Architecture and Implementation of Multimodal Plug and Play

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

Stefan Rapp
Sony Corporate Laboratories Europe
Hedelfinger Straße 61
D-70327 Stuttgart, Germany
rapp@sony.de

Gregor Möhler
Sony Corporate Laboratories Europe
Hedelfinger Straße 61
D-70327 Stuttgart, Germany
moehler@sony.de

Michael Strube
European Media Laboratory GmbH
Schloss-Wolfsbrunnenweg 33
D-69118 Heidelberg, Germany
michael.strube@eml.villa-bosch.de

ABSTRACT

This paper describes the handling of multimodality in the Embassi system. Here, multimodality is treated in two modules. Firstly, a modality fusion component merges speech, video traced pointing gestures, and input from a graphical user interface. Secondly, a presentation planning component decides upon the modality to be used for the output, i.e., speech, an animated life-like character (ALC) and/or the graphical user interface, and ensures that the presentation is coherent and cohesive. We describe how these two components work and emphasize one particular feature of our system architecture: All modality analysis components generate output in a common semantic description format and all render components process input in a common output language. This makes it particularly easy to add or remove modality analyzers or renderer components, even dynamically while the system is running. This plug and play of modalities can be used to adjust the system's capabilities to different demands of users and their situative context. In this paper we give details about the implementations of the models, protocols and modules that are necessary to realize those features.

Categories and Subject Descriptors

H.5.2 [**Information Systems**]: User Interfaces – *interaction styles, theory and methods.*

General Terms

Algorithms, Human Factors, Theory.

Keywords

Multimodal, dialog systems, multimodal fusion, multimodal fission.

ICMI '03, November 5-7, 2003, Vancouver, British Columbia, Canada.
Copyright 2003 ACM 1-58113-621-8/03/0011…$5.00.

1. INTRODUCTION

Traditional consumer electronics used to be clearly separated from computers. Nowadays the border between both areas is becoming more blurred. The Internet can be surfed on the TV and the stereo can be operated from the computer. However this merging does not only have its bright sides. People not experienced to computers and novel interfaces are easily left behind due to the complex controls and incompatibilities between different systems. Therefore the Embassi project aims at bringing together psychologists, computer scientists and engineers from consumer electronics to develop a system with the goal of lowering the technical barrier of modern consumer electronics and shifting the focus back to the user. Embassi consists of a consortium of 19 partners from industry and academia partially funded by the German Federal Ministry for Education and Research.

In this paper we outline our approach to natural interaction between humans and machines. In Embassi multimodality is managed by a Polymodal Input Module (PMI), which merges speech, gestures and input from a graphical user interface, and a Polymodal Output Module (PMO), which decides upon the modality used for the output, i.e., speech, an animated life-like character and/or the graphical user interface. In the following sections we outline the Embassi architecture, the modality fusion performed by the PMI and the presentation planning performed by the PMO.

2. THE EMBASSI ARCHITECTURE

The architecture of the Embassi system comprises a considerable set of agents. These agents are grouped into layers that deal with information at different levels of abstraction. Between each layer a well-defined protocol ensures that components can be added and removed easily. The agents communicate with each other using a subset of the agent communication language KQML [4] containing content expressed in XML-syntax [16] conforming to a DTD that describes the underlying ontology used in the system. The intelligence of the Embassi system is spread across multiple agents. These agents can enlist the help of other agents to accomplish a task cooperatively. As a result, new features can be implemented by incrementally adding new agents to the system. As a difference to the architecture presented in [18] Embassi uses a message-driven communication paradigm with a clear pipelined organization instead of a general blackboard as a central data structure.

The components shown in Figure 1 can be grouped into user input related (I, F, PMI), user output related (PMO, R, O), the generic dialog manager (D), the agents assisting the user in performing special tasks (A), the abstract representations of physical devices controlled by the system (X), and the context manager (C), a general repository for system-wide information, such as user profiles, user state, and the status of all connected devices.

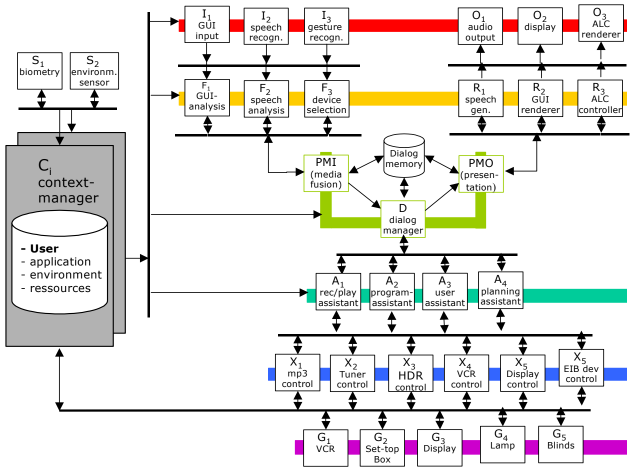


Figure 1: Architecture of the Embassi multimodal assistance system

When a combination of user events, such as a gesture and an utterance takes place, each type of event is dealt with by a dedicated I-component. Each event is transformed into a representation that hides device-specific details. As a result, one I-component that processes gestures picked up by a camera can be replaced by another component, which processes for instance pen input.

