

Comprehensive Framework for MultiModal Meaning Representation

Ashwani Kumar, Laurent Romary

▶ **To cite this version:**

Ashwani Kumar, Laurent Romary. Comprehensive Framework for MultiModal Meaning Representation. fifth International Workshop on Computational Semantics (IWCS-5), Jan 2003, Tilburg, Netherlands. inria-00525258

HAL Id: inria-00525258
https://inria.hal.science/inria-00525258

Submitted on 11 Oct 2010

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A Comprehensive Framework for MultiModal Meaning Representation

Ashwani Kumar and **Laurent Romary**
Laboratoire LORIA, B.P. 239
54506 Vandoeuvre-lès-Nancy, France
{kunar, romary}@loria.fr

Abstract

Most of the existing content representation language designs are anchored to a specific modality (such as speech). This paper describes the rationale behind the definition of MMIL, the interface language intended to be used as the exchange format between modules in the MIAMM[1] project, where we try to build a juke box incorporating multiple modalities such as speech, Haptics and Graphics. This interface language has been conceived as a general format for representing multimodal content both at lower (e.g. linguistic analysis) and higher (e.g. communication within the dialogue manager) levels of representations.

1. Introduction

Very often, we use multi-dimensional perceptual and cognitive spaces, such as listening, watching other's gestures, hand movements etc, to communicate with others. As a consequence, it seems a natural trend to try to communicate with a machine in a similar manner as we interact with others. Indeed, one of the profoundest motivations behind developing multi-modal interactive systems is to overcome artificiality of man-machine interaction. Consequentially, there has been a surge in efforts towards assessing the influence of multiple modalities on naturalness and complexity in understanding and processing, of such interactions and defining suitable computational frameworks and architectures, so as to realize fully extensible and easy to use multi-modal interactive systems.

Researchers such as Nigay and Coutaz (1993), have attempted to define the term multi-modality:

Literally, *multi* refers to 'more than one' and the term *modal* may cover the notion of 'modality' as well as that of 'mode'.

- Modality refers to the type of communication channel used to convey or acquire information. It also covers the way an idea is expressed or perceived, or the manner an action is performed (Coutaz, 1992).
- Mode refers to a state that determines the way information is interpreted to extract or convey meaning.

According to Charwat (1992), *modality* is defined as follows:

"*Perception* via one of the three perception-channels. You can distinguish the three modalities: visual, auditory, and tactile (physiology of senses)."

However, human physiology provides for some more senses:

Sensory Perception	Sense Organ	Modality
Sense of sight	Eyes	Visual
Sense of hearing	Ears	Auditory
Sense of touch	Skin	Tactile
Sense of smell	Nose	Olfactory
Sense of taste	Tongue	Gustatory
Sense of balance	Organ of equilibrium	Vestibular

Table 1.1: Different senses and their corresponding modalities (taken from Silbernagel, 1979)

Though the sense of smell and taste are not very communicative in nature, sense of balance is important in applications such as virtual reality environments.

As can be seen from the above mentioned definitions, multi-modal communication can be construed as information exchange where the type and nature of information being processed is governed by the modality in use, while mode determines the context in which informational content is extracted. It is crucial to note here that multi-modal communication is different from multimedia communication, as both use multiple communication channels but multi-modal systems strive towards building a coherent meaning representation of the exchanged information. However, the information exchanged using various modalities might be imprecise, terse or ambiguous and there might be no common understanding of content across various modalities and modules within the multi-modal system. Within these constraints, a common framework for meaning representation is required so as to:

- structure information in a form that is granular, abstract and typological, which can be used in information extraction applications;
- provide for a modular Multi-Modal system design (e.g dialog and navigation systems);

[1] www.miamm.org

- enable multi-level annotation of multi-modal corpora (for a sample annotator see Milde, 2002);
- be useful for multimodal language acquisition systems (see Dusan and Flanagan, 2001).

Any such meaning representation must accommodate temporal constraints imposed on information processing through various components of the multimodal system (Coutaz, 1992) and also be able to support fusion of multi-modal inputs and the fission of multi-modal outputs at the semantic level (Bunt and Romary, 2002). Bunt and Romary 2002, present seminal discussion on multi-modal content representation, possible objectives, basic constraints and a general methodology for defining a generic representation framework. In this paper we delve more deeply into relevant issues concerning multi-modal meaning representation and evaluate our proposed representation framework against constraints identified in Bunt and Romary, 2002.

2. Information Content and Exchange

Every modality plays a substantial role in interaction with a system and the types of roles themselves are identified by the nature of modality in use. In the following we talk about types of information encoded within modality *symbols* and the issues related to their exchange across multi-modal system's architecture:

2.1 Multi-input/output

- Speech/Text:

  Linguistic utterances attempt to articulate relational mappings at the level of rule-based syntax and semantics, between lexical *symbols*, entities, intensional and extensional properties pertaining to the utterance. Inherently these expressions are *propositional* and *referential* (for detailed discussion on *propositional* and *functional* aspects of linguistic utterance, see Bunt, 2000). They also possess high expressive power when conveying temporal information, quantifications and negations. By virtue of properties of closed semantic relations and pragmatic overloading[4], linguistic expressions tend to be discrete in nature.
- Tactile:

  Tactile inputs and pressure feedback tend to be semi-discrete in nature. An input can have some particular *information content* and increasing little pressure might increase/decrease the *information content* or it may completely change the perceived meaning. For example: A typical signal could be perceived like a *wake up* signal, while increasing amplitude of the signal (and in turn pressure) might strengthen the signal or could transform its perception to be like an *alarming* signal.
- Gesture:

  Gestural inputs encode morphological properties pertaining to human physiology such as eye gaze, facial expressions, hand pointing, head and body movement. These articulations are typically continuous in nature as the expressed symbols are visual cues that have continuous mapping to the entity or the morphological property they anchor to.
- Haptics:

  Haptic interaction is closely related to visualizations and hence the graphic modality. Typical haptics interaction procedures include: texture, hardness, global shape & volume, local shape & contour etc. The modality encodes non-arbitrary relational mappings and is continuous in nature.
- Graphic:

  Image-like inputs tend to be analogue, iconic and continuous in the sense that the representation allows for continuous variations of the *iconic* properties(e.g shape, colour and texture). Unlike linguistic expressions, graphic inputs have non-arbitrary relational mappings and potentially rich spatial information.



As a first approximation, all modalities can be represented along a scale which represents a continuum along the property of arbitrariness of the mapping relation between the representation and the represented phenomenon: Graphic and linguistic inputs make two ends of the spectrum.

[4] In the utterance, *go to the mirror and fix your hair*, the hearer derives from *to the mirror* and *fix your hair*, a location in front of mirror that will enable him to see his hair. (see di Eugenio and Webber, 1996)

2.2 Management

Each modality requires unique handling and as evident from discussion in previous section each modality has different way of associating relational mappings, which in turn govern the way *meaning* is interpreted. Hence it is very important to have suitable management strategies which allow to: integrate imprecise, ambiguous and incomplete inputs; construct cohesive content and be able to fragment the content along various modality dimensions so as to allow for multi-modal generation of output. In turn management strategies constrain the design for a meaning representation by following requirements:

1. Inputs that encode same *meaning* should have identical representations
2. Ultimate objective is to have a complete meaning representation, but it should also support proper percolation of modality based symbolic units such as syntactic linguistic or morphological properties of gestures, so as to enable better interpretation at later stages, if preliminary stages have failed to extract much information from those units.
3. Representation should allow for mechanisms to determine correspondence between different input streams. For some streams, such as speech, the passage of time assigns a natural ordering to things. But in a graphic manipulation, things can happen in any order, and it is always possible to go back and revisit an earlier part of the image.
4. Even if different streams provide contradictory/ambiguous information, representation should have features to encode relevant information about these ambiguities.

Besides, management of multimodal information should not be anchored on a particular modality (such as speech). However, it is important to maintain traces of *modality* information, which at various stages of management plays crucial role in encapsulating content, which otherwise could be potential source of ambiguity. For example: a linguistic imperative, *Select* and click of a SELECT button, are same at pragmatic level, while for preliminary modules, their information contents are very different. Also, framework should not be limited by the constraints of semantics, as is the case in a W3C effort to formalise a semantics-based Markup language (http://www.w3.org/TR/nl-spec/) and in the semantics-based framework proposed by Chai et. al.,2002. Semantics-based approaches tend to be either application specific or specific modality centric, which limits usability of such a framework in general and diverse scenarios .

Apart from constraints specifically imposed by the *modality*, information exchange is also constrained by module and sub-module requirements. Within a cohesive exchange framework, structural units and their informational content in an input to a module, conform to functional requirements of the particular module and to some extent, ideally any module can be characterized by the kind of information structures it is receiving. In reference to architecture in Figure 3.1, while information at lexical level is not desired by Action Planner, at times, it is common for multimodal fusion to have lexical information percolated with the output of semantic parser. For example: in the utterance, *show me all the songs from 1991 to 1999*, lexical information of *1991* is used by Fusion module to extract quantified temporal information using temporal knowledge, while in the output to Action Planner lexical information must be filtered and replaced by the quantified information.

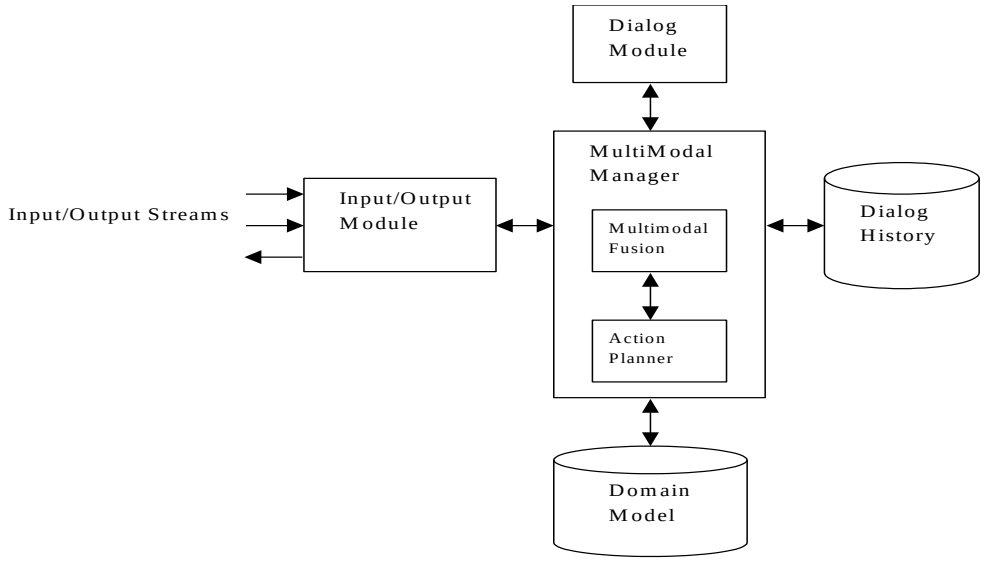


Figure 2.1: A Simplified MultiModal System Architecture

These requirements emphasize the need for a unified representation language which should: be pervasive throughout the system architecture, have proper mix of modality based *symbols* (e.g lexical, semantic

