An Animated Interface Agent Applying ATMS-based Multimodal Input Interpretation

Yasuyuki Kono, Takehide Yano, Tomoo Ikeda,
Tetsuro Chino, Kaoru Suzuki, and Hiroshi Kanazawa
{*kono,yano,tommy,chino,suzuki,kanazawa*}*@krl.toshiba.co.jp*

Kansai Research Laboratories, Toshiba Corporation
8-6-26 Motoyama-Minami, Higashi-Nada, Kobe 658-0015, Japan
TEL: +81 78 435 3104 FAX: +81 78 435 3161

Abbreviated title: ATMS-based Multimodal Interpretation

This manuscript has not been published elsewhere and has not been submitted simultaneously for publication elsewhere.

Summary

Two requirements should be met in order to develop a practical multimodal interface system, *i.e.*, (1) integration of delayed arrival of data, and (2) elimination of ambiguity in recognition results of each modality. This paper presents an efficient and generic methodology for interpretation of multimodal input to satisfy these requirements. The proposed methodology can integrate delayed-arrival data satisfactorily and efficiently interpret multimodal input that contains ambiguity. In the input interpretation, the multimodal interpretation process is regarded as hypothetical reasoning and the control mechanism of interpretation is formalized by applying ATMS (Assumption-based Truth Maintenance System). The proposed method is applied to an interface agent system that accepts multimodal input consisting of voice and direct indication gesture on a touch display. The system communicates to the user through a human like interface agent's 3D motion image with facial expressions, gestures and a synthesized voice.

1 Introduction

Humans can efficiently communicate with each other by using various means of communication such as language (voice), gestures, facial expressions and their combination. Thus we can say that human-human communication is essentially multimodal. A multimodal human-computer interface (MMIF), which enables a user to communicate with a computer without paying special attention to input methods, *i.e.*, an MMIF which accepts such input that anybody may give to his/her human counterpart is required (Maybury, 1994). As is said in (Maybury, 1993), human-computer interaction (HCI) technology should be promoted by developing practical MMIF technology, and various prototypes have been built to date.

In general, an MMIF reasons users' intentions from a set of fragmentary information which is entered using a variety of modalities. It receives multiple recognition results, which come asynchronously from the recognition modules, for example, a command by voice, or direct indication by a pointing device. Then the MMIF analyzes the relations among them, integrates, and finaly interprets them. Two requirements should be satisfied in order to develop a practical MMIF which can integrate and interpret multimodal input (MM integration/interpretation), *i.e.*, (1) integration of delayed-arrival data each of which arrives at different time, and (2) elimination of ambiguity in recognition results of each modality. We employed an assumption-based truth maintenance mechanism, ATMS (deKleer, 1986a; deKleer, 1986b), and formulated the control mechanism of MM input integration/interpretation in order to build a generic framework and efficiently cope with these problems.

We have developed a multimodal interface agent system that accepts multimodal input, namely the user's utterances and finger gestures on a touch screen such as pointing and circling. The system communicates to the user through a human-like interface agent's 3D motion image accompanied with a synthesized voice and non-verbal messages, namely facial expressions and hand gestures. Through developing, testing, exhibiting the system, we have obtained diverse knowledge about the multimodal HCI.

2 Multimodal interface

2.1 Multimodal interface agent system

Our goal is to establish an HCI technology and methodology which enables natural and efficient communications with computers. Considering conventional HCI methods, *e.g.*, GUI, we think that more natural human-computer communication can be realized by taking account of the four key points shown below:

Media integration that makes HCI similar to human-human natural communications, that is, integration of messages from various media, and

interpretation of the result of the integration.

Media complementation , *i.e.*, the ambiguities and incompleteness of recognition results from a certain mode should be compensated by the results from other modes, since unambiguous and complete recognition in a single mode cannot be expected.

Media allocation that enables appropriate allocation and switching of communication media according to the nature of handled information and the system user.

Non-verbal message processing such as facial expressions and gestures. Humans effectively communicate with each other by utilizing numerous non-verbal messages of various kinds. Non-verbal media can play an important role in human-computer interaction.

The first two points mainly concern the interpretation of user input and the last two the generation of system outputs. All four points involve tough problems, and no concrete methodology to process them has been established so far except the method based on typed feature structure unification presented by Cohen *et al.* (1997). Many of the existing MMIFs, *e.g.*, (Bolt, 1980; Kobsa, 1986; Stock, 1991; Koons *et al.*, 1993), depend not only on domains but also on modalities, that is, each MMIF is related to a specific combination of modalities and domains. A concrete and generic methodology that can better cope with the above points is required to construct a practical MMIF.

We have developed an efficient and generic multimodal integration/interpretation technology that enables media integration/complementation employing the ATMS, a hypothetical reasoning framework. We applied the method to a multimodal interface agent system that does work as a secretary. Figure 1 shows a screen-copy of the system. It is a kind of interface agent (Maes, 1997) which provides advice concerning office work on demand (Nakayama *et al.*, 1997). The system accepts multimodal input such as the user's utterance and gesture on a touch screen, *e.g.*, pointing and circling. The system communicates to the user through graphical representations and the interface agent's 3D motion image with facial expressions, gestures, and synthesized voice. The system can implicitly notify users of five kinds of its internal status with the agents' non-verbal messages as shown in Figure 2. The agents' mouth synchronously moves with synthesized voice.

In the situation shown in Figure 1, the user can, for example, select and view a document on the list. For instance, the user can tell the agent "Show me this minute book" while circling/pointing around the title of the document to which he/she wants to refer. The same gesture inputs cause a different system reaction if the user's words are different; for instance, the system shows different document for "Show me this minute book" and "Show me this report" in Figure 1.

Figure 3 shows the overall constituent modules and configuration of the multimodal interface agent system. The recognition module for each modality, for example, the Voice Recognition Module, accepts the user's input to the corresponding modality, performs its recognition process, and asynchronously sends the recognition result to the MM Input Integration/Interpretation Module. The MM Input Integration/Interpretation Module integrates the given inputs, derives a multimodal interpretation result, and sends the derived user's request to the application program, namely the know-how retrieving system. When the reply from the know-how retrieving system comes, the MM Input Integration/Interpretation Module sends the display output information to the Information Display Module, and the output text and non-verbal output information to the Agent's Motion/Voice Generator. Both the Voice Recognition Module and Voice Synthesizer run on UNIX workstations and all other modules run on Windows-NT based PCs. Each module is an independent software process. The modules run collaboratively, communicating with each other through KQML (Finin *et al.*, 1997) based messages.

2.2 Multimodal reference resolution

One of the most important technologies in developing MMIFs is the method for interpretating user's multimodal input. The major task of the MM Input Integration/Interpretation Module is to resolve the reference in a given multimodal input, called multimodal reference resolution(Neal *et al.*, 1991). Reference resolution is the problem of determining the object(s) referred to by verbal/non-verbal expressions.

Figure 4 shows the multimodal reference resolution process. The user asks "How can I contact this person?" while circling around the female person on the touch screen in this example. The MM Input Integration/Interpretation Module replaces noun phrases with referring expressions, *i.e.*, "this person", by "Ms. Nakayama" by resolving the referent. Then the know-how retrieving system receives a natural language text acceptable to the system, "How can I contact Ms. Nakayama?"

Verbal expressions that can be resolved by the MM Input Integration/Interpretation Module are noun phrases consisting either of [deictic word, adjective, noun] such as "this red car", or of [deictic word, noun] such as "this person." When a voice recognition result comes from the Voice Recognition Module, the MM Input Integration/Interpretation Module searches for noun phrases which fit either form above. If a suitable noun phrase is found, the system tries to resolve the reference expression by integrating it with gesture recognition results.

Figure 5 shows the structure of the Knowledge Base, a semantic network that contains the whole knowledge for solving MM input interpretation problems (see Appendix A for detail). The lower left area in Figure 5 indicates the semantic network of knowledge about verbal expressions. Knowledge about concepts, namely conceptual knowledge, is shown on the right side. Each ellipse in these

areas indicates a class particular knowledge belongs to, and each symbol in the ellipses indicates a knowledge element. Verbal/conceptual knowledge can also be represented as links among certain elements of these classes. The figure demonstrates an example of the reference resolution process, when the user speaks "this red one" while circling on the upper-left picture, namely a blue and a red car placed in a garage. Receiving such a multimodal input, the MM Integration/Interpretation Module performs problem solving by following links between nodes, which means that the system searches for the resolution, and the red car on the right side is determined as the multimodal resolution.

3 Control mechanism for multimodal interpretation

3.1 Requirements

Our objectives are to meet the following two requirements in designing the MM integration/interpretation architecture.