At the next level of abstraction, F-components transform the user events into a semantic representation of the user action that reflects the intention of the user independently of the type of action. Input expressed in different modalities has to be processed concurrently. Specifically, in our implementation, after recognizing speech (I_2), a word hypothesis graph is analyzed by a chunk parser (F_2) that extracts semantic concepts and the associated syntactic structure. At the same time, key events on the remote control (I_1) are analyzed in the GUI-analysis component (F_1). Also the data originating from pointing gesture recognition (I_3) is processed concurrently. The agent sends the recognition result to the device selection component (F_3) that maps it to one of several consumer electronic devices. The device positions in the room must be known to the system and are kept in the context manager (C_i). All F components use the same semantic representation as output, so called discourse representation structures (DRSs) [9,11] that were originally developed for natural language processing. A similar architecture is used within the SmartKom project [22]. However the SmartKom architecture lacks a clear grouping of agents, which results in a loss of clarity compared to the Embassi architecture.

The PMI component, further detailed in section 3, integrates the semantic representations of the F-components by resolving references between the modalities and past entities of the dialog history. This process is also referred to as media fusion. From the integrated semantic representation, the dialog manager infers the intention of the user (possibly by triggering further interactions with the user) and delegates the task of performing the desired operation to one of the assistants. This assistant can enlist the support of other agents in order to complete its job. The actual devices are not controlled directly by the agents but via the corresponding X-component, which provides an abstract representation of the features commonly found in devices of its class. For the demonstrator, an MP3 player, VCRs, an AV hard disk drive, tuner and display devices as well as a fan, lighting systems and blinds can be operated.

The dialog manager then formulates a reply to the user in an abstract amodal form. The message is sent to the PMO (described in section 4), which decides how to present it to the user. To that end, it considers the context of the dialog, user preferences, and the content of the message in order to set up an appropriate multimodal presentation. The rendering agents, e.g. text generation, program selection GUI and the animated life-like character (ALC), generate output according to the system intent and content formulated by the dialog manager and the setup imposed by the PMO. The result is a dynamic re-balancing between audio, graphical user interface, and the ALC. Finally, the O-components control the output devices that are responsible for speech generation and the display of graphics on a TV set.

3. MODALITY FUSION

In this section we present the way in which multimodal input is combined in the Embassi system. As stated above, users are free to choose any of the three supported modalities alone resulting in unimodal input, or they can combine modalities to form a multimodal input. The modality fusion component, called PMI, performs a semantic integration in which it considers concepts of any modality. The integration done takes place on a semantic level. Any further integration on a sensor or lexical level can be added on the I- or the F-level.

3.1 Merging of semantic structures

To give a simple example, a user might say ”switch on” while pointing to a specific lamp. As the example as depicted in Figure 2 suggests, the semantic analyses for all modalities are uniformly represented according to a common formalism (discourse representation structures (DRSs) [9,11]) and a common ontology (domain concepts). The output of the modality fusion is the combination of the command and the object specification as shown to the right of Figure 2. The semantic role ‘has-lamp’ that the object has for the action command ‘TurnOnEIBDevice’ is retrieved from a projection of the ontology included in the PMI module. It states which kind of commands can be associated or linked with domain objects if a semantic role is not given already.[1]

[1] The examples are taken from the real system. As a consequence, some aspects of the ontological modeling must remain unexplained in this paper. The interested reader is pointed to [23] that explains some design considerations for the ontology and the separation between dialog manager and assistant components and the downloadable version of the demonstrator that includes a complete description of the ontology in form of an XML-DTD.

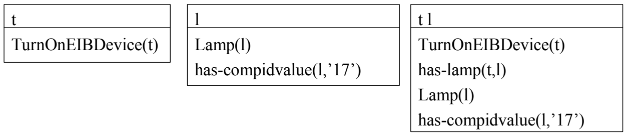


Figure 2: Fusion of verbal utterance and pointing gesture

Apart from the simple combination of a command with an object, more complex fusions are possible in our system. In Figure 3, a combination of the verbal utterance 'record the news tomorrow evening' with a selection of a specific broadcast channel, 'ARD' on the GUI is shown. Although the naturalness of such an interaction is debatable, the example shows the co-specification of an object by means of two modalities. As a result, the dialog manager receives a merged analysis, namely to 'record the news on channel ARD tomorrow evening'. It can be seen in Figure 3, that some aspects of the object (a program event) are specified from speech, such as the genre being news ('Nachrichten'), and the time interval where it is to be broadcast, being around 7 pm tomorrow. Also the command, namely to record it, is included in this DRS from speech analysis. From the GUI, the specification of program event objects of broadcast channel 'ARD' is received. As both the specification from the speech analysis and from the GUI is done following the same ontology, the two structures can be unified straightforwardly to form the combined analysis shown to the right.

