Magic Lounge Architecture

A First Outline

Laila Dybkjær and Niels Ole Bernsen
The Maersk Mc-Kinney Moller Institute for Production Technology
Odense University, Denmark
laila@mip.ou.dk, nob@mip.ou.dk

Magic Lounge is a virtual meeting application which will integrate a number of technologies most of which are already known, such as www and video conferencing. In addition, Magic Lounge will integrate the innovations necessary to achieve the envisioned Magic Lounge functionality as described in [Dybkjær and Bernsen 1998]. Figure 1 outlines the physical Magic Lounge environment, i.e. how users may be connected to, and present in, Magic Lounge.

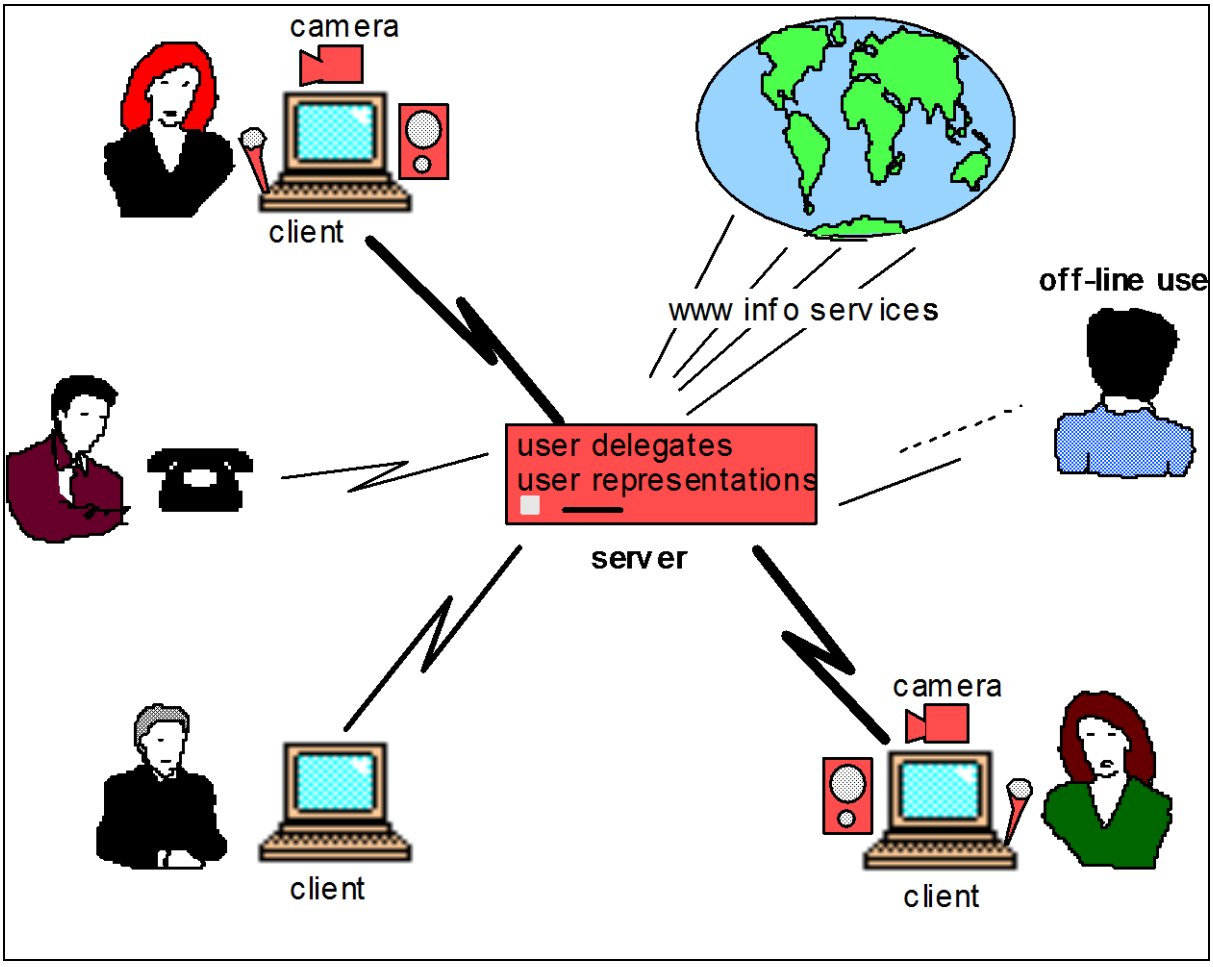


Figure 1. Magic Lounge allows multiparty human-human, human-computer, and human-human-computer communication. Users may be connected to the ML server via computer,

telephone and PDA (not shown) or they may have sent a delegate to the server. Bandwidth may vary and a computer may have more or less equipment (camera, loudspeaker, microphone, etc.). The figure does not show all the functionalities available in ML.

