MIAMM – A Multimodal Dialogue System Using Haptics

Norbert Reithinger, Dirk Fedeler, Ashwani Kumar, Elsa Pecourt, Christoph Lauer, Laurent Romary

Chapter 1

MIAMM – A MULTIMODAL DIALOGUE SYSTEM USING HAPTICS

Norbert Reithinger, Dirk Fedeler
DFKI – German Research Center for Artificial Intelligence
Saarbrücken, Germany
{Norbert.Reithinger, Dirk.Fedeler}@dfki.de

Ashwani Kumar
LORIA, Nancy, France
Ashwani.Kumar@loria.fr

Christoph Lauer, Elsa Pecourt
DFKI – German Research Center for Artificial Intelligence
Saarbrücken, Germany
{Christoph.Lauer, Elsa.Pecourt}@dfki.de

Laurent Romary
LORIA, Nancy, France
Laurent.Romary@loria.fr

Abstract In this chapter we describe the MIAMM project. Its objective is the development of new concepts and techniques for user interfaces employing graphics, haptics and speech to allow fast and easy navigation in large amounts of data. This goal poses challenges as to how can the information and its structure be characterized by means of visual and haptic features, how the architecture of such a system is to be defined, and how we can standardize the interfaces between the modules of a multi-modal system.

Keywords: Multimodal dialogue system, haptics, information visualization.

1. Introduction

Searching in information services for a certain piece of information is still exhausting and tiring. Imagine a user who has a portable 30 GB MP3 player. All titles are attributed with the most recent metadata information. Now he wants to search through the thousands of titles he has stored in the handheld device, holding it with one hand and possibly operating the interface with the other one, using some small keyboard, handwriting recognition or similar means. He can enter the search mode and select one of the main music categories, the time interval, or a large number of genres and format types. Scrolling through menu after menu is neither natural nor user adapted. Even recent interface approaches like the iPod navigation solve these problems only partially, or even negate it, like the iPod shuffle, which defines randomness and lack of user control as a cool feature.

Basically, we have two problems here: a user request that must be narrowed down to the item the user really wants, and the interface possibilities of such a small device. If we apply a (speech-) dialogue interface to this problem, the dialogue to extract exactly the title the user wants might be very lengthy. On the other hand, a menu-based interface is too time consuming and cumbersome due to the multitude of choices, and not very usable on a mobile device, due to dependence on graphical input and output.

The main objective of the MIAMM project (http://www.miamm.org/)[1] is to develop new concepts and techniques in the field of multimodal interaction to allow fast and natural access to such multimedia databases (see [Maybury and Wahlster, 1998] for a general overview on multimodal interfaces). This implies both the integration of available technologies in the domain of speech interaction (German, French, and English) and multimedia information access, and the design of novel technology for haptic designation and manipulation coupled with an adequate visualisation.

In this chapter we will first motivate the use of haptics and present the architecture of the MIAMM system. The visualization of the data guides the haptic interaction. We introduce the main concepts that use conceptual spaces and present the current visualization possibilities. Then we introduce the dialogue management approach in MIAMM, that divides into multimodal fusion and action planning. Finally, we give a short introduction to MMIL, the data exchange and representation language between the various modules of the system.

[1]The project MIAMM was partially funded by the European Union (IST-2000-29487) from 2001 – 2003. The partners are LORIA (F, coordinating), DFKI (D), Sony Europe (D), Canon (UK), and TNO (NL). The responsibility for this contribution lies with the authors.

2. Haptic Interaction in a Multimodal Dialogue System

2.1 Haptic as a New Modality in Human-Computer Interaction

One of the basic senses of humans is the haptic-tactile sense. In German, to understand can be uttered as *begreifen* – to grip – indicating that you really command a topic only after thoroughly touching it. How things feel like, how parts of a mechanism interact, or which feedback an instrument provides, are important cues for the interaction of humans in their natural and technological environment. Not surprisingly, the tactile and sensory motoric features of a new product are traditionally included in the design decisions of manufacturers, e.g. of carmakers. Also in areas like remote control haptics is commonly considered as an important interaction control and feedback channel[2].

Therefore, it is surprising that this modality only recently gains attention in the human computer interaction community. One can speculate whether the disembodied world of zeroes and ones in the computer distances us too much from the real world. However, with the advent of advanced graphical virtual worlds, getting embodied feedback is more and more important. A forerunner of this trend, as in many other areas, is video gaming where the interaction with the virtual world calls for physical feedback. Over the last years, force-feedback wheels and joysticks provide the players with feedback of his interaction with the game world.

While these interactions manipulate virtual images of real scenes, our goal in MIAMM is to interact in complex and possibly unstructured information spaces using multiple modalities, namely speech and haptics. Speech dialogue systems are nowadays good enough to field them with simple tasks. The German railway, for example, split their train timetable service in 2002 in a free-of-charge speech dialogue system and a premium cost, human operated service.

Haptic interaction in dialogue systems is rather new, however. Basically we are facing the following challenges:

- How do we visualize information and its structure?
- Which tactile features can we assign to information?
- How can we include haptics in the information flow of a dialogue system?

We will address these questions in the sections below.

To give an impression of the envisioned end-user device, the (virtual) handheld MIAMM appliance is shown in Figure 1.1. The user interacts with the

[2] See e.g. http://haptic.mech.nwu.edu/ for references.

device using speech and/or the haptic buttons to search, select, and play tunes from an underlying database. The buttons can be assigned various feedback functions. Haptic feedback can also provide e.g. the rhythm of the tune currently in focus through tactile feedback on the button. On the top-right side is a jog dial that can also be pressed. All buttons provide force feedback, depending on the assigned function and visualization metaphor.

Figure 1.1. The simulated PDA device.