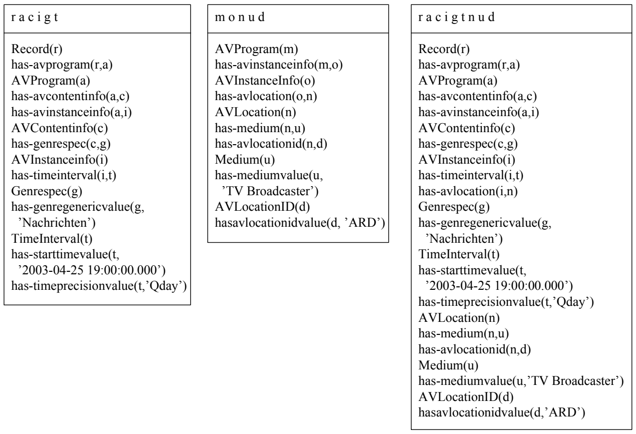


Figure 3: Fusion of verbal utterance and program selection

The two examples shown are simplified for brevity in two respects. Firstly, the DRSs are more elaborate in the real system. The messages have been slightly edited to hide some irrelevant details. Also they were converted from the XML representation actually used to a more readable form. Secondly, there can be more than one DRS coming in from each F-component. All input to the PMI can generally be ambiguous and ordered by a score. Of course this property is mainly due to the speech processing, but could also be beneficial to the other modalities if they suffer from uncertainty during recognition or analysis. For example, in our system, there is a video based pointing gesture tracking that can issue ambiguous references to devices, e.g. if there is a lamp very close to a VCR, say. Then, through the modality fusion, the ambiguity can potentially be resolved, for instance, if a verbal command 'record the news there' can be combined with the VCR but not with the lamp. In the system there is an alternative means to select devices. It works with detecting a user directed laser beam in a detector field attached to the device. Here, the triggering of a device is unambiguous, and hence the analysis can always only be a description of a single device.

3.2 Synchronization of concurrent input

Besides the homogeneity of the filter components with respect to the ontology as well as the syntax of the messages issued to the PMI, also the protocol is the same for all the analysis components. As all the components in the Embassi demonstrator run in parallel, care has to be taken for synchronizing the concurrent input streams to the PMI. We defined a protocol in which the result of any modality analysis are sent directly to the PMI and in which it is ensured that all relevant data from all considered modalities are gathered through queries to the other modality analyzers before further processing. More concretely, the protocol is as follows. Whenever one analyzer, say F_x, sends input describing a user interaction in the interval (t_a,t_b), then all connected further filter components, say F_y and F_z, receive a query for information relevant to (t_a,t_b). In case F_y has pending information relevant for this interval, F_y simply awaits until the analysis for this input is finished and sends input to the PMI as soon as it is finished with it. If F_z, say, has no user input relevant to (t_a,t_b), it immediately responds with an empty hypothesis graph covering (t_a,t_b). Finally, when F_y has concluded on its analysis it responds with a hypothesis graph to the PMI that in general can cover a larger interval, say (t_a,t_c), where $t_a<t_b<t_c$. Then the considered time interval in the PMI is extended to (t_a,t_c), and all components that have not given information on the interval extension (t_b,t_c), i.e. F_x and F_z are asked again. Let us assume that they all promptly respond with an empty graph indicating no further available user input. Then, the synchronization is finished and the PMI proceeds with the semantic unification step as described above. The protocol is suitable for our purpose of supporting plug and play of modalities in three respects: firstly, it ensures that all modality analyzers are considered in the integration, even if processing can not be done in real time (as is often the case with natural language processing). Secondly, the protocol avoids race conditions (replies and independently delivered information share the same message format). Finally, the protocol is completely symmetrical, thus it does not impose the presence of a specific modality analyzer that triggers an interaction.

3.3 Dynamic integration of analyzers

Also registration and deregistration of F-components is straightforward with the synchronization protocol. Whenever an F-component is inserted into the system, it is required to send an empty hypothesis graph to the PMI that enlists this analyzer for future synchronization queries. Whenever an F-component leaves the system, on the synchronization message following the subsequent user interaction, the PMI receives an error message from the router, that the F-component is no longer available and excludes it from further synchronization queries.

3.4 Corpus based rescoring of hypotheses

From a linguistic viewpoint, the main task of the PMI is to resolve anaphoric expressions. Early multimodal dialogue systems constrained the user with respect to the words or gestures [15] or allowed only for two modalities to be used, usually pointing gestures and speech [7]. These systems had methods for merging input from different modalities. However, they lacked methods for determining the correct interpretation for ambiguous input. In contrast, the Embassi system allows for input by speech, by pointing gestures and by a graphical user interface using a remote control. As said before, in principle arbitrary input modalities can be hooked up to the system since all input modules use the same semantic formalism. However, this increases the possibility of ambiguous interpretations, which was avoided in earlier dialog systems.

We investigated the relevant linguistic phenomena in a pilot study where we analyzed textual transcripts[2] of about 50 hypothetical human-machine-dialogs that developers of the project contributed. We set up a list of 14 relevant phenomena, including individual, deictic, discourse-deictic and expletive personal or demonstrative pronouns, synonyms and hyperonyms, synecdoche and metonymy, indirect anaphora (bridging) and proper nouns. For multimodal interactions, especially deictic expressions are important (such as a `that' which refers to an object identified by another modality). But a deictic multimodal interpretation must always be checked against a discourse-deictic reading, where the referenced item is a constituent introduced in the dialogue before. For example, a `there' could refer to a VCR that was switched on immediately before the sentence ‘record the eight o’clock news there’ was uttered, and a pointing gesture to the HDR could have erroneously been interpreted as an intended input. Consequently, the PMI must not only consider anaphoric expressions referencing to another modality but also to the same modality, context, and, irrespective of the used modality, the dialog history. For that purpose, the PMI was given read access to the context manager and dialog history in the implementation architecture.