This paper provides a first sketch of the Magic Lounge architecture (Section 1). Section 2 concludes the paper.

Throughout this paper, ML is being used as an abbreviation for Magic Lounge.

1. ML architecture description

In what follows, we first present the overall idea of the ML architecture. Then each part of the architecture (Figure 2) is discussed to clarify their functions. Finally, deployment is discussed to get an overview of which existing software could be included and which parts have to be implemented from scratch.

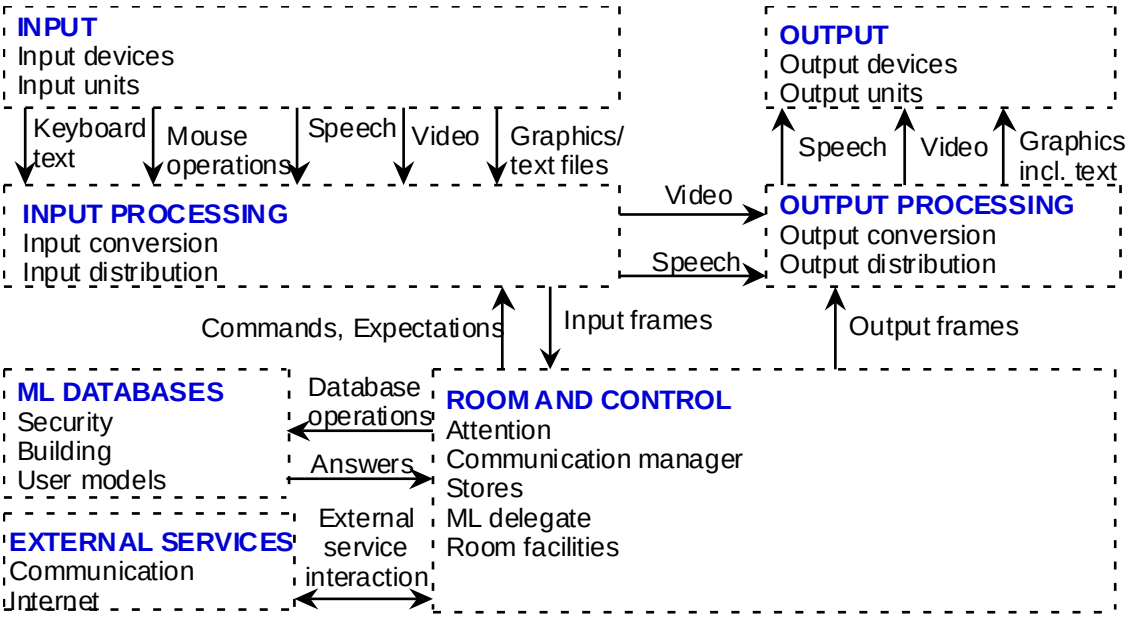


Figure 2. Overview of the ML architecture. A more detailed view is presented in Figure 3.

Overall idea

The ML architecture roughly consists of:

- Input: units, devices, decoding/processing, distribution.
- Output: distribution, encoding/processing, devices, units.
- Rooms: interaction management, facilities, local context and stores.
- ML structure (databases): user models, a building with rooms, security.
- External services: web pages, databases, etc.

Figure 2 only shows a single room (‘room and control’). In reality there will be a number of ‘room and control’ boxes corresponding to the number of active rooms. Each room has a set of facilities which may be rather comprehensive (cf. Figure 3). However, which facilities each

individual user actually has access to depends on how s/he is connected to ML. The central control part is the communication manager which distributes input to the relevant recipients and sends output to the output processing part. In addition, the 'room and control' part contains an attention part which keeps track of the focus of the ongoing interaction, a local room-dependent context in terms of a history ('stores'), explicitly user-stored information, and an ML delegate which represents ML and may actively participate in the communication.

The 'room and control' part may draw on several ML internal databases: security, building, user models; and on external services: Internet and communication services.

To begin with, there is a hall plus the possibility to create new rooms based on a generic model. We may later include several different templates.

The overall idea is that ML provides a framework: input, output, interaction management, interaction history, interaction focus and access to context. New facilities can be added when they are ready.

