstation with the rest of the session participants.
- Annotation tool shared among participants.
- Pointer service.
- Personal notebook for every participant, not session shared.

From the technical point of view, IRI-h is based on:

- A simple custom-made session control protocol. Only a main session is supported, sub-sessions within the main session are not supported.
- A token-based interaction protocol, that is to write to a shared service the participant needs to receive a token from the service's token manager.
- The Java Media Framework as the API used to implement real time media capturing and rendering, as well as RTP (Real-Time Transport Protocol) packetization.
- Gateway servers that are introduced to solve the network heterogeneity problem. A gateway is responsible for classifying whether a session participant is multicast-enabled or disabled by means of a multicast capability test. If the participant is found to be multicast-disabled, the gateway listens to the session's multicast streams and forwards any received multicast packets to the multicast-disabled participant through one or more UDP tunnels. On the other hand, any incoming unicast packets from the multicast-disabled participant are forwarded to the multicast channel. Some participants may be multicast-enabled and yet request gateway support, due to bandwidth constraints.

There are certain overlaps between the IRI-h and ET&L II projects. In particular:

- They are both striving for cross-platform nature and network heterogeneity.
- The Java Media Framework is used to capture, render and transmit time-based media (audio/video). At least we will start with JMF as an implementation framework, as long as its performance proves satisfactory.

However, discrete differences exist also between the two projects:

- ET&L II will support the creation of sub-session within a main session. This will allow private communication between participants and thus provide tutoring and working in groups capabilities.
- From the beginning of ET&L II and at the end of every implementation stage the test bed will consist of the targeted access technologies (LAN, cable networks and ADSL as described later). We won't just work on a LAN until project completion and then implement intermediary components to achieve network heterogeneity.

In the past, we have established a good co-operation relationship with the developers of IRI-H. One of our collaborators visited the IRI-H team for an extended visit, and we plan to share knowledge and technology between the two projects.

3.2 Video conferencing systems

3.2.1 Microsoft Netmeeting

Microsoft Netmeeting ([39]) is a video conferencing tool, offering point-to-point communication with video/audio. Apart from this, it offers:

- Whiteboard tool
- Application Sharing
- Remote Desktop Control
- File transfer
- Text-based chat

The main drawback of Microsoft Netmeeting is that it doesn't support multipoint audio/video communication. Its multipoint capabilities are constrained to the whiteboard tool and to application sharing. Therefore it can't be used as distance learning delivery platform. Moreover, because of its origin (Microsoft) it is not cross platform, but limited to the Windows operating system.

3.2.2 Click to Meet

Click to Meet ([40]) is a complete end-to-end solution for rich media communications. *Click to Meet* provides a framework for group communications using live, interactive voice, video and data collaboration as well as streaming technologies. *Click to Meet* supports T.120 standards for data sharing, so users can collaborate online using popular tools to chat, send files, draw on a whiteboard, and share applications. It is completely automated. Behind the

scenes, the system securely connects users and seamlessly assures that all network resources will be available for the duration of each call. If a meeting is scheduled with a roam-based conferencing system, *Click to Meet* will automatically connect that endpoint at the appropriate time, allowing the users in that room to attend their meeting without ever touching a dial-pad or keyboard. With *Click to Meet*, you can support:

- **The widest range of endpoints**, at varying bandwidth rates – whether you’re using clients from First Virtual Communications or Microsoft, or endpoints from PictureTel, Polycom, as well as other H.323 vendors.
- **Multiple server environments**, including Windows NT, Windows 2000, Sun Solaris, and Linux operating systems.
- **Diverse applications**, from high-end corporate conferencing and application sharing to video instant messaging, both one-to-one and group conferencing.
- **Flexible customisation** that will fit your network topology.
- **Easy scalability** as conferencing usage grows – by linking multiple conference servers together.
- **User-friendly interface** through web-based graphic interface that provides easy-to-use system administration and makes multipoint conferencing easier than ever.
- **The latest advanced features**, including Microsoft Exchange compatibility for easy scheduling and meeting management, continuous presence for viewing multiple windows at the same time, and streaming media integration to extend a conference live to a larger, view-only audience; or to record a conference for playback later.

Drawbacks

- Designed for and tested over the US backbone, mainly taking advantage of the existing ATM infrastructure. In the showcase of e-learning systems based on Click to Meet there are not any already installed systems mentioned, that use a cable or an ADSL network as their underlying network infrastructure.
- Not explicitly aimed at education. It is purely a conferencing system, that doesn’t provide the mechanisms to “simulate” the real classroom.
- H.323 ([20]) is used by the application. This is a conferencing standard that covers most of the demands of a today’s conferencing system. This also means that the system is not that modular and there are restrictions in the selection of codecs and session control protocol.

4. Technical approach

4.1 Introduction

The functional goals described in chapter 3 will be mapped into technical terms in this section. We must mention that, whenever possible, we will try to use off-the-shelf components to implement the several modules of the system’s architecture. This way we will take advantage of the reliability and performance of existing tools and spare developing time, i.e. we don’t want to “reinvent the wheel” for modules, that are quite standardized and well

implemented. The selection of the off-the-shelf components will be based on a weighted decision in terms of performance, stability and cost. We would like for the delivery platform to be as inexpensive as possible and thus reachable by a wide number of students, as generally the case with knowledge and education should be. However, in some instances we think that the existing methods and tools are not capable of serving the system's services. In such cases we will research on new methods and develop new tools that will satisfy the project's needs.

The technical approach is structured in three layers: communication layer, middleware and application layer.

As mentioned in the objectives' chapter, our delivery platform is targeted on access technologies already provided by commercial service providers, such as ADSL, LAN and cable networks. This constraint on the communication layer forces us to follow a bottom-up design procedure of the system's architecture: we just describe the "fixed" communication layer specification and then give a detailed description of the implementation of the other two layers, middleware and application layer.

4.2 Communication Layer

The ET&L II project is not intended to be a project, that will just promote the current state of research and will be practically usable only in a high tech laboratory. On the contrary the project is aimed, after coming to life, to be used straight at the university to extend the educational process. Therefore we will not make high assumptions on the underlying network, such as network-level quality of service (QoS) and resource reservation guarantees. We will just rely on access technologies that are today trivial in a university's infrastructure (support on campus participants) and that are already provided by ISPs (Internet Service Providers) in most of the developed European countries, including of course Switzerland (support remote participants). Particularly, the following three access technologies will be used as a test bed:

- 1) **LAN** (wired and wireless), providing up to 11Mbps up/downstream.
- 2) **Cable network:** in Switzerland there are at the time of writing two types of services provided, a service with 256Kbps downstream and 64Kbps upstream and a service with 512Kbps downstream and 128Kbps upstream (private services). (Cablecom Provider)
- 3) **ADSL (Asymmetric Digital Subscriber Line):** also making two types of connections available, one providing 256Mbps downstream and 64Kbps upstream capabilities and the other providing 512Mbps downstream and 128Kbps upstream (private service) capabilities. (Swisscom-Bluewin Provider)

Thus, even with the low-end type of connection, the user still has enough bandwidth for transmission/reception of a good-quality audio and video stream.

