International Journal of Advanced Research in IT and Engineering

ISSN: 2278-6244

SOFT COMPUTING IN INTELLIGENT SYSTEMS

Himanshu Sekhar Acharya*

Abstract: *In this paper we will describe an intelligent multi-modal interface for a large workforce management system called the smart work manager. The main characteristics of the smart work manager are that it can process speech, text, face images, gaze information and simulated gestures using the mouse as input modalities, and its output is in the form of speech, text or graphics. The main components of the system are a reasoner, a speech system, a vision system, an integration platform and an application interface. The overall architecture of the system will be described together with the integration platform and the components of the system which include a non-intrusive neural network based gaze tracking system. Fuzzy and probabilistic techniques have been used in the reasoner to establish temporal relationships and learn interaction sequences.*

Keywords: *Multi-modal, Neural network, Reasoner*

*Asst. Prof., Comp. Sc., KIIMS,CUTTACK

1. INTRODUCTION

Soft computing techniques are beginning to penetrate into new application areas such as intelligent interfaces, information retrieval and intelligent assistants. The common characteristic of all these applications is that they are human-centred. Soft computing techniques are a natural way of handling the inherent flexibility with which humans communicate, request information, describe events or perform actions.

A multi-modal system is one that uses a variety of modes of communication between human and computer either in combination or isolation. Typically research has concentrated on enhancing standard devices such as keyboard and mouse, with non- standard ones such as speech and vision.

An Intelligent multi-modal system is defined as a system that combines, reasons with and learns from, information originated from different modes of communication between human and the computer.

The main reason for using multi-modality in a system is to provide a richer set of channels through which the user and computer can communicate.

The necessary technologies for such systems are:

- AI and Soft Computing for representation and reasoning
- User interfaces for effective communication channels between the user and the system

In this paper we will describe the development of an intelligent multi-modal system referred to as the smart work manger (SWM) for a large-scale work force scheduling application.

2. RELATED WORK

Research in human-computer interactions has mainly focused on natural language, text, speech and vision primarily in isolation. Recently there have been a number of research projects that have concentrated on the integration of such modalities using intelligent reasoners. The rationale is that many inherent ambiguities in single modes of communication can be resolved if extra information is available . Among the projects reviewed in the references are CUBRICON from Calspan-UB Research Centre, XTRA from German Research Centre for AI and the SRI system from SRI International.

The main characteristics of the SWM are that it can process speech, text, face images, gaze information and simulated gestures using the mouse as input modalities, and its output is in the form of speech , text or graphics. The main components of the system are the reasoner, a speech system, a vision system, an integration platform and the application interface.

3. ENGINEERING ISSUES

Intelligent multi-modal systems use a number of input or output modalities to communicate with the user, exhibiting some form of intelligent behaviour in a particular domain. The functional requirements of such systems include the ability to receive and process user input in various forms such as:

- typed text from keyboard,
- mouse movement or clicking,
- speech from a microphone,
- focus of attention of human eye captured by a camera,

The system must be also able to generate output for the user using speech, graphics, and text.

A system, which exhibits the above features, is called a multi-modal system. For a multi- modal system to be also called intelligent, it should be capable of reasoning in a particular domain automating human tasks, facilitating humans to perform tasks more complex than before or exhibiting a behaviour which can be characterised as intelligent by the users of the system.

Given these requirements for intelligent multi-modal systems, it becomes obvious that such systems, in general, are difficult to develop. A **modular** approach is therefore necessary for breaking down the required functionality into a number of sub-systems which are easier to develop or for which software solutions already exist. Other requirements for such systems are concurrency, a communication mechanism and distribution of processes across a network.

4. ARCHITECTE

The overall architecture of SWM with the various modules and the communications between them is given in Fig. 1.