Our approach is thus similar to the one taken in [8], but differs in an important aspect. Whereas Johnston uses a unification based grammar for the multimodal integration, we use a corpus based approach similar to the one presented in [13] for anaphora resolution in spoken language. Instead of manually deriving task independent unification grammars, we apply machine learning techniques to that task. The idea is to train classifiers that decide, which linguistic phenomenon can be applied to each of the possible antecedents, that is, whether one merged DRS is to be preferred over another merged DRS, or whether an unmerged DRS is to be preferred. Practically, the classifiers decide on the scoring of the hypotheses, that is, they can favor a hypothesis with an otherwise lower score over a higher scored one. The decision is based on some simple features we extract from the syntactic and semantic (and, in the future, maybe also prosodic) structure. In doing so, we hope to achieve a concentration on the linguistic phenomena that are especially relevant to our task.

In order to establish a corpus on which to train our machine learning algorithms, we conducted two collection efforts. In the first campaign, we collected about 3000 uni- and multimodal interactions of 42 subjects, roughly balanced with respect to gender and age. The subjects were exposed to a living room setting in our studio where they were prompted to perform dialog steps in order to operate consumer electronic equipment. Each prompt consisted of a small user goal, such as switching on a device, selecting a movie from an EPG or recording a movie on a VCR. The subject was completely free to choose among different modalities (speech, free pointing with the finger, remote-control) as well as choosing the concrete form (e.g. lexical choice). About 20 such prompts made up one continuous story. We collected about 170 stories. The user interactions were picked up by several microphones, two cameras for the gestures, an additional camera for gaze analysis and lip reading as well as a PC logging the key press events on the remote control.

In a second campaign, we exposed an early prototype to the subjects. It could treat a small but core part of the system (browsing an electronic program guide (EPG) and selecting a program for recording), either by GUI, speech or a multimodal combination of both. This time, we recorded 65 age and gender balanced subjects. Audio was captured as for the first campaign. As enough video data was collected already in the first campaign, we avoided to collect several video streams and only captured the overall scene on DV tape and MPEG-4 for annotation purposes. For annotation of the system's activities we developed an XML-based centralized logging facility. The campaign was combined with an evaluation of the influence of different output strategies [10]. Different subject groups had to interact with three versions of the system, each having specific output capabilities, i.e. GUI only, GUI and synthesis, GUI and synthesis and animated face. The subjects were asked to solve 3 tasks: record a specific program, browse and select any interesting program for recording, and again record a specific program. These data were annotated using an annotation tool [12], which produces a set of XML files for each multimodal dialog. From these XML files data for feeding machine learning algorithms with training data are generated. The evaluation on testing data provides information on which model (i.e., which set of features in which order) performs best and which features yield significant impact on the results. For the task of multimodal integration a feature, which contains information about the time course of the input modalities, appears to be important. Another important feature concerns the linguistic realization of the referring expression (e.g., personal pronoun or demonstrative pronoun?). Finally, the model, which performs best on the testing data, is used within the PMI module to decide upon the integration of the different modalities.

3.5 Implementation

The PMI module is implemented in a combination of Java and Prolog. In the Java part, all communication within the communication framework is handled including KQML message management and XML parsing of the message content. Also the synchronization and analyzer registration algorithm as described above is completely contained in the multithreaded Java part. All DRSes contained in the edges of the hypotheses graphs that are received are converted to logical form and inserted into the prolog engine. The possibility of merging DRSes is checked for by prolog predicates. Through backtracking, all possible merged and unmerged DRSes are reported to the Java process, which then can apply the rescoring classifier and convert back from logical form to the KQML/XML based message format according to the common system ontology.

4. PRESENTATION PLANNING

The task of the presentation planning component PMO is to provide output that is coherent and cohesive. The PMO can make use of different rendering agents, which are situated on different output devices situated at different locations (Figure 1). We use ALCs [19] to provide an intuitive and natural interface to the dialog system. Moreover stand-alone speech-synthesis agents are used if graphics are not available or applicable. Custom GUIs of different sizes and

[2] Relevant non-speech events have been included in the transcript by means of a textual description in angle brackets in this study.

types are used in order to display information graphically. Each of those rendering agent types can be implemented on different output devices, which differ with respect to resource restrictions and to locations. Another challenge of the Embassi scenario is that each of those devices can be plugged in and out at run-time.

In the remaining part of the section we illustrate the presentation task in Embassi. Afterwards we explain the rendering agent service descriptions that serve to describe the syntax and semantics of a multimodal rendering agent and enable a dynamic plug-and-play of output components. Then we give an overview of our heuristic rule-based approach to presentation planning and provide examples. We conclude with an overview of the implementation of the presentation planner.