For all participants' unicast communication will be used since, unfortunately, the ISPs in Switzerland do not support IP multicasting over ADSL and cable networks. This means, that we have to use unicast as a means of exchanging media streams with participants of the virtual classroom, that use such access technologies to attend the course. Moreover unicast will simplify the implementation of reliable services and allow for user and not group resolution at the QoS application adaptation service (consider that in case multicast communication were used, the multicast-enabled participants would have to be optimally grouped in terms of average actual downstream bandwidth and thus a user would have to accept the bandwidth of his group's minimally network capable participant).

4.3 Middleware

We refer in this chapter to the application's communication subsystem or middleware, the part of the application that will process communication-relevant tasks and thus make them transparent to the main application.

4.3.1 Session Announcement (session lookup service)

We need to provide a medium to advertise to the students the available sessions and their description, concerning course name, conference time schedule, lecturer's name, content description, delivery media, information needed to receive those media (addresses, ports, formats etc.) and session administrator's contact information. The Session Description Protocol (SDP), as defined in [11], will be used to describe a conference. SDP is purely a format for session description – it does not incorporate a transport protocol, and is intended to use different transport protocols as appropriate, including the Session Announcement Protocol (SAP [18]), Session Initiation Protocol (SIP [17], [19]), Real-Time Streaming Protocol (RTSP [35]), electronic mail using the MIME extensions ([12], [13]) and the *Hypertext Transport Protocol* (HTTP [14]). In ET&L II the transport protocol will be HTTP, i.e. the session advertisement will be implemented in a common website.¹ The student, after viewing the available sessions in the website and selecting the session he is interested in participating in, he will follow the "join" link of the session of his interest. This action will automatically transmit an SDP message to the student. Taking into account that during installation time the conferencing application will have been registered in the student's workstation as a media handler for the MIME content type "application/sdp", the application will be automatically launched, as soon as the SDP message arrives at the student's workstation. The processing of the SDP message then provides all the needed information (IP address and port number of session manager etc.) for the candidate participant to make a request to join the session.

Note that by using a standardized format to describe the session, we will be able to easily interface the announcement website with existing brokerage systems. The only constraint is that the latter should provide an SDP formatted description of the courses available online. Moreover we could build a tool to automatically convert SDP formatted session description into HTML tables in order to produce with minimal effort the content of the announcement site.

The Java Specification Requests Group 141 ([15]) is in the developing stage of a Java SDP API, whose final release is announced for December 2001.

4.3.2 Session Management

Implementing a proper session management scheme is critical, because it completely defines the capabilities and limitations of the interaction between lecturer and attendees. In these days the undisputable favourites to dominate the session protocols' race are the IETF Session Initiation Protocol (SIP) and the ITU H.323 recommendation, which actually is a suite of conferencing protocols including a session control protocol. However, these two protocols are aimed mostly to IP Telephony signalling, i.e. they implement a lot of services

¹ In the same website all the software needed to take part in a session will be available for downloading, ex. conferencing application, Java Media Framework and codecs, along with installation instructions.

(user mobility, call forwarding etc.) that are redundant in a learning content delivery platform. The need for a specific session control protocol for university lectures-like conferences led the Communication Systems Group at TIK to the design of a new conference control architecture ([16]). We describe its main features in the following.

A. Session Structure

The following terminology will be used to describe the conference control architecture:

- **Session:** a session denotes a communication group
- **Root Session:** the session, where all the participants are subscribed. Teaching-aid tools are shared only in the main session.
- **Sub-session:** Sessions may be "forked" into sub-sessions. Such sessions can be constructed to support personal instruction between lecturer and student, to support tutoring, group working and to allow whispering between students (virtual desk-neighbors) without intervening in the main flow.
- **Session Holder:** The session holder controls a session.
- **Session Manager:** The session holder of the root session.
- **Parent Session:** The term parent session denotes the origin of a sub-session.
- **Floor:** The floor of a session denotes the transmitted information.
- **Floor Holder:** The floor holder provides the source of information within a session.
- **Participant:** A participant joined a session.
- **Session Participants' Register (SPR):** a list of every session's participants. One and only one member of the session is marked as the session holder and is the one who should be contacted by a new user to join the session.

Four virtual communication channels (Figure 1) model the flow of data between the session participants:

- **SAC:** The session announcement channel provides the session description to join a session at session join time. Moreover it carries session description content during the course, such as information about newly registered participants, information to handle late session joins etc.
- **FDC:** Through the focused data channel, audio and video data are transmitted/received. An FDC exists per participating workstation.
- **ICC:** Through the interaction control channel, floor control is bilaterally coordinated. It ensures that every time only one participant is the floor holder and has write access to the teaching-aid tools. A participant that intends to interact with the session initiates the ICC. It may be useful to have discrete tokens for floor control and write-access to the annotation tool to allow for one-to-one tutoring, where the tutor instructs and the student executes.

- **TAC:** The teaching-aids channel provides the flow for teaching aids, e.g. presentation and annotation tools. A TAC exists only in the root session.

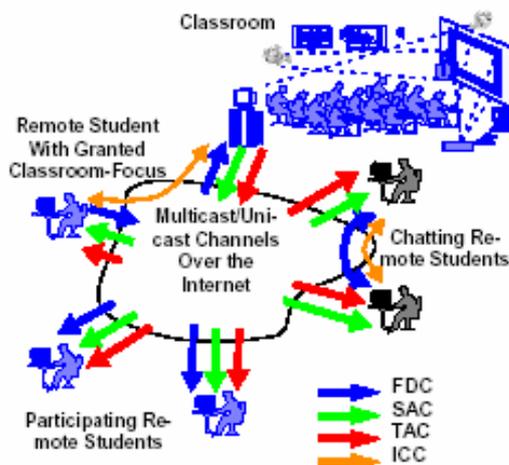


Figure 1– Virtual Communication Channels carry out the flow of data during a session

B. Session Establishment

A session is established if a participant sends a *session request* message to the session manager (lecturer). If the callee is willing to offer admission to the participant and establish a session, a *session grant* message is returned. Otherwise, the session request is *rejected*. Both, the session request message and the session grant message are manually initiated. The caller retrieves the session participants register (SPR) via the web-based interface (the description of a session, including the list of actual participants). After having selected an entry, he receives the address of the callee by selecting the appropriate entry and a session request message is automatically sent to the callee, as already described in the session announcement section.

After the session manager has sent the session grant message, he sends a registration message to the main session's SPR. This message extends his/her entry in the SPR by the description of the new participant.

Session participants provide continuous video or audio streams to the floor. To deal with bandwidth constraints only floor holders are allowed to send their streams. By that, session participants are not continuously aware of other attendees. Registered participants are marked within the SPR of the session holder. The initially established channel to send a session request message to the session holder becomes the ICC (described later) during the interaction period.

