What are Multimodalities made of ? Modeling Output in a Multimodal Dialogue System

Christian Elting
European Media Laboratory GmbH
Villa Bosch
Schloss-Wolfsbrunnenweg 33
D 69118 Heidelberg, Germany
christian.elting@eml.villa-bosch.de

Abstract

In order to produce coherent multimodal output a presentation planner in a multimodal dialogue system must have a notion of the types of the multimodalities, which are currently present in the system. More specifically the planner needs information about the multimodal properties and rendering capabilities of the multimodalities. Therefore it is necessary to define an output multimodality model that can properly describe the available renderers in sufficient detail and on the other hand keep a level of abstraction that enables the presentation planner to support a large set of different renderer types. In this paper we present our approach for such a multimodality model.

Keywords

multimodal, modality models, modality theory, presentation planning

Introduction

After the beginning of the information age the average user was suddenly able to access vast amounts of data and choose between a lot of different media like music or video. People all around the world came a step closer together connected by the internet as the new means of communication. But this age did not only have its bright sides. Following Moore's Law the new technologies also exploded on us and many users, who were not experienced with the new devices and technologies got left behind. Therefore it becomes increasingly important these days to make not only a step forward on the information highway but rather a step upwards in order to give everyday users a better view and control over the powerful technologies that are at their hands.

The German project EMBASSI (Herfet, Kirste, & Schnaider 2001) is a multimodal dialogue assistance which aims at connecting and simplifying everyday technologies. EMBASSI focuses on a car scenario, a home scenario and an ATM scenario. In this paper we illustrate our model for output multimodalities that models the multimodal renderers of the EMBASSI system. This is necessary in order to generate a coherent multimodal presentation that fits the current situative context and the current content. The system includes

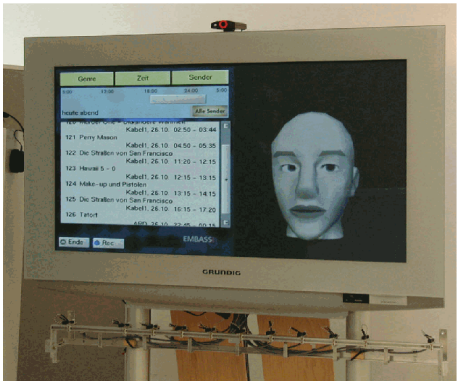

Figure 1: The first EMBASSI demonstrator.

haptic, graphical and acoustical output including several TV GUIs (fig.1), an avatar[1] as well as several PDA GUIs.

Definitions

We follow the definitions in (Bernsen 2001). The term *medium* denotes the "physical realisation of information at the interface between human and system". Examples for media are acoustics, graphics and haptics. With the term *modality* we refer to a "mode or way of exchanging information between humans and machines in some medium". We use the term *unimodality* for modalities that consist only of one basic modality as given in the taxonomy of unimodal output modalities in (Bernsen 2001). Examples for output unimodalities are a speech synthesis or a scroll-text. Moreover we use the term *multimodality* for modalities that are composed of one or more unimodalities. An example for an output multimodality is an avatar that consists of mimics, gestures and speech.

Motivations

It is crucial for a multimodal dialogue system to provide a coherent and cohesive multimodal output. In the EMBASSI system the presentation planner has to come up with a choice of multimodalities that serves the current situative context,

[1] We use the Term *avatar* for an animated anthropomorphic assistant.

the content to be displayed and the preferences of the current user. In order to do so the presentation planner has to be able to assess the rendering capabilities of a multimodality. The rendering capabilities are especially connected to the physical resource limitations of the output device (e.g. screen resolution, volume). By means of a proper description of the output multimodalities being present in the system the presentation planner can judge which output multimodalities fit best the current situation and the current content. Moreover even previously unknown output multimodalities that can be properly described by the model can be seamlessly integrated into the presentation.

Requirements for modeling output multimodalities

In (Bernsen 2001) the following requirements for modeling output unimodalities were identified that also hold for modeling output multimodalities. The first requirement is the *completeness* of the model meaning that every output multimodality possible can be described by the model. The second requirement concerns the *uniqueness* of the model which says that every output multimodality can only be decribed in one way by the model. Moreover the model has to fulfill the *relevance* criterium that the model only captures the most important differences between output multimodalities. Finally the *intuitiveness* requirement says that the multimodality model should be easy to operate with and correspond to the notions of the interface developers.