4.1 Presentation task

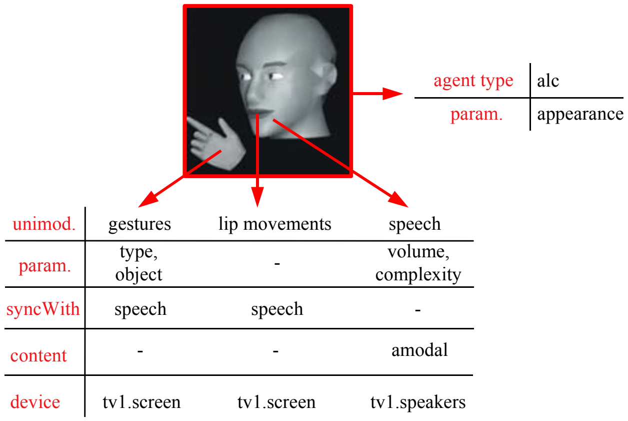
In the following we give an overview of the presentation task in Embassi that the PMO is facing. A speech act is used to identify the abstract goal of the presentation. The content of the presentation can consist of TV show content to be presented or of a general system message (e.g. that the last task has been fulfilled successfully). In order to generate coherent and cohesive presentations it is also necessary to know about the current context of the presentation. In our case the presentation context is given by the information provided by face-finder sensors, which detect the number of users present in front of an output device. Moreover the history of multimodal interactions has to be taken into account in order to generate output that is coherent in time. However, multimodal dialog systems need not only be adaptive, but also adaptable. Therefore in Embassi the user can influence the generation by means of output preferences, which are part of the user's profile [20]. Output preferences include rendering agent-specific preferences (e.g. the appearance of the ALC) as well as more general presentation-specific preferences (e.g. concerning the graphical layout).

The biggest challenge in Embassi is to adapt the presentation according to the dynamic set of available rendering agents. In a situation in which the user is currently working on a PDA, it might be meaningful to present an important message by means of the small PDA-GUI. However if the PDA is currently switched off, the TV set GUI might be a better alternative. Moreover the presentation content has to be adapted according to the rendering capabilities of a rendering agent. E.g. if a certain PDA-GUI could only render speech then only a limited amount of linguistic information can be transmitted at once due to the dynamic nature of speech. Finally a presentation also has to be adapted to the resource limitations of the device the output is currently rendered on.

We define a *presentation situation* as the tuple consisting of the speech act, the content, the sensory context, the interaction history, the output preferences and the set of available rendering agents for the current presentation task.

4.2 Rendering Agent Service Descriptions

In order to deal with a dynamic set of rendering agents we use a model that allows rendering agents to describe their functions and restrictions to the PMO [25]. This serves to describe the syntax and semantics of rendering agents and allows the automatic inclusion of previously unseen rendering agents into a presentation at run-time.



agent type	alc
param.	appearance

unimod.	gestures	lip movements	speech
param.	type, object	-	volume, complexity
syncWith	speech	speech	-
content	-	-	amodal
device	tv1.screen	tv1.screen	tv1.speakers

Figure 4: Service Description of an animated life-like character agent[3]

Figure 4 illustrates the rendering agent service description for an ALC that is rendered on a TV set. The agent type identifies the overall type of the rendering agent. The set of multimodal parameters contains the multimodal parameters that have to be set for a presentation and that parameterize the complete multimodal output (e.g. the appearance of the ALC). The set of unimodalities contains the single output unimodalities that form the output multimodality (e.g. gestures, lip movements or speech). For each unimodality we keep track of the following information. The set of unimodal parameters are the parameters that have to be set for each unimodality before a presentation can take place (e.g. speech volume). Moreover it is necessary to keep track of the synchronizations that have to be conducted for each unimodality. In this example the unimodalities gestures and lip movements have to be synchronized with speech. Speech itself does not need to conduct any further synchronization. Moreover it is important to keep track of the types of content that each unimodality can process. A map unimodality can only render geographical information whereas a speech unimodality can render arbitrary content. Another required piece of information is the output device on which the output is rendered. This is necessary in order to take care of the resource restrictions of the device and its location relative to the user. Finally it is necessary to set layout parameters, that establish the layout of the presentation for those rendering agents that present content on the same device and in the same medium. One possible layout for an ALC and a GUI is a vertically split screen.

Figure 5 illustrates how rendering agent service descriptions are used to support the dynamic addition and retraction of new rendering agents. In this example the output of the Embassi system is being displayed by means of a GUI rendering agent that displays graphical output on the screen of a TV set. When a new PDA is wirelessly connected to the agent platform, the rendering agent located on the PDA sends a rendering service description to the PMO. This service description serves to adapt the presentation to the type and capabilities of the rendering agent as well as the resource restrictions of the output device. After the analysis of this service description the PMO chooses to reduce the content of the presentation due to the limited screen size of the PDA compared to

[3] Copyright ALC picture © 2003 Center for Graphic Data Processing (ZGDV), Darmstadt, Germany .

the TV set and sends an adapted presentation back to the rendering agent on the PDA. This information can then be displayed accordingly.

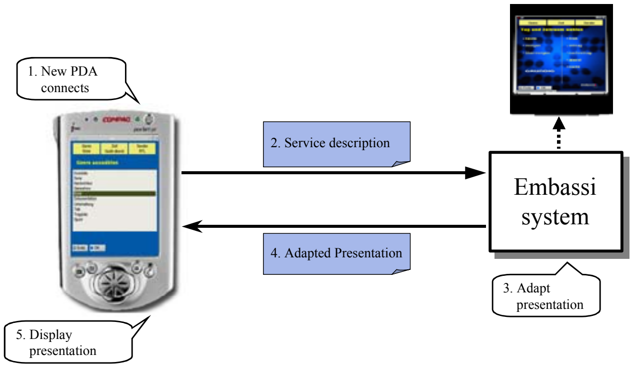


Figure 5: Dynamic addition of a PDA-GUI rendering agent [4]

4.3 Heuristic Rule-Based Presentation Planning

The goal of the PMO is to plan a presentation that fits the current presentation situation. To achieve this the PMO uses a rule-based planning approach, which is common among automatic presentation systems (e.g. [21]). To do so the PMO uses a distance measure to score a particular presentation in relation to all other possible presentations. As any latency of the system answers should be avoided this process should happen in near real-time.

