University of Geneva
Center for
Computer Science

Date: March 14, 2003

Number: 03.01

Computer Vision and
Multimedia Laboratory

Technical Report
Vision

Meeting Record Modelling for Enhanced Browsing

Stéphane Marchand-Maillet*

marchand@cui.unige.ch

http://viper.unige.ch/

Latest version at: **http://vision.unige.ch/publications/report.html**

*This work was supported by the NCCR IM2 and EU-IST M4 project.

Meeting Record Modelling

for Enhanced Browsing

Author	Stéphane Marchand-Maillet marchand@cui.unige.ch	*Viper* group http://viper.unige.ch
Document Identification	Technical Report 03.01	Computer Vision and Multimedia Laboratory – University of Geneva
Version	Revised	March 14th, 2003

This document presents a model for meeting and meeting data. Typically, several perspectives are combined, in order to arrive to an exhaustive model. The expected outcome is a complete model for meeting data annotation and its use within a meeting browsing web tool.

We start by identifying meeting activities. That is all possible activities happening during a meeting. The figure below presents these activities as an activity diagram gathered around a central transition state since we assume that any combination of activities can be envisaged.

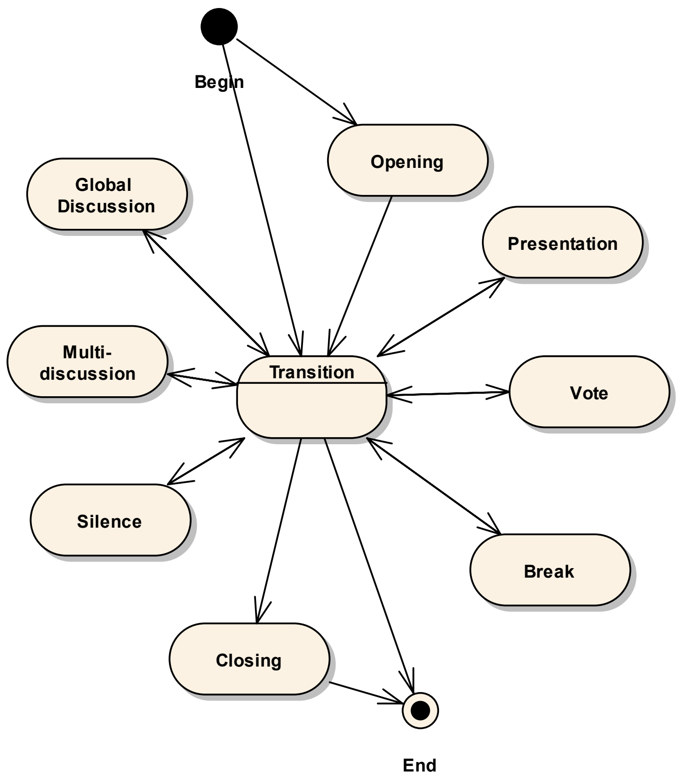


Fig. Meeting activities

Clearly, the above figure show a factored view of meeting activities. We think that any activity can fit within this model. Examples of instance of the activities listed above are:

- ✓ Lecturing = presentation
- ✓ White-board = presentation or global discussion

The next step we take is to identify all possible activities that a meeting participant can have during the meeting. The following diagram shows this again as a form of centralised state diagram where three clusters appear clearly.

The first group of activity identifies all modes of interaction that a participant can have with other members of the meeting. It is the view that other can have from this participant at every time. Ideally, each activity corresponds to a well-define timeframe.

The second group of activities shows the level of participation (involvement) a participant can have during the meeting. Here again, these are canonical activities that can correspond to different interpretations. Each of these should correspond to a unique timeframe, while parallel to the interaction described previously.

Finally, we list the possible physical states that a meeting participant can have. In the context of a "classical" meeting, this goes down to the list given below.