2.2 The Architecture of the MIAMM System

The timing of haptic interaction is another, not only technical challenge. Let's consider the physiology of the sensory motoric system: the receptors for pressure and vibration of the hand have a stimulus threshold of 1μm, and an update frequency of 100 to 300 Hz [Beyer and Weiss, 2001]. Therefore, the feedback must not be delayed by any time consuming reasoning processes to provide a realistic interaction: if the haptic feedback reaction of the system is delayed beyond the physiological acceptable limits, it will be an unnatural interaction experience.

Therefore, processing and reasoning time plays an important role in haptic interaction that has to be addressed in all processing stages of MIAMM. In 2001, the participants of the Schloss Dagstuhl workshop "*Coordination and Fusion in Multimodal Interaction*"[3] discussed in one working group architectures for multimodal systems (WG 3). The final architecture proposal follows in major parts the "standard" architecture of interactive systems, with the consecutive steps mode analysis, mode coordination, interaction management, presentation planning, and mode design. For MIAMM we discussed this reference architecture and checked its feasibility for a multimodal interaction system using haptics. We came to the conclusion that a more or less pipelined architecture does not suit the haptic modality. For modalities like speech, no immediate feedback is necessary: you can use deep reasoning and react in the time span of about one second.

As a consequence, our architecture (see Figure 1.2) considers the modality specific processes as modules which may have an internal life of their own: only important events must be sent to the other modules, and modules can ask about the internal state of other modules.

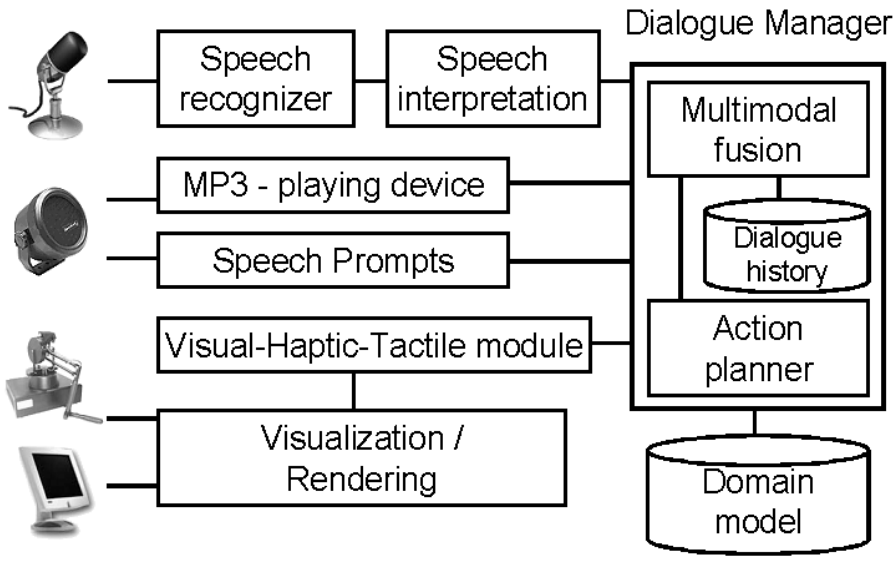

Figure 1.2. The MIAMM Architecture.

The system consists of two modules for natural language input processing, namely recognition and interpretation. On the output side we have a MP3-player to play the tunes, and pre-recorded speech prompts to provide acoustic

[3] See http://www.dfki.de/∼wahlster/Dagstuhl_Multi_Modality/ for the presentations.

feedback. The visual-haptic-tactile module (VisHapTac) is responsible for the selection of the visualization and for the assignment of haptic features to the force-feedback buttons. The visualization module renders the graphic output and interprets the force to the haptic buttons imposed by the user. The results are communicated back to the visual-haptic-tactile module. The dialogue manager consists of two main blocks, namely the multimodal fusion which is responsible for the resolution of multimodal references and of the action planner. A simple dialogue history provides contextual information. The action planner is connected via a domain model to the multi-media database. The domain-model inference engine facilitates all accesses to the database.

In the case of the language modules, where reaction time is important, but not vital for a satisfactory experience of the interaction, every result, e.g. an analysis from the speech interpretation, is forwarded directly to the consuming agent. The visual-haptic and the visualization modules with their real-time requirements are different. The dialogue manager passes the information to be presented to the agent, which determines the visualization. It also assigns the haptic features to the buttons. The user can then use the buttons to operate on the presented objects. As long as no dialogue intention is assigned to a haptic gesture, all processing will take place in the visualization module, with no data being passed back to the dialogue manager. Only if one of these actions is e.g. a selection, it passes back the information to the dialogue manager via the visual-haptic-tactile module autonomously. If the multimodal fusion needs information about objects currently in the visual focus, it can ask the visual-haptic agent.

The whole system is based partly on modules already available at the partner institutions, e.g. speech recognizers, speech interpretation or action planning, and modules that are developed within the project. The haptic-tactile interaction uses multiple PHANToM devices (http://www.sensable.com/), simulating the haptic buttons. The graphic-haptic interface is based on the GHOST software development kit provided by the manufacturer of the PHANToMs. The 3-D models for the visualizations are imported via an OpenGL interface from a modelling environment. The inter-module communication is based on the “Simple Object Access Protocol” (SOAP), a W3C recommendation for a lightweight protocol to exchange information in a decentralized, distributed environment. However, since the protocol adds a significant performance penalty to the system, we developed a solution that uses the message structure of SOAP, but delivers messages directly, if all modules reside in the same execution environment.

3. Visual Haptic Interaction – Concepts in MIAMM

The aim of the Visual Haptic Interaction (VisHapTac) module in the MIAMM system is to compute the visualization for the presentation requested by the dialogue management. Therefore, it has to find an adequate way to display a given set of data and to provide the user with intuitive manipulation features. This also includes the interpretation of the haptic user input. To do this VisHapTac has to analyse the given data with respect to predefined characteristics and it has to map them to the requirements of visualization metaphors. In the next paragraphs we show briefly what visualization metaphors are, which metaphors we use in the MIAMM project, and which requirements they have to fulfil. We discuss also which data characteristics are suitable for the system, where they come from and how they influence the selection for a visualization metaphor. A general overview on visualization techniques is to be found e.g. in [Card et al., 1999].