However in order to satisfy the *completeness* requirement the model would have to cover all n-modalities with $n > 0$. Then again the model would most likely not fulfill the intuitiveness and relevance criteria. Therefore we modified the completeness criterium to *sufficient completeness*. This requirement means that not every output multimodality has to be modeled but only those types of output multimodalities that are relevant to the domain.

Output Unimodality Model

A multimodality basically consists of a set of unimodalities that are connected by special relations like sychronisation or coordination needs. Therefore a multimodality model has to be based on a proper unimodality model. In (Bernsen 2001) we find an elaborate taxonomy of output modalities for the media acoustics, graphics and haptics. Bernsen identified five basic features of output unimodalities: linguistic (li) resp. non-linguistic (-li), analogue (an) resp. non-analogue (-an), arbitrary (ar) resp. non-arbitrary (-ar) and static (sta) resp. dynamic (dyn). He then produced all possible combinations of these features and conducted a pragmatic reduction of the set according to the relevance of the unimodality classes to interface design. The result of this procedure is a set of 20 unimodality classes (*generic level*) that are grouped into four super classes (*super level*). Bernsen further expands the unimodality hierarchy into the *atomic level* which produces 46 unimodality classes. Finally Bernsen sketches a possible further extension into the *subatomic level* which might be conducted when necessary depending on the prevailing interface domain. We use this unimodality taxonomy as the basis of our multimodal taxonomy.

Output Multimodality Model

The output multimodality model is used to describe the properties and rendering capabilities of output multimodalities in the EMBASSI system. The model consists of four distinct parts. The *set of unimodalities* containes the single unimodalities of the multimodality. The *multimodal relations* describe the multimodal relations between the single unimodalities of the renderer. The *multimodal type* serves to precise the type of the multimodality in question. The *assignment to physical output devices* is necessary to model the physical resource restrictions.

Set of Unimodalities

The set of unimodalities contains all the unimodalities that the multimodality is composed of. The unimodalities are modeled as given in the unimodality taxonomy in (Bernsen 2001). A speech synthesis unimodality for instance is modeled as a (li, -an, -ar, dyn, aco; freetext) renderer. Here we use the atomic property "freetext" to distinguish between renderers that render arbitrary content and renderers that only render partial sets of the content (e.g. only object lists). In this sense the property "freetext" resembles Bernsen's "spoken discourse" property. A simple avatar composed of a head and a hand for pointing gestures can be modeled by the unimodalities gesture (li, an, -ar, dyn, gra; gesture), mimic (li, an, -ar, dyn, gra; mimic) and speech (li, -an, -ar, dyn, aco; freetext). Bernsen includes only gestures in the category (li, an, -ar, dyn, gra). However we argue that also mimic can transport simple linguistic information (emotions or emphasis) and therefore should be included in this category.

Multimodal Relations

A multimodality usually does not only consist of single unimodalities that all render output independant of each other. Usually these unimodalities have to synchronize or coordinate their output. For instance the lip movements of the avatar have to be synchronized with the speech of the avatar. Therefore it is necessary for a multimodality model to take care of these multimodal relations, which have to be defined for each unimodality of a multimodality.

We can identify several multimodal relations that exist between the unimodalities of a multimodality. The most important ones are the necessary *synchronizations* between a unimodality of the multimodality and other unimodalities. Synchronization means coordination of the output of (dynamic) unimodalities in time. Apart from the synchronizations it is also necessary to characterise the multimodal referring expressions that can be generated by a multimodality. In (André & Rist 1994) the following types of referring expressions are identified. *Multimodal referring expressions* are referring to world objects. Imagine for instance an avatar pointing to a button of a remote control and saying "you have to push the arrow buttons in order to switch the program". In this case the avatar's gesture renderer, the avatar's speech renderer and the GUI's picture renderer would all make a reference to the same world object "arrow button". *Anaphoric referring expressions* also refer to world objects, but in an abbreviated form. They are implicitly or explicitly

introduced during the discourse, so that the user can resolve the abbreviation. An example for an anaphoric referring expression is an avatar saying "You have to push this in order to switch to the next program" while the GUI displays a remote control with a highlightened button. The third type of referring expressions are *crossmodal referring expressions*. Here not world objects are referenced, but another part of the multimodal presention (e.g. a picture as in the linguistic reference "As can be seen in the picture…").