or graphic information), be generic enough to accommodate simplistic management strategies such as fusion and fission, have a flat relational representation (see Requirement 3 above) and provide for appropriate identifiers if the input stream is ambiguous or uncertain for specific modality processing (see Requirement 4 above), so that proper resolution strategies could be applied at later stages.

2.3 Integrative Architectures

Apart from accommodating efficient management of information streams, the representation language must also allow for seamless information exchange. The language should not only carry information about input and output streams, but should also facilitate cross-talk between interacting modules. The most basic requirement is of modularity so as to accommodate various component levels of a multi-modal system and at the same time should not be too constrained and application specific. By modularity, we mean that there should be clear characterization of levels and sub-levels of representation, which are common across all modalities and architecture. Also, representation must be incremental in nature so as to facilitate complete content representation as the user-system interaction advances. This type of representation framework serves as inter-lingua across various modalities and the system architecture (e.g as in Figure 3.1). However, it is important to note here that clear distinction should be made between such a representation language and the protocols adopted (e.g the SOAP[5] based approach in the MIAMM project) for integration and information exchange. By no means, the language should be expected to facilitate exchange level integration across the architecture. However, it is desired for any efficient integration scheme to build specifications, so that exchange level is layered on top of language framework.

As evident from a simplified multimodal architecture in Figure 2.1,

- domain model
- dialog history model
- dialog model

modules are some essential components of a multimodal system. The representation should provide for information levels that can be used to integrate content on incremental basis with these modules (for detailed discussion on information levels related to these models, see Kumar, 2002). Moreover, it should be flexible at the structural level so as to cater for specific requirements of these models. These requirements pertaining to specific models can vary extensively e.g A particular model might require that entries describing the constituting structure be deleted or updated regularly, while some other models are constrained to accommodate all incoming information (see Wahlster 1988). In case of the previously mentioned example, *show me all the songs from 1991 to 1999*, dialog model would be interested in knowing about *illocutionary* content, such as *dialog acts*[6] in the utterance, while *dialog history* and *Action Planner* might differ along the way temporal information about *from 1991 to 1999* is integrated into respective structure models. It is essential for *dialog history* to understand the temporal information as some *startpoint*, *endpoint* and their *temporal relation* with the pertinent context. However, it suffices for Action Planner to integrate the temporal information in form of some uniformly represented *temporal duration*. Besides, it is expected that the language should facilitate communication between Multimodal manager and Domain model. The exchanged structures should pertain to some abstract representation of information being sought and should not try to be just a database query language. Conclusively, the representation should have proper mix of granular and abstract information structures, so as to enable multilevel integration throughout a multimodal system architecture.

3. Sample Representation Language

3.1 Introduction

Above mentioned objectives in Section 3 are the underlying motivations behind the definition of Multi-modal Interface Language (or MMIL), which is the central representation format of the MIAMM[7] software architecture as it accounts for the transmission of data between the dialogue manager and both the multi-modal inputs and outputs and the application. 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. In the following sections, we describe various components of MMIL and evaluate it against earlier discussed objectives and those outlined in Bunt and Romary 2002.

[5] See e.g. http://www.w3.org/TR/soap12-part0

[6] see e.g. a summary of current practices in http://www.dfki.de/mate/d11/chap4.html

[7] Modalities used in MIAMM are Speech, Graphics and Haptics. For more information see Reithinger et. al. 2002 and www.miamm.org.

The information streams that can be potential components of the MMIL language are the following ones:

1. **Word and phoneme lattice**: This is the information uniformly provided by the continuous speech recogniser as well as by the phoneme recogniser when needed by the Name recognition task. Each element in such a lattice structure is characterized by a starting and an ending point (either expressed by means of time stamps, or by reference to explicit temporal nodes), a phonetic or lexical label, and a score.
2. **Dependency representation forest**: This information structure represents the output of any structural analysis module, whether implemented as a fully-fledged syntactic parser or as a template (or automata) based shallow parser. It comprises two types of information: first the basic linguistic units that have been recognized on the word lattice together with their morpho-syntactic descriptions and second the various dependencies that have been identified between these units, either to express predicate-argument relations, or when some of these units are modifiers of others. This information structure, even if produced by a syntactic process, already corresponds to a basic semantic representation of the spoken data. Such a structure is provided for each elementary utterance on the part of the user, and is organized as a forest to account for the various ambiguities that may be left at this stage of the recognition process.
3. **Dependency representation**: This structure is the linguistic component of the feedback provided to the user by the Dialog Manager in response to a user's query or intervention. It is very close to what is represented in the dependency representation forest, but without presenting any alternative (it is a non-ambiguous structure). In addition to linguistic descriptions, it may contain references to graphical or haptic events with which it has to be synchronized.
4. **Word/phoneme sequence**: this information structure represents the output generation module as a sequence of words, or phonemes when proper names are to be uttered, possibly annotated by prosodic cues (e.g. intonation, pauses or sub-sequence to be stressed). This sequence has to be explicit enough for a speech synthesizer to generate to operate and may keep the information needed for the speech signal to be synchronized with the graphic-haptic generation.
5. **Visual-haptic semantic representation**: this information structure is the non-verbal output of the dialogue manager in response to any multi-modal input from the user. It contains a high-level (i.e. non GUI specific) representation of the intended layout of objects to be presented to the user, the dimension to be associated to haptic keys, and the reactive links between dimensions and sub-groups of the graphical layout. This structure also provides the necessary update to the dialog manager, when the graphical layout has been modified as a consequence of haptic actions.
6. **Graphic-haptic layout**: this structure provides the basic commands to be send to both the GUI and the haptic component to obtain a haptic and visual feedback to the user, together with the necessary associations between both types of data to account for the reactive feedback of the graphical data upon haptic action.

3.2 Levels of representation – events and participants

Given the variety of levels (phone, word, phrase, utterance, dialogue) 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 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.

Events and participants are described by two types of properties:

- 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.

3.3 Design Framework

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. Since the actual XML format is potentially complex, but above all, may require 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[8] project for terminological data representation (see Romary, 2001). We thus specify the various descriptors (or *data categories*) 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, 2002, *MMIL Requirement Specification*).

3.3.1 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 multi-modal 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 will be 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 4.1, using the UML formalism. Furthermore, the representation shows an additional level for the representation of the temporal information associated with events.

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

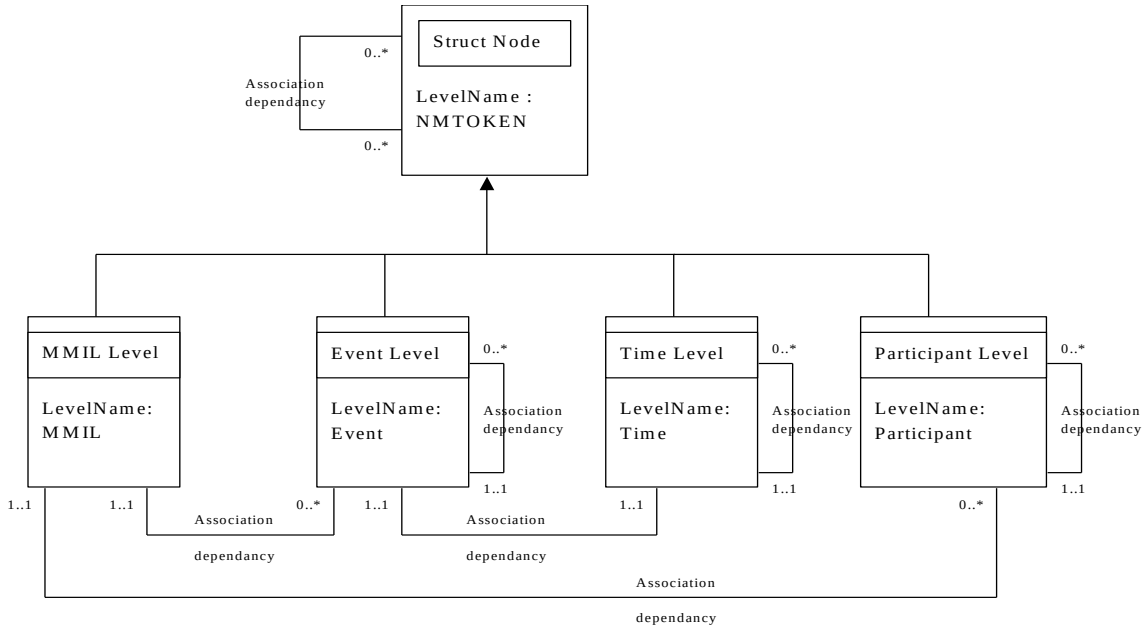


Figure 4.1: UML diagram representing the MMIL information meta-model.

3.3.2 MMIL XML overall structure

The overall MMIL XML structure is organized along the following elements:

<mmilComponent>, which implements the MMIL level

Content model (data categories excluded) : (event | participant)*

Attributes for this element are:

- id: uniquely identifies the MMIL sample.

<event>, which implements the Event level

Content model (data categories excluded) : (event*, tempSpan)

Attributes for this element are:

- id: uniquely identifies the event within the current MMIL structure

<participant>, which implements the Participant level

Content model (data categories excluded) : (participant *)

Attributes for this element are:

- id: uniquely identifies the participant within the current MMIL structure

<tempSpan>, which implements the time level

Content model (data categories excluded) : void

Attributes for this element are:

- id: uniquely identifies the time information within the current MMIL structure

Note: These unique Ids are intended to be used within the structure only. The MMIL identification data category is there to identify events and participants globally within the MMIAM architecture.

3.3.3 Position and duration of events

As shown on the UML diagram figure 4.1, any event may contain a specific level (temporal information) describing its temporal position and/or duration in time. This information is numerical and is complementary to possible relations that may hold between events through specific data categories.