Sub-sessions are established in a similar manner. A participant sends a session request message to whom he wants to build a sub-session with. In case the callee grants the sub-session establishment, a new SAC and SPR are constructed and its SAC's address and the SPR are added to the sessions' tree. The callee becomes the sub-session's holder.

C. Session Join

Joining a session requires the address of the appropriate SAC. Traversing down the tree of sub-sessions in the applications session control window, starting at the root session's SPR or at the SPR of the current session - if a sub-session should be joined - retrieves this information. The session description being transmitted on the SAC contains the following information:

- Session's SAC-address, i.e. IP address and server port number of session holder.
- Master session's TAC-address, i.e. i.e. IP address and teaching-aids server process port number of session manager.
- Parent session's SAC-address.
- Real time media channel address.
- Session holder.
- Floor holder.
- Validation checksum of the SAC-data.

The information to connect to a SAC is used to perform join operation into the SAC (also into the SPR) of the session and, if applicable, a similar subscription to the TAC of the session. Receiving the information transmitted on the SAC provides the address of the floor, i.e. the FDC. The received data is used to set the focus of the participant to the current floor holder. Joining a sub-session changes the focus of the participant.

Note that although the session holder manages sub-sessions, the flow of time-based media between sub-session participants runs through the session manager, the holder of the main session. The reason is that the participant cannot unicast his streams to every member of the sub-session because of bandwidth constraints. Therefore he/she transmits real-time media packets via unicast to the session manager and the session manager then takes over the transmission of the packets to the rest of the sub-session's participants.

Certain measures should be taken to solve the late join problem, i.e. a participant enters the root session at a time, when the course has already started. The session control protocol should have the means to fully synchronize the new participants' teaching-aid tools with the actual state of the respective tools, as the rest of the session members views them. A manager logging all the activities in every TAC channel could be adopted and upon late join events will provide the SAC of the session with the needed information to initialise the late arrived participant's tools.

D. Session Termination/Leave

A session close is either performed individually by the session participant or, for all members of the session, by the session holder. The protocol provides the information to reconnect to the parent session or return to the root session.

If the session holder decides to close the session, this intent is signaled to the session before the SAC is closed. This session close-announcement is sent to allow the participants to have their focus automatically set to the parent session or to the root session otherwise. If a participant intends to cancel its participation in a sub-session, he/she either returns to the

parent or root session (a leave operation on the current FDC and SAC is committed) and reconnects to the appropriate SAC and FDC as described above. If a registration in the current session was once performed, the participation is de-registered if a session close is performed by the session participant.

E. Interaction Control

Interaction control is performed by using the ICC. A participant that intends to acquire the floor establishes an ICC to the session holder. If a new session must be created, the channel established to send a session request to another participant becomes the ICC between the session participants (see section B).

Through the ICC, a floor request is sent to the session holder. The session holder manually decides whether he/she wants to grant or refuse the floor to the requester. Refusing a floor request results in the termination of the ICC. If the floor is granted, a notification called floor grant message, is sent through the ICC to the requester. Besides the notification sent by the ICC, the change of the floor holder is, of course, announced on the SAC. The interaction policy imitates the known protocols of traditional university lectures: if the session holder intends to explicitly ask a participant, he or she asks the question by audio and video to the session and expects the callee to establish an ICC and follow the procedure as described above.

Until an interaction process is either closed by the session holder or canceled by the participant, the ICC remains established. Closing the ICC denotes the termination of the interaction. Using the ICC, request-reply like dialogues are possible. If the session holder intends to ask a participant explicitly, the session holder articulates the question on the floor and addresses the participant by voice – as done nowadays in traditional university-like lectures – and expects that the participant responds by a floor request message as described above.

In future approaches we plan to design and implement a social protocol to be used for interaction control in sub-sessions with small numbers of participants. Using this, floor control will be used only for video signal transmissions: while only one participant will be visible in the sub-session floor, every participant will be able to send audio streams through the sub-session's FDC without having to request and receive a token in advance. This modus of interaction will bring distance tutoring and group working closer to real face-to-face interaction.

4.3.3 Video Codec

WaveVideo is a wavelet-based, hierarchical coding and compression method for full color video. It includes basic video compression algorithms based on wavelet transformations, an efficient channel coding, filter architecture for receiver-based media scaling, and error-control methods to adapt video transmissions to the wireless environment. The tagged wire format allows filtering of quality attributes like color, resolution, frame rate and image quality. Using a joint source/channel coding approach, WaveVideo offers a high degree of error tolerance on noisy channels, still being competitive in terms of compression. Adaptation to channel conditions and user requirements is implemented on three levels. The coding itself features spatial and temporal measures to conceal transmission errors. Additionally, the amount of introduced error-control information is controlled by feedback. The intra- and

inter-frame coder and decoder have a simple and efficient design, which enables WaveVideo to run as a software-only solution.

The WaveVideo codec is already implemented in native C and on diverse platforms (Microsoft Windows NT, Linux, Sun Solaris and NetBSD).

4.3.4 Audio Codec

We intend to use a standard audio codec with high compression ratio and very good quality, almost close to CD-quality. A standard much this criteria is the one provided by the MPEG. Other candidate audio codecs originate from the ITU-T: G.711 ([31]) based on pulse code modulation digitizing algorithm using logarithmic coding, G.721 ([same as G.711]) using adaptive differential PCM coding and G.722 ([32]) based on sub-band ADPCM.

4.3.5 Real-Time Media Handling

Time-based or real-time media are termed any data that change meaningfully with respect to time. Audio clips, MIDI sequences, movie clips, and animations are common forms of time-based media. A key characteristic of time-based media is that it requires timely delivery and processing. Once the flow of media data begins, there are strict timing deadlines that must be met, both in terms of receiving and presenting the data. For this reason, time-based media is often referred to as *streaming media*: it is delivered in a steady stream that must be received and processed within a particular timeframe to produce acceptable results.

The format in which the media data is stored is referred to as its *content type*: QuickTime, MPEG, and WAV are all examples of content types. A *media stream* is the media data obtained from a local file, acquired over the network, or captured from a camera or microphone. Media streams often contain multiple channels of data called *tracks*.

Common operations on streaming media are:

- Capturing
- Processing (ex. applying effects, transcoding, de/multiplexing)
- Rendering (often also called playback)
- Streaming

A. Capturing – Processing - Presenting

In this section we will see how to take advantage of **the Java Media Framework version 2.1.1** (JMF [41], [42], [43]) functionality to do capturing and processing. In the following section we will look at how JMF enables streaming media through the use the RTP protocol.

The JMF design goals are:

- Be easy to program
- Support capturing media data
- Enable the development of media streaming and conferencing applications in Java.