Concerning multimodalities the presentation planner has to know what kinds of references to world objects (by means of multimodal/anaphoric referring expressions) or to presentation objects (by means of crossmodal referring expressions) the multimodal renderer can generate and with which unimodalities this has to be coordinated. In order to model the generation of multimodal (possibly anaphoric) references to world objects we introduce the predicate *coordWORefWith (<multimodality>.<unimodality>)* . Here *<multimodality>* stands for an abtract multimodal type (which will be defined in detail in the next section and *<unimodality>* denotes a unimodality of this multimodality. In case of a freetext renderer within a GUI the possible multimodal references to world objects can be described by the following expression:

$$\begin{aligned} & coordWORefWith(GUI.image) \\ & \text{OR } \big[(coordWORefWith(Avatar.speech) \\ & \text{AND } coordWORefWith(Avatar.gesture)\big) \\ & \text{XOR } coordWORefWith(Speech.speech)\big] \end{aligned} \quad (1)$$

Here we use the AND-, OR- an XOR-relations to distinguish between synchronizations and coordinations that can take place independant of each other (OR-relation), that have to be conducted simultaneously (AND-relation) and that exclude each other (XOR-relation). The multimodal references to world objects by the GUI freetext renderer can therefore be coordinated with a GUI image renderer (e.g. by referring to an object within the image) possibly together with either an avatar reference (by speech and gestures) or by speech of the speech synthesis alone. The possible crossmodal references to objects within the presentation are described by means of the predicate $genCMRefTo$ by the following expression:

$$\begin{aligned} & genCMRefTo(GUI.image) \\ & \text{OR } genCMRefTo(GUI.freetext) \end{aligned} \quad (2)$$

In this example the GUI freetext renderer can refer to presentation objects generated by the GUI image renderer ("As you can see in the picture…") or by another GUI freetext renderer ("As described in the last section…").

In case of a dynamic renderer it is also necessary to characterize the unimodalities with which the output will possibly have to be synchronized. The following expression describes the necessary synchronizations of a dynamic GUI list-text renderer[2]:

$$\begin{aligned} & syncWith(Avatar.speech) \\ & \text{OR } \big[syncWith(Avatar.gesture) \\ & \text{XOR } syncWith(Speech.speech)\big] \end{aligned} \quad (3)$$

The dynamic display of a list has therefore to be synchronized with avatar gestures and if necessary with either avatar speech or the speech synthesis renderer.

Multimodal Type

With the current multimodality model it is not yet possible to properly model multimodalities and the different information flows that multimodal render agents are connected with. For instance in order to adapt the multimodal output according to the user's preferences the presentation planner must have a notion what kind of multimodality an "avatar" actually is in order to be able to set the output preferences accordingly. The multimodal type is also necessary in order to apply psychological presentation knowledge (e.g. on when and how to use an avatar multimodality). Furthermore a multimodal type is also necessary to model the information flow that each renderer needs to follow. For instance a multimodality of GUI type might need an information flow of the following form. First an abstract layout is given to the GUI that is to be rendered by the GUI. Afterwards the GUI returns a detailed layout in form of screen coordinates. Then the GUI is ready to receive an amodal setup to be rendered. After that the GUI awaits a proper execution command which triggers the GUI's presentation. An avatar or a speech synthesis might need a completely different information flow. Therefore it is necessary for a multimodality model to identify the multimodal that is given in our case by a set of unimodalities, the necessary synchronizations and the possible references to be generated by the unimodalities. Also note that multimodal types make it easier to formalize the necessary synchronizations and coordinations of a unimodality. This is why we already made use of multimodal types in the "multimodal relations" section.

Our approach here is to assign to each multimodal type a set of instances of the multimodality model. Each instance in turn is a tuple that consists of a set of unimodalities, the synchronizations, the necessary coordinations for multimodal references to world objects and the possible crossmodal references of the unimodalities.

In the following we provide an example of how two instances of the "GUI" multimodality type look like.

$$\begin{aligned} & \text{GUI } := \{ \\ & \big[(< list >, < sync_1 >, < refWO_1 >, < refCM_1 >), \\ & (< text >, < sync_2 >, < refWO_2 >, < refCM_2 >)\big], \\ & \big[(< list >, < sync_3 >, < refWO_3 >, < refCM_3 >), \\ & (< img >, < sync_4 >, < refWO_4 >, < refCM_4 >)\big], \\ & \ldots \} \end{aligned}$$

The term $< list >$ stands for an instance of the unimodality model namely a (li, -an, -ar, sta, gra; list) list-text renderer. $< text >$ stands for a (li, -an, -ar, sta, gra; freetext) freetext renderer and $< img >$ for a (-li, an, -ar, sta, gra) image renderer. The term $< sync >$ denotes the corresponding synchronization expression, $< refWO >$ denotes the expression for references to world objects and $< refCM >$ the expression for crossmodal references. The example

[2]To be precise a list itself is already a multimodality. We see it here as a renderer of graphic text that can only render limited content that has a list syntax.

shows two possible GUI instances namely a GUI consisting of a list-text renderer and a freetext renderer as well as a GUI consisting of a list-text and an image renderer.