Figure 1.3. The wheel visualization.

3.1 Visualization Metaphors

A visualization metaphor (based on the notion of Conceptual Spaces, see [Gärdenfors, 2000] is a concept for the information presentation related to a real world object. Manipulating the presented data should remind the user to the handling of the corresponding object. An example is a *conveyor belt* where things are put in a sequence. This metaphor can be used for presenting a list of items. Scrolling up or down in the list is then represented by turning the belt to one or the other side.

For the MIAMM project we use the following visualization metaphors (as presented in [Fedeler and Lauer, 2002]):

Figure 1.4. The timeline visualization.

3.1.1 The visualization metaphor "conveyor belt/wheel". The wheel visualization displays a list that can endlessly be scrolled up and down with the haptic buttons. The user can feel the clatter of the wheel on the buttons. The "conveyor belt/wheel" metaphor is used as described above with a focus area in the middle of the displayed part of it. It is suitable for a one-dimensional, not

necessarily ordered set of items. So it is one of the less restricted visualization metaphors, which means that the wheel is a good candidate to be the default visualization for every kind of incoming data, when there is no good criterion for ordering or clustering information. Also for a small set of items (less than 30) this metaphor gives a good overview of the data.

3.1.2 The visualization metaphor "timeline". The timeline visualization is used for visualizations, where one data dimension is ordered and has sub-scales. One example is date information with years and months. The user stretches and compresses the visible time scope like a rubber band using the haptic buttons, feeling the resistance of the virtual material. Usually, in the middle of the visualized part of the timeline a data entry is highlighted to show the focussed item. The user can select this highlighted item for the play list, or can directly play it, e.g., by uttering "*Play this one*".

Figure 1.5. The lexicon visualization.

3.1.3 The visualization metaphor “lexicon”. The lexicon visualization displays a sorted set of clustered items similar to the “rolodex” file card tool. One scalar attribute of the items is used to cluster the information. For example, the tunes can be ordered alphabetically using the singer’s name. Each item is shown on a separate card and separator cards labelled with the first letter divide the items with different first letters. Since only one card is shown at a time detailed descriptions of the item can be presented using this visualization. The navigation in this visualization is similar to the wheel. The user browses through the cards by rotating the rolodex with the buttons and the dial. A stronger pressure increases the speed of the rotation.

Figure 1.6. The map visualization.

3.1.4 The visualization metaphor “map/terrain”. The map or terrain visualization metaphor clusters information according to the main characteristics – in the example figure according to genres and subgenres – and groups them in neighbourhoods. A genetic algorithm with an underlying physical model generates the map. It guarantees that different characteristics are in distant areas of the map, while common structures are in a near neighbourhood.

The user navigates through the map with the buttons, “flying” through the visualization. He can zoom into the map and finally select titles. This visualization is especially useful to present two-dimensional information, which has inherent similarities. Distance and connections between the separate clusters can be interpreted as relations between the data.

3.2 Data Characteristics

The basic step when choosing a visualization metaphor is to characterise the underlying data. Some important characteristics are:

- Numeric, symbolic (or mixed) values;
- Scalar, vector or complex structure;
- Unit variance;
- Ordered or non-ordered data sets;
- Discrete or continuous data dimensions;
- Spatial, quantity, category, temporal, relational, structural relations;
- Dense or sparse data sets;
- Number of dimensions;
- Available similarity or distance metrics;
- Available intuitive graphical representation (e.g. temperature with colour);
- Number of clusters, that can be built and how the data is spread over them.

The domain model of MIAMM is the main source of this information. It models the domain of music titles utilizing some of the MPEG-7 data categories. In the description of the model the applicable data characterization for each information type are stored. Additional information, for instance about how many possible items there are for an attribute, has also to be examined. This can be used, e.g. for clustering a data set.

The visualization metaphors, too, have to be reviewed in order to get information about their use with the various characteristics, which therefore define the requirement for a visualization metaphor. Requirements are strongly depending on the virtual objects a visualization denotes. As an example, the virtual prototype with the “conveyor belt” metaphor as it is shown above can display about ten items, so the list should be limited to about 30 items to be manageable for the user on a PDA while the map visualizes the whole database.

3.3 Planning the Presentation and the Interaction

When a new presentation task is received from the dialogue manager it has to be planned how the content data will be displayed and how the user will interact with the visualization using the haptic buttons. This planning process consists of the following steps:

1 The incoming data has to be analysed with respect to the characteristics stored in the domain model. Also the size of the given data set is an important characteristic as some visualization metaphors are to be preferred for small data sets, as shown in the example above. It has to be examined whether the data can be clustered with respect to the different attributes of the items. To estimate how useful the different kinds of cluster building are, the number and size of the clusters is important. For instance, a handful of clusters with the data nearly equally spread between them can give a good overview of the presented information.

2 A mapping has to be found between the characteristics of the data and the requirements of the visualization metaphors. Therefore a kind of constraint solver processes this data in several steps.

   (a) The necessary characteristics and requirements are processed first. They are formulated as constraints in advance as they only depend on the non-dynamic part of the visualization metaphors (see above: "*data characteristics*").

   (b) Strongly recommended information – if available – is added. This could be user preferences or information for the coherence of the dialogue. One example is to use the same visualization metaphor for the same kind of data.

   (c) If there are additional preferences like button assignment – e.g., using the index finger for marking and not the thumb – they are processed in the last step.

In addition to the selection of a metaphor, a list of configurations and meta information is computed which will be used for further initialising the visualization. Then the content data is reformulated with respect to the selected visualization metaphor including the additional information and provided to the following sub module.