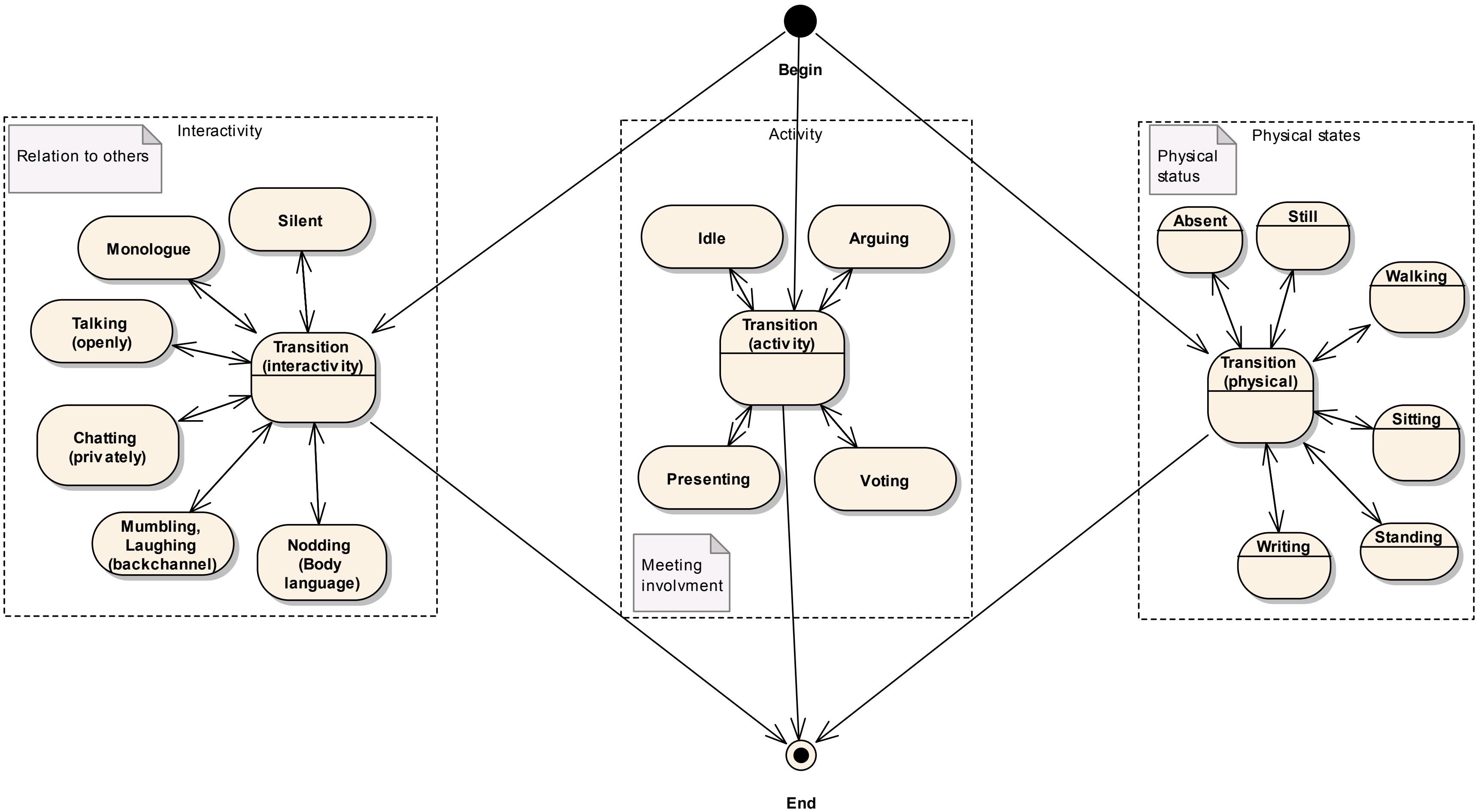


Fig. Participant activities

The combination of activities and states from the above three groups allows for characterising a number of activities. For example the activity “Note-taking” corresponds to a combination where the participant is silent (or nodding his/her head), idle, and writing.

The following complete data model is an attempt to create “low-cost” (non-redundant) relationships between the above-defined entities. It is both an outlook of the global information a meeting can contain and the base for the structure of a data repository storing this information.
Note that this data model proposes ways of extension via inheritance mechanisms. The idea is to specify whatever can be in a first approach and to extend where necessary.
As further information, the data structure below is a first step towards the definition of a meeting ontology using entities and relationships between them.

Meeting browsing use cases:
Browsing is related to querying. When a user browse data, it is often in the context of fulfilling an information need. It is therefore important to capture possible information needs into a set of browser use cases so as to be able to evaluate the validity and completeness of the proposed data model and to guide the browser design.
Derived actors for a browsing system:

- ✓ Participant: A person that is physically present in the meeting. May have various backgrounds (project member, visitor,…)
- ✓ Customer: may be a participant of a project (aware of the topic), absent at a meeting or a person unaware of the project
- ✓ Analyst: The entity in charge of the post processing

<u>Derived categories/questions/tasks</u>
Personal questions (based on memory):

- ✓ Where (next to whom) did I seat?
- ✓ Whom did I talk to (at that time/about what)?

Questions on participants:

- ✓ Who was there, what was his/her role?
- ✓ Who talked to whom (at a given time)?
- ✓ Who presented?
- ✓ Who entered/left (at a given time/at what time)?
- ✓ What was the participant layout (who seat next to whom)?
- ✓ Was there any external participant (NetMeeting)?
- ✓ What/when were the external communications of participants (phone,email) during the meeting?

Questions on topics:

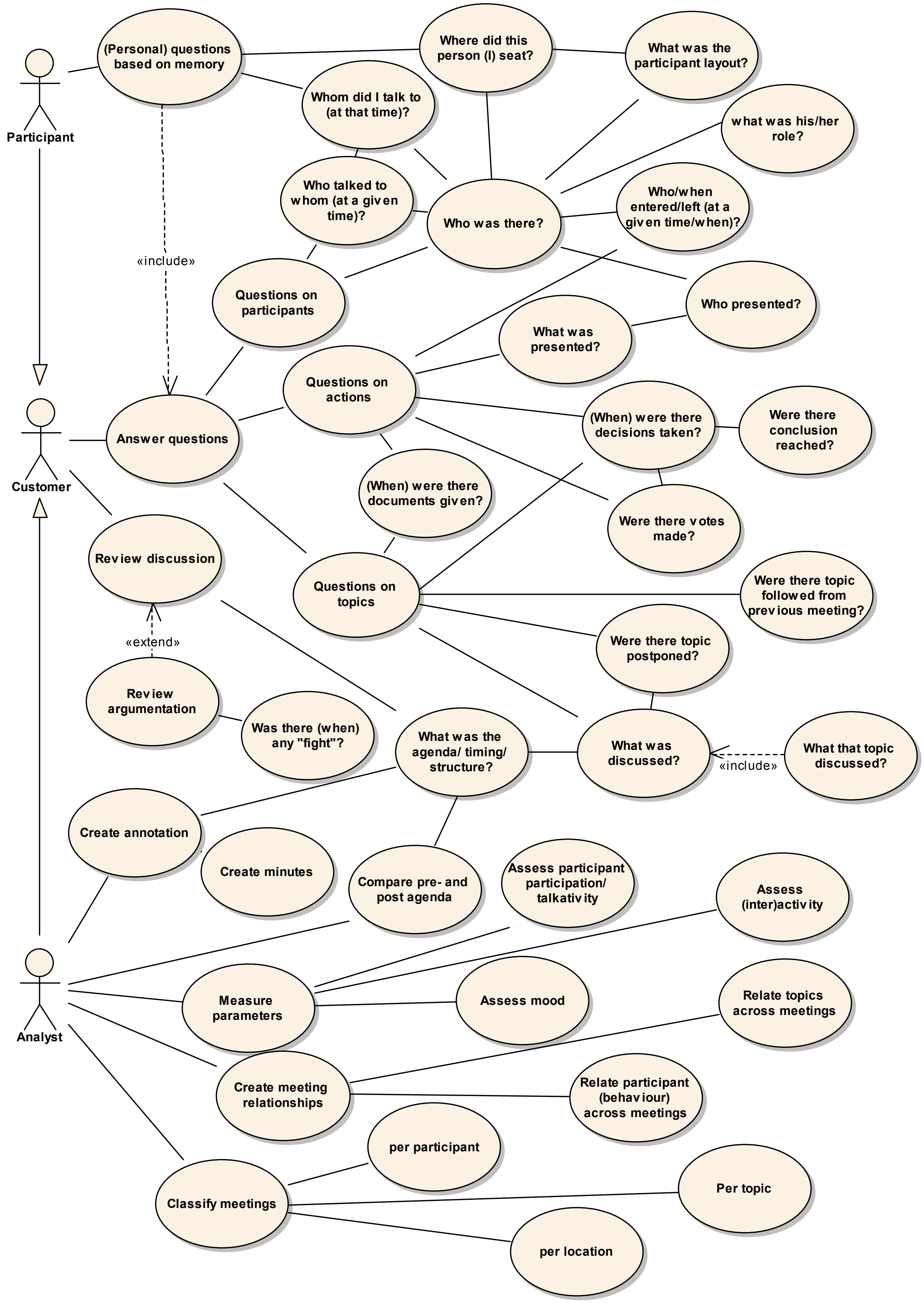
- ✓ What was discussed?
- ✓ Was this topic discussed?
- ✓ Was there any conclusion reached (which ones)?
- ✓ What was the timing, post-agenda, meeting-structure?
- ✓ Were there (and when) topics continued from previous meetings?
- ✓ Were there any topic postponed?
- ✓ Was there (and when) any “fight”?

Questions on actions:

- ✓ Were there (and when) documents given (before, during, after the meeting)?
- ✓ Were there (and when) votes made?
- ✓ Were there (and when) decisions taken?
- ✓ Were there (and when) any presentations?
- ✓ Was there any break?

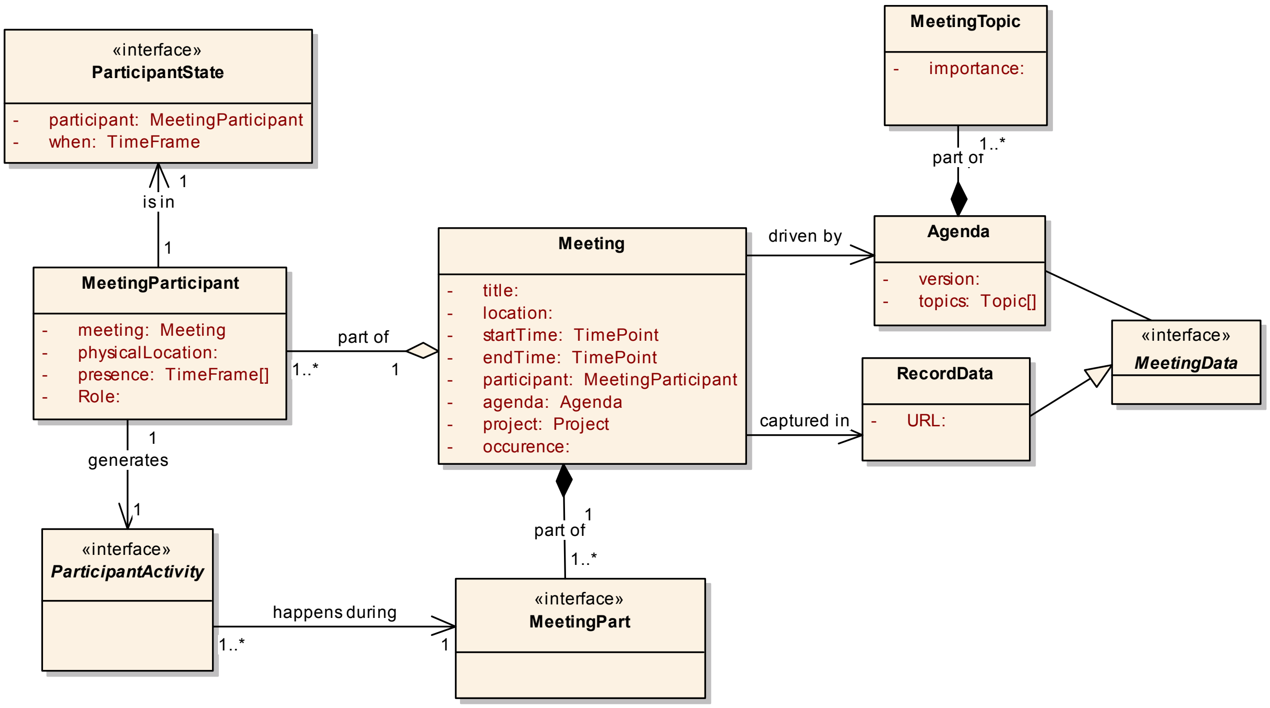
Tasks:

- ✓ Relate participants / topics across meetings
- ✓ Assess meeting (inter)activity
- ✓ Assess mood
- ✓ Assess participant talkativity / participation
- ✓ Derive post-meeting agenda
- ✓ Create minutes
- ✓ Compare pre- and post-meeting agendas
- ✓ Classify meetings
- ✓ Create meeting relationships (per participants/per topics/per location/per date/…)


Participant
Customer
Analyst
«include»
«extend»
«include»
(Personal) questions based on memory
Where did this person (I) seat?
What was the participant layout?
Whom did I talk to (at that time)?
what was his/her role?
Who talked to whom (at a given time)?
Who was there?
Who/when entered/left (at a given time/when)?
Questions on participants
Who presented?
What was presented?
Questions on actions
Answer questions
(When) were there decisions taken?
Were there conclusion reached?
(When) were there documents given?
Were there votes made?
Review discussion
Questions on topics
Were there topic followed from previous meeting?
Were there topic postponed?
Review argumentation
Was there (when) any "fight"?
What was the agenda/ timing/ structure?
What was discussed?
What that topic discussed?
Create annotation
Create minutes
Compare pre- and post agenda
Assess participant participation/ talkativity
Assess (inter)activity
Measure parameters
Assess mood
Relate topics across meetings
Create meeting relationships
Relate participant (behaviour) across meetings
per participant
Classify meetings
Per topic
per location

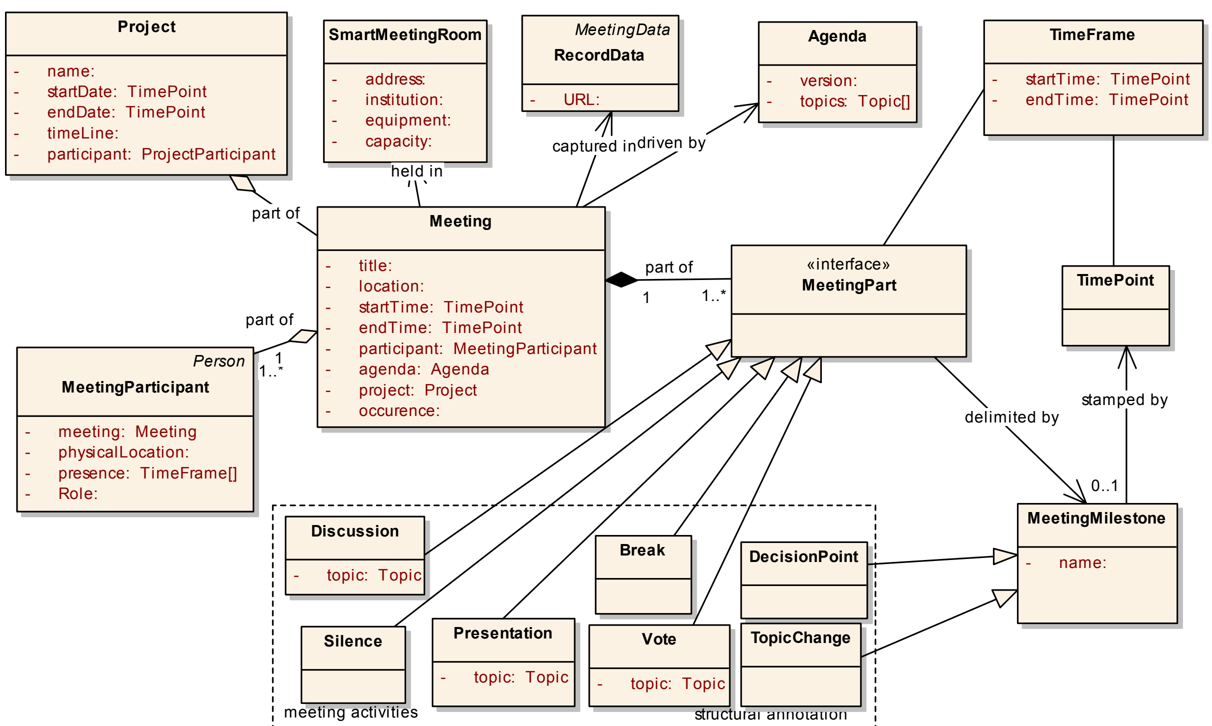
Meeting data modelling

We now derive the data model of the meeting, combining several levels.



