Using Multimedia Document Communication services on top of T.120 protocols[1]

Jaime Delgado, delgado@ac.upc.es
José Jesús Acebrón, acebron@ac.upc.es
Telematic Applications Group
Computer Architecture Department
Universitat Politècnica de Catalunya (UPC)
Barcelona, Spain

Abstract

One of the most relevant problems in electronic publishing is the interoperability between different systems in order to interchange multimedia information. Part of these problems could be solved by providing standardized multimedia document communication services such as storing, distribution, manipulation, and even more complex ones such as joint presentation/ viewing or joint synchronous editing. This paper deals with the technical issues of designing such services based on standardized protocols, such as the ITU-T T.120 Recommendations. Focus is made on adding multipoint facilities to the document communication services taking advantage of the T.120 Recommendations.

The paper discusses how to use the facilities provided by the T.120 Recommendations to design multimedia document communication services such as Joint Presentation/Viewing (PV) or Joint Synchronous Editing (SE) that run on heterogeneous environments. The fast development and penetration of the T.120 Recommendations make them an adequate platform in which to base other applications. The facilities of T.120 cover, between other features, multipoint communication, token management, conference control, etc.

The T.120 multipoint communication Recommendations are the base for that design, and are described briefly. These services could be based only on MCS, but the paper discusses how to map the service onto the complete T.120 specifications, which also include GCC, that provides a wide variety of conferencing management functions that facilitate the development of document communication services.

The paper presents PV and SE, as examples, describing the rules to be followed during a PV or SE session. The paper also analyzes the most relevant issues of the services: start-up synchronization, late joining, early leaving, co-ordination, update, closing and final document copy. These aspects may cause problems in the design and implementation.

The proposed design uses the conferencing capabilities specified in GCC to support PV and SE services. The paper discusses how to solve the relevant issues of PV and SE using the facilities of GCC (and T.120, in general).

[1] This work has been partly supported by the Spanish government (TIC95-0903-C02-01) and has been partly developed in collaboration with ETSI (European Telecommunications Standards Institute) sub-technical committee TE2.

1. Multimedia document communication services

This section provides a description of the two types of multimedia document communication services: basic and complex. The basic services can be classified into stand-alone and non stand-alone services. The complex services are based on the basic ones.

Services can act on whole multimedia documents or on document fragments. Document fragments are parts of documents with an entity, such as chapters, sections, footers, pages, images, etc. These fragments are the minimal piece that can be manipulated and interchanged.

1.1 Basic services

These services can not be decomposed into other services. Into this category, two types of services can be classified:

Stand-alone services: these basic services can be used alone or in combination with other services to provide more complex ones. These services are: storing, distribution, retrieval, storing and retrieval, and manipulation.

Non stand-alone services: these services must be used in combination with other services, since it makes no sense to use them alone. These services are: pointing, multi-pointing and token-interchange.

These basic services are specified in [ETS1] and [T.190].

1.2 Communication protocols for basic services

For the provision of the basic services, existing communication protocols such as DFR [DFR] and DTAM-DM [DTAM-DM] can be used.

DFR (Document Filing and Retrieval) provides for managing documents and other objects inside remote document stores in distributed systems.

DTAM-DM defines a general set of operations and a protocol for remote multimedia document manipulation, which can be used for accessing documents with an arbitrary internal representation. DTAM-TK defines another ASE (Application Service Element) for application token interchange.

1.3 Complex services

Some complex document communication services are:

- Joint multimedia document presentation/viewing (PV) with 1 or n users presenting (PV1, PVn).
- Joint synchronous editing (SE).
- Sequential multimedia document production (SP).
- Asynchronous multimedia document production (AP).

These services are being specified by ETSI (in a 2nd part of [ETS1], [ETS2]) and ITU-T (in the T.190 [T.190] series of Recommendations). This paper deals with the PV and SE services.

1.4 Communication protocols for complex services

In some applications, like PV or SE, there is also a need to interconnect more than two users. In this case, it would be very helpful to have multipoint communication services available like those provided by the ITU-T T.120 series of Recommendations [T.120], as introduced in 2.