The main metaphor for the user is the room. All interaction management and all facilities are coupled to the individual room. This requires that the user is always connected to a room and therefore that an ML server always has at least one public room (the hall).

We have to consider what the individual user should be able to do on his own and what can only be done in a group. Individual activities should probably not be logged.

Input

Input units

The units from which ML can be accessed include telephone, mobile phone, PDA and computer. The computer is primarily assumed to be a PC but access from, e.g., a Sun workstation should also be allowed. In fact, ML only requires that the computer can run jdk1.1.

Input devices

The input devices or channels attached to the input units described above include microphone (in a telephone or coupled to a computer), mouse, keyboard and camera. For each input channel a driver is available which will handle the input (keyboard text, mouse operations, speech, video, graphics/text files) arriving from the relevant unit plug.

Output

Output units

The units to which ML can deliver output include telephone, mobile phone, PDA and computer. The computer is primarily assumed to be a PC but access from, e.g., a Sun workstation should also be allowed. In fact, ML only requires that the computer can run jdk1.1.

Output devices

The output devices or channels attached to the output units include loudspeaker (in a telephone or coupled to a computer) and display. For each output channel a driver is available

which will handle the arriving output (speech, video, graphics incl. text) and channel it to the relevant plug on the unit.

Input processing

Input conversion

The raw input may be converted to recognise more meaningful or abstract tokens. 'Expectations' will be available to support recognition. Coordination of input from more than one device is supported.

Specifically, speech input and speech combined with gesture will be converted before the contents are further distributed. In general, speech input will be converted to text for inclusion on the magic board. Speech commands will be recognised and parsed. Speech along with gesture is recognised and interpreted in combination with the gesture. In all three cases, results are returned to the input distribution.

Input distribution

The task of input distribution is to send the received raw input to the relevant handlers for further processing. All speech and speech combined with mouse clicking will be sent to input conversion (see above). Speech including speech in video conferencing will, moreover, be sent directly to 'output processing' and so will the video part. All other input, including the converted input, will be packaged into input frames and sent to 'room and control'. Input frames include information on the user who sent the information, on the device(s) through which it was sent, and the actual input message.

Input distribution may receive commands from 'room and control'. The contents of the input frames will only be decoded in 'room and control'. If, for instance, the input frame sent to 'room and control' turns out to be a request to open a video conference link to somebody, then 'room and control' has to ask the input distribution to open a direct connection to output processing.

Output processing

Output conversion

People who only have a sound connection to ML, e.g. via an ordinary telephone, can only receive acoustic output. Text and graphics output is therefore being converted. Text is converted to speech, and graphics is converted to text which may then be converted to speech. Graphics converted to text may also be needed for people connected via a PDA and who therefore only have a low resolution display. The converted results are returned to 'output distribution'.

Output distribution

The task of output distribution is to decode the received output frames and send the contents to the relevant handlers either for further processing (conversion) or for channelling it to the users. Output frames include information on the users who are to receive the information, the device(s) through which it should be received, and the actual output message.

Room and control

Attention

Attention includes focus tracking and expectations based on focus. From the communication manager attention will know about input/output focus, and from the room facilities it will know which facility is currently being used and the topic in focus. Attention delivers expectations for use in the input conversion (at least speech recognition) and by the ML delegate.

Communication manager

The communication manager is the central 'event' handler. It encompasses input frame handling, facility management and output frame handling. Input frame handling is the delivery of frames to the relevant facility or to facility management. Facility management will start and stop a facility if requested by the input frame. Output frame marshalling wraps output into a frame with the relevant information (see 'output distribution').

Stores

The stores include both short term logging of, e.g., the latest 250 input/output events, and long term stores, e.g., user defined notes and lists.

A frame log will be maintained based on input from the communication manager. We may consider to reduce the number of frames stored to, e.g., the last 250 ones.

Moreover, the 'stores' will contain snapshots of the magic board**.**

We should consider to allow users to store information explicitly in some way, e.g. it should be possible to store a particularly interesting part of the board in a separate file and not just along with all the other information on the board. This is here called 'user-defined archives'.

ML delegate

The ML delegate is a personification of the ML (in the PP called communication manager). It will allow the ML system to participate actively in discussions and, e.g., answer questions asked by a user to all participants in a room. To support the ML delegate in following a conversation it can draw on 'expectations', it may consult the 'stores' and it may interact with the facilities to which the other users also have access.