- Provide access to raw media data
- Enable the development of custom, downloadable demultiplexers, codecs, effects processors, multiplexers, and renderers (JMF *plug-ins*)
- Maintain compatibility with JMF 1.0 Capturing is accomplished easily within the JMF: the developer just queries the available capture devices (microphone for audio, camera for video) using the CaptureDeviceManager Object and then creates a DataSource object. A DataSource in JMF encapsulates both the location of media and the protocol and software used to deliver the media. Then a processor object to encode the captured audio and video signals - using the codecs discussed in sections 5.3.3 and 5.3.4 respectively - has to be constructed and initialised with the created DataSource object as its input source. A Processor Object in JMF is a specialized media player: it takes a DataSource as input, performs some user-defined processing on the media data and then outputs the processed media data either to a presentation device (headphones, display) or to a DataSource object (to allow streaming or further processing by another processor). Our delivery platform will use two custom developed Processors to implement video and audio signal compression.
 - Encoding Processors –one for audio and one for video - will be implemented to encode the data from the capturing DataSource - using the WaveVideo encoder and an audio encoding standard respectively- and to store the compressed data in another DataSource to allow subsequently RTP packetization and streaming. This Processor will be activated, whenever the participant becomes the actual sender in the session.
 - Decoding Processors –one for audio and one for video - will be implemented to decode real-time media delivered from the network (after RTP depacketization) - using the WaveVideo decoder and an audio decoding standard respectively- and send the uncompressed data to a presentation device. These Processors will be activated, whenever the participant is acting as a receiver. Audio-video stream synchronization is performed easily by using JMF's methods.

B. RTP Packetization

At the time of writing this proposal, the Internet standard for transporting real-time data such as audio and video is the Real-Time Transport Protocol (RTP). RTP is defined in IETF RFC 1889 ([36]), a product of the AVT working group of the Internet Engineering Task Force (IETF).

RTP enables you to identify the type of data being transmitted, determine what order the packets of data should be presented in and synchronize media streams from different sources. RTP data packets are not guaranteed to arrive in the order that they were sent: in fact, they are not guaranteed to arrive at all. It is up to the receiver to reconstruct the sender's packet sequence and detect lost packets using the information provided in the packet header. While RTP does not provide any mechanism to ensure timely delivery or provide other quality of service guarantees, it is augmented by a control protocol (RTCP) that enables you to monitor the quality of the data distribution. RTCP also provides control and identification mechanisms for RTP transmissions.

RTP-streams reception and transmission is accomplished in JMF through the use of the JMF RTP-API.

A Player or Processor handles the presentation of an incoming RTP stream: in our case the already described decoding Processors. To receive and present the multiple streams (RTP sessions¹), we use an RTPManager, a JMF object that is capable of handling multiple RTP sessions. Subsequently the extracted streams are fed to the decoding Processors to decode the audio/video media and present them via presentation devices.

An RTPManager object is used to initialize and control multiple RTP sessions to stream time-based media across the network. The data to be transmitted are compressed from the encoding Processors and output into DataSource objects. These DataSource objects are then passed to the RTPManager and transmission can be started (or controlled) through the use of RTPManager's methods.

Note that we have to implement custom packetizer and depacketizer plug-ins for the WaveVideo format, as this format is not a standard one and thus this payload type is not supported by the existing RTP specification.

Below we present two diagrams showing the complete handling process of real-time media, one is showing the handling on the sender's side and the other the handling on the receiver's side.

¹ There is some confusion on the term session, which is sometimes used for a conference and sometimes for a single media stream transported by RTP. In this document we use the term with both meanings, but in the RTP section we will use it only with its second meaning to resolve ambiguity.