To achieve this objective, a first step is to extend existing applications (like DTAM-DM) in order to be able to cope with T.120. However, it is also important not to modify the current functionality of those standards, but only to add the possibility to interconnect with multiple systems.

2. Multipoint communication specifications: T.120

A key difference between basic and complex services is that, in the latter case, they could be simplified using multipoint communication. To add this, a good approach is to use the ITU-T T.120 Recommendations, that are described in the following sections. How to use them to provide multimedia document communication services is detailed in further sections (see sections 3.3.2 and 4).

2.1 Description of the T.120 Recommendations

2.1.1 The T.120 series

The T.120 protocols provide a means of telecommunicating all forms of Data/Telematic media between two or more multimedia terminals, and of managing such communication. T.120 thus provides a multipoint data communications service that has application in all forms of multimedia communication.

The T.120 protocols are suitable for use on all types of network: PSTN (Public Switched Telephone Network), ISDN (Integrated Service Digital Network), CSDN (Channel Switched Digital Network), PSDN (Packet Switched Digital Network), B-ISDN (Broadband ISDN), LANs (Local Area Networks). T.120 provides the capability for seamless interworking of applications between terminals connected to different networks.

The T.120 protocols can handle one or more simultaneous "conferences"; any terminal may participate in more than one of these if authorized to do so; the convener of any one conference may control the participation in that conference and the information which flows in that conference.

2.1.1.1 Multipoint Multimedia Communication

Traditional services have been constrained to point-to-point operation. In order to support group activities such as meetings, conferences, etc., involving physically separated participants, there is a requirement to join together more than two locations; the term multipoint communication simply describes the interconnection of multiple terminals. Normally, a special network element, known as a Multipoint Control Unit (MCU), is required in order to provide this function.

Figure 1 gives an example of a multipoint conference configuration.

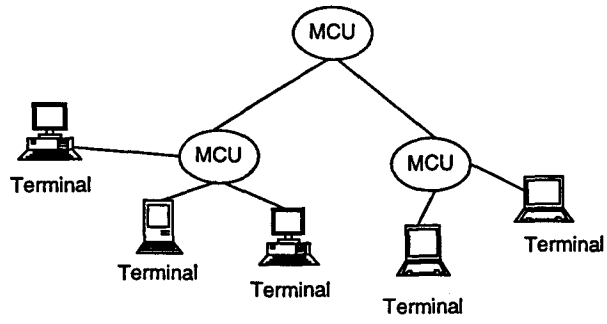


Figure 1: Example of multipoint conference configuration showing Multipoint Control Units (MCUs) and conferencing terminals

2.1.1.2 The T.120 Protocol Suite - System Model

The T.120 model comprises the communications infrastructure and the application protocols that make use of it. Figure 2 shows the full model with both standardized and non-standardized applications. T126 and T.127 standard application protocols are not considered in this paper.

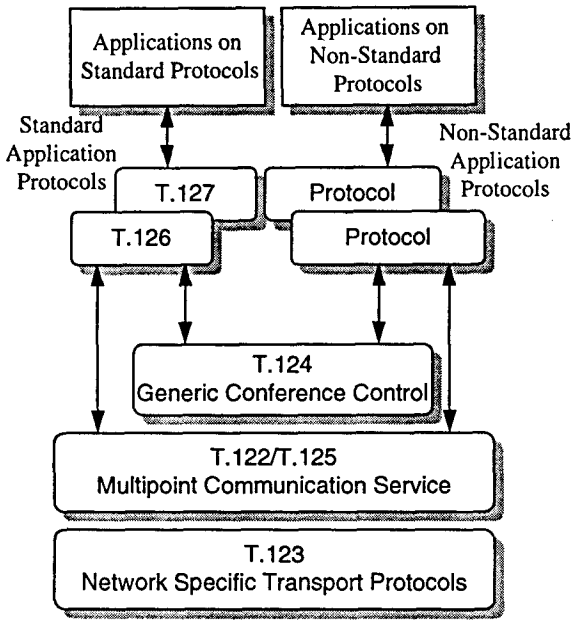


Figure 2: T.120 System Model.