Integration of delayed-arrival data: Recognition result data from certain modalities are likely to arrive at the MM integration/interpretation module after the start of the integration/interpretation process because of differences in the calculation time for the recognition process of each input modality. A framework in which delayed-arrival data is accepted, efficiently re-integrated and interpreted is required.

Elimination of ambiguity in recognition results: Generally, two or more recognition result candidates are received by the MM integration/interpretation module as the input from each modality, since a correct recognition rate of 100% cannot be expected. The MM integration/interpretation module needs to perform its process efficiently, identifying the most plausible candidate for each ambiguous input element.

It will be easier to extend the acceptable modalities of an MMIF if there is a generic MM integration/interpretation architecture. Parsing technology, which is studied in several fields such as natural language analysis, provides an effective basis for MM integration/interpretation. Studies aiming at a generic framework for MM integration/interpretation often utilize parsing technology for natural languages exemplified by the system developed by Koons *et al.* (1993) and MMDCG (Shimazu *et al.*, 1994). When the delayed-arrival data problem occurs, however, almost all of the interpretation (parsing) processes have to be performed again in these systems, since most of them do not have special functions for incremental parsing. Moreover, most of them have to repeatedly perform the interpretation process for the number of MM input candidates, *i.e.*, they are inefficient in coping with the ambiguity of the recognition result of each

modality. A control mechanism of the problem-solving process is needed for solving the above problems. This control mechanism should carry out functions such as "detect and reuse the data which are not affected by delayed-arrival" or "reuse the usable data derived by testing other MM input candidates."

3.2 Problem-solving control with ATMS

The above requirements of the multimodal integration/interpretation process suggest that the ATMS (deKleer, 1986a; deKleer, 1986b) is appropriate for the core module of the controlling mechanism of multimodal integration/interpretation. In other words, the ATMS and an inference system work collaboratively in ATMS-based problem-solving systems. The inference system executes problem-solving and informs the ATMS of its inference process. The consistency among data dealt with by the inference system is managed by the ATMS. The ATMS holds and revises a set of valid assumptions, which is the origin of the inference and data derivation process by the inference system. When a contradiction is encountered, the ATMS computes the set of assumptions responsible for the contradiction by tracing the derivation paths back from the contradiction to that assumptions. When an assumption is denied, the data which rely on it is no longer held and hence are automatically denied by the ATMS. In addition, the ATMS can avoid repeating calculations which have previously been performed. This function improves the efficiency of the inference process.

The information given by the inference system takes the following form:

$$N1, N2, ..., Nk \Rightarrow D,$$

which means that data D derived from a set of data N1,N2,...,Nk. N1,N2,...,Nk is called the justification of D .

The data dealt with in the inference system is classified into three categories, namely premised data, assumed data, and derived data. A premise is a data that is true in any context. An assumed data is the one produced with an assumption that holds independently of any other data. A derived data is the one inferred from other data.

Following each justification back from a certain derived data, one finally reaches a set of assumptions or premises. That is to say, a set of assumptions on which an individual data depends can be calculated. Such a set of assumptions is called an environment. One of the major tasks for the ATMS is to record justifications received from the inference system and to calculate and watch a consistent environment for a set of data. When a contradiction is encountered, the ATMS calculates the no-good environment, which is the cause of the contradiction, and records it in the ATMS (hereafter called the no-good record). Every environment included in the no-good record can be regarded as an inadequate combination of the assumptions. The ATMS maintains the consistency of the inference process by using the no-good record. The inference system selects a new consistent environment which does not include the no-good record

elements and continues inference. A situation in the problem-solving process is called a context, which is defined by the set of the data that hold in that situation. An environment which derives all the data included in the context is called the characteristic environment of the context. When an inconsistency is encountered, the ATMS calculates and records the environment of $\perp$. The inference system ceases solving the problem in the contradictory context and transfers it to a new and consistent characteristic environment. With regard to the nodes which have been derived before that time, the ATMS determines whether each node holds (in) or does not hold (out) in the new characteristic environment. A new context is composed of a set of "in" nodes. [1]

3.3 MM interpretation based on hypothetical reasoning

The methodology to efficiently derive MM interpretation results which meets the two requirements mentioned in Section 3.1 is established by regarding the MM integration/interpretation process as a control of the problem-solving based on the ATMS. In this case, the requirements mean integration of delayed-arrival data and elimination of ambiguity in recognition results.

Figure 6 shows the detailed configuration of the MM input integration/interpretation architecture which we have constructed. The recognition module of each modality, *e.g.*, the Voice Mode Recognition Module, accepts user's input into the corresponding modality, performs its recognition process, and informs the MM Input Integration/Interpretation Module of the MM input element that contains the obtained recognition result, the original input time and the unique ID of the element. An MM input element generally contains two or more recognition result candidates of the corresponding modality. The expression forms of MM input elements differ for each modality.

The Knowledge Base is a semantic network that contains all the knowledge required to solve MM input interpretation problems, *e.g.*, the definitions of target objects each of which can be referred to by users, target classes to which target objects belong, the attributes and values of target objects, and word surfaces which express them. Figure 4 summarizes the process running in the MM Input Integration/Interpretation Module. The module first obtains a set of MM input elements, called an MM input (MMI), by asynchronously receiving MM input elements from the recognition modules of each modality and determining the set of MM input elements which should be integrated. The MM Input Integration/Interpretation Module next generates a set of MMI candidates. Each MMI candidate is generated by selecting one candidate at a time from each MM input element which composes the MMI. Then, MMI candidates are analyzed one by one by referring to domain knowledge in the Knowledge Base until a plausible interpretation is obtained. This is called the MM input interpretation process. When the interpretation of an MMI is identified, it is sent to the

[1] It is premised not on parallel type of ATMS, *i.e.*, Basic ATMS (deKleer, 1986a), but on gone-round type, *i.e.*, DDB-Guided ATMS (deKleer, 1986b).

Application Module, namely the know-how retrieving system. The Application Module accepts the MMI as user's input and generates output information. The MM Input Integration/Interpretation Module receives the output demand from the Application and performs feedback to the user through an output-mode module. In this way, the user is able to interact with the Application Module by utilizing various modalities.

Our current system supports two input modalities, i.e., speech and gesture modalities. The Voice Recognition Module accepts continuous speech of a sentence, and generates a word-lattice. Then it provides the MM Input Integration/Interpretation Module with n-best sentence candidates of the utterance, which are generated by applying grammatical constraints to the lattice. Each candidate is composed of the sentence, that is, a list of words, and its score. The Gesture Recognition Module accepts referring gestures such as pointing or circling on a touch screen, and also provides the MM Input Integration/Interpretation Module with n-best candidates of referred object(s). Each candidate is a set of referable objects displayed on the screen. Gesture recognition result candidates contain neighborhood objects with lower score in addition to objects directly touched or surrounded.

The number of MMI candidates, which is obtained from a multimodal input composed of one speech input and one gestural input, is the cross-product of spoken candidates and gestural candidates. Each MMI candidate is initially scored, taking into account both speech and gestural scores. Then the candidates are sorted in descending order and interpreted in that order.

The ATMS is notified of details of the MM input integration and interpretation process, and records them. When a contradiction is encountered as a result of failure of MM input analysis *etc.*, or when data managed by the ATMS reach a certain state, *e.g.*, when the analysis of an MMI candidate is completed, the Environment Management Module generates a new environment for problem-solving and instructs the ATMS to switch to it. The MM Integration/Interpretation Module continues problem-solving in the new environment.

The integration and interpretation process is governed by production rules and a rule interpreter (see Appendix). The rule interpreter evaluates the condition part of each rule like PROLOG, i.e., it searches the unifiable knowledge base element or unifiable and "*in*" ATMS node for each element in the condition. If an interpretation, i.e., a set of unifications, satisfies all the conditions of a rule, the rule is executed.

When a multimodal input containing referring expressions which should be solved is detected, the rule interpreter is given the following goal that is composed of the list of MMI elements and unique id of the MMI, e.g., MMI#1:

$$referent\_object(\_ObjectList, MMI\#1)$$

The rule interpreter selects an MMI candidate from the sorted list of MMI candidates one by one. The interpreter tests the candidate by repeating the following cycle:

1. When the current environment is contradictory, the interpreter attempts to apply rules in the Contradiction Resolution Rule Set (see Appendix E) to resolve the contradiction. If any of the rule set has not fired, i.e., contradiction resolution has failed, the interpretation of the current MMI candidate is terminated.

2. The interpreter attempts to apply rules in the Contradiction Detection Rule Set (see Appendix D) to check whether the current environment is contradictory or not.

3. The interpreter attempts to apply rules in the Multimodal Referent Resolution Rule Set (see Appendix C).

unless more than one of the following three conditions is satisfied:

- the given goal is satisfied,
- no rules have been executed in a cycle,
- contradiction resolution fails.