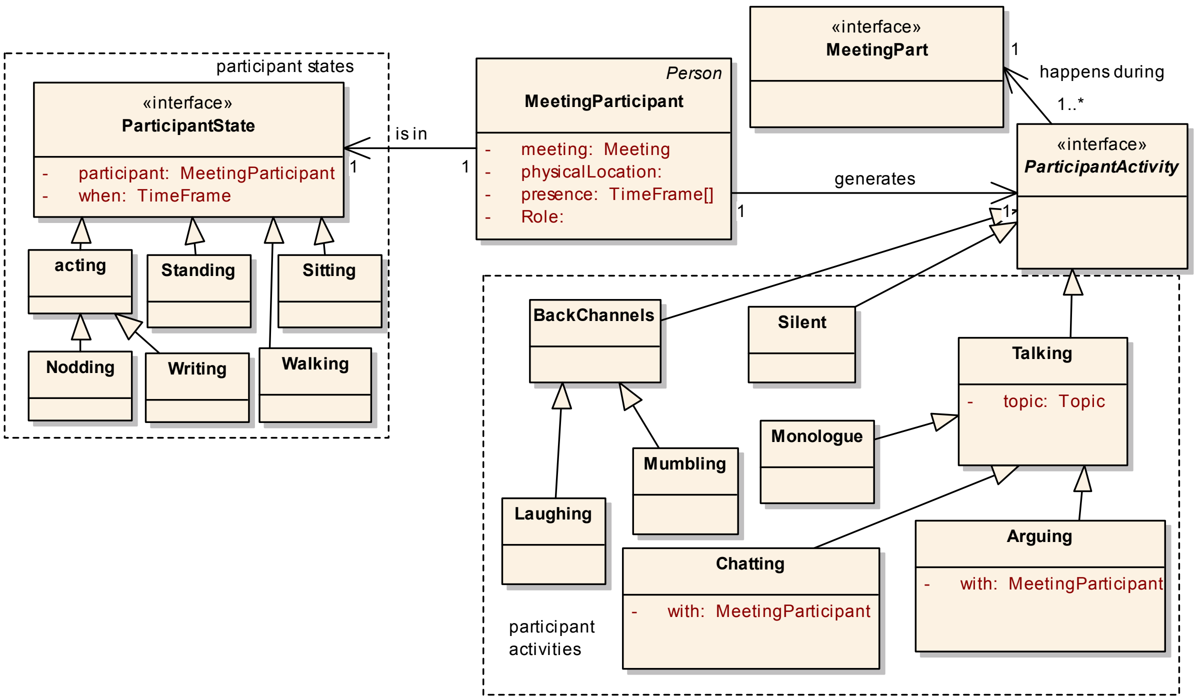
At the uppermost level, we essentially define key data structures as the meeting, which is composed of meeting parts, driven by an agenda (possibly in several versions) and recorded into record data. Both then become part of the data stored wrt a meeting, the meeting data. Meeting participants take part to the meeting and can, at any moment, be in a given (physical) state performing a given activity (meeting involvement).

We now take a closer look at the model centred around the meeting.



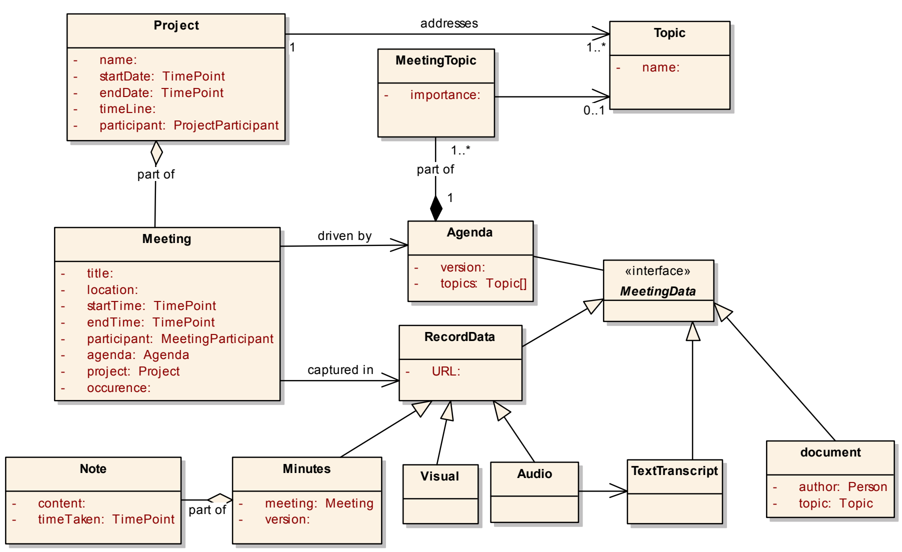
A meeting is part of a project, held in a given meeting room that will capture the data. The meeting is driven by an agenda that relates to topics, which then become meeting topics. This gives link to the external world (see below).
A meeting may be structured in terms of its temporal activities. These are defined by the meeting states presented earlier. Their delimiters are said to be meeting milestones that will form key points for the structuring of the meeting. Examples of these milestones are decision points and topic change points. They all relate to a specific time point, and their combination form meeting timeframe that will be referred to during information search and labelled using meeting activity titles.

Meetings are made by participants, we now take a close look at them.



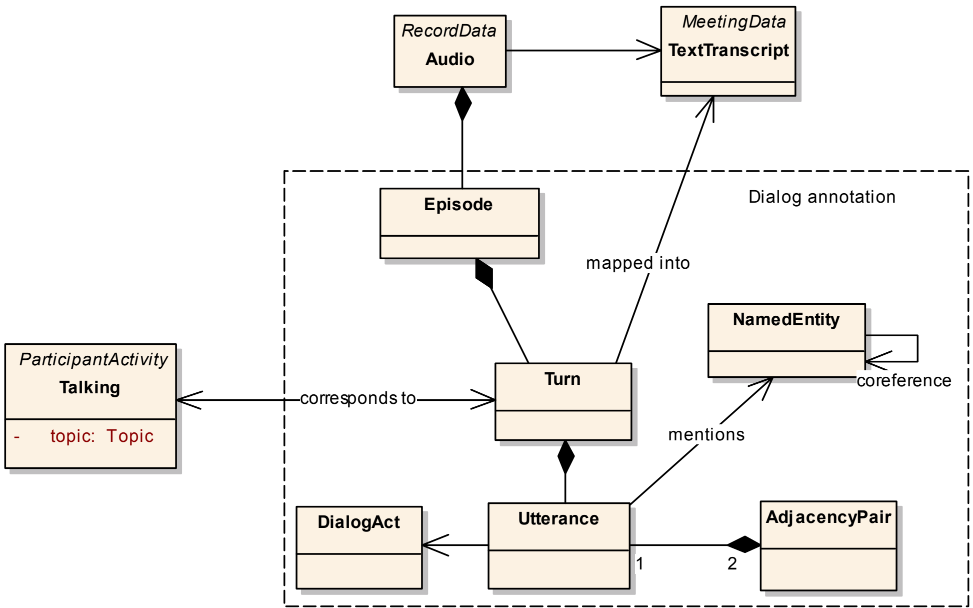
At any point in time (during a meeting part), the participant is in a given physical state, corresponding to the earlier analysis. Similarly, the participant is involved in a given interactive process that defines his/her current meeting involvement. The model is intuitive in that it defines a participant as being silent, talking or "emitting backchannels" (emitting sound that is not speech – laught, mumbling, etc).