On the presentation planning level we do not distinguish between rendering agents but introduce the concept of *multimodal configurations* (MMC). A MMC is a logic multimodality that is composed of one or more rendering agents. MMCs represent meaningful output multimodalities that can be efficiently used to build multimodal presentations. An “Assistant”-type MMC is made up of a rendering agent for the graphics part of an ALC and a combined text generation and speech synthesis agent. A “SystemVoice”-type MMC consists of a combined text generation and speech synthesis agent. Moreover a “ProgramSelectionGui”-type MMC is made up of a single GUI rendering agent. In the Embassi system we do not synchronize GUI output with speech, therefore we do not need to include speech into “Program Selection Gui”-type MMCs. We call a set of MMCs a *presentation*.

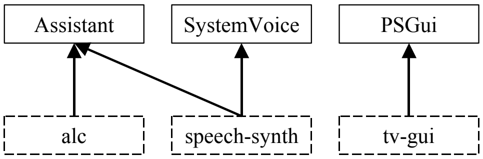


Figure 6: Building MMCs from rendering agents

Figure 6 illustrates how MMCs are built from a rendering agent set consisting of a graphics-only ALC rendering agent, a speech synthesis agent and a GUI rendering agent. Note that the speech synthesis agent can be used in an “Assistant”-type MMC, in which speech is synchronized with graphics, as well as in a “SystemVoice”-type MMC, in which speech is rendered without further synchronizations.

The search space for the presentation planning problem is exponential over the number of available MMCs resp. rendering agents. A presentation can be generated by an arbitrary set of MMCs. Therefore the power set of all MMCs has to be analyzed, which is exponential over the number of MMCs. Moreover, for a certain set of MMCs all possible combinations of parameter values have to be analyzed, which is also exponential over the average number of different parameter values. Therefore, there is a need to apply proper heuristics to reduce the search space.

Scoring Type	*Scoring Items*
presentation	{}, {A}, {P}, {S}, {A, P}, {P, S}, {A, S}, {A, P, S}
MMC	{A, P, S}
unimodalities	{gestures, lip-movements, speech, graphic-text}

Table 1: Scoring complexity

Our approach is illustrated in table 1 for three MMCs A, P and S. Configuration A contains the unimodalities gestures and lip-movements, P consists of the unimodality graphic-text and S consists of the unimodality speech. We first use a distance measure to score all possible sets of MMCs (presentations), which has the complexity 2^n for n MMCs. Then we score every single MMC according to how well it fits to the current presentation situation, which has complexity n. After that we score each unimodality of each MMC, which has the complexity nu, if u is the average number of unimodalities of an MMC and no sharing of unimodalities takes place. Therefore the selection of a presentation has the complexity 2^n+n+nu for n given MMCs. We call this process the selection process.

Here we applied the following heuristics. For a given set of MMCs we do not calculate a score for all possible combinations of unimodalities of a presentation. We argue that if the scores for a set of MMCs and for every single unimodality within the set are sufficiently high, then it is also very likely to find a proper set of unimodalities that fits the current presentation situation. Therefore we omit scoring all possible combinations of unimodalities within the set.

Furthermore we do not consider all possible combinations of unimodal, multimodal and layout parameter values. Instead we argue that for the presentation hypothesis containing the best combination of scores concerning the set of MMCs, the single MMCs and the unimodalities of the MMCs it is very likely that a convenient combination of parameters exists. Therefore we only set unimodal, multimodal and layout parameter values for the best presentation hypothesis according to the selection process. This has the complexity $nu+n+l$ for l layout parameters. We call this process the parameterization process. Consequently the complete algorithm consisting of the MMC selection process and the parameterization process has the complexity $2^n+2n+2nu+l$. For $n=10$ available MMCs with an average of $u=10$ unimodalities for each MMC and a total of $l=10$ layout parameters this results in 1.254 presentations to be inspected, which is still a sound complexity for near real-time processing.

[4] Copyright GUI pictures © 2003 Grundig GmbH, Nuremberg, Germany.

4.4 Example

In the following section we provide an example of the presentation planning algorithm. In the current presentation situation the speech act "message-warning" indicates that an important warning message for the user should be rendered (in this case that her favorite TV show is about to start). The sensory context data indicates that the user is currently located in the kitchen. Currently an "Assistant"-type MMC, which is located on a TV set in the living room, is available as well as two "SystemVoice"-type MMCs located in the living room and the kitchen and a "ProgramSelectionGui"-type MMC located in the kitchen.