The next step in the processing is the visualization/rendering module, which computes a visualization from a graphics library of metaphors and fills in the configuration data containing the content ('what to show') and the layout ('how to show'), including the layout of the icons on the PDA's screen. It then initiates the interaction, i.e. it provides the call-backs that map the user's haptic

input to the visualization routines. If the user presses the button, the tight coupling of graphic elements to functions processing the response enables the immediate visual and haptic-tactile feedback.

4. Dialogue Management

4.1 Architecture of the Dialogue Manager

The Dialogue Manager (DM) plays a central role within the MIAMM architecture, as it is the module that controls the high-level interaction with the user, as well as the execution of system internal actions like database access. Its tasks are the mapping of the semantic representations from the interpretation modules onto user intentions, the update of the current dialogue context and task status on the basis of the recognized intentions, the execution of the actions required for the current task (e.g. database queries), and finally the generation of an output through the output layers such as speech, graphics and haptic feedback.

The DM is required to cope with possibly incomplete, ambiguous or wrong inputs due to linguistic phenomena like anaphora or ellipsis, or to errors in previous layers. Still in these situations the DM should be able to provide an appropriate answer to the user, resolving the ambiguities or initiating a clarification dialogue in the case of errors and misunderstandings. Multimodality poses an additional challenge, as inputs in different modalities, possibly coming asynchronously, have to be grouped and assigned a single semantic value.

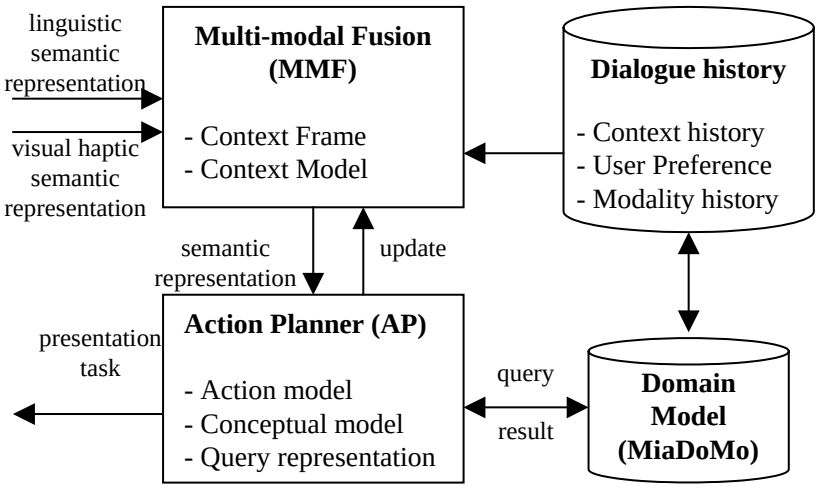

Figure 1.7. Functional architecture of the Dialogue Manager.

Based upon these functional requirements, the DM is decomposed in two components (see Figure 1.7): the multimodal Fusion component (MMF) and the Action Planner (AP). Semantic representations coming from the Speech Interpretation (Spin) and the Visual Haptic Interaction (VisHapTac) modules are first disambiguated and fused by MMF, and then sent to AP. AP computes the system response and sends the required queries to the corresponding modules. Queries to the MIAMM database and to the devices are done through the domain model (MiaDoMo). The AP also sends presentation tasks to VisHapTac, to the MP3 Player, or activates speech prompts. All data flowing between modules, including communication between the DM components, is defined using MMIL, the data interchange format in MIAMM (see Section 5).

The underlying motivations for the decoupling of DM and MMF are first to account for modularity within the DM design framework to enable an integrative architecture, and second to provide for sequential information flow within the module. This aspect is crucial in multimodal systems, as the system cannot decide on action execution until all unimodal information streams that constitute a single message are fused and interpreted within a unified context. The functionality and design of the dialogue management components are outlined in the next two sections.

4.2 Multimodal Fusion

MMF assimilates information coming through various modalities and submodules into a comprehensive and unambiguous representational framework. Ideally, output of the MMF is free from all kinds of ambiguities, uncertainties and terseness. More specifically, MMF:

- Integrates discursive and perceptual information, which at the input level of MMF is encoded using lexical and/or semantic data categories as specified by the MMIL language;
- Assigns a unique MMILId, each time a new object enters into the discourse. This id serves as an identifier for the object within the scope and timeline of the discourse;
- Resolves ambiguities and uncertainties at the level of semantics;
- Updates the dialogue history, triggered by the user's utterances and various updates from other modules within MIAMM architecture.

Effectively, from a functional point of view the design of MMF can be divided into three mechanisms, which are further described in the following subsections.

4.2.1 Interpretation. This is the first step towards analysis of the semantic representation provided by Speech and VisHapTac layers, so as to identify semantically significant entities (discursive and perceptual) in the user's input. These discourse entities serve as potential referents for referring expressions. For example: in the user's utterance *show me the list* MMF identifies relational predicates such as /subject/, /object/ etc. and corresponding arguments such as *show*, *the list* etc. as semantically significant entities and these discourse entities are accommodated into the *live*[4] discourse context. Essentially, every information unit within the MMIL semantic representation serves as a cognitive model of an *entity*[5]. A typical minimal representation for an entity contains:

- A unique identifier;
- Type category for the entity.

Type is derived from a set of generic domains organized as type hierarchy, which is established in the Domain Model. We incorporate these representations into a cognitive framework named as *Reference Domains* [Salmon-Alt, 2000], which assimilates and categorizes discursive, perceptual and conceptual (domain) information pertaining to the entities. On the basis of the information content within the structures representing these entities, a reference domain is segmented into zero, one or more partitions. These partitions map access methods to reference domains and are used for uniquely identifying the referents.

Usually, perceptual and discursive prominence of the entities enables to single out a particular entity within a partition. Effectively, these prominence attributes are incorporated by the specific operation of *assimilation* on the pertinent reference domains. Triggered by discursive cues (e.g. prepositions, conjunctions, quantified negations, arguments of same predicate), assimilation builds associations (or disassociations) between entities or sets. Assimilation could be perceptually triggered (e.g. graphics and haptics triggers) as well.