This level is implemented in MMIL using the <tempSpan> element, which may contain the data categories: /Starting point/, /Ending Point/ and /Duration/. (See Romary 2002, *MMIL Technical Specification* for the description of these data categories.)

Examples:

```
<tempSpan startPoint="2002-05-31T13:20:00" endPoint="2002-05-31T13:20:01.1"/>,
```

indicates that the corresponding event occured (or is to occur) between the indicated dates (here, 1.1 seconds separate the starting and the ending point).

```
<tempSpan startPoint="2002-05-31T13:20:00" duration="P1.1S"/>,
```

indicates that the corresponding event began (or is to begin) at date 2002-05-31T13:20:00 and lasted (or will last) 1.1 seconds.

```
<tempSpan duration="P1.1S"/>,
```
indicates that the corresponding event lasted (or will last) 1.1 time units.

It is important to note here that some *events* can have *static* temporal position in the sense that they can just be *states* and at representation level, such a view of temporal positioning provides a uniform framework to incorporate events, states and meta-actions (e.g cross-talk between sub-modules).

3.3.4 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 *participants* :
  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.
- Relational Data Categories (to express dependencies)
  - Relations between *events* and *participants* :
    relation mappings between *events* and *participants*, using the knowledge available at certain stage of processing. For a complete meaning representation, these relational mappings must reflect the pragmatic relations between modality *symbols* and entities (events or participants) they anchor to.
  - Relations between *events* :
    propositional aspects and temporal relations among *events*.
  - Relations between *participants* :
    /similarity/ relationships.

One of the main trade-offs in designing a data category specification for a project like MIAMM is to account for both the generality of some basic categories, which one would not want to redefine from one application to another, and the fact that it is impossible to avoid application specific categories, which are, in particular, strongly related to the underlying domain model. The issue here is to determine whether it is possible to widely agree upon a set of reference data categories from which a project like MIAMM would just have to pick up those that are relevant for its context and complement them with some idiosyncratic ones. Such general categories could be part of a so-called data category registry, which could be made widely accessible, and which would be maintained at an international level. Comparing our own constraints in the MIAMM project which the work acheived in Verbmobil (Wahlster, 2000) and more recently in Smartkom, with the M3L language, it appears that an important "core" set of such categories could be identified. Still, it is only by working together with similar other initiatives that an agreement can be reached on precise definitions, conditions of use, etc.

3.3.5 Exemplary Illustration

According to the specifications, the representation of semantic content (output of a semantic Parser) of a simplistic utterance: "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"
</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). Also, the representation expresses a speech event(`e0`), the speaker(`user`), the addressee(`system`) and that event `e1` is prepositional content of `e0`. It can be interesting to identify how a specific difference is made between mode, as expressed by the predicative event (`e1`), and the communicative act, conveyed by the speech event proper.
The representation allows argument under-specification: even if participants "`p0`" and "`User`" are not fully identified at semantic level, information about their reference status and type is made available so that following modules can adopt relevant resolution strategies. After resolving the meaning of *the song* (using the under-specified structure of "`p0`") and "`User`", Fusion Module outputs:

```
<mmilComponent >
      <event id="e0">
            <evtType>speak</evtType>
            <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>
            <MMILId>tune10</MMILId>
            <objType>tune</objType>
            <refStatus>resolved</refStatus>
      </participant>
      <participant id="User">
            <MMILId>pers007</MMILId>
            <objType>User</objType>
            <refStatus>resolved</refStatus>
      </participant>
      <relation
            type="propContent"
            source="e1"
            target="e0"/>
      <relation
            type="subject"
            source="System"
            target="e1"/>
      <relation
            type="object"
            source="p0"
            target="e1"
</mmilComponent>
```

In this sequential fashion, information can be exchanged seamlessly, without adding redundancy or losing some information from one part of architecture to another. Within a specific module, the representation is made richer

in content, which can potentially mean filtering certain data category values e.g as fusion module identifies *the song* as *tune10*, /refType/ is not significant anymore and is not provided in following representations.

4. Typical usages of MMIL structures

4.1 Two views on temporal structures

Allowing incrementality in content representation structures throughout varions stages of processing can at times lead to contradictory choices. For instance, the analysis of the temporal adjunct in an utterance such as*show me all the tunes from 1991 to 1999* can be both viewed from a linguistic point of view where the approximate beginning and end of an interval are being specified, or from a processing point of view, when the dialogue manager for instance must take a precise decision as to which tunes he should retain and actually present back to the user. Still, it should be possible to retain the fact that both views correspond to the same underlying concept and thus avoid duplicating information (in dialogue history for instance).

The first view is dealt with by assuming that temporal expressions are not assigned to numerical values at parsing stage and are thus systematically represented by means of elementary events and relations between them. For instance, *from 1991 to 1999* will be represented as a temporal circumstance for the main event associated to the propositional content of the utterance, represented by means of three events: one for the period itself, one for the starting event (*1991*), and one for the ending event (*1999*).