2.1.2 T.122/T.125 - Multipoint Communications Service (MCS)

MCS provides a general multipoint connection-oriented data service. It collects point-to-point transport connections and combines them to form a Multipoint Domain. Within that domain, a large number of logical channels can provide one-to-one, one-to-many and many-to-one data delivery. Nodes within an MCS Domain are hierarchically organized in a tree structure. Data delivery normally follows the most efficient path to the nodes that are to receive the data, but a mechanism is also provided, if there is a need, to guarantee that data from multiple source nodes is received in the same sequence at all nodes. MCS acts as a resource provider to the layers above, providing channels and token resources on demand. A large number of tokens are provided for applications to use for co-ordinating events and processes.

The MCS protocol is specified in T.125. It includes rules on how to use the transport service.

2.1.3 T.124 - Generic Conference Control (GCC)

GCC provides a set of services for setting up and managing a multipoint conference. It provides access control and arbitration of capabilities. GCC facilities are used by applications to co-ordinate independent use of the MCS channels and tokens within the same Multipoint Domain. Nodes can join and leave meetings at any time and GCC facilities can be used to query a node to find a desired meeting. GCC also makes a centralized registry facility available to applications in order to identify dynamically assigned channels and tokens.

It defines a service on top of MCS. The service provides for Conference establishment and termination, Conference roster, Application roster, Remote actuation, Conference conductorship, Bandwidth control, Application registry, etc.

GCC is intended to be used by specific applications, like still image conferencing, high level audio and video control, multipoint binary file transfer, and other standardized or proprietary applications.

These applications using GCC may also make use directly of MCS at the same time. The GCC Recommendation details how the MCS primitives are used.

3. Design of the PV and SE services

3.1 Description of the presentation/viewing service

The PV complex services consist in the remote presentation or joint viewing of a document, where a user (the presenter) presents the document to other users (the viewers).

The basic principles of PV are:

- Several users co-operate to jointly present/view one or more documents. Every user has a full copy of the document(s).
- One user (the initiator) will initiate the PV session.
- The user presenting the document at a given time is called presenter, the other users are the viewers.
- The presenter communicates to all the viewers.
- Users may establish several one-to-one associations or one one-to-several association.
- When a presenter points to a part of the document, this operation is sent to all viewers.
- Optionally, the initiator, or another user, may take the role of moderator of the PV session, who is in charge of distributing the PV token, introduced below, if any.
- The moderator can control who is allowed to present the document. The moderator can be the initiator, a pre-defined user, or the moderator role may change during the PV session.

The PV complex services can be categorized according to the number of users that can present a document, the two following services are considered: PV1, with only 1 presenter, and PVn, with n presenters.

The PV1 complex service consists on the remote presentation or joint viewing of one or more documents, where only one user presents the documents to the other users (no PV token is needed).

The PVn complex service consists on the remote presentation or joint viewing of one or more documents, where several users present the documents to the other users.

In PVn, all users are able to present the document to the other users, but at a given time, only one user can be presenting, while the others are viewing.

The main differences with PV1 are:

- more than one user may present;
- a PV token is needed to control who is presenting.

3.2 Description of the joint synchronous editing service

The following are the principles of the Joint Synchronous Editing (SE) service:

- Several users have to edit a multimedia document (or documents) together.
- An initiator user starts the joint synchronous editing session.
- The initiator of the session could also be the moderator of the service.
- The initiator invites all the users that have to participate in the document editing. When the session has started, no more users are allowed to entry the session.
- If the document already exists (it is not a new document), every user retrieves or receives a copy of the document that is to be edited. When the users have their copies of the document(s), the editing can start.
- Only one user can be editing (the first editor is the initiator) the document at a given time. If another user wants to edit the document, he/she must request the current editor (or the moderator, if exists) for permission. This will be usually done using some token mechanism.
- The following steps are followed once the editing session has started:
  1. The current editor (or the moderator, if exists) can give editing permission to another user.
  2. The user can edit the document. The updates of the document are notified to all the other users in the session.
  3. Another user can ask for permission to edit the document. The current editor is responsible for giving the permission when he/she considers it adequate.
  4. Steps 1-3 are repeated during the session.
- Users that are no more interested in the editing can leave the session at any time. Of course, the editor should not leave without giving editing permission to another user.
- The initiator can finish the session at any time, regardless if he/she has the editing permission or not. It would better to ask first for the editing permission, then finish the session.