For example, the user can command *play this one*, while haptically selecting an item from the displayed play list. The haptic trigger would entail assimilation of the participants of type /tune/ into a single reference domain and modifying its status to /infocus/. In other scenarios, when we have different kind of visualizations such as galaxy, perceptual criteria such as proximity and similarity can lead to grouping of contextual entities. Depending on the type of trigger, an entity or a set can be made prominent but it does not necessarily lead to a focussed domain (as in the case of conjunctions).

[4]Live discourse context refers to a unified representation framework which is a contextual mapping of user's recent utterances and system's responses.
[5]An entity represents an object, event or a state.

4.2.2 Dialogue progress and context processing. For the dialogue to progress smoothly, the reference domains, realized from the semantic representations, as outlined in the previous section, must be integrated in a procedural fashion. These mechanisms must reflect the continuity of the dialogue progress and should entail certain inference mechanisms, which could be applied upon such an integrated framework, so as to achieve the ultimate goal of fusing asynchronous multimodal inputs.

Inherently, task-oriented dialogues are characterized by an incremental building process, where with the perceived cognition of system's knowledge and awareness, the user strives towards fulfilling certain requirements which are necessary for the task completion. Indeed, these interactions go beyond simple slot-filling (or menu based) task requirements. At the level of dialogue progress, we can construe task-oriented dialogues as composition of several states named as *context frames*, which are individually constructed through incremental process. These states might be realized during a single utterance or can span several dialogue sequences. Dialogues are modelled as combination of incremental building and discrete transitions between these context frames. This is complimentary to the *information state* theories, prevalent in the literature. Indeed, the idea is to form a content representation in form of context frames, which have strong *localized* properties owing to highly correlated content structures, while across several such frames there is not much correlation.

The basic constituting units within a context frame representation are:

- A unique identifier, assigned by the MMF;
- Frame type, such as terminal or non-terminal;
- Grounding status about the user's input, based on the dialogue acts and the feedback report from the AP;
- Reference domains at various levels, as described in Section 4.2.1.

4.2.3 Reference resolution and fusion. Reference resolution strategies vary from one referring expression to another in the sense of differing mechanisms to partition the particular reference domain. One or more (in case of ambiguity) of these domains in the live context frame is selected and restructured by profiling the referent. The selection is constrained by the requirement of compatibility between the selected contextual domain and the under-specified domain constructed for the expression being evaluated [Salmon-Alt, 2000]. This entails restructuring mechanism at the level of context frames named as *merging* [Kumar et al., 2002], where the under-specified reference domains are integrated within the live context frame until the frame acquires the status of a discrete state, in which case it is pushed to the dialogue history.

The dialogue history comprises of the following three components, whose precise updating and retrieval processes are controlled by the MMF:

- Context History: is a repository of resolved semantic representations, in form of sequential context frames.

- Modality History: is a repository of modality interactions, which could not be integrated into the live context (possibly, because of the temporal *lead* of the modality event). If the modality history stack is not empty, all the member frames, which are within some time limit as compared to the live context frame, are tried for merging into the context frame. Besides, there are heuristics for deleting frames, if they remain unconsumed for long time and hence, rendered out of context.

- User Preferences: is a repository user's preferences built over the course of current and previous discourses.

In the output produced by MMF, all the pending references are resolved (in the worst case, few potential referents are provided) and the ensuing goal representation is passed to Action planner.

4.3 Action Planner

Task oriented cooperative dialogues, where both partners collaborate to achieve a common goal, can be viewed as coherent sequences of utterances asking for actions to be performed or introducing new information to the dialogue context. The task of the action planner is to recognize the user's goal, and to trigger the required actions for the achievement of this goal. The triggered actions can be internal, such as database queries and the updating of the internal state of the system, or external, like communication with the user. In other words, the action planner is responsible for the control of both the task structure and the interaction structure of the dialogue.

4.3.1 Interaction and task structure. The interaction and task structure are modelled as sequences of *communicative games*, that may include embedded sub-games. Each of these *communicative games* consists of two moves, an initiative move (I) and a response move (R), one of them coming from an input channel and the other going to an output channel (from the point of view of the AP). Each application goal, be it a user goal or an internal sub-goal, corresponds to one *communicative game*. Figure 1.8 shows a fragment of a sample dialogue from the MIAMM domain. This interaction consists, on the top level, of a *communicative game*, a simple "display game" including U1

and SI, and is played by the user[6] that makes the request, the AP that passes the request to the VisHapTac, and by the VisHapTac that displays the desired song list. This game includes an embedded "clarification game" (S1 and U2), and a "query game", which is played internally by the AP and the domain model.

U1: "Show me music"
S1: "What kind of music are you looking for?"
U2: "I want rock of the 80's"
S2: (shows the query results as a list)

Figure 1.8. Sample dialogue.

Interactions are thus viewed as joint games played by different agents, including the user and all the modules that directly communicate with the AP. The moves in each game specify the rules to play it. This approach allows the identification and unified modelling of recurrent patterns in interactions.

4.3.2 Interaction and task flow. To initiate the appropriate communicative game that will guide the interaction, the AP first has to recognize the overall goal that motivates the dialogue, i.e. it has to map a semantic representation coming from the MMF to a suitable application goal. These semantic representations include actions to be performed by the system, as well as parameters for these actions. The setting of a goal triggers the initiation of the corresponding "communicative game".