Room facilities

All facilities are presented together in this section. They may draw upon all other parts of the ML framework such as 'attention' and 'stores'. Each facility is started and stopped by the communication manager. Communication in general goes via i/o frames from the communication manager. Facilities receive input frames by having set up a number of event handlers. The communication manager sends each input frame to every facility that has a matching input frame handler. A facility may bypass the input frames (e.g. video output from the web to ensure sufficient speed).

Room facilities include web browsing, email, fax, video conferencing, search, filtering, general application sharing, (the possibility to open a telephone line from within ML), magic board summary, magic board (editable), non-editable boards, representations (upload, view, who are

present, etc.), room handling (save, delete, etc.), change room and link to other information spaces. Which facilities are available to a particular user depends on the unit via which s/he is connected to ML and the available channels. It must be possible to add or remove facilities independently.

The important task is to define APIs for the rest of the ML system that the facilities may use, in particular a protocol for how the facilities present themselves on the Magic Board (if at all). We should consider if CORBA has something similar to Ole/COM Compound Document structure.

ML databases

Security

The security database contains all password files and authorisations such as who has access rights to what. This database is used by 'room and control' whenever a user wants to change room or perform any other operation which requires certain rights.

We may consider if we need an 'administration room'. Only system administrators would have access to this room. From here they may clean up, give new passwords to users who forgot their own, etc. The most important facility in this room would be a control board which has all the crucial information on authorisations and passwords.

Building

The building contains all the rooms and may also be called a room database. This is the place in which any room is saved and from which it may be retrieved. It is used by 'room and control' whenever an operation is performed which involves saving a room, deleting a room or retrieving a room. Moreover, it keeps track of which rooms are in use and by which users.

User models

The user models include a representation of each ML user at least in terms of a text file with his/her name. However, the user is allowed to upload much more fancy representations: video clips, pictures, sounds, avatars, ...

The user may also define a delegate which represents the user and which can be sent to a certain room at a certain time to participate actively in a meeting while the user is off-line.

Any user who wants to register to ML will be asked to give his/her full coordinates. This will enable other ML users to easily contact a person, e.g., via email or telephone.

Finally, the user models include a list of users currently logged on and how. Where they are for the moment is known by the list of rooms (see 'building').

The user models are drawn upon by 'room and control' and can be updated from there.

External services

Communication

If we want to include email and fax we need an external service which can handle this kind of communication with the world outside ML.

Internet

It has already been decided to include web browsing. This is an external service outside ML and accessed via the Internet. We may consider to include other external services accessed via the Internet, such as databases and ftp.

Deployment

Platforms

As already mentioned, ML must be accessible via (stationary or mobile) telephone, PDA and computer (PC (including Mac) or workstation). A multi-platform architecture is therefore needed. JDK1.1 (or later) in principle satisfies this requirement. There are still problems with Macs but they will probably be solved before we have finished ML. To glue components together we could use CORBA or COM/DCOM. Both have advantages and drawbacks. COM/DCOM is more well-established and used in practice. For the moment however, it only supports Windows, including Windows95 and WindowsNT (which will probably be by far the most common platforms used by our users, but not by all of them). There is a chance that in 1-2 years there will also be a COM/DCOM version which supports unix. CORBA is intended to be platform-independent and is cleaner but even though there is a number of running applications, the development tools are more sparse, and CORBA has not yet proven its adaptabililty to floods of applications and to dynamically and chaotically changing environments. It is difficult to tell which, if any, of these we should choose.

Three-layered architecture

Basically, we have a three-layered architecture in which we may distinguish among user interface, business services and data services. These three layers are related to Figure 2 as follows:

The user interface includes what the user perceives. This is not described in any detail in Figure 2. We may consider the input/output units and devices part of the interface, but otherwise nothing has been indicated.

Business services are services which primarily exist at ML run-time. Thus the business services in Figure 2 include all input and output processing, i.e. input conversion, input distribution, output conversion and output distribution. Moreover, it includes all room facilities, the communication manager, attention, the ML delegate and the stores. The stores contain data and one might argue that they belong to the data services. However, in contrast to the other data services (see below), the stores are room specific and only live while the room lives to which they belong.

The data services are services which not only exist at run-time and which have a longer life than a room. The ML databases (security, building, user models) are of this kind. We shall also consider the external services (communication, Internet) data services. They are different from the ML databases since they have a life outside ML. However, seen from ML's point of view they may be seen as acting at the same level as the databases.