Of course by using multimodality types we limit the number of multimodalities that can be modeled by the presentation planner and therefore abandon the completeness requirement in favor of the "sufficient completeness" requirement. On the other hand the relevance requirement for the multimodality model demands that all the relevant features of multimodalities are to be incorporated into the model. As the type of the multimodality is in our view an important information we act according to this requirement if we incorporate it. Furthermore in order to properly realize an information flow between different types of multimodal renderers agents it is crucial to get information about the protocols used. Therefore incorporating multimodality type information into the model also serves the intuitiveness requirement. However each multimodal type can comprise an arbitrary number of multimodalities. Therefore the multimodal type achieves to provide enough detail for presentation planning on the one hand and still keep a level of abstraction on the other hand.

Assignment to Physical Output Devices

In order to adapt the multimodal output to the current situative context it is especially important to know where the physical output devices are located. When the user leaves a room the presentation planner has basically two choices. Either the planner can render the complete content acoustically in order to still reach the perception of the user in the next room or the planner can (which might be the better alternative in most of the situations) switch the multimodal user interface to devices that are located in the room the user just entered. In order to do so it is necessary that the location of the output device of each unimodality of a multimodality is known. Of the physical output device the location of the device, the medium, in which the output is rendered (graphics, acoustics, haptics) and the resource restrictions of the device (screen resolution, volume, etc.) are known.

Apart from the device location where the output is rendered also the resource restrictions of the device are important for presentation planning. Imagine for instance that a comprehensive list of movie titles should be displayed. This can be rendered by the static list-text renderer of the GUI on the 800x600 TV screen. But when the user leaves the room with a PDA with a screen resolution of 320x240 and the presentation switches to the PDA it is not not possible anymore to display the whole list statically. Therefore the GUI render agent has to inform the presentation planner that the rendering failed due to the resource restrictions. The presentation planner now has the possibility to choose a new output strategy for the same unimodal renderer (e.g. by first presenting movie genres instead of movie titles) or to switch to another renderer on the PDA (e.g. dynamic scrolling text).

Case Study

In the following we provide an example of how an avatar multimodality, a GUI multimodality and a speech synthesis multimodality can be described by the multimodality model. Afterwards we show how these information can be used during presentation planning.

Table 1: The avatar unimodalities.

ID	type
"gesture"	(li, an, -ar, dyn, gra; gesture)
"mimic"	(li, an, -ar, dyn, gra; mimic)
"speech"	(li, -an, -ar, dyn, aco; freetext)

Table 2: The GUI unimodalities.

ID	type
"list"	(-li, an, -ar, sta, gra; list)
"list_dyn"	(-li, an, -ar, dyn, gra; list)
"freetext"	(-li, an, -ar, dyn, gra; freetext)

Multimodality modeling

The avatar unimodalities are shown in table 1. The GUI unimodalities are presented in table 2 and the speech synthesis unimodality in table 3 respectively. The avatar consists of unimodal gesture and mimic (including lip movements) renderers as well as a speech renderer. The GUI consists of a static freetext renderer, a static renderer for list-text and a dynamic renderer for list-text that displays the list item by item. The multimodal speech synthesis renderer consists only of a single unimodality namely speech.

1. **Multimodal relations for Avatar.gesture**
   The following expressions show the necessary synchronizations (expression (4)), the necessary coordinations for multimodal references to world objects (expression (5)) and the possible crossmodal references (expression (6)) for the avatar gestures.

$$\begin{aligned} &[\, syncWith(Avatar.speech) \\ &\text{AND } syncWith(Avatar.mimic)\,] \\ &\text{OR } [syncWith(Avatar.speech) \\ &\text{AND } syncWith(Avatar.mimic) \\ &\text{AND } syncWith(GUI.list\_dyn)] \end{aligned} \quad (4)$$

   Expression (4) shows that the avatar gestures have to be synchronized with the avatar speech and mimic and possibly with a dynamic display of list-text.

$$\begin{aligned} &coordWORefWith(Avatar.speech) \\ &\text{OR } coordWORefWith(GUI.list) \\ &\text{OR } coordWORefWith(GUI.list\_dyn) \\ &\text{OR } coordWORefWith(GUI.freetext) \end{aligned} \quad (5)$$

Table 3: The speech synthesis unimodalities.