The subsequent flow of the game is controlled by means of non-linear regression planning with hierarchical decomposition of sub-goals, as used in the SmartKom project [Reithinger et al., 2003; Wahlster, 2005]. Each *communicative game* is characterized by its preconditions and its intended effects. On the basis of these preconditions and effects the AP looks for a sequence of sub-games that achieve the current goal. For example a "display game" requires a list of items and has the effect of sending a display request with this list as its parameter to VisHapTac, whereas a "database query game" requires a set of parameters to do the query and sends the query to MiaDoMo. If the preconditions are not met, AP looks for a game that satisfies them. After successful completion of this game, the triggering game is resumed. *Communicative games* specify thus a partially ordered and non-deterministic sequence of actions that lead to the achievement of a goal. Execution of system actions is interleaved with planning since we cannot predict the user's future utter-

[6]User is here an abstraction over the speech interpretation and visual haptics interaction modules. All inputs reaching the action planner pass through the multimodal fusion component. There they are fused and integrated. The action planner does not know which input layer the inputs originally came from.

ances. This strategy allows the system to react to unexpected user inputs like misunderstandings or changing of goals.

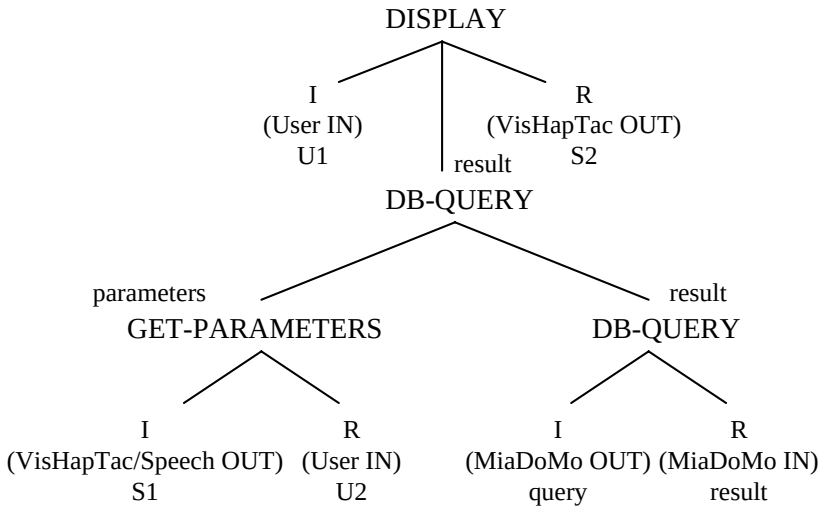

Figure 1.9. A sample communicative game.

Figure 1.9 illustrates the "display game" shown in Figure 1.8, spanning from U2 to U3. The names of the *communicative games* are written in capitals (DISPLAY, GET-PARAMETERS and DB-QUERY). Each game includes either an initiative-response (IR) pair or one or more embedded games. In this example the top-level game DISPLAY, includes an embedded game DB-QUERY, that itself includes two embedded games, GET-PARAMETERS and DB-QUERY. The leaves indicate moves, where I and R say if the move is an initiative or a response, and the data in brackets defines the channel from/to which the data flows. The arrows connecting *communicative games* show dependency relations. The label of the connecting arrows indicates the data needed by the mother-game that induced the triggering of a sub-game providing this data.

The GET-PARAMETERS game sends a presentation task to the VisHapTac, asking the user for the needed parameters. Similarly the DB-QUERY game sends a database query to the MiaDoMo. In both cases, the DM waits for the expected answer, as coded in the response part of the game, and provides it to the triggering game for further processing.

5. The Multimodal Interface Language (MMIL)

5.1 Design Framework

The Multimodal Interface Language (MMIL) is the central representation format of the MIAMM software architecture. It defines the exchange format

of data exchanged between the modules of the MIAMM system. It is also the basis for the content of the dialogue history in MIAMM, both from the point of view of the objects being manipulated and the various events occurring during a dialogue session. Therefore, the MMIL language is not solely dedicated to the representation of the interaction between the user and the dialogue system, but also of the various interactions occurring within the architecture proper, like, for instance, a query to the domain model. It provides a means to trace the system behaviour, in continuity as what would be necessary to trace the man-machine interaction. As a result, the MMIL language contains both generic descriptors related to dialogue management, comprising general interaction concepts used within the system and domain specific descriptors related to the multimedia application dealt with in the project.

This ambitious objective has a consequence on the design of the MMIL language. The language is formulated using XML: Schemata describe the admissible syntax of the messages passed through the system. Since the actual XML format is potentially complex, but above all, required some tuning as the design of the whole system goes on, we decided not to directly draft MMIL as an XML schema, but to generate this schema through a specification phase in keeping with the results already obtained in the SALT[7] project for terminological data representation, see [Romary, 2001]. We thus specify the various descriptors (or *data category*) used in MMIL in an intermediate format expressed in RDF and compatible within ISO 11179, in order to generate both the corresponding schema and the associated documentation, see [Romary, 2002a].

5.2 Levels of Representation – Events and Participants

Given the variety of levels (lexical, semantic, dialogue etc.) that the MMIL language must be able to represent, it is necessary to have an abstract view on these levels to identify some shared notions that could be the basis for the MMIL information architecture. Indeed, it can be observed that most of these levels, including graphical and haptic oriented representations, can be modelled as *events*, that is temporal objects that are given a type and may enter a network of temporal relations. Those events can also be associated with participants which are any other object either acting upon or being affected by the event. For instance, a lexical hypothesis in a word lattice can be seen as an event (of the lexical type), which is related to other similar events (or reified dates) by temporal relations (one hypothesis precedes another, etc.) and has at least one participant, that is the speaker, as known by the dialogue system.

Events and participants may be accessible in two different ways. They can be part of an information structure transferred from one module to another

[7] http://www.loria.fr/projets/SALT

within the MIAMM architecture, or associated to one given module, so that it can be referred to by any dependency link within the architecture. This mechanism of *registration* allows for factorisation within the MIAMM architecture and thus lighter information structures being transferred between modules.

Two types of properties describe events and participants:

- Restrictions, which express either the type of the object being described or some more refined unary property on the corresponding object;
- Dependencies, which are typed relations linking two events or an event to one of its participants.