Distribution on client/server

Probably most business services as well as most data services will be placed on the server. We suppose that we cannot place anything on a telephone client and hardly much, if anything at all,

on a PDA. This speaks in favour of putting as much as possible on the server. The following distribution seems to be a reasonable one: all the ML databases must be on the server. The external services are as such not part of ML but via the ML server they can be used from ML. The conversion tools should also be on the server. Most of them concern speech and are important for communication via the telephone on which we cannot place any services. Moreover, it would probably be very expensive to equip all clients with, e.g., speech understanding, speech-to-text and text-to-speech. On the other hand, if we want to use speaker-adaptive recognisers these will have to be on the clients. Some of the room facilities may have to be on both client and server. Local representations of users will be on the client. It seems reasonable that a computer (PC) user should be able to save a copy of a room on his own client.

Use of existing software

We have to consider which existing software we will have to incorporate in ML. It is important for our work to know which modules we "only" need to create interfaces to and which modules we will develop from scratch. In the following we discuss some of the modules to include and for which we "only" need to develop interfaces.

Input conversion

Speech-to-text: Dragon systems offer a large-vocabulary, continuous speech speech-to-text system (at least for English; no Danish). The question is if they will come up with tools/environments which make it possible to incorporate it as a module in ML. IBM also offers a large vocabulary, continuous speech speech-to-text system, ViaVoice (for several languages but no Danish). The price is about 100$.

Speech recogniser (language adaptable): For speech recognition (to be used in, e.g., speech-driven search) we may consider the recogniser from Entropics, Cambridge. Entropics is a company which primarily develops speech tools. For instance, HTK (Hidden Markov Model ToolKit) from Entropics is a well-know tool. The speech recogniser comes with a vocabulary of 3-4000 words (English). However, through use of the HTK tool new words may be added or a brand new vocabulary in another language (e.g. Danish or German) may be inserted instead. The embedding of the recogniser in a larger system through APIs is also supported by Entropics. Right now we are investigating GrapHvite and HTK at MIP.

Speech and gesture: LIMSI has developed a map system which understands combined speech and gesture (pointing) input. This system will be included.

Output conversion

Text-to-speech: Several speech synthesisers are available in several languages. We could, e.g., check what Infovox offers. Maybe it would be an advantage to allow the use of a Danish synthesiser for Danish users, a French one for French users, etc., without any reprogramming other than selecting one module instead of another to be the one to be used at run-time.

Graphics-to-text: We have to find out what exists in this field.

Room facilities

Web browsing: A web browser from Hotjava has already been included in the November'97 demonstrator.

Email: If we decide to include email (it is not in the PP but may be a very convenient thing to have), we should consider which existing email system it would be most appropriate to build into ML. We should not develop one ourselves.

Fax: The situation and the argumentation for inclusion of fax is the same as for email.

Video conferencing: Also as regards video conferencing we should investigate what is on the market and what could possibly be incorporated in ML.

Search: Speech-driven search is a research area. We have to watch closely what happens in the field during the next year or two. The consortium does not seem to have any existing software to throw in. Is there something which we could buy?

Filtering: This facility enables the user to retrieve a well-defined subset from the stores. A user may, e.g., ask to see all contributions made by, e.g., Peter during a particular meeting and saved on the magic board.

Application sharing: Video conferencing systems often come with some kind of application sharing which in some cases is pretty advanced. When we investigate video conferencing systems (see above) we should at the same time keep an eye on the extent of application sharing offered by the individual systems. If we can get what we need in this way it would be fine but we may have to add something ourselves.

Open a telephone line: If a user, being in ML, has to call another user who has not arrived and who is not logged in, it would be very practical if the user can call the second user over the telephone without having to log out from ML.

Magic board summary**:** We have to find out what to do.

Magic board: A preliminary text-only version exists which will be improved and enhanced. It must be extended to allow graphics and "magic beans", i.e. applications like crossword puzzles.

Non-editable boards: These have to be developed from scratch.

Representations: For the moment you can see who is present in a room (textual representation only). Upload facilities, facilities which allow the user to choose how to see other users, delegate implementation, etc., have to be developed.

Room handling: A minimum of room handling has been included already. It is, e.g., possible to save a room. This facility will be further extended and improved.

Change room: This is possible in the current version (November'97 demonstrator) but has to be improved.

Link to other information spaces: We have to find out what to do. We probably cannot build on existing software.

Stores: This could be as simple as a file browser with the ability to store/retrieve objects from the Magic Board. It could also support simple databases (off-the-shelf software).

Other existing software

There may be other existing software which we could benefit from using and which was not mentioned above. This will have to be considered in the next iteration of the architecture specification.