During the selection process we first assign the best scores to combinations of MMCs that combine dynamic graphic animations with acoustics. This is due to the facts that the current message is a warning message that should be immediately perceived by the user. This is the case for the combination of the "SystemVoice"- and the "ProgramSelectionGui"-type MMCs, which therefore receives the best score. The same is true for the "Assistant"-type MMC in the living room. Then the scores for every single MMC are calculated. As there are no specific differences in using each MMC in a presentation (e.g. due to MMC-related output preferences) each MMC receives the same score increment. Afterwards every single unimodality of the MMC is scored. Speech unimodalities receive a lower score due to the output preferences. Additionally, acoustics and dynamic graphics, which are rendered in the room that the user is currently located in, are preferred as they are easily perceivable for the user. As a result the best-scored presentation is the presentation that consists of a single MMC, namely the "ProgramSelectionGui" in the kitchen that can display dynamic graphics. Afterwards the parameterization process starts. For the dynamic display of text the type of animation has to be chosen as well as the text complexity, which are both set accordingly for a warning message.

If the user would have been in the living room instead of the kitchen the "Assistant"-type MMC in the living room would have been chosen as it involves graphics and dynamics in the room the user is currently located in. Although this choice contradicts the output preferences, the algorithm favors an easily perceivable warning message over accordance with user preferences.

4.5 Implementation

As a difference to existing presentation planning systems (e.g. [1, 17, 14]), that use classical AI planning approaches and tools [3, 5, 6], we chose to implement a special purpose expert system. The reason for that is the complexity of the presentation task. In order to properly formulate this data in a knowledge-based system, we chose to rely on object-oriented programming and complex object type-checking, which could not be provided by the other systems to the extent needed.

We implemented Java classes to represent a rendering agent service description. When the set of available rendering agents changes (e.g. due to an output device being disconnected) the corresponding set of rendering agent service descriptions is updated accordingly. When a new presentation task is sent to the PMO from the dialog manager the PMO first starts to build a set of available MMCs from the set of available rendering agents. After that the presentation planning starts by building the power set of all MMCs, which represents all presentation hypotheses. Then all the matching rules stored in the rule base are applied to the hypotheses to score sets of MMCs, single MMCs and single unimodalities. After that the best presentation is chosen for parameterization. During the scoring process the choice of proper distant measures is crucial. For instance a simple score increment for well-suited unimodalities would have the effect that the most complex presentations, which make use of many unimodalities, are preferred over less complex presentations. We are currently evaluating different distance measures, which provide a more stable scoring of sets of MMCs, MMCs and unimodalities.

In the parameterization process parameterization rules are applied to set the unimodal, multimodal and layout parameters necessary for the presentation. After that the fully instantiated presentation is passed on to a protocol handler, which translates the presentation into a proper KQML protocol for the PMO and the rendering agents. Finally, the protocol is executed and the multimodal presentation is rendered accordingly.

A fault-tolerant combination of a message buffering system and a state machine ensures a stable information flow. This is crucial as three different protocols are executed in parallel. The first protocol is the output protocol between the dialog manager, the PMO and the rendering agents. The second protocol is conducted with the context manager, which contains the user output preferences as well as the sensory data and which can trigger notification messages at any time. The third protocol is conducted with the router to check the set of rendering agents available for output.

However, the PMO can only heuristically parameterize a presentation, as it cannot foresee the exact resources needed by the media objects, which are generated by the rendering agents. Therefore a protocol set by the PMO can also fail (for instance because it is not possible to place all pictures into the layout proposed by the PMO). In this case the rendering agents acknowledge the failure of the rendering process and the PMO chooses the second best presentation or repeats the parameterization process with a new set of constraints.

5. CONCLUSION

The information society of today is growing fast. Nowadays users show not only a great variety of age, experience and cultural background but also special physical and cognitive needs. Therefore it is especially imperative for multimodal dialog systems to allow a great degree of flexibility in handling and configuring such a system. In this paper we gave details on the handling of multimodality in the Embassi system, which allows an individual configuration of multimodal input and output components at run-time. In this paper we described the general architectural approach taken in Embassi and the two modules PMI and PMO that are responsible for multimodal fusion and fission respectively. We illustrated the semantic protocol used by the PMI to fuse the asynchrous input from the modality analysis components and how a special registration protocol supports the dynamic addition and retraction of analyzers at run-time. Moreover we have shown how the PMO uses agent service descriptions to support the corresponding dynamic addition or retraction of output components and how these service descriptions are exploited during the presentation planning process.

PMO and PMI are an integral part of the implementation of the Embassi software demonstrator part of which is also available for download[24]. The overall system architecture and the two modules PMI and PMO have already been proven useful in another (company internal) project and will most likely be further developed in other contexts.

6. ACKNOWLEDGMENTS

We would like to acknowledge the work of Georg Michelitsch, Robert Porzel, Claudia Settele-Raspl, Yun Ding, Holger Geisen and Christoph Müller, that have been involved in the conceptual design, implementation and/or experiments of the PMI/PMO components. This work was partly funded by the German federal ministry of education and research as part of the Embassi project (grants IL 904 S 8 and 01 IL 904 D/2) as well as by the Klaus Tschira foundation.

7. REFERENCES

[1] André, E., Finkler, W., Graf, W., Rist, T., Schauder, A., Wahlster, W., WIP: The Automatic Synthesis of Multimodal Presentations. In: Mark Maybury (Ed.), Intelligent Multimedia Interfaces, pp. 75–93, AAAI Press, 1993.