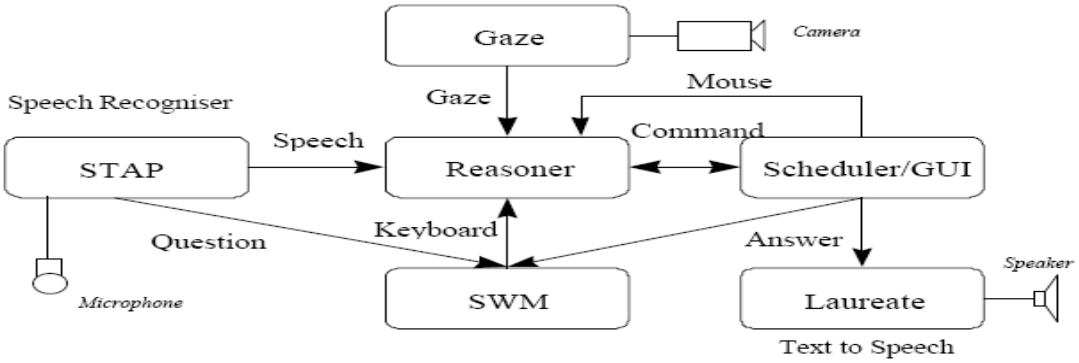

Figure 1 - An overview of SWM Architecture

5. THE REASONER

The main functions of the reasoner are two folds. First it must be able to handle ambiguities such as *give me this of that*. Second it must have the capabilities to deal with often-conflicting information arriving from various modalities. The capabilities of the reasoner are to a large extent dependant upon the capabilities provided by the platform on which the reasoner is implemented. The platform used for the reasoner is CLIPS, which is a well known expert system shell developed by NASA with object oriented, declarative and procedural programming capabilities and the fuzzyCLIPS extension. The reasoner handles ambiguities by using a knowledge base that is being continually updated by the information arriving from various modalities. The structure of the reasoner is shown in Fig. 2. There are five modules in the reasoner: fuzzy temporal reasoning, query pre- processing, constraint checking, resolving ambiguities (WIZARD) and post-processing.

The **fuzzy temporal reasoning module** receives time-stamped events from various modalities and determines the fuzzy temporal relationship between them. It determines to what degree two events have a temporal relationship, such as before, during or overlapping. Using the certainty factors (CF) of *fuzzy*CLIPS the temporal reasoner can answer questions such as:

what is the CF that event 1 took place before event 2 what is the CF that event 1 took place just_before event 2 what is the CF that event 1 took place during event 2

what is the CF that event 1 is overlapping with event 2

The relationship with the highest CF will be chosen as the most likely relationship between the two events. This relationship can be used later by the reasoner to resolve conflicts between, and checking dependency of, the modalities.

In the **query pre-processing module** a sentence in natural language form is converted to a query which conforms to the system's pre-defined grammar. Redundant words are removed, keywords are placed in the right order and multiple word attributes are converted into single strings.

The **constraint checking module** examines the content of the queries. If individual parts of the query do not satisfy pre-defined constraints then they are replaced by ambiguous terms (*this, that*) to be resolved later, otherwise the query is passed on to the next module.

The **WIZARD** is at the heart of the reasoner and is the module that resolves ambiguities. The ambiguities in this application take the form of reserved words such as *this* or *that*, and they refer to objects that the user is or has been talking about, pointing at or looking at. The ambiguities are resolved in a hierarchical manner as shown in Fig3.