If more than one of the rules in the corresponding rule set in a certain step has been executed by the end of the step, the above cycle is re-started from the first step.

When an MMI candidate satisfies the given goal, the final score of the candidate is calculated by subtracting the penalty, which is calculated from the interpretation cost of the candidate and other heuristics, according to the initial score. When the interpretation of a candidate finishes, i.e., success or failure, the next MMI candidate is selected from the list of candidates and tested as described above. If the initial score of the newly selected candidate is below the given threshold lower than the minimal final score of successful candidates, interpretations of subsequent candidates are canceled. When any candidate of the list has not satisfied the given goal, the multimodal integration/interpretation process itself ends in failure.

3.3.1 Treatment of delayed arrival

At the beginning of the MM input integration process, the initial value of $\mathcal{S}$, which is the entire set of MM input elements, is set as empty. Then, the following operations are repeatedly applied until either (1) MM input interpretation succeeds, or (2) MM integration reaches time-out, *i.e.*, no recognition result of any modalities has arrived by a certain time.

1. When the recognition result of one of the modalities, namely an MM input element, has newly arrived, the ID of the element is appended to $\mathcal{S}$.

2. If $\mathcal{SS}$, an arbitrary subset of $\mathcal{S}$, has not yet been analyzed, the module assumes $\mathcal{SS}$ as MMI and performs MM analysis. [2]

3. $\mathcal{S}$ is set as empty if a time-out occurs. The module outputs $\mathcal{SS}$ and removes $\mathcal{SS}$ from $\mathcal{S}$ when MM input analysis has been successful. [3]

Each assumption generated in the MMI integration processes is represented by one of the following two forms:

integrate($\mathcal{SS}$, Mmi#) It is assumed that all the MMI elements in the list of MMI element $\mathcal{SS}$ (1st argument), which is the subset of the entire set of MMI elements until then, are integrated as an MMI. An MMI ID (2nd argument) is assigned to the integration.

no_omission($\mathcal{M}$, $\mathcal{SS}_m$, Mmi#) It is assumed that only MMI elements in the list of MMI element $\mathcal{SS}_m$(2nd argument) are the MMI elements of the modality $\mathcal{M}$(1st argument) as the MMI of MMI ID Mmi#(3rd argument). This type is assumed for each input modality that is linked to the system.

Let us assume an MMIF that accepts both voice inputs in natural language, vIn, and referring gesture inputs, gIn, *e.g.*, circling and pointing, on a touch screen. Assume a case in which a speech recognition result V#1 is first obtained from voice modality, because it has taken time to recognize gIn. The system assumes only V#1 as MMI elements for an MMI, and the following three assumptions are generated and added to the current environment by applying Rule C.1:

integrate([V#1], MMI#1)
no_omission(vIn, [V#1],MMI#1)
no_omission(gIn, [], MMI#1)

A data expressing MMI is next derived by the following justification, and control is handed over to the MMI interpretation process (see Rule C.5): [4]

$$\begin{array}{ll} & \textbf{integrate}([\text{V\#1}], \text{MMI\#1}) \\ \& & \textbf{no\_omission}(\text{vIn}, [\text{V\#1}], \text{MMI\#1}) \\ \& & \textbf{no\_omission}(\text{gIn}, [], \text{MMI\#1}) \\ \Rightarrow & \mathit{integrated\_input}([\text{V\#1}], \text{MMI\#1}) \end{array}$$

An MMI candidate is selected in due order and analyzed in the MM input interpretation process. In this process, an assumption for each MMI element of the MMI candidate is generated, and the interpretation process advances in such a way that derivations are made from these assumptions. The ATMS is

[2] Efficiency of generation and tests of $\mathcal{SS}$ can be improved by applying heuristics, for example by preferentially unifying neighboring time stamps of MMI elements.

[3] In order to prevent MMI elements which have not been used for analysis and which are noises in most cases from accumulating in $\mathcal{S}$, MMI elements older than a predetermined period are also removed from $\mathcal{S}$.

[4] Although input-time information is included in the predicates, it is not described here for simplification.

notified of the detailed process and stores it. The process is shown in Section 3.3.2 in detail using an example.

When the analysis of a certain MMI candidate is completed and the next MMI candidate is selected, the Environment Management Module calculates and sets up a new environment appropriate for analyzing the new MMI candidate. The data which are assumed or derived in the former interpretation process become available for the MMI Integration/Interpretation Module by the operation. Thus, the system can search for the MMI candidate that can draw an appropriate interpretation, efficiently employing data generated in its former problem-solving.

Let us assume that the delayed input G#1 arrives from the gesture modality, gIn, while MMI interpretation is in progress in the above example. The following two data are assumed:

integrate([V#1,G#1], MMI#1)
no_omission(gIn, [G#1], MMI#1)

Consequently, the following two contradictions are encountered (see Rule D.3 and Rule D.2):

$$\begin{array}{ll} & \textbf{integrate}([\text{V\#1}, \text{G\#1}], \text{MMI\#1}) \\ \& & \textbf{no\_omission}(\text{gIn}, [], \text{MMI\#1}) \\ \Rightarrow & \bot \end{array}$$

$$\begin{array}{ll} & \textbf{integrate}([\text{V\#1}], \text{MMI\#1}) \\ \& & \textbf{integrate}([\text{V\#1}, \text{G\#1}], \text{MMI\#1}) \\ \Rightarrow & \bot \end{array}$$

In order to resolve these contradictions, the Environment Management Module generates a new environment for problem-solving that does not contain neither **no_omission**(gIn, [], MMI#1) nor **integrate**([V#1], MMI#1) and contains **integrate**([V#1, G#1], MMI#1) (see Rule Rule E.4 and Rule E.2). The ATMS is notified of the newly generated environment and shifts the current environment to it. By the transfer, the data which should not stand any more are automatically removed from the context, and data that are not shaded remain in the new context, so that they can be referred to without re-calculation, (*e.g.*, only data derived from the assumption **no_omission**(vIn, [V#1], MMI#1) is calculated). The MMI integration/interpretation process continues integrating delayed-arrival data in this way. The state of reasoning data that can be reused is saved in this process.

3.3.2 Treatment of ambiguity in recognition results

A more detailed MM interpretation process, which follows the MM integration, is described here by referring to another example to demonstrate that the process can efficiently determine the result, *i.e.*, the referent(s), coping with the ambiguities in recognition results in each modality. Although the user's utterance and the system's internal symbols are originally written in Japanese,

we represent them in English here for readers' convenience. Figure 7 shows a screen copy of the reference resolution sub-system, after the user has spoken the sentence "What is this person doing?" in Japanese while touching the point indicated by "x" on the touch screen. [5] The task of the MMIF in this case is to replace the referring expression "this person" with the name of the referred target object "Ms. Nakayama", to construct the resolved sentence "What is Ms. Nakayama doing?", and to send the sentence to the application program. The target object corresponding to "Ms. Nakayama" is not the first-place candidate for the gesture recognition result, but the second-placed or lower in this case, because the object corresponding to the touched point is a notice board. MM interpretation would fail, if the system tested only the first-place candidates for both voice and gesture modality. Therefore, candidate pairs which contain the second-place candidate or lower have to be searched for. The MMI Interpretation Process generates MMI candidates from a given MMI and tests them one by one until a plausible candidate is found.

In such a case, the following three assumptions are first generated (see Rule C.1):

integrate([V#1,G#1], MMI#1)
no_omission(vIn, [V#1], MMI#1)
no_omission(gIn, [G#1], MMI#1)

and the following node is derived from them by applying Rule C.5:

integrated_input([V#1,G#1], MMI#1)

On testing a certain MMI candidate, an assumption is generated for each modality in which elements of the MMI candidate are contained. Data derived from the assumptions differ depending on the modality. Let us assume that the speech recognition result V#1, which is composed of n-best sentences, i.e., [(what, is, this:md#2, person:nn#7, doing), (whom, should, I, send, this:md#2, pamphlet:nn#11)], [6] is obtained as a voice recognition result. Wh also assume that G#1, which is a set of candidates of a location indicated by the user's pointing gesture, is composed of [Location#20064, Location#20016, Location#20032]. That is, the notice board, the female person (Ms. Nakayama), and the desk in Figure 7 in this order is obtained as a gesture recognition result. [7] The first MMI candidate, namely [(this:md#2, person:nn#7), Location#20064], is selected and the following three assumptions are generated and added to the environment to analyze the MMI candidate by applying Rule C.2, Rule C.3, and Rule C.4, respectively:

vIn_sentence([md#2,nn#7],V#1)
gesture_location(Location#20064, G#1)

[5] The "x" is not displayed on the screen. It is indicated for readers' convenience.