We will not consider the figure of a session moderator, but in some applications it can be useful to have such an element. The main function of the moderator (as in usual meetings) is to decide to give the turn (token in our case) to the most adequate participant in each case, following some criteria (it is not necessarily a FIFO policy). For example, in a SE session, if several users ask for the token to edit a video, the moderator can decide to give the token to the most expert.

3.3 Some critical aspects in the PV and SE services

This section describes those aspects that are critical for the design of the PV and SE services. Each critical aspect is assigned a number CAx, where x is the number of the critical aspect, that is referenced when solutions are presented.

3.3.1 Description of the critical aspects

The critical aspects studied are described in Table 1. From this table, it can be seen that PV has less critical aspects than SE. Figure 3 illustrates the critical aspects in SE.

3.3.1.1 Initial document synchronization (CA1)

When a user joins (by invitation) the service, he/she may have the corresponding multimedia document or documents. If not, the user must retrieve the documents before the conference starts.

The initial document synchronization can be done using any document transfer mechanism (such as DFR, DTAM-DM or FTP), initiated by the user (the simplest case) or by the initiator.

3.3.1.2 Joining and leaving the session (CA2)

Management of the users that are joining and leaving the presentation or editing session implies a complex management of the service.

In the case of PV, if the users join the service later, the problem of the initial document synchronization arises. The users can leave the PV session at will.

Table 1: Description of the critical aspects

CA	Critical Aspect	PV	SE	Description
CA1	Initial document synchronization	✓	✓	When the session starts, every user should have the same document copy where the editing takes place.
CA2	Joining and leaving the session	✓	✓	When can users join and leave the session.
CA3	User co-ordination	✓	✓	How are the users synchronized to perform the presentation or editing actions.
CA4	Document updates	-	✓	How are the several copies of the document updated when a user makes editing actions.
CA5	Finishing the session	-	✓	Under what conditions can the session be finished.
CA6	Final valid copy	-	✓	After finishing the session, which copy of all is to be considered the final document.

✓: indicates if this critical aspect is relevant for that service.

In the case of SE, the following aspects are to be taken into account:

- It makes not much sense to join a document service once it has started. For other types of services, later joining may be meaningful, but not in our case.

- A user should be able to leave the session, e.g. a user has finished his/her modifications in the document and is no more interested in the evolution of the document. But that should not happen under certain circumstances:
  - the user has any exclusive resources (such as tokens);
  - the user is the initiator (or the moderator) of the session.