ID	type
speech	(li, -an, -ar, dyn, aco; freetext)

The avatar gesture renderer can take part in the generation of a multimodal reference to a world object that is generated together with the avatar speech renderer and the list-text and freetext renderers of the GUI (expression (5)). This can for instance be a multimodal reference to a TV show title that is also displayed graphically as a list-text item and presented acoustically by speech.

$$\begin{aligned} & genCMRefTo(GUI.list) \\ & \text{OR } genCMRefTo(GUI.list\_dyn) \\ & \text{OR } genCMRefTo(GUI.freetext) \end{aligned} \quad (6)$$

The avatar gesture renderer can only produce crossmodal references (i.e. pointing gestures to other presentation parts) in the graphics medium. Therefore it can only render crossmodal references to the GUI unimodalities (expression (6)).

2. **Multimodal relations for Avatar.mimic**
In expresssion (7) the necessary synchronizations for the avatar mimic (including lip movement) are presented. No references to world objects or crossmodal references can be generated by the avatar mimic in this example.

$$syncWith(Avatar.speech) \quad (7)$$

The avatar mimic (especially the lip movements) have to be synchronized with the avatar speech.

3. **Multimodal relations for Avatar.speech**
Expressions (8)-(10) show the necessary synchronizations, coordinations for the generation of references to world objects as well as the possible crossmodal references respectively for the avatar speech unimodality.

$$\begin{aligned} & \big[syncWith(Avatar.mimic) \\ & \text{AND } syncWith(Avatar.gesture) \big] \\ & \text{OR } \big[syncWith(Avatar.mimic) \\ & \text{AND } syncWith(Avatar.gesture) \\ & \text{AND } syncWith(GUI.list\_dyn) \big] \end{aligned} \quad (8)$$

As shown in expression (8) the avatar speech has to be synchronized with the mimics and the gestures of the avatar. Optionally the avatar speech can also be synchronized with the dynamical display of the GUI list-text renderer, i.e. the avatar is reading the names of the items that are displayed one by one by the list-text renderer.

$$\begin{aligned} & coordWORefWith(Avatar.gesture) \\ & \text{OR } coordWORefWith(GUI.list) \\ & \text{OR } coordWORefWith(GUI.list\_dyn) \\ & \text{OR } coordWORefWith(GUI.freetext) \end{aligned} \quad (9)$$

Together with the avatar gesture renderer and the GUI renderers the avatar speech renderer can generate multimodal references to world objects like TV programs (expression (9)).

$$\begin{aligned} & genCMRefTo(GUI.list) \\ & \text{OR } genCMRefTo(GUI.list\_dyn) \\ & \text{OR } genCMRefTo(GUI.freetext) \end{aligned} \quad (10)$$

Expression (10) shows that the avatar speech renderer can generate crossmodal references to the GUI list-text renderers ("You can see the title in the list on the TV screen.") or to the GUI freetext renderers ("The text on the left gives you the contents of the movie.").

4. **Multimodal relations for GUI.list**
In expression (11) the necessary coordinations for the generation of references to world objects for the static GUI list-text renderer are given. This static renderer does not need to conduct any synchronizations nor can crossmodal references be generated in this example.

$$\begin{aligned} & coordWORefWith(GUI.freetext) \\ & \text{OR } \big[\big(coordWORefWith(Avatar.speech) \\ & \text{AND } coordWORefWith(Avatar.gesture) \big) \\ & \text{XOR } coordWORefWith(Speech.speech) \big] \end{aligned} \quad (11)$$

The static list-text renderer can render multimodal references to world objects together with the avatar speech and gesture renderers (e.g. to TV programs as world objects). Additionally the static list-text renderer can generate references to world objects with the stand-alone speech synthesis renderer. The same is true for the GUI freetext renderer that can for instance linguistically reference TV shows.