[2] Bernsen, N. O. and Dybkjær, L.: "A Theory of Speech in Multimodal Systems", In: Dalsgaard, P., Lee, C.-H., Heisterkamp, P. and Cole, R. (Eds.): Proceedings of the ESCA Tutorial and Research Workshop on Interactive Dialogue in Multi-Modal Systems, Irsee, Germany, June 1999, pp. 105-108.

[3] CLIPS - C Integrated Production System, Version 6.3, http://www.ghg.net/clips/CLIPS.html, 2003.

[4] Finin, T., Fritzson, R., McKay, D., McEntire, R., KQML as an Agent Communication Language, Int. Conf. on Information and Knowledge Management, Maryland, MD, 1994.

[5] JAM: A BDI-theoretic Mobile Agent Architecture, Huber, M. J., Int. Conf. on Autonomous Agents Agents'99, Seattle, WA, 1999.

[6] Jess – The Expert System Shell for the Java Platform, Version 6.1, http://herzberg.ca.sandia.gov/jess/, 2003.

[7] Johnston, M., Cohen, P. R., McGee, D., Oviatt, S. L., Pittman, J. A., and Smith, I.: Unification-based multimodal integration. In Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and of the 8th Conference of the European Chapter of the Association for Computational Linguistics, Madrid, Spain, 7–12 July 1997, pages 281–288.

[8] Johnston, M., "Unification-based Multimodal Parsing", In: Proceedings COLING-ACL 1998.

[9] Kamp, H., Reyle, U. From Discourse to Logic, Vol I, Kluwer, Dordrecht, 1993.

[10] Krämer N C, Nitschke J: Ausgabemodalitäten im Vergleich: Verändern sie das Eingabeverhalten der Benutzer? In R. Marzi (Hrsg.), Bedienen und Verstehen. 4. Berliner Werkstatt Mensch-Maschine-Systeme. Düsseldorf: VDI-Verlag, 2001.

[11] Ludwig, B. et. al. "Context and Content in Dialogue Systems" In: Proceedings of the Third International Workshop on Human-Computer Conversation. Bellagio, Italy, July 3-5, 2000.

[12] Müller, C. and Strube, M.: MMAX: A tool for the annotation of multimodal corpora. In Proceedings of the 2nd IJCAI Workshop on Knowledge and Reasoning in Practical Dialogue Systems, Seattle, Wash., 5 August 2001, pages 45–50.

[13] Strube, M, Rapp, S, Müller, C: The Influence of Minimum Edit Distance on Reference Resolution, In: EMNLP '02, Philadelphia, PA, USA. July 6-7, 2002, pp. 312-319.

[14] Towns, S., Callaway, C., Lester, J., Generating Coordinated Natural Language and 3D Animations for Complex Spatial Explanations, AAAI Workshop on Representations for Multi-modal Human-Computer Interaction, Madison, WI, 1998.

[15] Wauchope, K., Everett, S., Perzanowski, D., and Marsh, E.: Natural language in four spatial interfaces. In Proceedings of the 5th Conference on Applied Natural Language Processing, Washington, D.C., 31 March – 3 April 1997, pages 8–11.

[16] World Wide Web Consortium. Extensible Markup Language, http://www.w3c.org/XML/, 2003.

[17] Zhou, M., Feiner, S., Top-down hierarchical planning of coherent visual discourse, Int. Conf. on Intelligent User Interfaces IUI97. Orlando, FL, 1997.

[18] Brøndsted, T., Larsen, L. B., Manthey, M., Mc Kevitt, P., Moeslund, T., Olesen, K. G., The Intellimedia WorkBench - an environment for building multimodal systems. H. Bunt, R-J. Beun, T. Borghuis, L. Kievit (eds.): Int. Conf. Cooperative Multimodal Communication,Theory and Applications, Tilburg, January 1998.

[19] Rieger, T., Berner, U., A scalable avatar for conversational user interfaces, 7th ERCIM User Interfaces for All Workshop, Paris, France, October 24-25, 2002.

[20] Hellenschmidt, M., Kirste T., Rieger, T., An agent based approach to distributed user profile management within a multi-modal environment, International Workshop on Mobile Computing IMC 2003, Rostock, Germany, June 17-18, 2003.

[21] Müller, J., Poller, P., Tschernomas, V., Situated Delegation-Oriented Multimodal Presentation in SmartKom, Workshop Intelligent Situation-Aware Media and Presentations (ISAMP), AAAI-2002, Edmonton, Canada, 2002.

[22] Wahlster, W., SmartKom: Fusion and Fission of Speech, Gestures, and Facial Expressions, Proc. of the 1st International Workshop on Man-Machine Symbiotic Systems, Kyoto, Japan, 2002.

[23] Forkl Y, Hellenschmidt M: Mastering Agent Communication in EMBASSI on the Basis of a Formal Ontology, ISCA Tutorial and Research Workshop , Multi-Modal Dialogue in Mobile Environments, June 17-21, 2002 Kloster Irsee, Germany

[24] OpenEmbassi for Linux, available from http://www.embassi.de/open_embassi.

[25] Elting, C., Möhler, G., Modeling Output in the EMBASSI Multimodal Dialog System, Int. Conf. on Multimodal Interfaces ICMI02, Pittsburgh, PA, October 14-16, 2002.