[6] The representation is simplified to focus on referring expressions in the sentences, although each word takes the form of "(word surface):(part of speech)#(word id)" in fact. Hereafter, the interpretation process is described in this simplified manner.

[7] A gesture recognition result is obtained by searching the set of pre-defined knowledge about locations of objects which is stored in the knowledge base described above.

deictic_word(G#1, md#2)

Then the following data are separately derived from the assumption **vIn_sentence**([md#2,nn#7],V#1) by applying Rule C.6:

vIn_word(md#2,V#1),
vIn_word(nn#7, V#1),

By analyzing the words, the following data are also derived from the same assumption **vIn_sentence**([md#2,nn#7],V#1):

modify(md#2, nn#7)
object_noun(nn#7, V#1)
verbal_modifier_noun([nn#7], V#1)

The interpretation process for the MMI candidate progresses in this way, and target object candidates that can be derived from the voice candidate, namely, three "persons" which are labeled male#202, male#203, and female#204, are obtained by Rule B.2, Rule C.9, and Rule C.14:

$$\begin{array}{ll} & \text{expression\_class}(\text{nn\#7}, \text{person}) \\ \& & \text{cc\_relation}(\text{male\_person}, \text{is}-\text{a}, \text{person}) \\ \& & \text{cc\_relation}(\text{female\_person}, \text{is}-\text{a}, \text{person}) \\ \& & \text{class\_object}(\text{male\_person}, [\text{male\#202}, \text{male\#203}]) \\ \& & \text{class\_object}(\text{female\_person}, [\text{female\#204}]) \\ \Rightarrow & \mathit{object\_of\_noun}(\text{nn\#7}, [\text{male\#202}, \text{male\#203}, \text{female\#204}]) \end{array}$$

$$\begin{array}{ll} & \mathit{object\_noun}(\text{nn\#7}, \text{V\#1}) \\ \Rightarrow & \mathit{singular\_object}(\text{nn\#7}, \text{V\#1}) \end{array}$$

$$\begin{array}{ll} & \textbf{vIn\_sentence}([md\#2, nn\#7], V\#1) \\ \& & \mathit{object\_of\_noun}(\text{nn\#7}, [\text{male\#202}, \text{male\#203}, \text{female\#204}]) \\ \Rightarrow & \mathit{vIn\_object}(\text{V\#1}, [\text{male\#202}, \text{male\#203}, \text{female\#204}]) \end{array}$$

On the other hand, the analysis of the candidate of the directly indicated location, namely gesture_location(Location#1, G#1), progresses and the candidate of the indicated object, notice_board#1, is obtained by the following derivation (see Rule C.11 and Rule C.12):

$$\begin{array}{ll} & \mathit{integrated\_input}([V\#1, G\#1], MMI\#1) \\ \& & \textbf{deictic\_word}(G\#1, md\#2) \\ \& & \mathit{vIn\_word}(md\#2, V\#1) \\ \Rightarrow & \mathit{deixis}(\text{G\#1}, \text{md\#2}, \text{V\#1}) \end{array}$$

$$\begin{array}{ll} & \textbf{gesture\_location}(\text{Location\#20064}, \text{G\#1}) \\ \& & \text{location\_object}([\text{notice\_board\#1}], \text{Location\#20064}) \\ \& & \mathit{deixis}(\text{G\#1}, \text{md\#2}, \text{V\#1}) \\ \Rightarrow & \mathit{deixis\_object}(\text{G\#1}, [\text{V\#1}], [\text{notice\_board\#1}]) \end{array}$$

These derivation processes are notified to the ATMS and are stored in it. Here, the system tries to resolve the referent by integrating the data generated above. However, the interpretation of the MMI candidate fails, since

notice_board#1, which is derived from gIn, is not contained in [male#202, male#203, female#204], which are obtained from vIn. Then, the following contradiction is derived (see Rule D.4):

$$
\begin{array}{ll}
 & deixis(G\#1, md\#2, V\#1) \\
\& & vIn\_object(\mathrm{V}\#1, [\mathrm{male}\#202, \mathrm{male}\#203, \mathrm{female}\#204]) \\
\& & gIn\_object(\mathrm{G}\#1, [\mathrm{notice\_board}\#1]) \\
\Rightarrow & \bot
\end{array}
$$

In order to resolve the contradiction, the following three assumptions which represent the MMI candidate being tested are removed from the current environment (see Rule E.3):

vIn_sentence([md#2,nn#7],V#1)
gesture_location(Location#20064, G#1)
deictic_word(G#1, md#2)

Data derived from these assumptions are also removed from the context, *i.e.*, they become "out", with the change of the environment. Then the location indicated by gesture input is changed from the first order candidate Location#20064, which is the symbol of the location indicating the notice board, to the second order candidate Location#20016, namely also indicating Ms. Nakayama, and the next MMI candidate [what, is, this:md#2, person:nn#7, doing, Location#20016] is selected. In this way, the following data are assumed and added to the environment:

vIn_sentence([md#2, nn#7],V#1)
deictic_word(G#1, md#2)

In this case, the following two data were assumed once in the past interpretation, since the noun phrase to be resolved that is selected for the voice candidate is the same as that of the last MMI candidate (see Rule C.2, Rule C.3, and Rule C.4):

vIn_sentence([md#2, nn#7],V#1)
gesture_location(Location#20016, G#1)
deictic_word(G#1, md#2)

Thus the following data, which have already been derived from the above assumptions, automatically return to the context, so that there is no need to re-interpret the voice modality candidate:

deixis(G#1, md#2, V#1)
object_singular(nn#7, V#1)
vIn_object(V#1, [male#202,male#203, female#204])

The assumption **gesture_location**(Location#20016, G#1) is generated and the candidate, a directly indicated object, is obtained by the following deriva-

tion in the interpretation of the MMI candidate (see Rule C.12):

$$\begin{array}{ll} & \textbf{gesture\_location}(\text{Location\#20016}, \text{G\#1}) \\ \& & \text{location\_object}([\text{female\#204}], \text{Location\#20016}) \\ \& & \textit{deixis}(\text{G\#1}, \text{md\#2}, \text{V\#1}) \\ \Rightarrow & \textit{deixis\_object}(\text{G\#1}, [\text{female\#204}]) \end{array}$$

Then female#204, which is the symbol of the referent indicating Ms. Nakayama, is readily resolved without re-interpreting the voice modality candidate (see Rule C.16):

$$\begin{array}{ll} & \textit{voiceIn\_object}(\text{V\#1}, [\text{male\#202}, \text{male\#203}, \text{female\#204}]) \\ \& & \textit{deixis\_object}(\text{G\#1}, [\text{female\#204}]) \\ \& & \textit{integrated\_input}([\text{V\#1}, \text{G\#1}], \text{MMI\#1}) \\ \& & \textbf{vIn\_sentence}([md\#2, nn\#7], V\#1) \\ \& & \textit{singular\_object}(\text{nn\#7}, \text{V\#1}) \\ \& & \textbf{gesture\_location}(\text{Location\#20016}, \text{G\#1}) \\ \Rightarrow & \textit{referent\_object}([\text{female\#204}], \text{MMI\#1}) \end{array}$$

The interpretation of the MMI candidate succeeds and the resolved sentence "What is Ms. Nakayama doing?" is obtained by replacing the referring expression "this person" with "Ms. Nakayama." Then the sentence is sent to the application program, so that the application is able to interact with the user without being concerned with referring expressions.

In the above example, the number of queries to the Knowledge Base Module to test the entire set of MMI candidates is reduced to about 1/5 comparing with that of the previous version of our multimodal referent resolution system that does not employ the ATMS-based mechanism. The number of times of set calculations, *e.g.*, intersection or union, is also reduced to about one half. The effectiveness of the method will be increased by the additions of supported modalities or by increase of alternatives in recognition results of each modalities.

4 Discussions

4.1 CG agent interface

The developed multimodal interface agent system has been displayed and demonstrated at several exhibitions, including "Tommorrow21 Toshiba Technology Exhibition" which was attended by over 60,000 people. We have created a humanoid user interface with a more realistic face (Suzuki *et al.*, 1996), which moves and utters in real time. We have found the following effects in humanoid agent interfaces:

Relaxation Audience's reluctance to interact with the systems seems to be reduced when the agent appears on the screen.

Gaze Control The user can recognize to whom he/she should talk and can naturally communicate with the computer gazing at the agent.

Furthermore, the following points have been found important:

Lifelikeness It is important that the agent looks alive. For instance, if the agent does not react at all until the voice recognition result comes, users become uneasy. Many members of the audience commented that they could not determine whether or not the agent was listening. Making the agent slightly, continuously and randomly move is effective for lifelikelyness. Blinking of the agent's eyes and face movements are especially effective. In a case of speech recognition failure, our agent says "Pardon?" with a puzzled expression on his face and a hand gesture, which corresponds to the status "pardon" in Figure 2. We have found that such clearly visible non-verbal messages are effective in informing the user about the system's internal status.