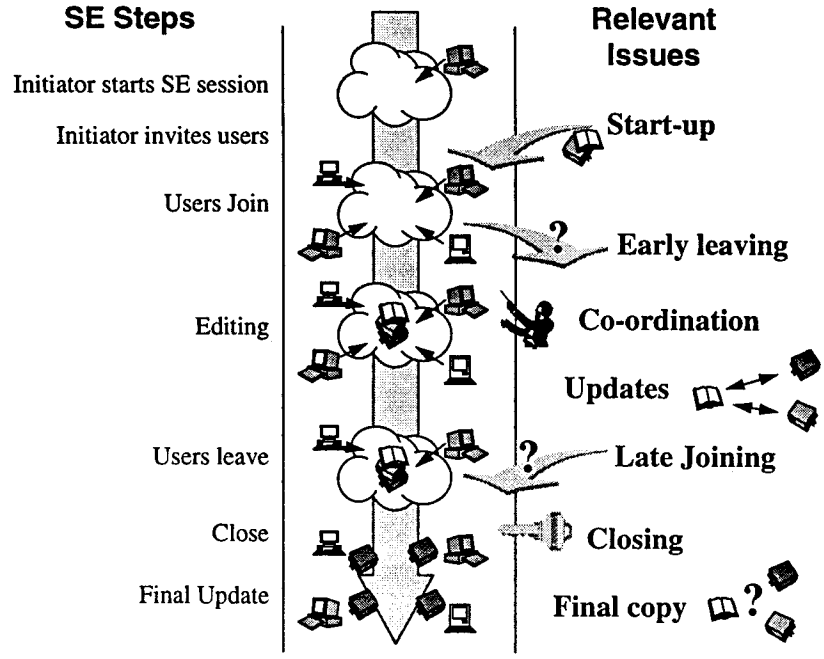


Figure 3: Critical aspects in a SE session

3.3.1.3 User coordination (CA3)

In PVn and SE , there must be some coordination mechanism to control the evolution of the session. In PVn, only one user may be presenting at a given time. In the joint synchronous editing session, only one user can be editing the document. If another user wants to present or edit, he/she must ask for permission.

An efficient method to coordinate the users is to have a token. Who has the token, can present (in PVn) or edit (in SE) the document. If a user wants to present/edit, he/she must ask the token owner (editor or moderator) for the token. This mechanism works, but some rules should be specified in order to guarantee that the token is always accessible or controlled. The following rules are to be followed:

1. the initiator of the editing session owns the token when the session starts;
2. the token owner presents/edits the document(s);
3. other users may request the token to the current owner of the token (editor or moderator);
4. the owner gives the token to a requesting user;
5. 2 to 4 are repeated for the users owning the token and presenting/editing;
6. the presentation/editing session is terminated regardless of who owns the token.

The problem of using a token is that it cannot get "out of control". The following situations are to be avoided:

- the user owning the token leaves the session without giving it;
- the token gets somehow duplicated by a missworking application;
- the system of the user owning the token crashes.

3.3.1.4 Document updates (CA4)

This critical aspect does not affect the PV service, since there are no document updates in this service.

In SE, when a user makes any editing action on the multimedia document, information should be sent to all the users in order to update the document copies they have. The modified document fragment can be sent. This can be done in a similar way as done in DTAM when a document fragment is written, but in a multipoint way.

A multipoint extension to DTAM could be considered as a good solution for that problem.

3.3.1.5 Finishing the session (CA5)

Finishing the session does not imply any problem in PV.

In the case of SE, the session should finish in a stable state. Some conditions are to be fulfilled in order to finish the session.

The first condition is that all the users that are still joining the session have received the same update information. If not, there may be inconsistencies between the different resulting multimedia document copies.

The second condition is that the token should be owned by the one responsible for finishing the session. In this way, it is guaranteed that no one is still editing the document.

The session initiator (or moderator, if exists) should be responsible for finishing the Joint Synchronous editing session.

3.3.1.6 Final valid copy (CA6)

The problem of final valid copy does not affect the PV service.

When the SE session finishes, there are many multimedia document copies, each for every user that participated in the session. There are two kinds of copies:

- final versions: at those users that were in the conference until it finished;
- intermediate versions: at those users that left the session before finishing.

All the final versions are valid, but sometimes, after finishing the session, it is difficult to control if a user had left the session or was there until finished. The copy of the session initiator (or moderator, if exists) contains all the updates. This is the most reliable copy.

A final document synchronization mechanism could also be used, similar to the initial synchronization, in order to store a final copy in a known document server.

3.3.2 Design decisions to solve the critical aspects

This section provides solutions based on the features provided by GCC to the critical aspects described in 3.3.1.