The following (partial) MMIL structure implements such a representation by, on the one hand, using the recursivity of the event level to decompose the main event (e3) corresponding to the temporal circumstance, and, on the other hand, by relating more precisely the three events with one another through the two "`startsWith`" and "`endsWith`" relations[2].

```
<mmilComponent>
	<event id ="e3">
		<evtType>TempPeriod</evtType>
		<event id="e4">
			<evtType>year</evtType>
			<lex>1991</lex>
		</event>
		<event id="e5">
		<evtType>year</evtType>
			<lex>1999</lex>
		</event>
	</event>
	<relation
		type="startsWith"
		source="e3"
		target="e4"/>
	<relation
		type="endsWith"
		source="e3"
		target="e5"/>
</mmilComponent>
```

At a later stage of processing, when temporal references have been fully computed (usually by means of instantiating prototypical values, or by considering more elaborate constraint solving algorithm), it is expected that a decision has been taken concerning the actual numerical span associated with the temporal circumstance. It is then possible to anchor events with precise time level information, which an be represented in a MMIL structure as follows:

```
<mmilComponent>
	<event id="e3">
		<evtType>TempPeriod</evtType>
		<tempSpan
			startPoint="1991-01-01T00:00:00"
			endPoint="1999-12-31T24:59:59"/>
	</event>

</mmilComponent>
```

As can be seen, the two levels of representation are not contradictory and can even be kept, if necessary, in one single full structure.

4.2 Notification of visual structure

One of the simplest applications of MMIL is to represent notifications from visual and haptic levels (VisHapTac in MIAMM) to the dialogue manager. In most cases, such a notification should basically contain the objects that

[2] See the discussion on data categories above.

are visually accessible by the user, combined with focussing information, for instance resulting from specific haptic input. As always in MMIL, such a notification is considered as an event (internal to the system architecture), and makes full use of the possible hierarchical decomposition of participants to express embedded sets. In the following example, the assumption is made that an initial set of object has been presented to the user ("set1"), from which a subset ("sub_set1") has been put into focus, and more specifically a selection on a tune has been made ("s6"):

```
<mmilcomponent>
       <event id="e0">
              <evtType>HGState</evtType>
              <visMode>galaxy</visMode>
       </event>

       <participant id="set1">
              <individuation>set</individuation>
              <cardinality>25</cardinality>
              <objectType>Tune</objectType>
              <participant id="s1">
                     <Name>song1</Name>
              </participant>
              <participant id=" s2">
                     <Name> song2</Name>
              </participant>
              <participant id=" s3">
                     <Name> song3</Name>
              </participant>
              <participant id="sub_set1">
                     <individuation>set</individuation>
                     <cardinality>3</cardinality>
                     <attentionstatus>inFocus</attentionstatus>
                     <participant id=" s4">
                            <Name>song4</Name>
                            </participant>
                     <participant id=" s5">
                            <Name>song5</Name>
                     </participant>
                     <participant id=" s6">
                            <MMILId>tune55</MMILId>
                            <attentionStatus>inSelection</attentionStatus>
                            <Name> song6</Name>
                     </participant>
              </participant>
          <participant id=" s7">
              <Name> song7</Name>
          </participant>
          <participant id=" s8">
              <Name> song8</Name>
          </participant>
          …….
          <participant id=" s25">
              <Name> song25</Name>
          </participant>
       </participant>
       <relation
              type="description"
              source="set1"
              target="e0"/>
<mmilcomponent>
```

5. Evaluation

In this section, we take up the set of requirements that have been stated by (Bunt and Romary, 2002) as an evaluation grid for MMIL. Doing so, we try to identify further work to be done on the language definition and its actual usefulness for a wider variety of applications.

5.1 Expressiveness

Our fundamental premise of having two levels or representations fits well with the meaning analysis done in section 2: where we differentiate between *propositional* and *functional* aspects of interactions expressed using various modalities. It is very difficult to adopt a uniform representation which can formulate multi-modal *symbols* along these properties. Hence it is imperative to have abstract levels which most optimally encode both these properties and along all the modalities. By no means, having such a level-based representation limits

expressiveness as: events describe any temporal action and situation (including *states*), participants describe any entities related with the events, and dependencies provide compact, computationally efficient and expressive role mappings. The *mode* attribute of *information exchange* supports this by indicating whether the utterance was input by speech, visual-haptics etc. The *timestamp* attributes (which are not explicitly represented in our samples, but are associated with any MMIL sample) of *information exchange* also provide for temporal coordination by indicating when exchange occurred. Further, events and participants can be sub-levelled which ensures that the representation does not compromise on resolution of granularity. For example: Sample representation of Visible set, illustrates how a partitioned set can be represented.

5.2 Semantic adequacy/richness

Representation structures defined within MMIL framework are motivated by first order predicate logic (FOL) and hence provide for formal deductive and inference mechanisms. Strictly speaking these structures do not conform to typical FOL semantics, but one should see a MMIL structure as an existential constructs on events and participants (à la DRT), restrictions and dependencies corresponding to predicates on the corresponding variables. For instance, the representation suggested for the sentence "Play the song" could be rephrased in FOL as:

∃ e0, ∃ e1, ∃ p0, ∃ User, ∃ System,
evtType (e0, speak) ∧
dialogueAct (e0, request) ∧
speaker (e0,User) ∧
addressee (e0, System) ∧
evtType (e1, play) ∧
mode (e1,imperative) ∧
tense (e1,present) ∧
individuation (p0, singular) ∧
objType (p0, tune) ∧
refType (p0,definite) ∧
refStatus (p0,pending) ∧
propContent (e0,e1) ∧
subject (e1,System) ∧
object (e1,p0).