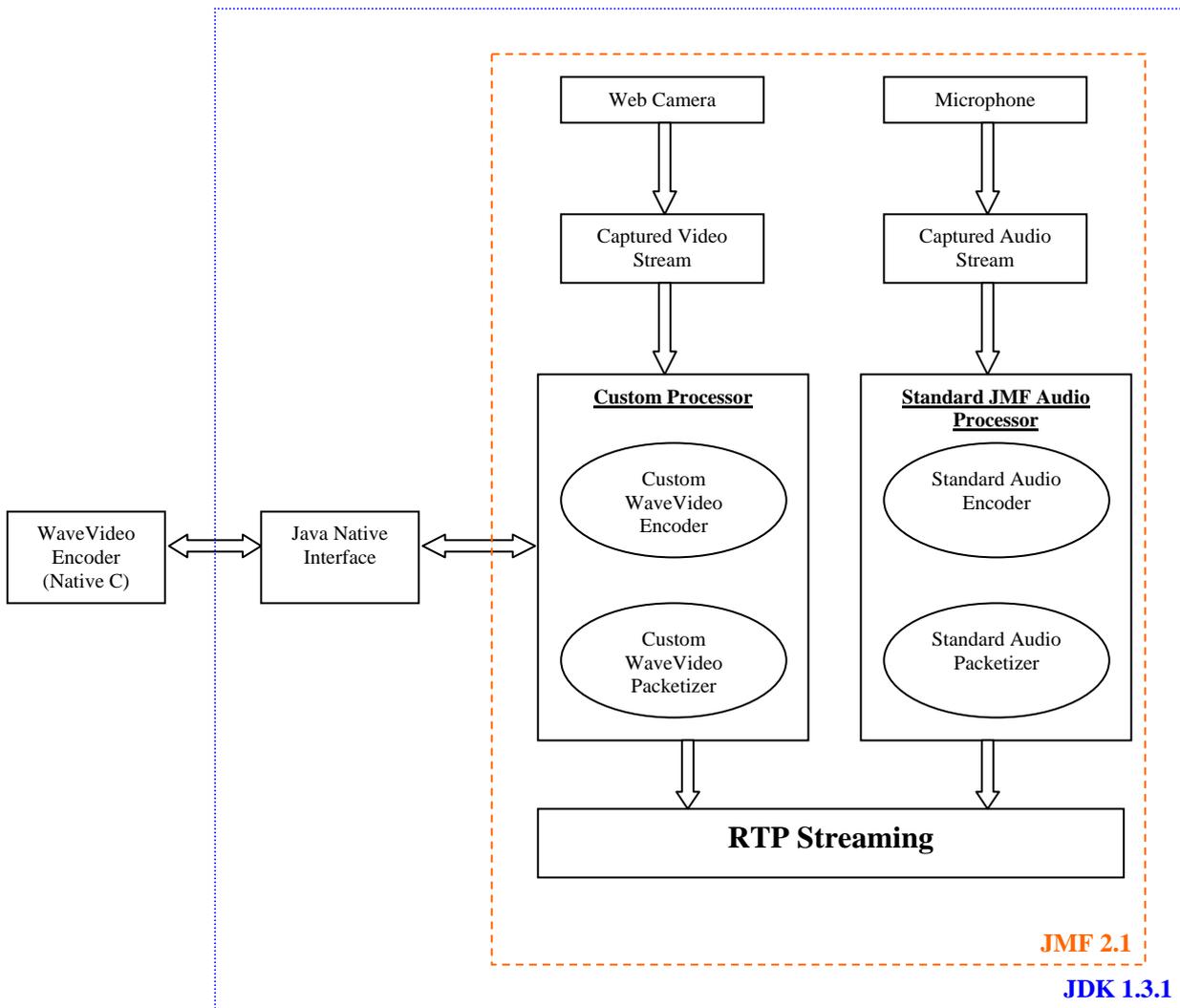


Figure 2– Real -Time media handling on the sender’s side

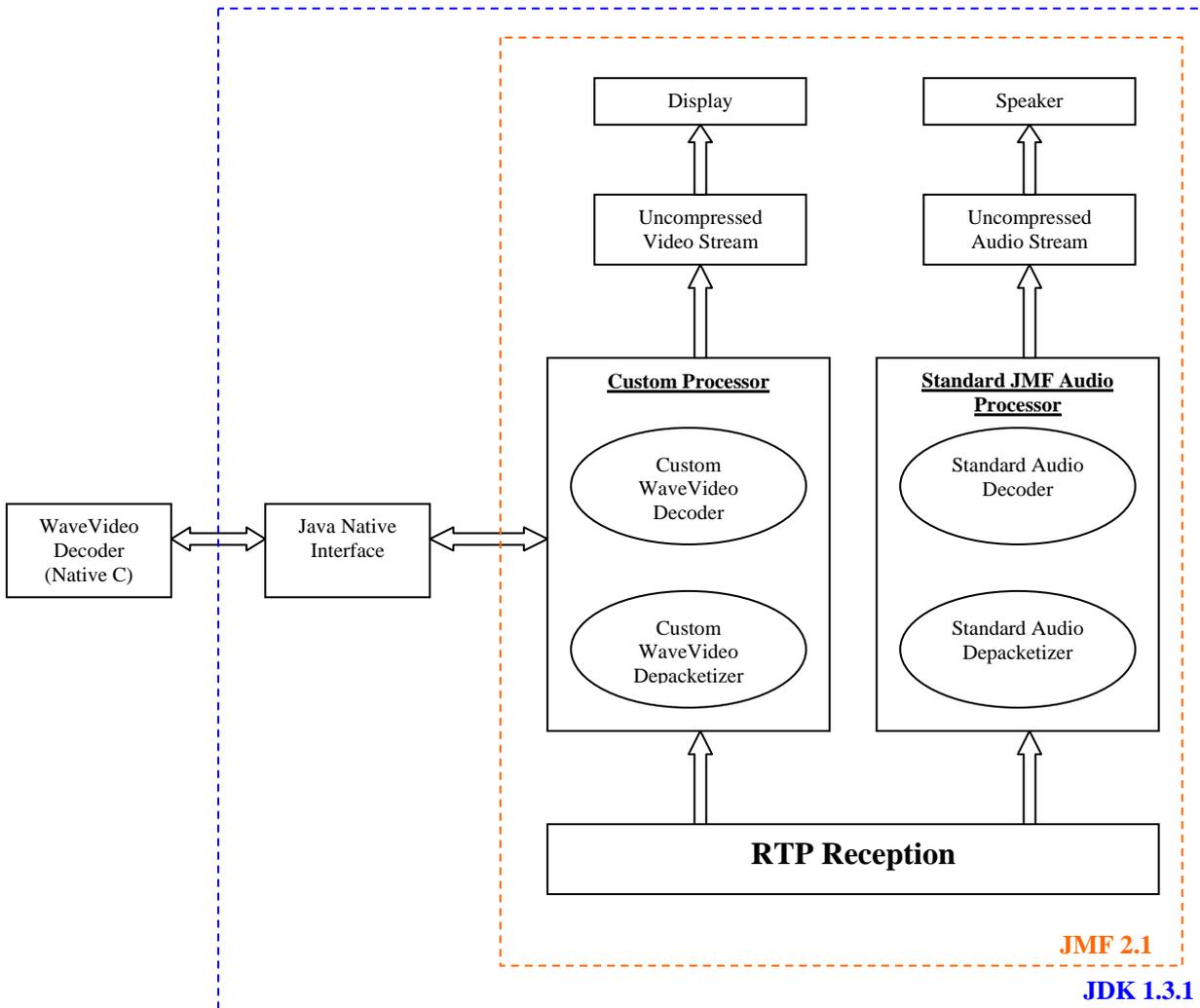


Figure 3 – Real-Time media handling on the receiver’s side

Important Note:

The decision to use JMF 2.1 as the media framework to implement real-time media handling and streaming is based on the fact, that this API completely provides all we need to implement these programming tasks. However, we should first test JMF’s performance before adopting it.

Some tests on packet loss, delay and delay jitter by RTP-streaming using JMF 2.1 have already been performed within the scope of a diploma thesis ([44]) in our department, reporting unacceptable results for some kinds of access technologies. However, the thesis just provided measurements and didn’t go further to find out which was the erroneous component (JMF or lower levels of the end systems) or where the bottleneck was. During the design stage of ETL&L II and before starting with the implementation, we will take the experiments on JMF’s RTP-streaming module a bit further, discover where the points of degradation are and try to resolve them. In case this is estimated to be more time-consuming than the learning cycle of other RTP APIs, we will use native RTP APIs (different for every platform) and maintain cross-platform nature by using Java to control them via the Java Native Interface.

4.3.6 Application Adaptivity

As a best effort network without any network-level QoS guarantees is assumed, we have to implement mechanisms to provide at least application-level QoS: adapt the application bandwidth demands on every participant's side dynamically. For this service network monitoring and reporting is needed, as well as a centralized decision manager. The approach is inspired by [22] and similar, but not identical, work can be found in [23], [24] and [25].

As already mentioned, data transport provided by RTP is augmented by RTCP, a control protocol, that provides monitoring of the data delivery, minimal control and identification functionality. The specification of RTP protocol (RFC 1889) defines five types of RTCP packets, from which three of them are of interest to us for this service:

- **Sender Report:** carries transmission and reception statistics from participants that are active senders.
- **Receiver Report:** carries reception statistics from participants that are not active senders.
- **SDES:** source description items, including a CNAME. This type of packet provides unique identification of a session participant

Compound RTCP packets are sent periodically from every receiver to the QoS agent. These contain all the necessary information for quality-of-service monitoring, such as number of packets lost, measure of interarrival jitter and timestamps needed to compute an estimate of round trip time (RTT) between the active sender and the receiver issuing the report. The QoS agent then analyses the reports of every receiver and the actual network state seen by him is classified as either unloaded, loaded or congested, which means that the packet-filtering module will either increase, hold or decrease respectively the streaming bandwidth to the particular receiver. The algorithm used to draw a decision about the network state of a receiver from the statistics gathered, as well the algorithm to compute the percent to which bandwidth will be increased or decreased on a media quality adaptation decision, are open research issues (during the project implementation. Moreover it is to be decided how often the QoS Agent will perform an analysis on the network state of every receiver and draw a decision about its network state (after a number of RTCP report receptions). This frequency should be different for every receiver and should allow for maximisation of the resolution of the statistics according to bandwidth constraints.