Balance Control The agent's looks should be balanced with other elements, *e.g.*, the quality of the synthesized voice and the reality of movement. Realistic voices and movements, high ability and intelligence are expected if an agent has a highly realistic appearance. Therefore, we have given the interface agent a rather comical appearance to avoid audience's disappointment.

4.2 Generality and efficiency of the method

The proposed multimodal input integration/interpretation method is designed to be domain independent. That is, the system can work on various domains only by changing domain knowledge represented in the semantic network contained in the Knowledge Base Module. The generality of the method has been examined by building two systems each of which works in different domains. One is an object retrieval system for maps that accepts sentences such as "Cheap hotels around here," and another is the interface agent system mentioned above. Both have successfully solved their own problems, *i.e.*, detected referents without changing the multimodal input integration/interpretation program. They differ from each other with respect to the domain definitions in the knowledge base including the set of locations of referable objects, and the vocabulary set for the voice recognition module.

Cohen *et al.* presented a method of multimodal integration/interpretation based on typed feature structure unification, and developed QuickSet, a framework for multimodal interaction(Cohen *et al.*, 1997; Johnston *et al.*, 1997; Johnston, 1998). Their approach is general and well-formulated on the basis of natural language processing. To cope with recognition ambiguity, it is required to examine each cross-product of recognition alternatives, e.g., spoken and gestural alternatives

(Johnston *et al.*, 1997). Johnston (1998) recently introduced the unification-based multimodal parser dealing with such kind of ambiguity. It is based on the chart parsing method (Kay, 1980), and prunes down the number of candidates by eliminating overlapping complete edges of lower probability from the edge list to be executed when higher probability complete edges are selected. Although our system generates the cross-product number of candidates as subjects of interpretation, it is only the worst case to examine all the candidates. By pruning analyses of hopeless (low-scored) MMI candidates as described in Section 3.3, the number of analyzed MMI candidates is significantly less than the worst case. Furthermore, the computational complexity is greatly reduced by the effectiveness of cache mechanism of the past interpretation data provided by the ATMS even in the worst case. That is, as the interpretation progresses to the lower-scored candidate, less rules need to be fired for the interpretation.

Our current system accepts only the task of multimodal referent resolution and does not support other types of multimodal command, e.g., generation of an object. Our architecture for rule interpretation is based on *typeless* unification of atomic sentences like PROLOG, which means that the inference process is controlled by applying limited constraints, i.e., predicate names and location of arguments. To increase supporting command types, a *typed* approach similar to QuickSet would be needed.

Besides, a set of objects can be selectable as a candidate of recognition result provided by our Gesture Recognition Module, if the set is preliminarily given as one of the subjects of referents. Consider the case in which one gestural input is obtained as shown in the example of Figure 7. If the noun phrase part of the speech candidate is "this person", "Ms. Nakayama" is obtained as the resolution result candidate. The gestural candidate that contains all the characters in the picture matches, if speech is "these persons". [8]

Oviatt *et al.* (1997) empirically determined that the most (43%) multimodal inputs were *not* simultaneous but subsequent ones, i.e., gesture first and then speech, among all the collected multimodal inputs each of which contain deictic term(s). Our system could correctly integrate and interpret subsequent multimodal input by changing the constraint of time stamps, which is referred to decide whether certain delayed inputs should be integrated or not. This provides evidence of our method's ability to cope with delayed-arrival data.

We have also found experimentally that the scoring algorithms of gesture recognition modules should be changed to some extent according to the domains for higher efficiency. Moreover, modality dependence remains in the formalism such as the grain sizes of assumptions and their controlling policy.

Although the proposed method itself is generic, the control method requires many portions depending on modalities. The developed system is controlled to resolve its problems in such a way that greater importance is given to speech

[8] To differentiate "person" and "persons" is difficult in English speech recognition. In Japanese, however, it is not so difficult, i.e., "person" in English corresponds to "hito" and "persons" to "hitotachi", and both "this" and "these" correspond a same word "kono".

recognition results than gesture recognition results. Such control allows the user to roughly indicate the desired object(s) by touch gesture and to exempt him/her from the precise indication. We believe that it is one of the most important advantages of media complementation to integrate (appropriate) candidates which are unfortunately placed in the lower ranks. On the other hand, the number of good effects of media complementation differs among media. In our case, voice recognition plays a more important role for the gesture recognition module than the gesture does, *i.e.*, lower rank candidates of speech recognition can hardly be rescued.

4.3 Grain size of assumptions

In general, there is a trade-off between the reusability of past problem-solving data and the degree of the side effects of changing the status of each assumption. If reusability is improved, *i.e.*, if the coverage of each assumption is enlarged, side effects will increase and testing of promising candidates will be avoided. Moreover, problem-solving repetition increases the number of ATMS nodes and the number of contradictory records. That causes certain overheads for using ATMS, such as node searches and contradiction evaluation.

Such problems are inevitably caused by the minimality and monotonicity of contradictions which are inherent characteristics of ATMS. Therefore, it is necessary for advanced HI systems to implement a mechanism, like "intelligent oblivion" of human beings. For example, the grain sizes of assumptions in human's inferences change dynamically and past assumptions disappear within an appropriate period.

4.4 Treatment of time lapse

The ATMS has been employed for problems in various fields because of its flexibility and usefulness since it was proposed by deKleer. Although there have also been many attempts regarding its utilization in the field of HI, *e.g.*, to extract linguistic information from speech recognition results with ambiguity (Nishioka *et al.*, 1991), most of them have not led to practical applications.

The ATMS is able to construct/reproduce any problem-solving context regardless of the reasoning sequence. It expands the flexibility of the reasoning sequence of PS. In solving a certain scale of problems, *i.e.*, tasks, it is often necessary to refer to former problem-solving data that was updated previously. If such problems are dealt with by hypothetical reasoning, difficulties and complexities will arise in managing context. This has barred the application of ATMS to practical tasks.

The proposed framework can deal with data whose validity is transformed in the following way. This function of the framework makes it possible to overcome the computational complexity of the past problem-solving data: [9]

[9] This method has not yet been implemented.

1. All the queries to the Knowledge Base Module are managed through the Environment Management Module.

2. The Environment Management Module generates an ATMS assumption which gives input time information to each reply from the Knowledge Base Module, and memorizes assumption IDs in its internal history.

3. When the knowledge base is updated, the Knowledge Base Module enumerates all the former queries that are influenced by the update and notifies other modules. The Environment Management Module assigns invalid flags to the corresponding internal records.

4. The Environment Management Module searches its internal history first when a query is received from the MMI Integration/Interpretation Module. If the same inquiry exists in the history and is valid, the ID of the corresponding ATMS node is replied. When there is no valid node in the history, the Environment Management Module queries the Knowledge Base Module.

The above function of, so to speak, a cache memory of the knowledge base enables problem-solving not only on the basis of the newest knowledge but also by referring to the knowledge of the former context. Moreover, a snapshot of each problem-solving context can be reproduced.

5 Concluding remarks

This paper has proposed a new multimodal input integration/interpretation method for MMIF based on hypothetical reasoning. Using this technology, the method is able to cope with delayed-arrival data and ambiguities in recognition results. Furthermore, it efficiently performs re-calculation. Although the proposed method itself is generic, the control method requires many portions depending on modalities. Generic control knowledge needs to be separated. The effectiveness of non-verbal messages are qualitatively evaluated by developing and testing a multimodal interface agent system employing the proposed method.

Building a practical MMIF on the basis of the experimental reference resolution system is a subject of future work in conjunction with the enhancement of acceptable verbal expressions and of acceptable non-verbal modalities.

References

Bolt, R.A. 1980. Put-that-there: Voice and gesture at the graphic interface. *Computer Graphics*, Vol.14, No.3, 262–270.

Cohen, P.R. 1994. Natural language techniques for multimodal interaction. *Trans. IEICE Japan*, J77-D-II, No.8, 1403–1416.

Cohen, P.R., Johnston, M., McGee, D., Oviatt, S., Pittman, J., Smith, I., Chen, L., and Clow, J. 1997. QuickSet: Multimodal interaction for distributed applications. In *Proc. ACM Multimedia'97*, New York, ACM Press.

deKleer, J.D. 1986a. An assumption-based truth maintenance system. *Artificial Intelligence*, Vol.28, 127–162.

deKleer, J.D. 1986b. Back to backtracking, controlling the ATMS. In *Proc. AAAI-86*, 910–917.