3.3.2.1 Conference Management

For the conference management, it should be considered that SE is a controlled service, that means:

- When the SE session is started, it is known who will participate. It is not required that everybody knew about the conference.
- The participants can leave at will, but cannot join once the service has started. The conference is locked after all the users are invited, it does not make sense to join the service after it has started (*solution to CA2*). No password is required if it is locked.
- Users behave as specified, e.g. nobody tries to edit without the token. It makes no sense to eject users.
- It is not useful to split or join SE conferences;
- The conference can be finished at any time by the initiator (or moderator, if exists) (*solution to CA5*)
- The initiator (or moderator) owns a final copy of the multimedia document (*solution to CA6*).

3.3.2.2 Co-ordination

The editing control (token) can be done in three different ways:

a) the conference is conducted: only the conductor can edit the document;

b) an MCS token is allocated (the GCC primitive `GCC-Registry-Assign-Token` is used for this purpose);

c) the application implements its own token mechanism.

Option b is selected (*solution for CA3*). It is better that changing the conductor of the conference, since it is not necessarily the same to be in turn in the edition, as being the conductor of the conference. Furthermore, the conductor mechanism would be useful if a moderator is to be supported in the conference. Option c is not chosen, because T.120 provides token mechanisms that do not need to be implemented by the applications itself.

3.3.2.3 Initial document synchronization

For this problem, there are two solutions, one based on a transmission over an MCS channel, and the other based directly on the GCC capabilities:

a) Transmission over an MCS channel. The document is broadcast over a channel and the interested users can retrieve it. This would be inefficient, because there may be few users interested in getting a copy.

b) Initial information for retrieving the documents (not the document itself) is passed as user data in the invitation primitive. The User Data parameter of the `GCC-Conference-Invite` indication that receives the user can specify the information.

Option b is selected (*solution for CA1*). There is still a problem to be solved: when do really the editing starts?

Before the editing starts, the initiator (or moderator, if exists) should be sure that all the users have already the documents. This can be solved using the `GCC-Conference-Invite.response` primitive for that synchronization. The users receive the `GCC-Conference-Invite.indication` primitive with the information about the documents to be retrieved. The user issues the `GCC-Conference-Invite.response` when he/she is ready for starting the session.

In the case of PV, the solution for CA1 could be applied for CA2.

3.3.2.4 Document updates

All the users in the SE session are interested in receiving the document updates being made by the editing user. The best solution is to broadcast it over a channel (*solution to CA4*). This functionality is done by an Application Service Element (ASE) inside what is called the Application Protocol Element (APE), therefore it is specific to the application how it is implemented. But resources are required, which will be reserved through GCC, therefore it is relevant to specify it at the GCC level.

The information transmitted may be DTAM-DM operations that specify the modifications on the multimedia document.

3.3.2.5 User access

GCC provides for the management of participants in a conference. Therefore the management of the users is easy. The best way to control the joining of the conference is to create a locked conference. The users are invited by the conference initiator. Once all the users are added, no one more can join the conference. Users can leave the conference at will, since it is not possible to control the conference leaving, and in some cases it is useful (*solution to CA2*).

4. Example: SE working on GCC

This section describes the steps followed during a whole SE session, taking into account the specifications given in 3.3. For the establishment of the session, the following actions are done (see Figure 4):

- The conference initiator is the top GCC provider.
- The initiator creates the conference (`GCC-Conference-Create`).
- The initiator invites the users to the session (several `GCC-Conference-Invite.req`).
- Each user receives an invitation indication (`GCC-Conference-Invite.ind`).
- If the user does not have the documents, he/she retrieves them.
- Each user accepts the invitation (`GCC-Conference-Invite.res`).