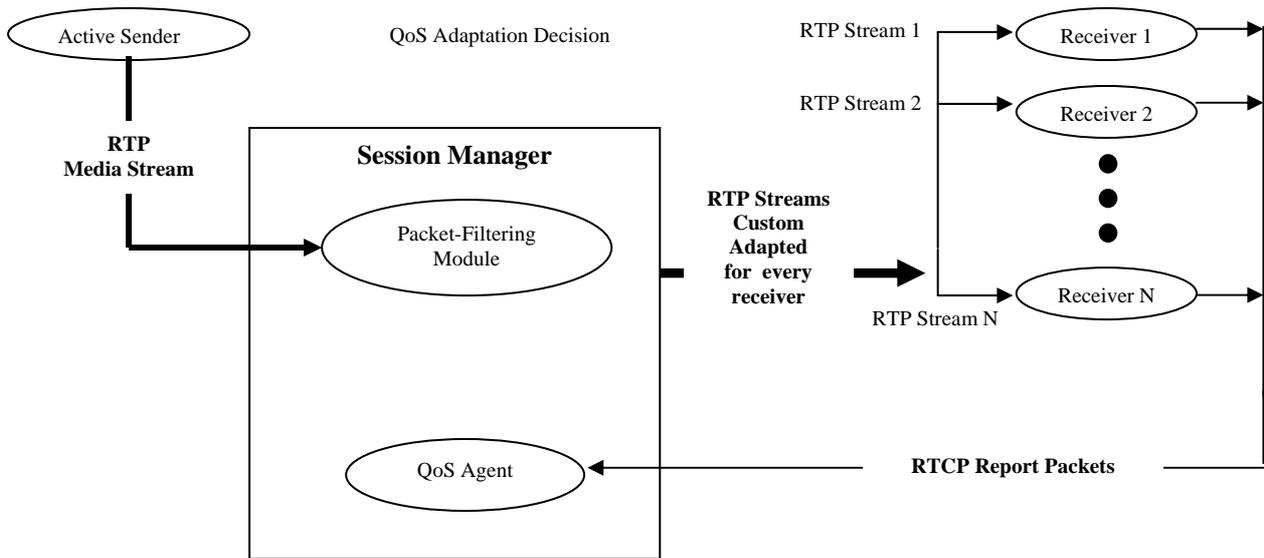


Figure 4 – Dynamic Application - Level Quality of Service Adaptation (process followed in a period of the iterative Report-Analysis-Decision-Adaptation Cycle)

The QoS adaptation decision is subsequently announced to the packet-filtering module, which is responsible for “executing” the decision. It keeps a list of records with all the participants and the actual quality and bit rate of real time data, that are transferred to them in RTP packets. When receiving a QoS adaptation message from the QoS Agent, the packet-filtering module maps the bandwidth increase or decrease decided into changes in the encoding of the media streams. Particularly:

- For audio streams, because most audio encodings have a single constant bit rate, the packet-filtering module can signal the active sender to switch to a lower quality audio encoder. However, because audio streams usually have not high bandwidth demands, this feature will not be supported.
- For video streams, normally the control module would map the bandwidth increase/decrease into video stream quality and bit rate parameters used by the encoder. Then for every group of receiver, that share the same bandwidth, the captured video signal would have to be re-encoded to consume the prescribed bandwidth and re-packetized. But here hides the power of the WaveVideo codec: it provides a layered encoding of the video signal, with each layer representing additional a detail level (frequency subbands). Then packet filtering is made easy by just analysing the video packet and dropping the sufficient enough detail layers to enable the video signal transmission with the available bandwidth. A similar router-based media scaling procedure is described in [21].

4.4 Application Layer

4.4.1 Application Graphical User Interface

In the present section we will try to give a rough description of the application’s graphical user interface (GUI) for the three kinds of the system’s users: the student, the lecturer and the administrator.

A. Student’s GUI

As the reader can see in figure 5, the student’s GUI consists of the following windows (explained as numbered in the figure):

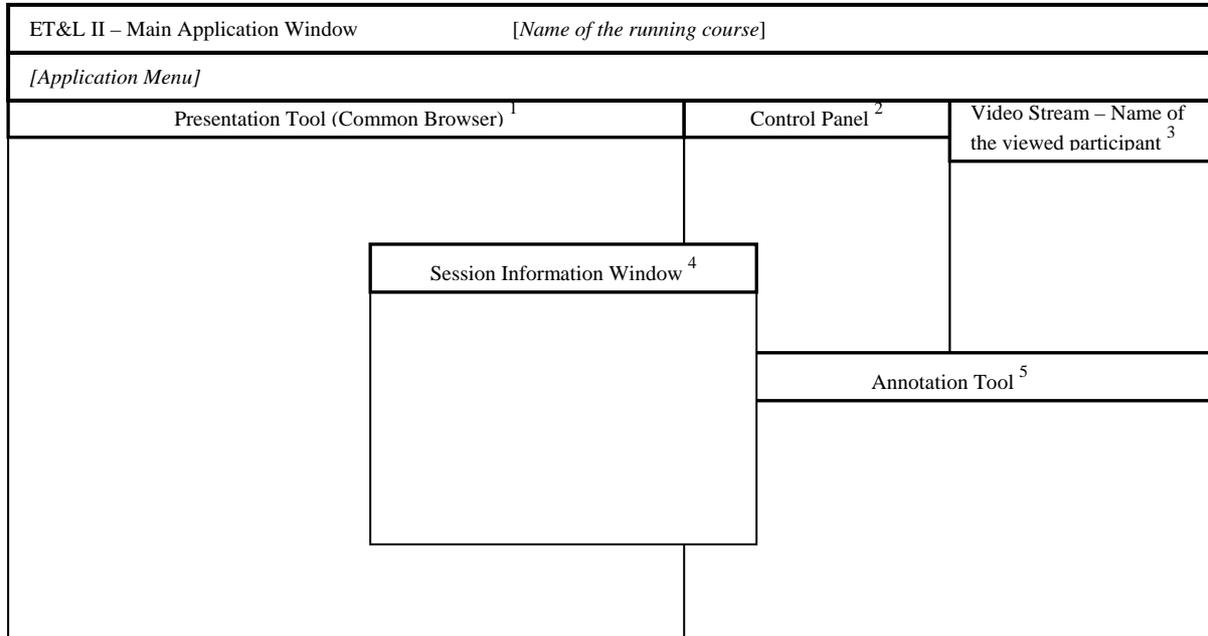


Figure 5 - Student’s Graphical User Interface

- 1: The Presentation Tool, which is a window encapsulating a common Web browser. You can read more on the decision to use a browser as a presentation tool in the next subsection.
- 2: The Control Panel, which will contain audio controls, a leave course button, a buttons to start/stop video stream transmission, one button to request floor control, an announcement text box, where incoming session establishments requests to the user will be announced and a session grant and a session deny button.
- 3: The Video Stream window, where the active floor holder is presented.
- 4: The Session Information Window, where a tree-form view of the available sessions and its participants are presented. The name of each session is accompanied by the name of the session holder and the floor holder of every session is highlighted.
- 5: The Annotations tool that is used among the participants to share notes. It is the blackboard of the distance-learning platform.