5. **Multimodal relations for GUI.list_dyn**
Expressions (12) and (13) show the necessary synchronizations and coordinations for the generation of references to world objects for the dynamic GUI list-text renderer. The dynamic list-text renderer cannot generate any crossmodal references.

$$\begin{aligned} & \big[\big(syncWith(Avatar.speech) \\ & \text{AND } syncWith(Avatar.gesture) \big) \\ & \text{XOR } syncWith(Speech.speech) \big] \end{aligned} \quad (12)$$

As shown in expression (12) the dynamic list-text renderer has to be synchronized with avatar gestures as well as with speech renderers (expression (12)).

$$\begin{aligned} & coordWORefWith(GUI.freetext) \\ & \text{OR } \big[\big(coordWORefWith(Avatar.speech) \\ & \text{AND } coordWORefWith(Avatar.gesture) \big) \\ & \text{XOR } coordWORefWith(Speech.speech) \big] \end{aligned} \quad (13)$$

The dynamic list-text renderer can generate the same references to world objects as the static list-text renderer (c.f. expression (11)).

6. **Multimodal relations for GUI.freetext**
In expressions (14) and (15) the necessary coordinations for the generation of references to world objects and the possible crossmodal references of the static GUI freetext renderer are displayed. This static renderer does not need to conduct any synchronizations nor can crossmodal references be generated.

$$coordWORefWith(GUI.list)$$

$$\begin{aligned} &\text{OR } coordWORefWith(GUI.list\_dyn) \\ &\text{OR } \big[(coordWORefWith(Avatar.speech) \\ &\text{AND } coordWORefWith(Avatar.gesture)) \\ &\text{XOR } coordWORefWith(Speech.speech)\big] \end{aligned} \quad (14)$$

The GUI freetext renderer can take part in the generation of multimodal references to world objects together with the other GUI renderers as well as together with the avatar or the speech synthesis (expression (14)).

$$\begin{aligned} &genCMRefTo(GUI.freetext) \\ &\text{OR } genCMRefTo(GUI.list) \\ &\text{OR } genCMRefTo(GUI.list\_dyn) \end{aligned} \quad (15)$$

The GUI freetext renderer can generate crossmodal references to GUI list-text renderers ("As you can see in title list on the left…") or to another GUI freetext renderer ("As described in the text above…") (expression (15)).

7. **Multimodal relations for Speech.speech**
   Expressions (16)-(18) show the necessary synchronizations and coordinations for the generation of references to world objects as well as the possible crossmodal references for the speech synthesis unimodality of the speech synthesis multimodality.

$$syncWith(GUI.list\_dyn) \quad (16)$$

As shown in expression (16) the stand-alone speech synthesis only needs to be synchronized with the dynamic list-text renderer.

$$\begin{aligned} &coordWORefWith(GUI.list) \\ &\text{OR } coordWORefWith(GUI.list\_dyn) \\ &\text{OR } coordWORefWith(GUI.freetext) \end{aligned} \quad (17)$$

Multimodal references to world objects can be generated by the speech synthesis together with any GUI renderer (expression (17)).

$$\begin{aligned} &genCMRefTo(GUI.list) \\ &\text{OR } genCMRefTo(GUI.list\_dyn) \\ &\text{OR } genCMRefTo(GUI.freetext) \end{aligned} \quad (18)$$

The stand-alone speech sythesis can generate linguistic crossmodal references to any GUI renderer (expression (18)).

Presentation planning

In the following we describe how the multimodality descriptions from the previous section can be exploited during presentation planning. The presentation planning is triggered by a presentation task that is sent to the presentation planner by the dialogue manager. This task consists of an output goal (e.g. "message_inform" or "message_error") and the content to be displayed. In this example the content consists of TV show information, which should be presented to the user.

First the presentation planner examines the available modalities and the corresponding multimodal types. The planner finds three multimodalities of types avatar, GUI and speech synthesis as modeled in the previous section. As the user expressed a preference for working with the system's avatar the planner favours an GUI-avatar combination over a GUI-speech combination. Multimodalities of GUI type are modeled as the default multimodalities for presenting TV show informations in the presentation knowledge of the planner. However the output device on which the avatar and the GUI render output is a small 320x240 screen. As it is most likely that the complete list of TV show titles cannot be displayed statically on this screen the PMO chooses to present the information dynamically with the GUI's dynamic list renderer $GUI.list\_dyn$.

Concerning the multimodality of type avatar the presentation knowledge of the planner indicates that it is required to use mimic together with speech. Therefore the planner chooses to use the $Avatar.mimic$ and the $Avatar.speech$ unimodalities. Moreover the presentation knowledge indicates that crossmodal references by means of pointing gestures are recommended to support the coreferences of $Avatar.speech$ and $GUI.list\_dyn$ to TV shows. Therefore the planner also uses the $Avatar.gesture$ unimodality to generate those references.