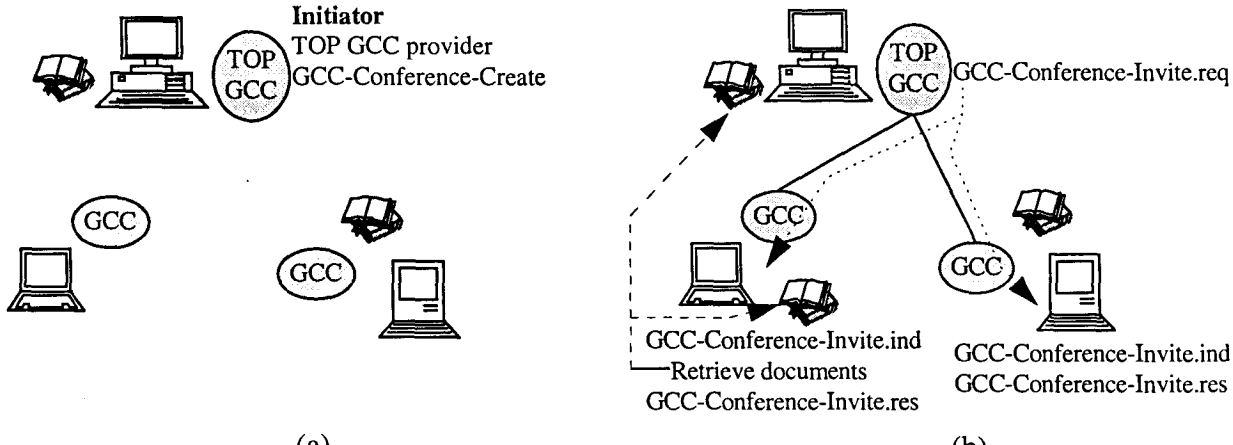


Figure 4: (a) Starting the SE session, (b) the initiator starts editing

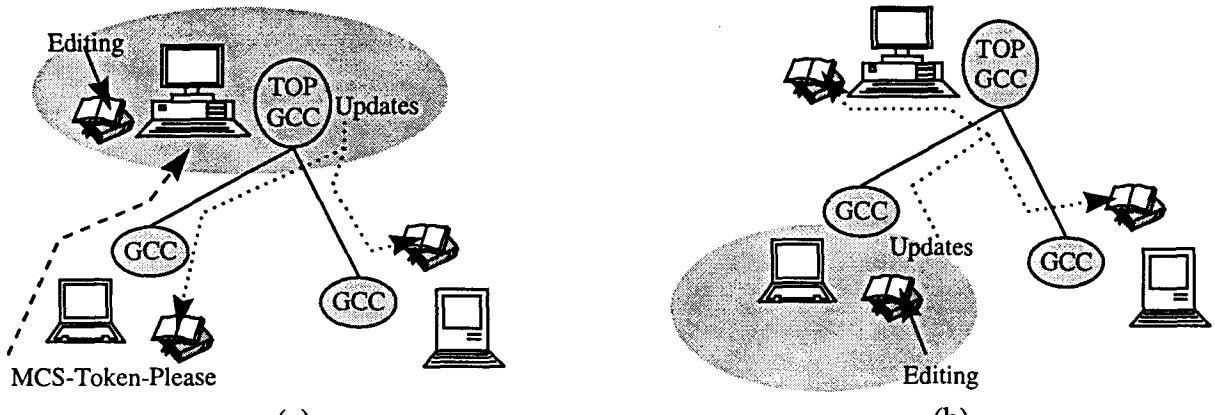
The session is ready for beginning. Now the next steps follow (see Figure 5):

- The initiator edits the document, the updates are sent to the other participants.

- Another user can request the token (`MCS-Token-Please`).
- The editor gives the token to the user (`MCS-Token-Give`).
- Now the user can edit the document, the updates are also distributed.

These steps are repeated during the session. In order to finish the session the following is done:

- The initiator (or moderator, if exists) requests the token.
- When he/she gets the token, he/she finishes the session (`GCC-Conference-Terminate`).



(a) (b)

Figure 5: (a) Initiator edits the multimedia document, a user requests the editing token, (b) another user gets the editing token

5. Conclusions

The paper has presented a solution to a specific problem in Electronic Publishing: How to allow several users to view/edit together a multimedia document. For this purpose, two complex services, Joint multimedia document Presentation/Viewing (PV) and Joint Synchronous Editing (SE), that are being standardized, are described, and a solution to provide multi-user facilities is proposed. PV and SE complex services require complex functionality from both the management and communication aspects. The communications problems can be solved using the T.120 multipoint communication provided by MCS. The session management problems, as described in the paper, can be solved using the Generic Conference Control (GCC).