A known problem in such systems is the overlapping of windows: in case the student opens up the session information window, as shown in the previous figure, it may overlap with windows that are “useful” for the lecture, such as the presentation or the annotations window. We will program the application in such a way, that all the control windows will open up in a specified space of the application’s main window to ensure that the opening of control windows won’t disrupt the student’s attention.

B. Lecturer's GUI

The lecturer's graphical user interface will be similar to the student's with some additional functionality. The rights of the lecturer are to be decided, e.g. whether the lecturer will have the right to kill sub-sessions between students, drop students out of the session (like he would do in a normal class with hot-tempered students). Below we describe the add-ons to the students GUI resulting the lecturer's user-interface:

- In the Presentation Tool, controls should be added to allow the lecturer to load the preferred presentation and to control its flow (next/previous slide, first/last slide, show table of contents etc.)
- In the Control Panel an announcement text box will be added, where incoming floor control requests to the lecturer will be forwarded, plus floor control grant/deny/cancel buttons to allow full control over the interaction during the session.
- In the annotations' tool controls have to be provided to the lecturer to load pre-made drawings or texts, to erase the whiteboards contents, to load new toolkits etc.

C. Administrator's GUI

For the administrator GUI certain controls should be provided to administer the virtual classrooms, such as create session description and administer announcement site. Moreover the GUI should enable an easy way to administer users, i.e. lectures and students, like dropping malicious users out of a session or handling a session manager's crash.

4.4.2 Teaching-Aid Tools

A. Presentation tool

We decided to use a common web browser embedded in the main application window as the presentation tool. The reasons for drawing such a decision are the following:

- Low design and implementation complexity and thus short implementation time. We only have to decide which browser to use (maybe use the OS's default browser) and research the Java technology that will carry out the embedding.
- Widely accepted content format. The HTML format is a widely spread standard and moreover most existing presentation and document processing applications support content exporting into HTML format. Additionally through the MIME mechanism we are able to open documents of other formats in a browser (e.g. Adobe Acrobat files). A special conversion tool (Presedia Producer [8]) could be used for the lecturers, who make use of Microsoft PowerPoint as their presentation production tool and wish to preserve the events offered by PowerPoint. This tool simply converts presentations from PowerPoint format into Macromedia Flash format.
- As the standardization trend for Reusable Learning Objects (RLOs, e.g. SCORM [9]) is using XML as a description language, using a web browser as a presentation tool means that we don't have to care for compatibility between RLO and presentation tool, just use a browser that supports the XML version

(XML and DTD parser) used by the learning objects. Such browsers exist already (e.g. CiTEC's DocZilla based on Mozilla browser)

The presentation tool – i.e. the flow of the presentation - will be operated by the lecturer and presented synchronously to all the participants of the main session. The advance of slides on the student's side will be done either automatically on lecturer's will by using push technologies or manually by the student by pressing a link on a sidebar of the presentation screen, that will contain a table of the presentation's contents.

Finally the downloading at once of the whole presentation material to the student's workstation during session join should be supported, as long as the student is joining the session before the course has started. This will save some network resources during the conference.

B. Annotation tool

For annotation (often called whiteboard) a Java Sharing Aware Application will be implemented, enabling text processing and basic drawing.

5. Dissemination

The goal of making the ET&L II platform known to the community researching distance learning, as well as to the educational community generally, is handled in this chapter.

In order to achieve this in to the most efficient grade, we will make use of high-publicity initiatives that cope with e-learning (or generally computer aided education) matters. As described in the work plan, during the dissemination task we will try to establish contacts to the aforementioned initiatives to disseminate our platform through their already existing affairs. Such initiatives in the ETH domain are:

- **ETH World:** a strategic initiative to prepare ETH Zurich for the information age. Its objective is to create a universal virtual communication and cooperation platform, supporting the activities of everyone working or studying at ETH. ETH World will augment the present physical locations, "Zentrum" and "Hönggerberg", by a virtual space, creating a third, virtual campus for ETH Zurich. This will be achieved by extending the physical infrastructure of ETH Zurich by means of advanced information and communication technology to form a novel "infostructure". In this environment, research groups and teaching and learning communities can cooperate without limitation of time or location. ETH World also supports new processes for the management and services of the university. In addition, the virtual space will help to strengthen the links between the alumni, business and industry firms and ETH. (...) Parallel to this, ETH World is being built through a growing number of individual projects, developing e-learning, research tools, information management, infrastructure elements and community building. Thus ET&L II is a project that perfectly fits the objectives of ETH-World. One of the existing ETH-World projects is of special relevance to ET&L II. The project *DEEDS (Distance Education at ETHZ: Delivery System)* aims at creating a universal portal for access to distance education content, which has been developed at ETH. The specific delivery platform that ET&L II develops will be integrated as a delivery platform supported by DEEDS, such that the DEEDS portal will allow seamless access to lectures that are offered through the ET&L II platform.

- **NET (Network for Educational Technology)**: a project organized by the rectorate of the ETH Zurich. NET looks into the integration of computer-based educational systems into the higher education domain, supports developers and users of such systems, coordinates related projects and encourages cooperation. The NET therefore has both a mandate to provide services to teachers and to transfer new ICT-based systems to practical usage. The ET&L project has already established a good working relationship with the NET; ET&L II will build upon these contacts to assure that the project's results will be accessible to and used by the educational community.

Apart from the above “ETH-internal” contacts establishment, during the project lifetime we will try to get advantage of the interaction with international research peers and similar initiatives to make the existence of our system internationally known. It is understood that the research results obtained during the project will be disseminated through the established channels (workshops contributions, conference and journal articles).

Last but not least we will contact industrial partners in order to discuss the possibility of an industrial exploitation of the developed platform. In case industrial interest is raised, an ET&L II follow-up project performed by industrial partners could take our platform further, by following a typical product development cycle (implement-evaluate-test).