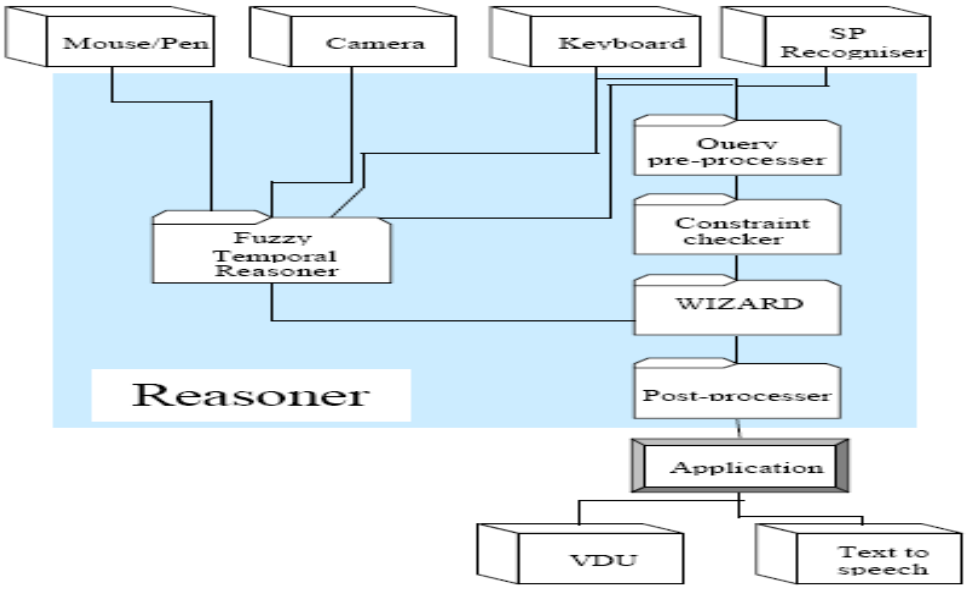

Figure 2 – The structure of the Reasoner

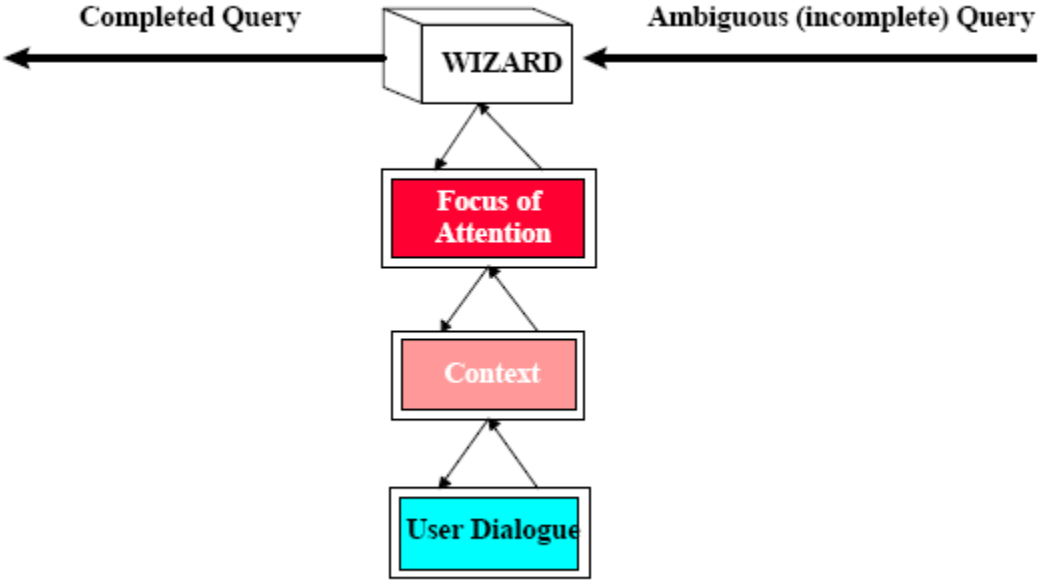

Figure 3 - Resolving ambiguities in the Reasoner

The context of the interactions between the user and the system, if it exists, is maintained by the reasoner in the knowledge base. When a new query is initiated by the user, it is checked against the context. If there are ambiguities in the query and the context contains relevant information then the context will be used to create a complete query that will be sent to the application interface for processing. Another mechanism used to resolve ambiguities is the focus of attention, which is obtained from the user when pointing with

The mouse or gazing at an object on the screen. At the same time there could be dialogue with the user through the speech recognizer or the keyboard. CLIPS is mainly used in a declarative mode in this system and therefore all the modules work in parallel. This can cause conflict between information arriving from various modalities. The conflict resolution strategy used in the reasoner is hierarchical as shown in Fig. 3, with the focus of attention having the highest priority and the dialogue system the lowest. This means that the dialogue system will act as a safety net for the other modalities if all

fails, or if inconsistent information is received from the modalities. In cases where text input is required however, the dialogue system is the only modality that will be called upon. In all other cases the dialogue system will be redundant unless all others fail in which case a simple dialogue in the form of direct questions or answers will be initiated by the system. The WIZARD sends the completed queries to the post-processing module.