The use of GCC provides solutions for the most critical aspects of the services, which are: initial document synchronization, joining and leaving of the session, user co-ordination, document updates, finishing the session and the final valid copy.

GCC is intended to provide functionality to a wide range of conferencing applications. PV and SE are conferencing-like services, but they are not the typical ones. This fact puts some restrictions in the use of GCC, and then not all the functionality of GCC is used.

The most important advantages of basing complex services on the T.120 series of Recommendations are:

- Standardized protocols: the communication is based on international Recommendations interworking between different systems and platforms. Most vendors are announcing the future development of T.120 compatible applications.
- Network independence: the use of T.120 Recommendations hides the network specific issues.
- Re-use of software: software tools and applications are emerging and built to support T.120 communications, that can be re-used for the multimedia document services.

A drawback of using the T.120 series of Recommendations could be some overhead on the system implementation, since they are not specific for multimedia document services. Nevertheless, the use of a "restricted" version of the T.120 may avoid that overhead. Work on specifying profiles for these Recommendations is starting in EWOS [EWOS].

Finally, and as a summary, the Joint multimedia document Presentation/Viewing and the Joint Synchronous Editing services have been specified based on existing communication mechanisms that provide a good platform for multipoint communication services.

These topics have been also discussed in ETSI [ETS1] [ETS2] and EWOS [EWOS].

The implementation of a T.120 testing platform being developed in the UPC will allow the testing of these services.

References

[DFR] ISO/IEC 10166: 1991, *Information technology - Text and office systems - Document Filing and Retrieval (DFR)*. (Amendments and Technical corrigenda also exist).

ISO/IEC ISP 12069, Parts 1 to 6: 1996, *Information technology - International standardized profiles ADFnn - Document Filing and Retrieval (DFR).*

[DTAM] ITU-T Recommendation T.435 (1995): *Document Transfer And Manipulation (DTAM) - Services and protocols - Abstract service definition and procedures for confirmed document manipulation.*

ITU-T Recommendation T.436 (1995): *Document Transfer And Manipulation (DTAM) - Services and protocols - Protocol specifications for confirmed document manipulation.*

ISO/IEC ISP 15121, Parts 1 to 3: 1996, *Information technology - International standardized profiles AODnn - Application profiles for Open Document Architecture (ODA): Profiles for interactive manipulation of ODA documents.*

[ETS1] ETS 300498-1:1996, *Open Document Architecture (ODA); ODA communication services; Part 1: Basic Services.*

[ETS2] ETS 300498-2:1997, *Open Document Architecture (ODA); ODA communication services; Part 2: Complex Services.*

[EWOS] EWOS EG SMMI 96/070, "Profiling ITU-T T.120 Recommendations", Jaime Delgado, José J. Acebrón, 22 March 1996.

[T.120] The ITU-T T.120 Series of Recommendations currently includes:

ITU-T Recommendation T.120 (1996), *Data Protocols for Multimedia Conferencing.*

[T.121] ITU-T Recommendation T.121 (1996), *Generic Application Template.*

[T.122] ITU-T Recommendation T.122 (1993), *Multipoint Communication Service for Audiographic and Audiovisual Conferencing Service Definition.*

[T.123] ITU-T Recommendation T.123 (1993), *Network Specific Data Protocol Stacks for Multimedia Conferencing.*

[T.124] ITU-T Recommendation T.124 (1995), *Generic Conference Control.*

[T.125] ITU-T Recommendation T.125 (1994), *Multipoint Communication Service Protocol Specification.*

[T.126] ITU-T Recommendation T.126 (1995), *Multipoint Still Image and Annotation Protocol.*

[T.127] ITU-T Recommendation T.127 (1995), *Multipoint Binary File Transfer Protocol.*

[T.190] ITU-T Recommendation T.190 (1995): *Telematic Services - Terminal Equipment and Protocols for Telematic Services - Co-operative Document Handling: Framework and Basic Services.*