As mentioned above, the meeting is recorded and associated with a number of data, as shown in the following diagram.

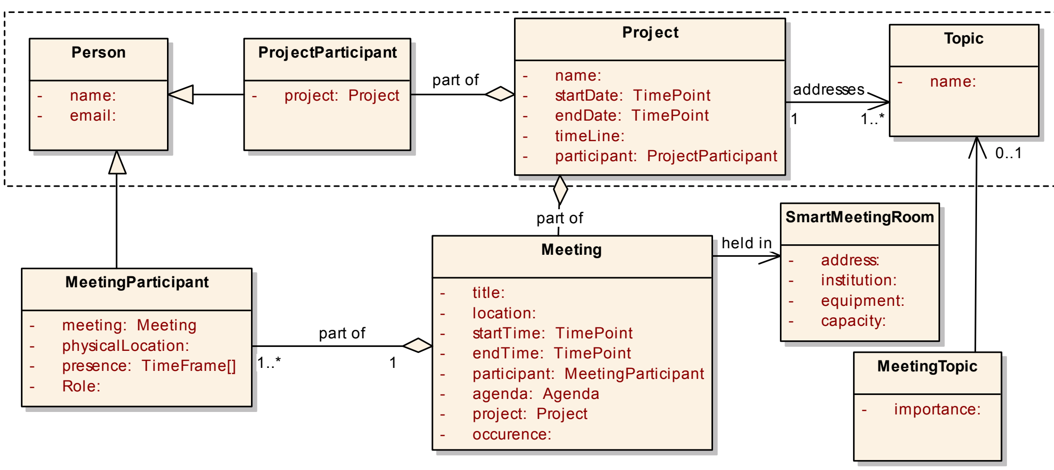


Typically, the meeting room will help in recording meeting data (video and audio). In parallel, these will be stored as data associated to the meeting, as will be documents (eg distributed during the meeting). The agenda, that is a composition of topics related (or not) to the project will be stored likewise. Finally, minutes, as a set of notes (taken or not during the meeting) and the text transcripts will also be stored in this repository.

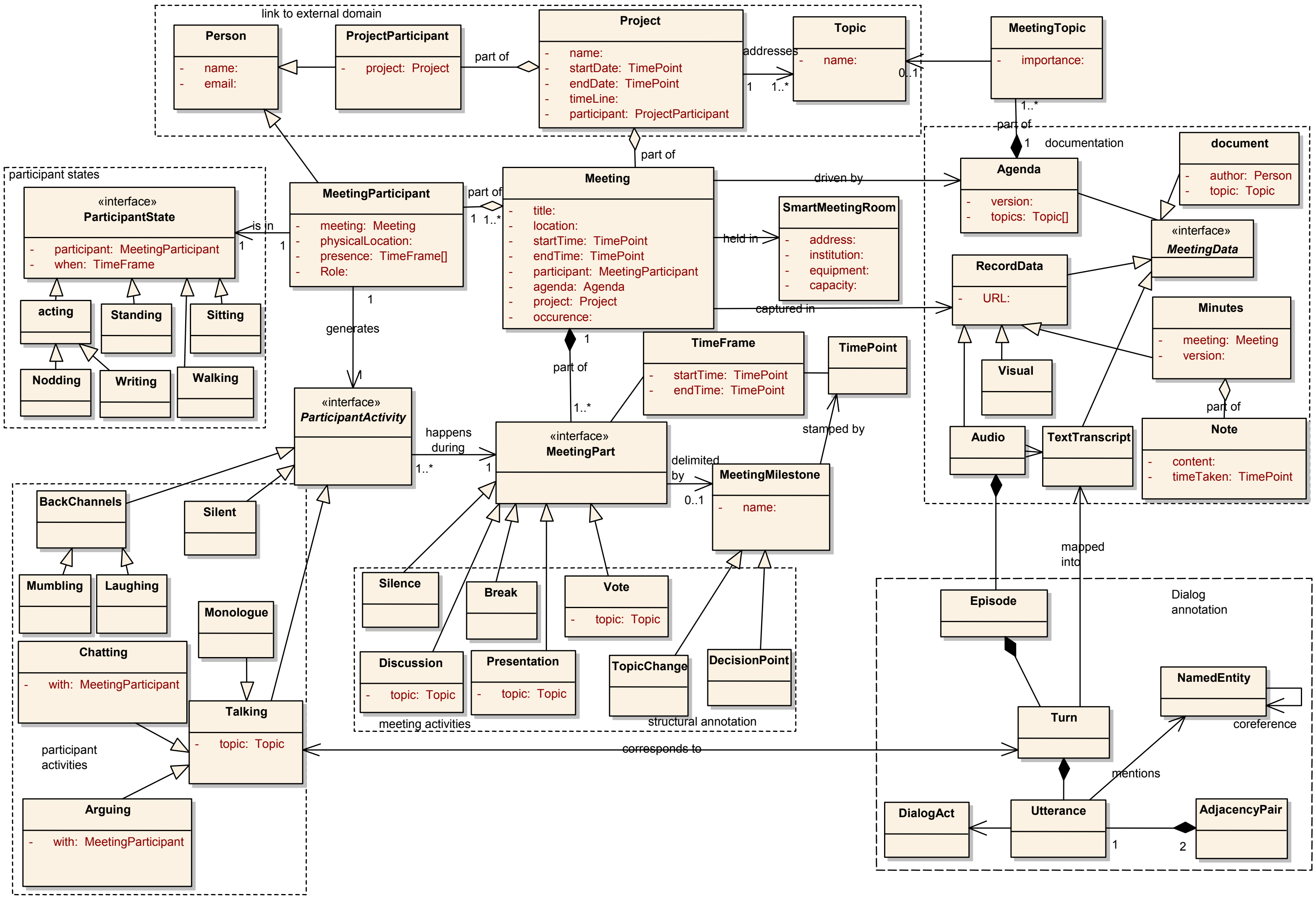
We eventually integrate the annotation of what was said in the meeting into this model.