Finin, T., *et al.* 1997. KQML as an agent communication language. In *Software Agents*, ed. J. Bradshaw, 291–316, MIT Press.

Johnston, M., Cohen, P.R., McGee, D., Oviatt, S., Pittman, J., and Smith, I. 1997. Unification-based multimodal integration. In *Proc. ACL'97*, 281–288, Morgan Kaufmann.

Johnston, M. 1998. Unification-based multimodal parsing. In *Proc. ACL'98*, Morgan Kaufmann.

Kay, M. 1980. Algorithm schemata and data structures in syntactic processing. In *Readings in Natural Language Processing*, ed. B.J. Grosz, *et al.*, 35–70, Morgan Kaufmann.

Kobsa, A. 1986. Combining deictic gesture and natural language for referent identification. In *Proc. COLING86*, 356–361.

Koons, D.B., Spaarrell, C.J, and Thorisson, K.R. 1993. Integrating simultaneous input from speech, gaze, and hand gestures. In *Intelligent Multimedia Interfaces*, ed. M.T. Maybury, 267–276.

Maes, P. 1997. Agents that reduce work and information overload. In *Software Agents*, ed. J. Bradshaw, 145–164, MIT Press.

Maybury, M.T. (Ed). 1993. *Intelligent Multimedia Interfaces*. MIT Press.

Maybury, M.T. 1994. Research in multimedia and multimodal parsing and generation. *Journal of Artificial Intelligence Review*, Vol.8, No.3.

Nakayama, Y., Manabe, T., and Takebayashi, Y. 1997. Development of a knowledge/information sharing system "Advice/Help on Demand". In *Proc. Interaction97*,Information Processing Society of Japan,103–110 (in Japanese).

Namba, Y., Tano, S., and Kinukawa, H. 1997. Semantic analysis using fusionic property of multimodal data. *Trans. Information Processing Society of Japan*, Vol.38, No.7, 1441–1453 (in Japanese).

Neal, J.G.and Shapiro, S.C. 1991. Intelligent multi-media interface technology. In *Intelligent User Interfaces*, ed. J.W. Sullivan, *et al.*, ACM Press.

Nishioka, S., Kakusho, O., and Mizoguchi, R. 1991. A generic framework based on ATMS for speech understanding system. *Trans. IEICE Japan*, Vol.E74, No.7, 1870–1880.

Oviatt, S., DeAngeli, A., and Kuhn, K. 1997. Integration and synchronization of input modes during multimodal human-computer interaction. In *Proc. CHI'97*, 415–422, ACM Press.

Shimazu, H., Arita, S., and Takashima, Y. 1994. Multi-modal definite clause grammar (MM-DCG). In *Proc. COLING-94*, 832–836.

Stock, O. 1991. Natural language and exploration of an information space, ALFresco interactive system. In *Proc. IJCAI91*, 972–978.

Suzuki, K., Yamaguchi, O., Fukui, K., Tanaka, E., Kuratate, T., and Matsuda, N. 1996. A multi-modal humanoid user interface (mmHUI) —An experimental narration agent (Rachel)—. *SIG-HI, Information Processing Society of Japan*, 96-HI-69,47–53 (in Japanese).

Takebayashi, Y., Tsuboi, H., Kanazawa, H., Sadamoto, Y., Hashimoto, H, and Shinchi, H. 1993. A real-time speech dialogue system using spontaneous speech understanding. *Trans. IEICE Japan*, Vol.E76-D, No.1, 112–120.

Appendix

Rule sets which are executed during multimodal integration/interpretation are described here. Some of the rules are implemented as C++ program codes. Both common primitive predicates in PROLOG, e.g., "member", "append", and "bagof", and generic set operations, e.g., "intersection" and "union", are described in the rules without explanation. Each rule is tested and executed in all applicable interpretations in each execution cycle.

Atomic sentences whose relation constants are written in **bold** are **assumption ATMS nodes**, and those in *italic* are *derived nodes*. The term "**PREMISE**" in rules in Appendix means generation of a premise node of the designated atomic sentence, and "**ASSUME**" means generation of an assumption node. "**DERIVE**" means derivation of the designated atomic sentence justified by all ATMS nodes in the condition field of the rule.

A Knowledge Base Structure

The domain knowledge is stored in the Knowledge Base. Each knowledge has one of the followings, and whole knowledge assertions constructs a semantic network.

- expression-npr(_ProperNounSurface,_ExpressionId)

  Each knowledge of this type links surface string of a proper noun word and its expression-id.
  ex.) expression-npr("Ms.Nakayama", npr#13)

- expression-npn(_ProNounSurface,_ExpressionId)

  Each knowledge of this type links surface string of a pronoun word and its expression-id.
  ex.) expression-npr("one", npn#2)

- expression-nn(_NounSurface,_ExpressionId)

  Each knowledge of this type links surface string of a noun word and its expression-id.
  ex.) expression-nn("person", nn#41)

- expression-mf(_ModifierFeatureSurface,_ExpressionId)

  Each knowledge of this type links surface string of a word that modifies features of objects and its expression-id.
  ex.) expression-mf("colored", mf#3)

- expression-mv(_ModifierValueSurface,_ExpressionId)

Each knowledge of this type links surface string of a word that modifies values of objects and its expression-id.
ex.) expression-mv("red", mv#9)

- expression-md(_ModifierDeixisSurface,_ExpressionId)

  Each knowledge of this type links surface string of a deictic word and its expression-id.
  ex.) expression-md("this", md#2)

- class(_ClassName)

  Each knowledge of this type declares a class of a concept.
  ex.) class(car)

- instance(_ClassName,_InstanceName)

  Each knowledge of this type declares an object of a certain class.
  ex.) instance(female, female#1)

- cc-relation(_ClassName1,_RelationName,_ClassName2)

  Each knowledge of this type declares relations between the specified two classes.
  ex.) cc-relation(female, is-a, human)

- expression-class(_ExpressionId,_ClassName)

  Each knowledge of this type links an expression and a class.
  ex.) expression-class(nn#41, human)

- expression-feature(_ExpressionId,_FeatureName)

  Each knowledge of this type links an expression and a feature.
  ex.) expression-feature(mf#3, color)

- expression-value(_ExpressionId,_ValueName)

  Each knowledge of this type links an expression and a value.
  ex.) expression-feature(mv#9, red)

- expression-object(_ExpressionId,_ObjectId)

  Each knowledge of this type links an expression and an object.
  ex.) expression-object(npr#13, female#1)

- class-feature(_ClassName,_ListOfFeatureName)

  Each knowledge of this type declares the set of features that each instance of the class has.
  ex.) class-feature(car, [size,color,age])

- object-feature-value(_ObjectId, _FeatureName, _ValueName)

  Each knowledge of this type declares the value of specified feature of the specified object.
  ex.) object-feature-value(notice_board#1, color, white)

- location-object(_ListOfObjectId,_LocationId)

  Each knowledge of this type declares the set of objects located on/in the specified location.
  ex.) location-object([female#204], Location#20016)

B Static Knowledge Generation Rule Set

When the system is started, the following rules are applied to all applicable knowledge-base elements.

[Rule B.1]

if expression_value(_ValueWordId, _ValueName) **and**
bagof(target_object_feature_value(_ObjectId, _FeatureName, _ValueName), _ObjectId, _LObjectOfValue)
then **PREMISE** object_of_value(_ValueWordId, _LObjectOfValue)

[Rule B.2]

if expression_class(_NounWordId, _ClassExp) **and**
collect all leaf classes which link to _ClassExp into _LLeafClass **and**
member(_LeafClass, _LLeafClass) **and**
class_object(_LeafClass, _LObjectOfClass)
then **PREMISE** object_of_noun(_NounWordId, _LObjectOfClass)

C Multimodal Referent Resolution Rule Set

[Rule C.1]

if the system decides that all MMI elements in _LMmiElementId are to be a multimodal input **and**
the set of MMI elements from gesture modality is _LGinId **and**
the set of MMI elements from voice modality is _LVinId
then **ASSUME integrate**(_LMmiElementId, _MmiId)
ASSUME no_omission(gIn, _LGinId, _MmiId)

ASSUME no_omission(vIn, _LVinId, _MmiId)

[Rule C.2]

if the system selects the candidate of sentence as _LWordId, list of word id, in _VInId, the MMI element in the current MMI
then **LET** all "*in*" **vIn_sentence**(_, _) be *out* **and**
ASSUME vIn_sentence(_LWordId, _VInId)

[Rule C.3]

if the system selects the candidate of referred location as _LocationId, in _GInId, the MMI element in the current MMI
then **LET** all "*in*" **gesture_location**(_, _) be *out* **and**
ASSUME gesture_location(_GInId, _LocationId)