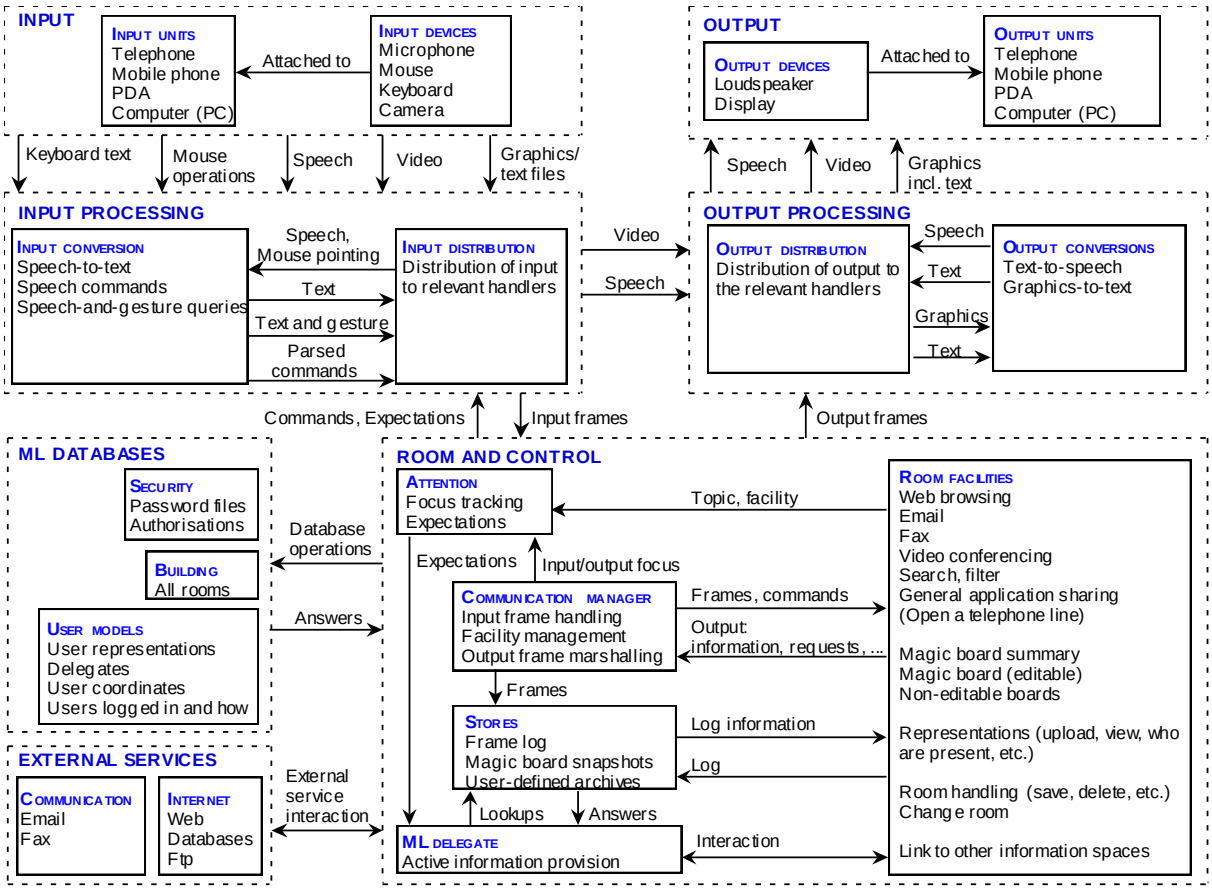


Figure 3. The ML architecture. Detailed logical view.

2. Concluding remarks

The sketch presented in Section 1 is a first draft of the ML architecture. It still needs considerable further specification. In order to produce a more detailed version closer to implementation it is necessary to investigate available software which may be used (as a basis) for the platform as well as the individual units. This is done in [Masoodian et al. 1998].