Still, such a representation does not provide a real "semantics" for the various predicates that it comprises. The clue to this, unless one would want to describe a complete set of axioms for these, is to define a reference description for predicates as data categories that would both uniquely identifies them and provide explicit definitions and application notes for them. To do so, we have used the framework provided by ISO 11179 for data element description, which provides a set of reference attribute to be used to specify so-called data elements (corresponding to our data categories).

For instance, the /individuation/ category is described in the MMIL specification as follows:

/Individuation/

Def: Indicates whether the current event or participant is an individual entity or should be seen as a set grouping together several entities.

Values:

- /Singular/, indicates that the entity is an individual object (implemented as *singular*)
- /Set/, indicates that the entity is an unordered set of entities (implemented as *set*)
- /Sequence/, indicates that the entity should be seen as an ordered sequence of entities (implemented as *sequence*)

Note: when the type of an entity described as a set is given, it applies to all the entities it is composed of.

Impl: <individuation>

Still, such a reference description is only valid within the project that has access to the specification. It can only have a stable status, if it were to be subsetted from a wider registry of reference data categories for semantic content representation. Such a direction is probably one to follow for our community.

5.3 Incrementality

By allowing, flexible and incremental representational structures of *information units*, MMIL ensures that multi-stage processing in multi-modal systems is feasible. Also, modularity in the representation allows that there is no restriction as on which stage of processing, fusion or fission takes place. Each module tries to interpret the content of the representation as per its desired processing objectives and its own available

knowledge constraints. If the module is able to resolve some ambiguity, corresponding ambiguity information from the representation is filtered out and resolved information is fused into the representation (see section 4.3.4).

5.4 Uniformity

Structures used within MMIL are pervasive across the complete MIAMM architecture and at no stage any specific structure is tailored for a particular module. This is ensured by maintaining subtly identified building blocks for events and participants and following a uniform strategy to filter out (or incremental addition) some information. For example:
A typical sequence of user's utterance in MIAMM scenario can be:

1) *Play the first song*.
2) *Download it*.

Output from Resolution module to Action Planner looks like:
1) *Play the first song*

```
<mmilComponent>
	……
	<event id="e1">
		<evtType>play</evtType>
		<mode>imperative</mode>
		<tense>Present</tense>
	</event>
	<participant id="p0">
		<individuation>singular</individuation>
		<MMILId>tune10</MMILId>
		<objType>tune</objType>
		<refStatus>resolved</refStatus>
	</participant>
		……
</mmilComponent>
```

2) *Download it*

```
<mmilComponent >
	…….
	<event id="e1">
		<evtType>download</evtType>
		<mode>imperative</mode>
		<tense>Present</tense>
	</event>
	<participant id="p0">
		<individuation>singular</individuation>
		<MMILId>tune10</MMILId>
		<objType>tune</objType>
		<refStatus>resolved</refStatus>
	</participant>
	….
</mmilComponent>
```

Even if, both the utterances have differing syntactic and semantic content, representation provided to Action Planner is uniform.
Also, our framework does not assume a default interaction scenario e.g collaborative discourse model. Repair based approaches such as outlined in Healey and Thirlwell (2002), can also be uniformly represented in MMIL format.

5.5 Under-specification and Partiality

Incomplete inputs, unresolved ambiguities and uncertainties are accounted by having under-specified structures, which provide information about the type of ambiguity and its status. For example: In the utterance "show me that", participant "p0" lacks exact information about /objType/, which is left to the Multimodal Fusion component to be understood.

```
<mmilComponent >
	….
	<event id="e1">
		<evtType>show</evtType>
		<mode>imperative</mode>
		<tense>Present</tense>
	</event>
	<participant id="p0">
		<individuation>singular</individuation>
		<refType>demonstrative</refType>
		<refStatus>pending</refStatus>
	</participant>
```

```
            …
      <relation
            type="object"
            source="p0"
            target="e1”
      <relation
            type="destination"
            source="p0"
            target="User”
</mmilComponent>
```

In a similar way, predicate ellipsis (e.g. *Yes, this tune.*) are dealt with by systematically introducing a void event that will be further specified by the Dialogue manager.

5.6 Openness

Conceptual and design framework used for MMIL is not bound to a particular theory of meaning or content. While characterising levels of representation and data categories, we have tried to conform to the ideas thought about in early sections of the paper and other pre-existing semantic theories. MMIL has been kept independent from any specific application framework, so that it can cope for instance with the various parsing technologies adopted for different languages (template based vs. TAG based parsing). This in turn provides, MMIL, some degree of genericity, which could make it reusable and extensible in other contexts. Besides, MMIL design is not normative to the extent of specifying data categories. As specified in ISO 16642, it is possible, when needed, to refine a given data category by means of additional descriptors. For example, if we consider that a similarity relation is expressed by a /similar/ relation between two participants as follows:

```
      <relation
            type="similar”
            source="id1”
            target="id2”/>
```

It is possible to express more precisely the set of dimensions along which the similarity search is to be made, as follows:

```
      <relationGrp>
            <relation
                  type="similar”
                  source="id1”
                  target="id2”/>
            <dimension>genre</dimension>
            <dimension>author</dimension>
      <relationGrp>
```

5.7 Extensibility

MMIL is highly extensible as it is eventually expressed as an XML schema format and hence inherits all the advantages of using XML, such as :

- An XML declaration, which, beyond identifying that the current document is an XML one, allows one to declare the character encoding scheme used in the document (e.g. iso-8859-1, utf-8, etc.);
- XML is both Unicode and ISO 10646 compatible, which means that, in the context of MIAMM, there is no limitation in the use of writing systems and languages in the content of the information exchanged within the dialogue system or with the external multimedia database;
- XML comes along with a specific mechanism, called *namespaces*, allowing one to combine, within the same document, mark-up taken from multiple sources. This very powerful mechanism, which is in particular the basis for XSLT and XML schemas, allows more modularity in the definition of an XML structure and also to reuse components defined in other context;
- XML provides a general attribute, ‘xml:lang’ to indicate the language used in a given element.