[Rule C.4]

if the system selects the referring word, _DeicticWordId, such as "this" or "that" referred by gesture input _GInId
then **LET** all "*in*" **deictic_word**(_, _) be *out* **and**
ASSUME deictic_word(_GInId, _DeicticWordId)

[Rule C.5]

if **integrate**(_LMmiElementId, _MmiId) **and**
no_omission(gIn, _LGinId, _MmiId) **and**
no_omission(vIn, _LVinId, _MmiId) **and**
append(_LGinId, _LVinId, _LMmiElementId)
then **DERIVE** *integrated_input*(_LMmiElementId, _MmiId)

[Rule C.6]

if **vIn_sentence**(_LWordId, _VInId) **and**
member(_WordId, _LWordId)
then **DERIVE** *vIn_word*(_WordId, _VInId)

[Rule C.7]

if **vIn_sentence**(_LWordId, _VInId) **and**
member(_WordId, _LWordId) **and**
the word of _WordId is the referred noun in the sentence
then **DERIVE** *object_noun*(_WordId, _VInId)

[Rule C.8]

if **vIn_sentence**(_LWordId, _VInId) **and**
object_noun(_ObjectWordId, _VInId) **and**
member(_ModifyWordId, _LWordId) **and**
the word of _ModifyWordId modifies the word of _ObjectWordId
then **DERIVE** *modify*(_ModifyWordId, _ObjectWordId)

[Rule C.9]

if *object_noun*(_WordId, _VInId) **and**
the word of _WordId is singular
then **DERIVE** *singular_object*(_WordId, _VInId)

[Rule C.10]

if *object_noun*(_WordId, _VInId) **and**
the word of _WordId is plural
then **DERIVE** *plural_object*(_WordId, _VInId)

[Rule C.11]

if *integrated_input*(_LInputId, _MmiId) **and**
member(_GInId, _LInputId) **and**
member(_VInId, _LInputId) **and**
deictic_word(_GInId, _DeicticWordId) **and**
vIn_word(_DeicticWordId, _VInId)
then **DERIVE** *deixis*(_GInId, _DeicticWordId, _VInId)

[Rule C.12]

if location_object(_LObjectId, _LocationId) **and**

gesture_location(_GInId, _LocationId) **and**
deixis(_GInId, _, _VInId)

then **DERIVE** *deixis_object*(_LObjectId, [_VInId], [_GInId])

[Rule C.13]

if **vIn_sentence**(_LWordId, _VInId) **and**
object_of_noun(_LObjectNoun, _NounWordId) **and**
object_of_value(_LObjectValue, _ValueWordId) **and**
member(_NounWordId, _LWordId) **and**
member(_ValueWordId, _LWordId) **and**
intersection(_LObjectNoun, _LObjectValue, _LReferents)

then **DERIVE** *verbal_object*(_LWordId, _LReferents)

[Rule C.14]

if **vIn_sentence**(_LWordId, _VInId) **and**
object_of_noun(_LObjectNoun, _NounWordId) **and**
member(_NounWordId, _LWordId) **and**
Rule C.13 is not applicable

then **DERIVE** *verbal_object*(_LWordId, _LObjectNoun)

[Rule C.15]

if **vIn_sentence**(_LWordId, _VInId) **and**
verbal_object(_LWordId, _LReferents) **and**
verbal_modifier_and_noun(_, _VInId)

then **DERIVE** *voiceIn_object*(_LReferents, _VInId)

[Rule C.16]

if GOAL is *referent_object*(_, _MmiId) **and**
voiceIn_object(_LReferents, _VInId) **and**
deixis_object(_LReferents, _LVinId, _LGinId) **and**
integrated_input(_LInputId, _MmiId) **and**
vIn_sentence(_LWordId, _VInId) **and**
singular_object(_WordId, _VInId) **and**
gesture_location(_GInId, _) [For all _GInId in _LGinId] **and**
member(_VInId, _LVinId) **and**
append([_VInId], _LGinId, _LInputId) **and**

count(_LReferent) = 1
then **DERIVE** *referent_object*(_LReferents, _MmiId)

[Rule C.17]

if GOAL is *referent_object*(_, _MmiId) **and**
voiceIn_object(_LReferents, _VInId) **and**
deixis_object(_LReferents, _LVinId, _LGinId) **and**
integrated_input(_LInputId, _MmiId) **and**
vIn_sentence(_LWordId, _VInId) **and**
plural_object(_WordId, _VInId) **and**
gesture_location(_GInId, _) [For all _GInId in _LGinId] **and**
member(_VInId, _LVinId) **and**
append([_VInId], _LGinId, _LInputId) **and**
count(_LReferent) ≠ 1
then **DERIVE** *referent_object*(_LReferents, _MmiId)

D Contradiction Detection Rule Set

[Rule D.1]

if **no_omission**(_Modality, _LId1, _MmiId) **and**
no_omission(_Modality, _LId2, _MmiId) **and**
_LId1 $\neq$ _LId2
then **DERIVE** $\perp$

[Rule D.2]

if **integrate**(_LMmiElementId1, _MmiId) **and**
integrate(_LMmiElementId2, _MmiId) **and**
_LMmiElementId1 $\neq$ _LMmiElementId2
then **DERIVE** $\perp$

[Rule D.3]

if **integrate**(_LAllElementId, _MmiId) **and**
no_omission(_M, _LIdM1, _MmiId) **and**
_LIdM2 = subset of _LAllElementId each of which is from _M **and**
_LIdM1 $\neq$ _LIdM2
then **DERIVE** $\perp$

[Rule D.4]

if *deixis*(_GInId, _DeiciticWordId, _VInId) **and**
voiceIn_object(_VInId, _LVoiceObject) **and**
deixis_object(_GInId, _LDeixisObject) **and**
_LVoiceObject $\neq$ _LDeixisObject
then **DERIVE** $\perp$

E Contradiction Resolution Rule Set

[Rule E.1]

if **no_omission**(_Modality, _LId1, _MmiId) **and**
no_omission(_Modality, _LId2, _MmiId) **and**
_LId1 ⊂ _LId2
then **LET no_omission**(_Modality, _LId1, _MmiId) be *out*

[Rule E.2]

if **integrate**(_LMmiElementId1, _MmiId) **and**
integrate(_LMmiElementId2, _MmiId) **and**
_LMmiElementId1 ⊂ _LMmiElementId2
then **LET integrate**(_LMmiElementId1, _MmiId) be *out*

[Rule E.3]

if the cause of the contradiction is resolution failure
then **LET vIn_sentence**(_LWordId,_VInId)
and/or **gesture_location**(_LocationId, _GInId)
and/or **deictic_word**(_GInId, _DWordId) be *out*

[Rule E.4]

if **integrate**(_LAllElementId, _MmiId) **and**
no_omission(_M, _LIdM1, _MmiId) **and**
_LIdM2 = subset of _LAllElementId each of which is from _M **and**
_LIdM1 ⊂ _LIdM2
then **LET no_omission**(_M, _LIdM1, _MmiId) be *out*