6. References

General

- [1] Technology Distance Education and National Development, University of British Columbia
- [2] Deborah Jasek: “Distance Learning Defined”, Center for Professional Development Texas Transportation Institute
- [3] ET&L Project Proposal, Professor B. Plattner, TIK-ETH Zurich, <http://www.tik.ee.ethz.ch/~etl/Docu/ProjectProposal.pdf>
- [4] “Swiss Virtual Campus”, <http://www.virtualcampus.ch/>
- [5] T. Walter, H. Hänni, M. Hildebrand, S. Hinni, P. Kunz, B. Plattner: “Das Teleteaching System Telepoly, Implementation und Erfahrungen, Praxis der Informationsverarbeitung und Kommunikation, No. 2/99, 1999
- [6] IEEE Learning Technology Standards Committee (LTSC) (<http://ltsc.ieee.org/>)
- [7] International Federation for Information Processing (IFIP), Technical Committee 3 (TC3) (<http://www.edu.ge.ch/cptic/prospective/projets/ifip/>)
- [8] Presedia Producer: <http://www.presedia.com/products/producer.htm>
- [9] “SCORM v1.2 (Sharable Content Object Reference Model), Advanced distributed Learning Network (ADLNet, <http://www.adlnet.org>)
- [10] DocZilla: A Mozilla-based SGML/XML/HTML browser developed by CiTEC (<http://www.doczilla.com>)

Session Announcement:

- [11] M. Handley, V. Jacobson: “SDP: Session Description Protocol”, Network Working Group, Request for Comments 2327, 1998
- [12] Freed, N. and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies", RFC 2045, November 1996
- [13] Freed, N. and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types", RFC 2046, November 1996
- [14] D. Eastlake, C. Smith, D. Soroka, “HTTP MIME Type Handler Detection”, RFC 2936, September 2000
- [15] Java Specification Requests 141, Java SDP API, Final Release announced for December 2001 (<http://www.jcp.org/jsr/detail/141.jsp>)

Session Management:

- [16] L. Ruf, Th. Walter, B. Plattner: “An Interaction Control Architecture for Large Chairperson-controlled Conferences Over the Internet”, Proc. of the 7th International Workshop, IDMS 2000, Enschede, The Netherlands, October 2000
- [17] M. Handley, H.Schulzrinne, E.Schooler, J. Rosenberg: “SIP: Session Initiation Protocol”, IETF SIP Working Group, Request For Comments 2543, July 2001
- [18] M. Handley, C. Perkins, E. Whelan: “SAP: Session Announcement Protocol”, IETF Network Working Group, Request For Comments 2974, October 2000
- [19] K. Singh, G. Nair and H. Schulzrinne: “Centralized Conferencing using SIP”, Columbia University
- [20] Packet-based multimedia communications systems (H.323), ITU-T Recommendation, 02/1998

Application Adaptivity:

- [21] R. Keller, S. Choi, M. Dasen, D. Decasper, G. Fankhauser and B. Plattner: “An Active Router Architecture for Multicast Video Distribution”, Infocom 2000, March 2000, Tel Aviv, Israel
- [22] I. Busse, B. Deffner, H. Schulzrinne: “Dynamic QoS Control of Multimedia Applications based on RTP”, GMD-Fokus, Berlin, May 1995
- [23] W. Zhu and N.Georganas: “JQoS: Design and Implementation of a QoS-based Internet Videoconferencing System using the Java Media Framework (JMF)”, Multimedia Communications Research Lab, University of Ottawa
- [24] SanKu Jo and P.Cantrell: “A dynamic QoS Control Scheme for Videoconferencing in a Heterogeneous Internet”, Lawrence Berkeley National Laboratory/Texas A&M University
- [25] D. Sisalem, A.Wolisz: “MLDA: A TCP-friendly Congestion Control Framework for Heterogeneous Multicast Enviroments, GMD FOKUS, Berlin, Germany

WaveVideo:

- [26] G. Fankhauser, Marcel Dasen, Nathalie Weiler, Bernhard Plattner and Burkhard Stiller: “The WaveVideo System and Network Architecture: Design and Implementation”, TIK, Technical Report No. 44, June 1998.
- [27] M. Dasen, G. Fankhauser, B. Plattner: “An Error Tolerant, Scalable Video Stream Encoding and Compression for Mobile Computing”, ACTS Mobile Summit 96, Granada Spain, November 27-29
- [28] 1996 Alexander Schwab: “Temporale Kompressions-Algorithmen in WaveVideo“, TIK-ETH Zurich, Diplomarbeit, 1997
- [29] Pankaj N. Topiwala: “Wavelet image and video Compression“, Kluwer Academic Publishers, 1998
- [30] M. Vetterli, J. Kovacevic: “Wavelets and Subband Coding“, Prentice Hall, 1995

Audio Codecs

- [31] Pulse Code Modulation (PCM) of voice frequencies. ITU-T Recommendation, 11/1988 (G.711)
- [32] 7 kHz audio coding within 64 Kbit/s. ITU-T Recommendation, 11/1988 (G.722)
- [33] Speech coders: Dual rate speech coder for multimedia communications transmitting at 5.3 and 6.3 Kbit/s. ITU-T Recommendation, 03/1996 (G.723)
- [34] Coding of speech at 16 Kbit/s using low-delay code excited linear prediction. ITU-T Recommendation, 09/1992 (G.728)

Real-Time Media Streaming

- [35] H. Schulzrinne, A. Rao, R. Lanphier: “Real-Time Streaming Protocol (RTSP)”, IETF Network Working Group, Request For Comments 2326, April 1998
- [36] H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson: “RTP: A Transport Protocol for Real-Time Applications“, IETF Audio-Video Transport Working Group, Request for Comments 1889, January 1996

Similar Approaches

- [37] A. Abdel-Hamid, S. Ghanem, K. Maly and H. Abdel-Wahab: “The Software Architecutre of An Interactive Remote Instruction System For Heterogeneous Netwrok Envirnonments”, 6th IEEE Symposium on Computers and Communications, Tunisia, 2001
- [38] K. Maly, C. Wild, C.M. Overstreet, H. Abdel-Wahab, A. Gupta, A. Youssef, E. Stoica, R. Talla and A. Prabhu: “Interactive Remote Instruction: Lessons Learned”, Computer Science Department, Old Dominion University, Norfolk Virginia, presented in WebNet96
- [39] Microsoft Netmeeting Website: “<http://www.microsoft.com/windows/netmeeting/>”
- [40] Click To Meet Website: “<http://www.fvc.com/products/clicktomeet.htm>”

Java Media Framework:

- [41] JavaTM Media Framework 2.0 Programmer's Guide, November 19, 1999, Sun Microsystems (<http://java.sun.com/products/javamedia/jmf/2.1.1/guide/JMFTOC.html>)
- [42] JavaTM Media Framework 2.0 API Specification, March 19, 2001, Sun Microsystems (<http://java.sun.com/products/java-media/jmf/2.1.1/apidocs/>)
- [43] Linden De Garmo: "Core Java Media Framework", Prentice Hall, 1999
- [44] J. Colombie and N. Sudarevic: "Jini at your Doorstep", Semester Thesis, TIK-Communication systems Group, 2001

Internet Access Providers

- [45] Bluewin ADSL 512 (http://www2.bluewin.ch/services/adsl/broadway_light_d.html)
- [46] Cablecom Comfort Hispeed Internet (<http://www.cablecom.ch/de/privinpr.html>)