From a technical point of view, dependencies can be expressed, when possible, by simple references within the same representation, but also by an external reference to an information structure registered within the architecture.

5.3 Meta-Model

From a data model point of view the MMIL structure is based on a flat representation that combines any number of two types of entities that represent the basic ontology of MIAMM, namely *events* and *participants*.

An *event* is any temporal entity either expressed by the user or occurring in the course of the dialogue. As such, this notion covers interaction event (spoken or realized through the haptic interface), events resulting from the interpretation of multimodal inputs or event generated by decision components within the dialogue system. For instance, this allows us to represent the output of the action planner by means of such an event. Events can be recursively decomposed into sub-events.

A *participant* is any individual or set of individuals about which a user says something or the dialogue system knows something about. Typical individuals in the MIAMM environment are the user, multimedia objects and graphical objects. Participants can be recursively decomposed into sub-participants, for instance to represent sets or sequences of objects.

Events and participants cover all the possible entities that the MIAMM architecture manipulates. They are further described by means of various descriptors, which can either give more precise information about them (restrictions) or relate events and participants with one another (dependencies). Both types of descriptors are defined in MMIL as Data Categories, but dependencies are given a specific status by being mostly implemented as <relation> elements attached to encompassing MMIL structure. Dependencies can express any link that can exist between two participants (e.g. part-whole relation), two events (temporal order), or between a participant and an event (“participants” to a predicate).

Events and participants can be iterated in the MMIL structure, which leads to the meta-model schematised in Figure 1.10, using the UML formalism. Furthermore, the representation shows an additional level for the representation of the temporal information associated with events.

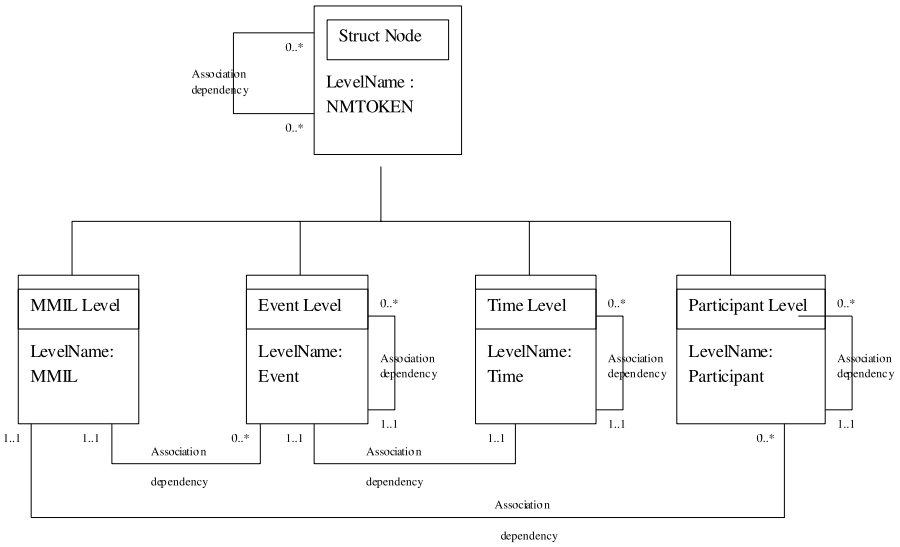

Figure 1.10. UML diagram representing the MMIL information meta-model.

5.4 Data Categories

Data category specifications are needed to identify the set of information units that can be used as restrictions and dependencies to instantiations of nodes from the meta-model. Following are the types of data categories incorporated within MMIL specifications:

- Data Categories describing both *events* and *participant*: general information such as identifiers, lexical value, attentional states, and ambiguities, about events or participants;
- Data categories for *events*: information pertaining to certain types of system-known events and functional aspect of user's expressions;
- Data categories for *participants*: exclusive information about participants such as generic types and other related attributes;
- Data categories for time level information: temporal positioning and duration for an event;

- Relations between *events* and *participants*: relation mappings between *events* and *participants*, using the knowledge available at certain stage of processing such as /object/, /subject/ etc.;

- Relations between *events*: propositional aspects and temporal relations among *events* such as /propContent/etc. ;

- Relations between *participants,* e.g., similarity relationships (see below).

5.5 Sample Illustration

Given those preliminary specifications, the representation of semantic content of a simple sentence like “play the song” would be as follows:

```
<mmilComponent>
         <event id="e0">
                  <evtType>speak</evtType>
                  <dialogueAct>request</dialogueAct>
                  <speaker target="User"/>
                  <addressee target="System"/>
         </event>
         <event id="e1">
                  <evtType>play</evtType>
                  <mode>imperative</mode>
                  <tense>Present</tense>
         </event>
         <participant id="p0">
                  <individuation>singular</individuation>
                  <objType>tune</objType>
                  <refType>definite</refType>
                  <refStatus>pending</refStatus>
         </participant>
         <participant id="User">
                  <objType>User</objType>
                  <refType>1PPDeixis</refType>
                  <refStatus>pending</refStatus>
         </participant>
         <relation
                  type="propContent"
                  source="e1"
                  target="e0"/>
         <relation
                  type="subject"
                  source="System"
                  target="e1"/>
         <relation
                  type="object"
                  source="p0"
                  target="e1"/>
         <relation
                  type="destination"
                  source="User"
                  target="e1"/>
</mmilComponent>
```

As can be seen from above, it is possible to mix information percolating from lower levels of analysis (like tense and aspects information) with more semantic and/or pragmatic information (like the referential status of the participant). Kumar and Romary [2003] illustrate and examine this representational

framework against typical multimodal representation requirements such as expressiveness, semantic adequacy, openness, uniformity and extensibility.