Figure 1: Multimodal Interface Agent System

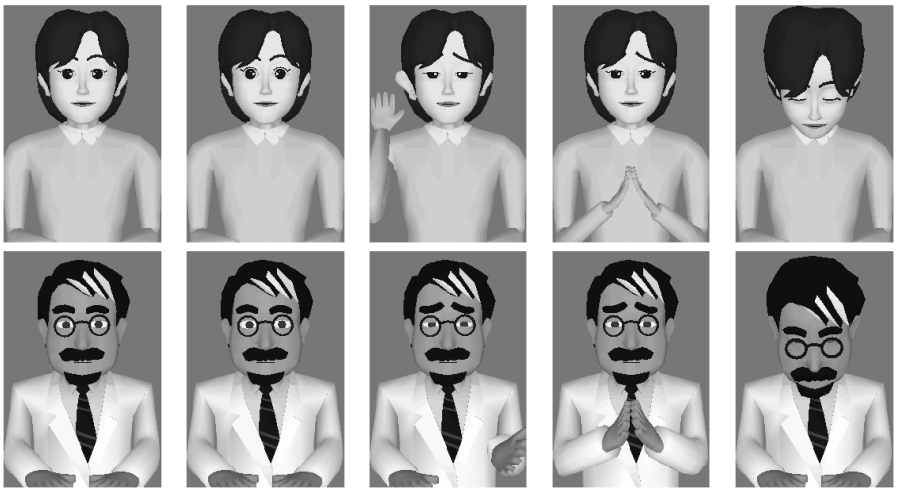

Figure 2: Screen images of the agents

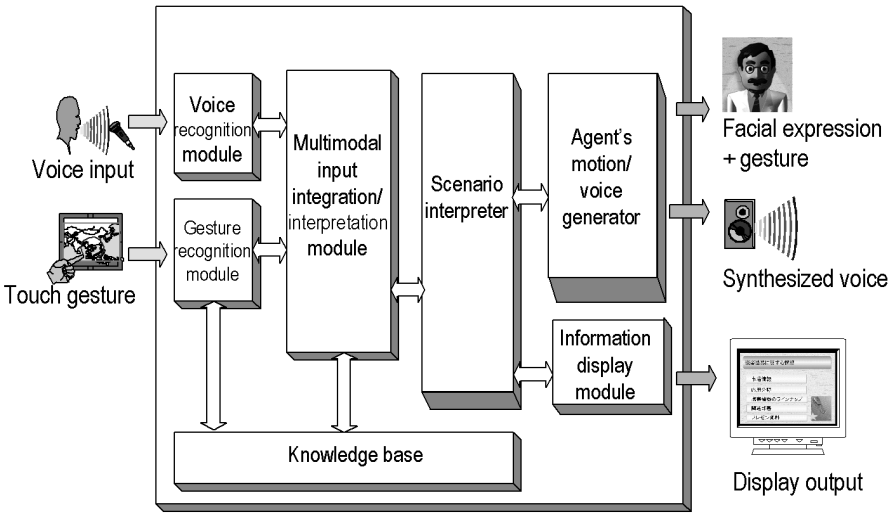

Figure 3: Overall configuration of multimodal interface agent system

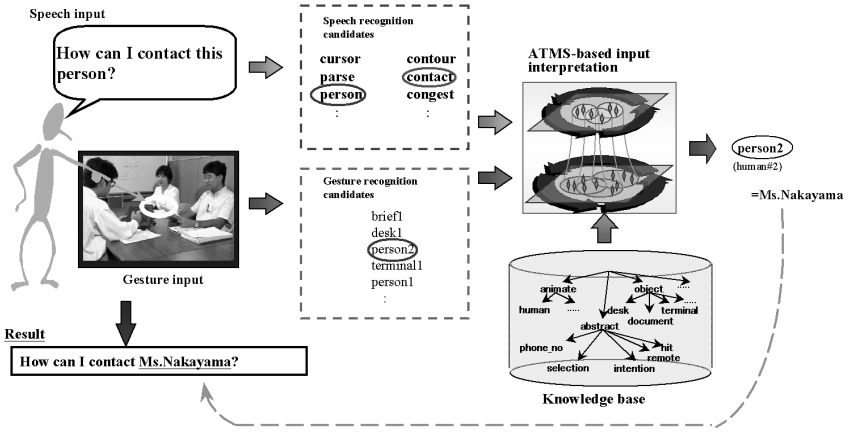

Figure 4: Multimodal reference resolution process

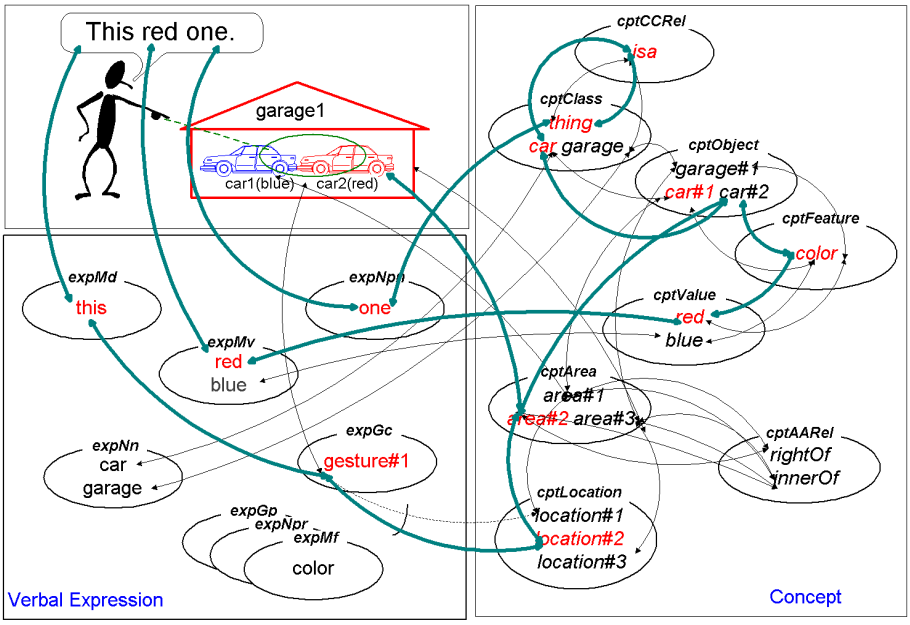

Figure 5: Knowledge base structure

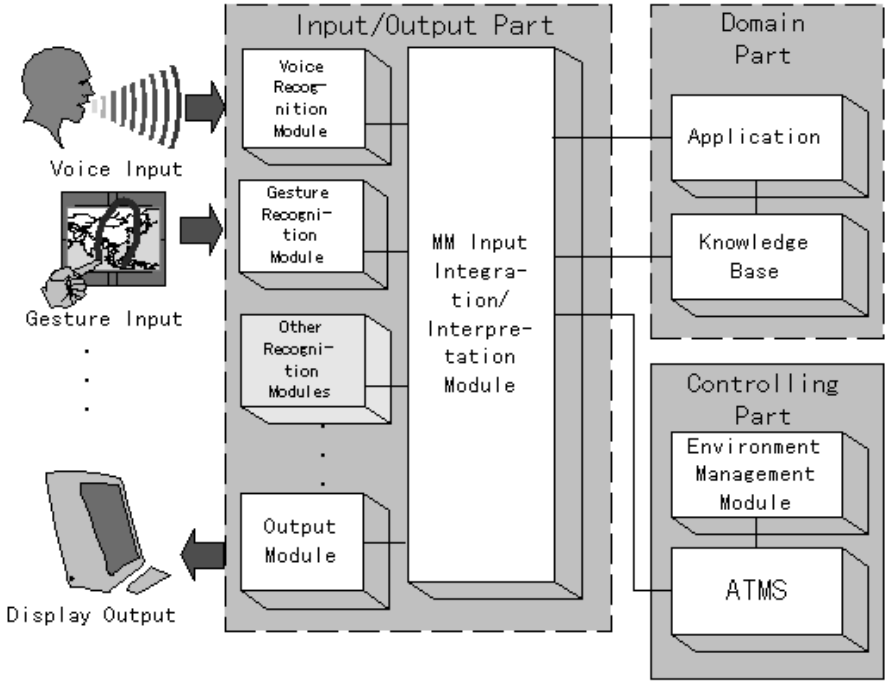

Figure 6: Detailed configuration of the multimodal input integration/interpretation module

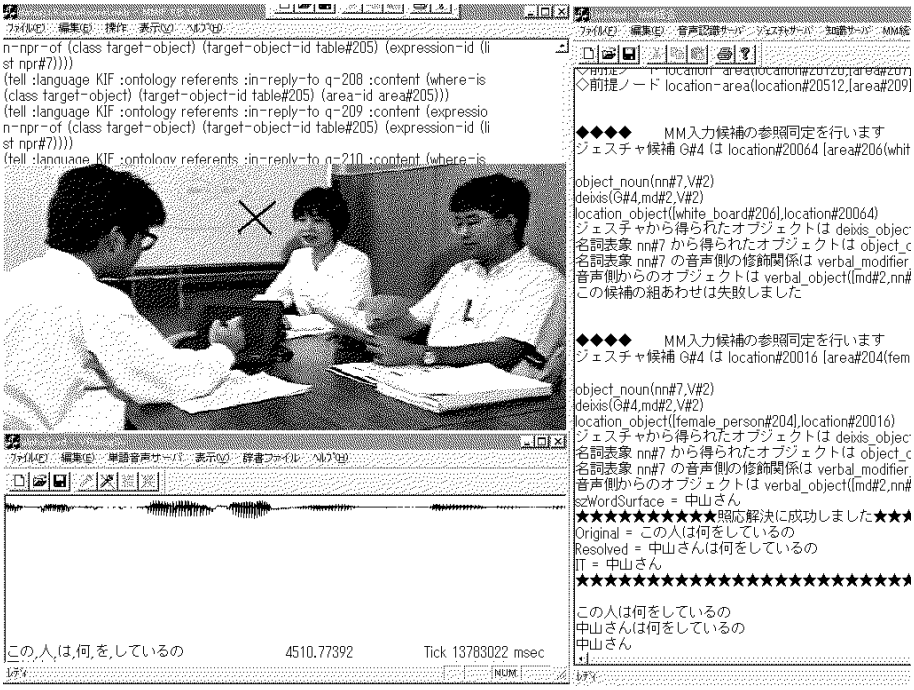

Figure 7: Example screencopy of MM reference resolution