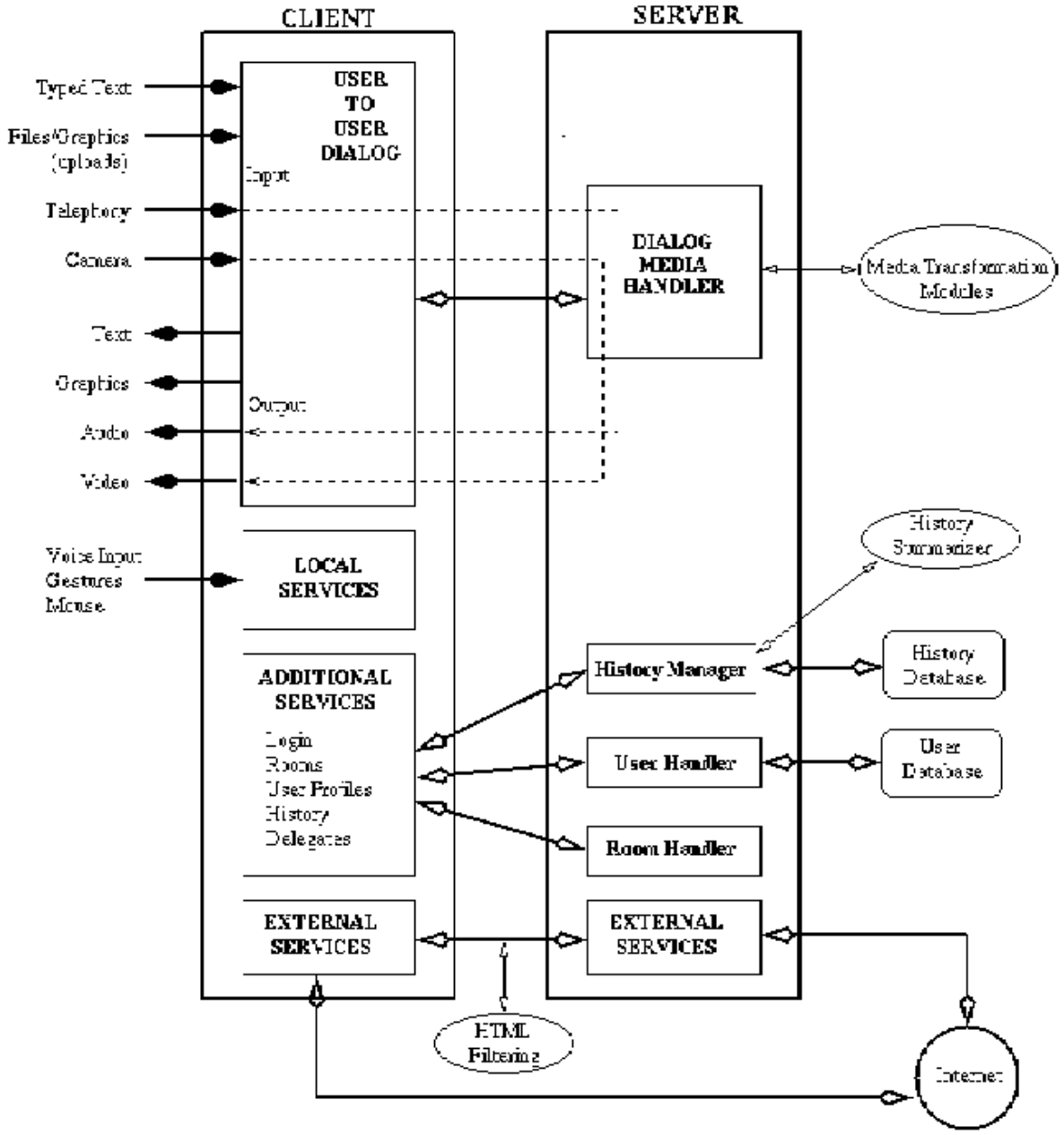
Acknowledgement: Hans Dybkjær actively participated in the discussion of the ML architecture and provided many good ideas and useful criticism. His support is gratefully acknowledged. Masood Masoodian provided many useful comments to the present draft report. We are also grateful to LIMSI and DFKI whose architecture diagrams are presented in the Appendices to this report.

References

Dybkjær, L. and Bernsen, N.O.: Magic Lounge Architecture. Scenarios and Use Cases. Deliverable 1.1, 1998.

Masoodian, M., Martin, J.C., Hauck, C. and Rist, T.: Architecture of the Magic Lounge Demonstrator for Year 1. Deliverable 1.1, 1998.