This structure is the direct mapping of the taxonomy defined by linguists. The integration points are the meeting audio record and corresponding transcripts in terms of the data produced and the participant "talking" activity in terms of identification.
As a key to extending and relating this model with other models, we give pointers to external entities.

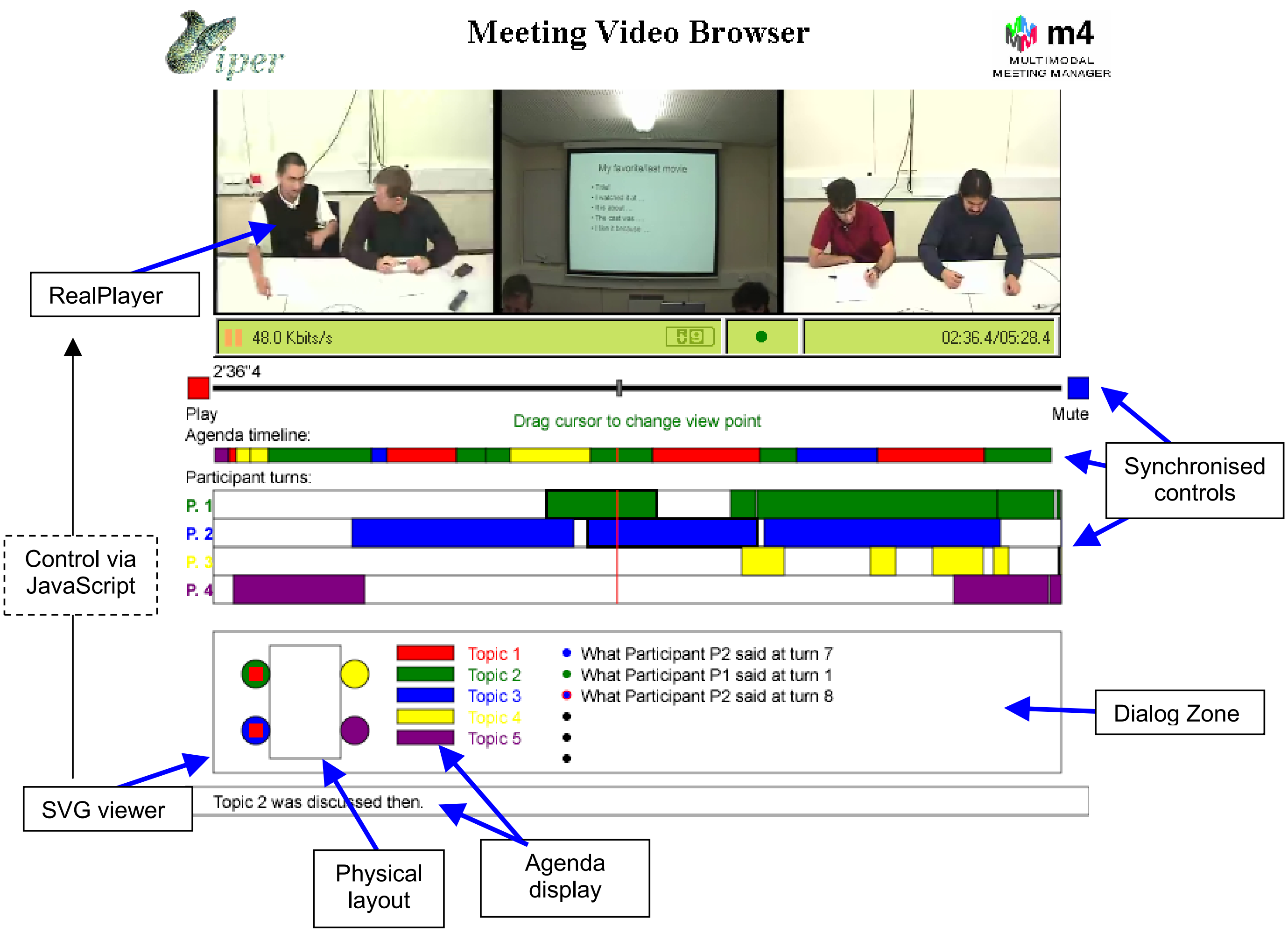


All this is then integrated into the complete meeting data model presented below.