6. Conclusion

As an undergoing work, the MMIL specification are not, at the time of this paper, fully stabilized. Still, the first experiments that we have conducted at various levels in the MIAMM architecture show that it can cover a wide range of the phenomena needed to be dealt with in a multimodal dialogue system. It remains to be shown whether MMIL, or any dialect based on similar concepts, can actually be used in other dialogue environment, and beyond in other types of applications such as information extraction for instance. As a whole, our experience shows that it is high time for our community to define some reference guidelines, and probably even more (one would think of a true reference registry of data categories), for multimodal semantic content representation.

7. References

Bunt, H.C., 2000. *Dialogue pragmatics and context specification*. In H. Bunt and W. Black, editors, Abduction, Belief and Context in Dialogue. John Benjamins Publishing Company, Amsterdam.

Bunt H., L. Romary 2002. *Towards Multimodal Content Representation*. In 'International Standards of Terminology and Language Resources Management', LREC 2002, Las Palmas (Spain).

Chai J., S. Pan and M. X. Zhou 2002. *MIND: semantics based Multimodal interpretation framework*, In Proceedings of the International CLASS Workshop on Natural, Intelligent and Effective Interaction in Multimodal Dialogue Systems, 28-29 June, Copenhagen, Denmark, 2002.

Charwat H. J. 1992. *Lexikon der Mensch-Maschine-Kommunikation*. Oldenbourg.

Coutaz J. 1992. *Multimedia and Multimodal User Interfaces: A Taxonomy for Software Engineering Research Issues*. In Proc. Second East-West HCI conference, pp.229-240 St Petersburg.

Dusan S. and J.L. Flanagan 2001. *Human Language Acquisition by Computers*, in Proceedings of the International Conference on Robotics, Distance Laerning and Intelligent Communication Systems, WSES/IEEE, Malta, pp 387-392

Eugenio, B. D., B. L. Webber 1996.*Pragmatic overloading in Natural Language instructions* International Journal of Expert Systems, Special Issue on Knowledge Representation and Reasoning for Natural Language Processing

Healey P.G.T. and M. Thirlw ell (2002), *Analysing Multi-modal Communication: Repair-based Measures of communicative Co-ordination*, In Proceedings of the International CLASS Workshop on Natural, Intelligent and Effective Interaction in Multimodal Dialogue Systems, 28-29 June, Copenhagen, Denmark, 2002.

Nigay L., J. Coutaz 1993. *A Design Space For Multimodal Systems:Concurrent Processing and Data Fusion*. In INTERCHI'93 Proceedings, pages 172--178, Amsterdam, the Netherlands.

Kumar, A. 2002. *Dialog Module Technical Specification*. Project MIAMM – *Multidimensional Information Access using Multiple Modalities.* EU project IST-20000-29487, Deliverable D5.1. LORIA, Nancy

Milde J.-T. 2002, *Creating multimodal, multilevel corpora with TASX*. In Proceedings of the International CLASS Workshop on Natural, Intelligent and Effective Interaction in Multimodal Dialogue Systems, 28-29 June, Copenhagen, Denmark, 2002.

Romary, L. 2001. *Towards an Abstract Representation of Terminological Data Collections - the TMF model*, TAMA 2001 – Terminology in Advanced Microcomputer Applications, Antwerp (Belgium), 1-2 February 2001.

Romary,L.,2002.*http://www.dfki.de/~wahlster/Dagstuhl_Multi_Modality/WG_4_Multimodal_Meaning_Representation/ppframe.htm*

Romary, L., 2002. *MMIL requirements specification*. Project MIAMM – *Multidimensional Information Access using Multiple Modalities.* EU project IST-20000-29487, Deliverable D6.1. LORIA, Nancy

Romary, L., 2002. *MMIL technical specification*. Project MIAMM – *Multidimensional Information Access using Multiple Modalities.* EU project IST-20000-29487, Deliverable D6.3. LORIA, Nancy

Silbernagel, D.. *Taschenatlas der Physiologie*. Thieme, 1979.

Reithinger N.,Lauer C. Romary L, *MIAMM: Multidimensional Information Access using multiple modalities*, In Proceedings of the International CLASS Workshop on Natural, Intelligent and Effective Interaction in Multimodal Dialogue Systems, 28-29 June, Copenhagen, Denmark, 2002.

Wahlster, W. 1988. *distinguising user models from discourse models*, Computational lingusitics, Volume 14, Number 3.

Wahlster, W. (Ed.), 2000, Verbmobil: Foundations of Speech-to-Speech Translation, Springer-Verlag.