Appendix A: Input from DFKI



Appendix B: Input from LIMSI

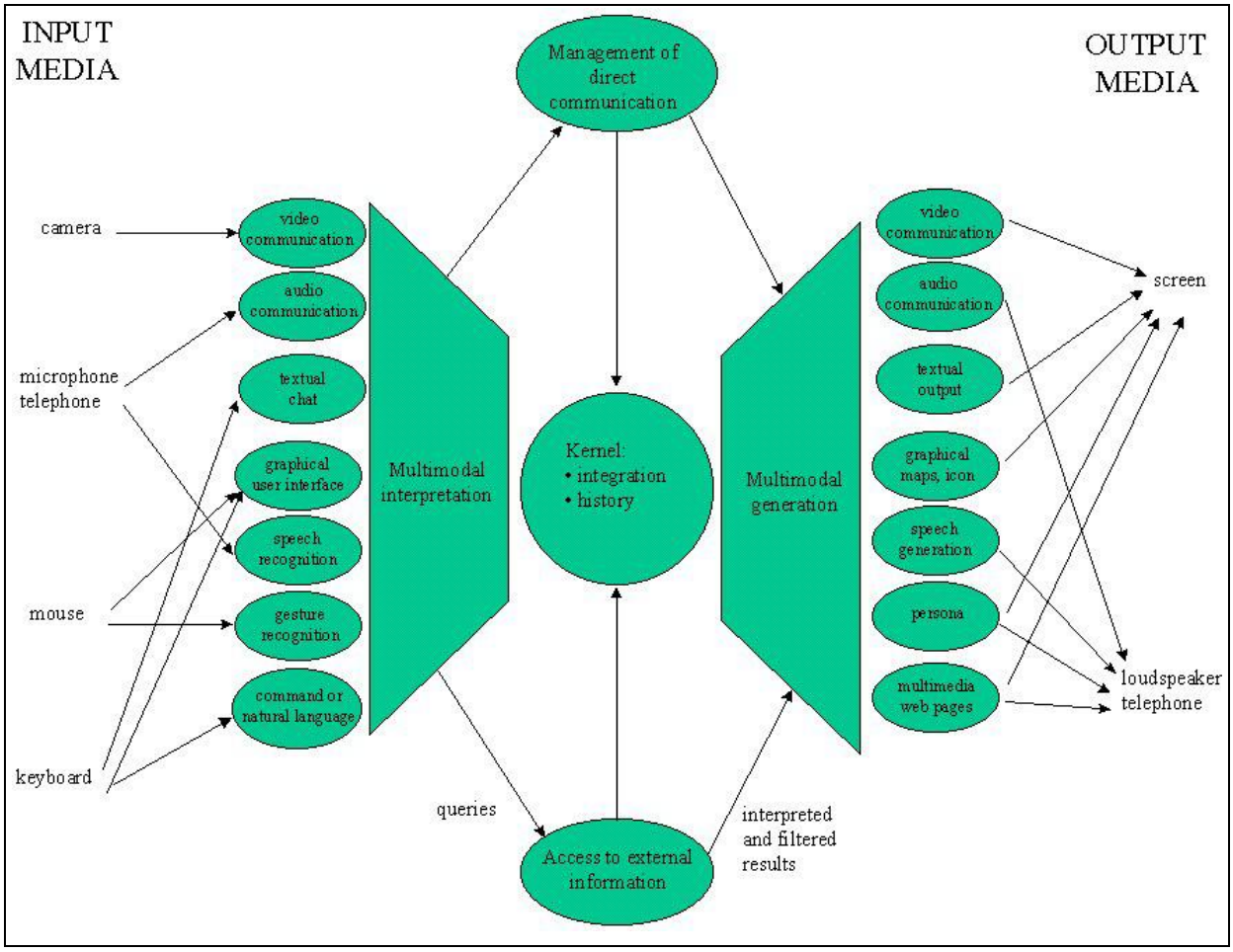


Figure 1. Suggestions for Magic Lounge functional architecture

The Magic Lounge has to integrate several media:

- several media are used as input (left-hand side);

- several media are used as output (right-hand side).

Specialised modules process each modality:

- each input media may be used in several ways (I call them input modalities), each of them being processed by at least one module;

- each output media may be used in several ways (I call them output modalities), each of them being processed by at least one module.

The Magic Lounge features four main functions:

Multimodal interpretation: in charge of integrating the information received on all the input modalities. Interpretation will take a small part in the processing of some media and modalities

(i.e., direct video communication). Interpretation will require more processing in some media and modalities (speech recognition). One key point of Magic Lounge is the transparency and useful integration of user-user communication and user-system communication (i.e., one user may combine speech and mouse gesture to get some information about another user displayed in a video window.

Multimodal generation: in charge of coordinating the information produced on all the output modalities. Similarly to the multimodal input, persona may make gestures towards information displayed about user-user communication.

Access to external information: in charge of accessing mono- or multimedia external information (web, databases...).

Management of direct communication: in charge of specific user-to-user direct communication functions.

Kernel : in charge of global functions such as integration and coordination of user/user and user/computer communication, history...