link to external domain
Person
- name:
- email:
ProjectParticipant
- project: Project
part of
Project
- name:
- startDate: TimePoint
- endDate: TimePoint
- timeLine:
- participant: ProjectParticipant
addresses
1
1..*
Topic
- name:
0..1
MeetingTopic
- importance:
1..*
part of
1
documentation
document
- author: Person
- topic: Topic
Agenda
- version:
- topics: Topic[]
«interface»
MeetingData
RecordData
- URL:
Minutes
- meeting: Meeting
- version:
part of
Note
- content:
- timeTaken: TimePoint
Visual
Audio
TextTranscript
participant states
«interface»
ParticipantState
- participant: MeetingParticipant
- when: TimeFrame
is in
1
1
acting
Standing
Sitting
Nodding
Writing
Walking
MeetingParticipant
- meeting: Meeting
- physicalLocation:
- presence: TimeFrame[]
- Role:
part of
1
1..*
Meeting
- title:
- location:
- startDate: TimePoint
- endDate: TimePoint
- participant: MeetingParticipant
- agenda: Agenda
- project: Project
- occurence:
part of
driven by
held in
SmartMeetingRoom
- address:
- institution:
- equipment:
- capacity:
captured in
1
generates
1
part of
1..*
TimeFrame
- startTime: TimePoint
- endTime: TimePoint
TimePoint
stamped by
«interface»
ParticipantActivity
happens
during
1..*
1
«interface»
MeetingPart
delimited
by
0..1
MeetingMilestone
- name:
BackChannels
Silent
Mumbling
Laughing
Monologue
Chatting
- with: MeetingParticipant
Talking
- topic: Topic
participant
activities
Arguing
- with: MeetingParticipant
Silence
Break
Vote
- topic: Topic
Discussion
- topic: Topic
Presentation
- topic: Topic
TopicChange
DecisionPoint
meeting activities
structural annotation
mapped
into
Dialog
annotation
Episode
NamedEntity
coreference
Turn
corresponds to
mentions
DialogAct
Utterance
AdjacencyPair
1
2

Database structure:
An actual database structure can be derived from the previous data model. TDB

Meeting Browser:
As an highlight of the capabilities SVG and SMIL to create a demonstrator, we propose an online Video Meeting Browser composed essentially of a RealPlayer driven by SVG within a Adobe SVG Viewer. These are standard technologies that are progressively incorporated in all browsers.
The browser is composed of a video viewing video and a control slider to scroll at any position.
The agenda is shown as a composition of topics, materialised with respective colours. Clicking at any position seeks within the top video.



Live version at http://viper.unige.ch/~marchand/projects/M4/player.html

As it was recognised at several occasions in this project, speaker turns are an essential cue for browsing. Here, speaker turns are displayed as consistent colour blocks and clicking on one of these sets the play at the beginning of the speaker turn (note that all camera remain synchronised). Whenever playing, the timeline is synchronised and active speakers are highlighted at their Physical location. Likewise, dialog transcripts appear as a roll-over fashion.

This initial browser shows that important information can be displayed and highlighted using these technologies. Surely, extra interactivity can be added, including:

- ✓ Complete interactivity: Clicking on any item triggers the playing at this location (mostly done)
- ✓ Displaying physical Smart Meeting Room equipments for complete understanding of the scene. This could be completed by an interactive view for synthesising unseen views.
- ✓ Display statistics: speaker "talkativity", topic importance. As percentages, clickable pie-charts…

- ✓ Add scalability. It is clear that the above interface fits well in a context of short meetings. For longer meetings with a high number of speaker turns, zooming facilities and hierarchical browsing should be considered.
- ✓ Enhance symbolic coding: In the proposed interface, both participants and topics are colour-coded. This may be confusing. There is a need for defining truly efficient symbolic coding for intuitive use of the interface. The use of a metaphor in this context can prove useful.
- ✓ Augment usability: propose more complete (interactive) help for functionalities.

The optimal aim would be to create an incremental process where some feedback (either data input or data validation) would be sent to the database during browsing. Starting with an empty interface (looking like a simple player), the user would then input metadata (as in data on data, structuring data) and readily use this data to describe and browse the raw data incrementally and hierarchically. We believe that this is definitely possible and manageable use web technologies such as SVG and SMIL. This would include:

- ✓ Validation of speaker turn boundaries. The seeding point could be the choice or random chunks and the user would be asked about the homogeneity of the chunk. Based on several answers, a process would then infer correct boundaries.
- ✓ Idem for topic selection. The link between speaker turns and agenda points would be created interactively.
- ✓ Keyframe/milestone marking. It is important that important meeting points (visual or otherwise) can be decided and fixed while browsing.

Remarks:

All the above can clearly be generated automatically from the data stored in a DB that would embed the data model proposed earlier. The data would be read and formatted on request by eg a servlet or a CGI-like script. For now, it is a Matlab script that generates random data.
Not only SVG allows for the creation of distributed interfaces but is a nice solution for the design of complex interfaces. The reuse of the above code is made direct with the use of opensource Java SVG interpreter such as the Apache Batik package.

References:

- ✓ M4 project mails and discussions.
- ✓ IM2 project mails and discussions.
- ✓ S. Renals and D. Ellis, Audio information access from meeting rooms. ICASSP 2003.
- ✓ W3C SVG web page : http://www.w3.org/Graphics/SVG/Overview.htm8
- ✓ Adobe SVG zone: http://www.adobe.com/svg/
- ✓ RealPlayer home page : http://www.real.com/
- ✓ SVG Essentials. J. David Eisenberg O'Reilly Publishers. February 2002.
- ✓ Apache's Batik homepage: http://xml.apache.org/batik/index.html