5.6 Additional Mechanisms

The sample illustration is very simple and obviously does not seem to be exhaustive and flexible enough for true multimodal interactions. Essentially, the MMIL design framework allows for certain additional mechanisms, see [Romary, 2002b] for details, which impart sufficient representational richness and integration flexibility within any kind of multimodal design:

- Alternatives and Ranges;
- Temporal positioning and duration;
- Refinements of data categories.

As specified in ISO 16642, it is possible, when needed, to refine a given data category by means of additional descriptors. Consider, e.g., that a similarity query is expressed by a /similar/ relation between two participants as follows:

```
<mmilComponent>
         ...
         <participant id="id1">
         ...
         </participant>
         <participant id="id2">
         ...
         </participant>
         <relation
                  type="similar"
                  source="id1"
                  target="id2"/>
         ...
</mmilComponent>


<mmilComponent>
         ...
         <participant id="id1">
         ...
         </participant>
         <participant id="id2">
         ...
         </participant>
         <relationGrp>
                  <relation
                           type="similar"
                           source="id1"
                           target="id2"/>
                  <dimension>genre</dimension>
                  <dimension>author</dimension>
         <relationGrp>
         ...
</mmilComponent>
```

It is possible to express more precisely the set of dimensions along which the similarity search is to be made, as illustrated immediately above.

6. Conclusion

The main objective of the MIAMM project was the development of new concepts and techniques for user interfaces employing graphics, haptics and speech to allow fast navigation in large amounts of data and easy access to it. This goal poses interesting challenges as to how can the information and its structure be characterized by means of visual and haptic features. Furthermore it had to be defined how the different modalities can be combined to provide a natural interaction between the user and the system, and how the information from multimodal sources can be represented in a unified language for information exchange inside the system.

The final MIAMM system combines speech with new techniques for haptic interaction and data visualization to facilitate access to multimedia databases on small handheld devices [Pecourt and Reithinger, 2004]. Interaction is possible in all three target languages German, French, and English. The final evaluation of the system supports our initial hypothesis that users prefer language to select information and haptics to navigate in the search space. The interaction proved to be intuitive in the user walkthrough evaluation [van Esch-Bussemakers and Cremers, 2004].

Nevertheless there are still open questions and further research is still needed to exhaust the possibilities that multimodal interfaces using haptics offer. This includes the conception of new visualization metaphors and their combination with haptic and tactile features, as well as the modelling and structuring of the data to take advantage of the expressivity of these modalities. The results of these investigations can provide interesting insights that help to cope with the problem of the constant growth of available information resources and the difficulty of its visualization and access.

References

Beyer, L. and Weiss, T. (2001). Elementareinheiten des Somatosensorischen Systems als Physiologische Basis der Taktil-Haptischen Wahrnehmung. In Grunewald, M. and Beyer, L., editors, *Der Bewegte Sinn*, pages 25–38. Birkhäuser Verlag, Basel.

Card, S. K., Mackinlay, J. D., and Shneiderman, B., editors (1999). *Readings in Information Visualization: Using Vision to Think*. Morgan Kaufmann Publishers Inc.

Fedeler, D. and Lauer, C. (2002). Technical Foundation of the Graphical Interface. Technical report, DFKI, Saarbrücken, Germany. Project MIAMM – Multidimensional Information Access using Multiple Modalities, EU project IST-20000-29487, Deliverable D2.2.

Gärdenfors, P. (2000). *Conceptual Spaces: The Geometry of Thought*. MIT Press, USA.

Kumar, A., Pecourt, E., and Romary, L. (2002). Dialogue Module Technical Specification. Technical report, LORIA, Nancy, France. Project MIAMM – Multidimensional Information Access using Multiple Modalities, EU project IST-20000-29487, Deliverable D5.1.

Kumar, A. and Romary, L. (2003). A Comprehensive Framework for Multi-Modal Meaning Representation. In *Proceedings of the Fifth International Workshop on Computational Semantics*, Tilburg, Netherlands.

Maybury, M. T. and Wahlster, W., editors (1998). *Readings in Intelligent User Interfaces*. Morgan Kaufmann Publishers Inc.

Pecourt, E. and Reithinger, N. (2004). Multimodal Database Access on Handheld Devices. In *The Companion Volume to the Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 206–209, Barcelona, Spain. Association for Computational Linguistics.

Reithinger, N., Alexandersson, J., Becker, T., Blocher, A., Engel, R., Löckelt, M., Müller, J., Pfleger, N., Poller, P., Streit, M., and Tschernomas, V. (2003). SmartKom: Adaptive and Flexible Multimodal Access to Multiple Applications. In *Proceedings of the Fifth International Conference on Multimodal Interfaces (ICMI)*, pages 101–108, Vancouver, Canada. ACM Press.

Romary, L. (2001). Towards an Abstract Representation of Terminological Data Collections - the TMF Model. In *Proceedings of Terminology in Advanced Microcomputer Applications (TAMA)*, Antwerp, Belgium.

Romary, L. (2002a). MMIL Requirements Specification. Technical report, LORIA, Nancy, France. Project MIAMM – Multidimensional Information Access using Multiple Modalities, EU project IST-20000-29487, Deliverable D6.1.

Romary, L. (2002b). MMIL Technical Specification. Technical report, LORIA, Nancy, France. Project MIAMM – Multidimensional Information Access using Multiple Modalities, EU project IST-20000-29487, Deliverable D6.3.

Salmon-Alt, S. (2000). Reference Resolution within the Framework of Cognitive Grammar. In *Proceedings of International Colloquium on Cognitive Science*, San Sebastian, Spain.

van Esch-Bussemakers, M. P. and Cremers, A. H. M. (2004). User Walkthrough of Multimodal Access to Multidimensional Databases. In *Proceedings of the Sixth International Conference on Multimodal Interfaces (ICMI)*, pages 220–226, State College, PA, USA. ACM Press.

Wahlster, W., editor (2005). *SmartKom: Foundations of Multimodal Dialogue Systems*. Springer, Berlin.