The synchronization expression of $Avatar.gesture$ (expression (4)) indicates that synchronizations have to be conducted with $Avatar.speech$ and $Avatar.mimic$. It is additionally possible to synchronize with $GUI.list\_dyn$. The presentation knowledge of the planner indicates that it is recommendable to synchronize dynamic GUI output together with avatar unimodalities if both generate references to the same objects (in this case TV show titles). Therefore the planner initiates synchronizations with $Avatar.speech$, $Avatar.mimic$ and $GUI.list\_dyn$ for the $Avatar.gesture$ unimodality. This is done by setting proper flags in the setup message for the avatar renderer. Corresponding synchronizations with $Avatar.gesture$, $Avatar.mimic$ and $GUI.list\_dyn$ are set for $Avatar.speech$ (c.f. expression (8)). Moreover the planner initiates a synchronization with $Avatar.speech$ for $Avatar.mimics$ (c.f. expression (7)). For $GUI.list\_dyn$ synchronizations with $Avatar.speech$ and $Avatar.gesture$ are initiated (c.f. expression (12)).

After all the synchronization needs are satisfied the presentation planner examines the possible referring acts. The $Avatar.speech$ renderer can generate references to the TV shows by rendering the titles of the shows. The presentation knowledge of the planner indicates that it is recommendable to coordinate these references with any other dynamic references to the same world objects. As it is possible to coordinate world object references from $Avatar.speech$ with $GUI.list\_dyn$ the planner initiates these coordinations for $Avatar.speech$ and $GUI.list\_dyn$ respectively. For dynamic unimodalities this implies an additional synchronization concerning the time the references take place. Furthermore the planner initiates a crossmodal reference from $Avatar.gesture$ to $GUI.list\_dyn$ (c.f. expression (6)). This will lead to an avatar pointing gesture to the GUI's list and support the coreferences to the TV show titles.

Next the planner has to determine which protocols the renderers support. A protocol is a sequence of agent messages between the planner, the renderer and possibly other

agents that has to take place prior to an output of the renderer. Usually the protocol consists of a setup message for the renderer to generate the modality-specific output and a trigger from the presentation planner to display the output.

After that the rendering parameters are set for the avatar and the GUI. We distinguish two kinds of parameters. Multimodal parameters are parameters that concern the multimodality as a whole (e.g. an avatar character). Unimodal parameters on the other hand concern only a single unimodality (e.g. the text complexity of the speech that the speech unimodality of the avatar synthesizes). In this example the avatar character and the GUI look-and-feel have to be set as multimodal parameters. This is done according to corresponding entries in the user profile. Unimodal parameters include the emotion to be expressed by $Avatar.mimic$ ("happy"), the text complexity of the speech rendered by $Avatar.speech$ (set to "high" as the user is classified as a beginner) and the type of gesture to be rendered by $Avatar.gesture$ ("pointing gesture"). The types of the unimodal parameters to be set by the planner (e.g. text complexity) are inferred from the categories of the unimodalities.

Finally the planner also determines by means of the device assignments which unimodalities have to perform layout coordinations. In this case the avatar unimodalities $Avatar.gesture$ and $Avatar.mimic$ have to coordinate their layout with $GUI.list\_dyn$ as they render output on the same output device. It is also possible that the planner already proposes an abstract layout to simplify these coordinations.

The results of the presentation planning process concerning the synchronizations, references and layout coordinations to be conducted as well the unimodal and multimodal parameters are put into proper setup messages for the render agents (which comply with the specific protocols) and are sent to the render agents. Afterwards the avatar and the GUI render their output and synchronize resp. coordinate their results. After these processes are finished successfully an acknowledgement is sent back to the presentation planner which then sends triggers to start the presentation. The avatar starts by reading the TV show titles. Within the GUI a vertically scrolling text is used to display information (title, start time, end time) for each show. Each time a title is mentioned by speech the corresponding information appears in the GUI. Additionally the avatar points at the GUI list to stress the coreferences between the list and the avatar speech. After the presentation is finished acknowledgements are sent back to planner which is then ready to process the next presentation task.

Discussion

As mentioned in the "requirements" section a multimodality model should fulfill the requirements of sufficient completeness, uniqueness, relevance and intuitiveness. The sufficient completeness can be obtained by a proper definition of the subatomic level in the unimodality taxonomy as well as by a proper definition of the multimodal types. With these two concepts any multimodality in the domain can be modelled. Furthermore the unimodal taxonomy has been shown to be complete in the media of acoustics, graphics and haptics (Bernsen 2001). Therefore also every multimodality in the domain covered by the multimodal types can be modeled by covering the unimodalities, the synchronizations and the referring acts. What follows is that the sufficient completeness claim is fulfilled by the model.

The uniqueness claim is also fulfilled as every multimodality can only be modeled in one way in the multimodality model. After the set of unimodalities, the synchronizations and referring acts of a multimodality have been identified and a proper multimodal type has been defined the modeling of multimodalities is unique in the domain modelled by the multimodal types.

However concerning the question whether the relevance claim and the intuitiveness claim have been fulfilled we are not yet able to give a final answer. We think however that by modeling sychronisations and referring acts as well as multimodal types we included the most important multimodal features into the model. We will implement a presentation planner, which exploits the multimodality models of the renderer agents. An evaluation of this planner will further investigate if the two claims have been fulfilled.

Related Work

The work on multimodality models currently seems to focus either on rather theoretical issues (Pineda & Garza 1999) or on very system-oriented approaches (e.g.(W3C)). We consider our approach to be rather a compromise between both theory and practice. In (Vernier & Nigay 2000) an output multimodality model is presented that is actually exploited by a multimodal presentation system. The multimodalities are modeled as a set of unimodalities that basically are characterized by Bernsen's unimodality properties (Bernsen 2001). These multimodalities can then be combined temporally, spatially and syntactically (e.g. formatted text) or semantically (e.g. by means of coreferences). However the model is restricted to graphic multimodalities.

Concerning the characterisation of unimodal modalities (Arens, Hovy, & Vossers 1993) have identified several modality characteristics like the "carrier dimension" or the "temporal endurance". However for our purpose characterisations like "default detectability" or "baggage" are not precise enough to model multimodalities properly. Therefore we opted for Bernsen's unimodality taxonomy which gives a complete und sufficiently detailed model for unimodalities.

Conclusion and Future Work

In this paper we presented our concept for modeling output multimodalities in a multimodal dialogue system. We elaborated on the different parts of the model and provided a comprehensive case study.

Future work on the multimodality model will include a complete definition of the "sub-atomic" level of the unimodality taxonomy of which we only gave a few examples in this paper (e.g. "list", "gesture" or "mimic") that can be used within the prevailing domain. Moreover we intend to define a device model in order to take care of the resource restrictions of the output multimodalities and the physical

location of the renderers. After the multimodality model has been formalized the output multimodality models will be used by the EMBASSI presentation planner (Elting & Michelitsch 2001) to properly distribute the content to be displayed among the multimodalities and provide a coherent multimodal presentation.

Acknowledgements

This work was funded by the Klaus Tschira foundation and the German Federal Ministry for Education and Research as part of the EMBASSI project (grant 01 IL 904 D/2).

References

André, E., and Rist, T. 1994. Referring to World Objects with Text and Pictures. In *Proc. of the 15th COLING*.

Arens, Y.; Hovy, E. H.; and Vossers, M. 1993. On the Knowledge underlying Multimedia Presentations. In Maybury, M., ed., *Intelligent Multimedia Interfaces*. AAAI Press/The MIT Press.

Bernsen, N. O. 2001. Multimodality in Language and Speech Systems - From Theory to Design Support Tool. In Granström., ed., *Multimodality in Language and Speech Systems*. Kluwer Academic Publishers.

Elting, C., and Michelitsch, G. 2001. A Multimodal Presentation Planner for a Home Entertainment Environment. In *Proceedings of the PUI'01 Workshop on Perceptive User Interfaces*.

Herfet, T.; Kirste, T.; and Schnaider, M. 2001. EMBASSI - Multimodal Assistance for Infotainment and Service Infrastructures. In *Proceedings of the EC/NSF Workshop on Universal Accessability: Providing for the Elderly*.

Pineda, L., and Garza, G. 1999. A Model for Multimodal Representation and Inference. In Neilson, I., and Paton, R., eds., *Visual Representations and Interpretations*. Springer-Verlag.

Vernier, F., and Nigay, L. 2000. A Framework for the Combination and Characterization of Output Modalities. In *Proceedings of DSV-IS2000*, 32–48.

W3C Recommendation: Synchronized Multimedia Integration Language (SMIL2.0). http://www.w3.org/TR/SMIL20/.