The **post-processing module** simply converts the completed queries in a form suitable for the application. This involves simple operations such as formatting the query or extracting key words from it.

Table 4 contains some examples of interactions with the reasoner and how it works.

Query	User Action	Reasoning Process
show me technician ab123 job locations on the map	none	Complete query, process command and sent to application
tell me the duration of this job	mouse is clicked or eyes are focused on a job	this job is ambiguous it is resolved using focus context is updated
show me this of that technician	no focus context is technician *ab123 end of day*	two ambiguities context is used to solve ambiguities
read me this	no focus no context	**everything** is ambiguous the user is asked to repeat the missing parts the context is updated

Table 1 -Examples of Interactions with the Reasoner

6. GAZETRACKING: TECHNICAL ISSUES

The initial objective for the gaze system is the capability of tracking the eye movements within slightly restricted environments. More specifically, the scenario is a user working in front of a computer screen viewing objects on display while a camera is pointed at the user's face. The objective is to find out where (which object) on the screen the user is looking at, or to what

context he is paying attention. This information in combination with other inputs provided by speech and other modalities would be most useful to resolve some real application tasks.

The general difficulties we have to face to build a gaze tracking system are as follows;

- imprecise data, or the head may pan and tilt resulting in many eye images (relative to the viewing camera) corresponding to the same co-ordinates on the screen,
- Noisy images, mainly due to change of lighting. This is typical in an uncontrolled open plan office environment,
- possibly infinitely large image set, in order to learn the variations of the images and make the system generalise better,
- Accuracy and speed compromise, for a real-time running system, more complicated computation intensive algorithms have to give way to simple algorithms.

Gaze Tracking: System description

Neural networks have been chosen as the core technique to implement the gaze tracking system.

- A three-layer feed-forward neural network is used. The net has 600 input units, one divided hidden layer of 8 hyperbolic units each and corresponding divided output layer with 40 and 30 units, respectively, to indicate the positions along x- and y- direction of the screen grid.
- Grey-scale images of the right eye are automatically segmented from the head images inside a search window, the images of size 40 □ 15 are then normalised, and a value between -1 and 1 is obtained for each pixel. Each normalised image comprises the input of a training pattern.
- The pre-planned co-ordinates of this image, which is the desired output of the training pattern, is used to stimulate two related output units along the x and y output layer respectively.
- The training data are automatically grabbed and segmented when one tracks with the eyes a cursor movement following a pre-designed zigzag path across or up and down the screen. A total of 2000 images is needed to train the system to achieve better performance.

- The neural network described has a total of 11,430 connection weights. The off-line training takes about half an hour on the Ultra-1 workstation. Once trained, the system works in real-time.

7. CONCLUSION

In this paper I have shown how such flexibility can be exploited within the context of an intelligent multi-modal interface. Soft computing techniques have been used at the interface for temporal reasoning, approximate query matching, learning action sequences and gaze tracking. It is important to note that soft computing has been used in conjunction with other AI- based systems performing dynamic scheduling, logic programming, speech recognition, and natural language understanding. I believe that soft computing in combination with other AI techniques can make a significant contribution to human-centered computing in terms of development time, robustness, cost, and reliability.

We plan to investigate the following improvements to the reasoner in the near future:

- The context can be extended to have a tree like structure such that the user is able to make reference to previously used contexts.
- The temporal reasoner can be used more extensively in conflict resolution.
- The grammar can be extended to include multiple format grammars.
- The dialogue system can be improved to become more user friendly.
- Different approaches can be used for resolving ambiguities such as competition between modalities or bidding.

8. REFERENCES

1. Azvine, B., Azarmi, N. and Tsui, K.C. 1996. Soft computing - a tools for building intelligent systems, BT Technology Journal, vol. 14, no. 4, pp. 37 45, October.
2. Bishop, C. 1995.Neural Network for Pattern Recognition. Oxford University Press.
3. Lesaint, D., C. Voudouris, N. Azarmi and B. Laithwaite 1997. Dynamic Workforce Management. UK IEE Colloquium on AI for Network Management Systems (Digest No.1997/094), London, UK.
4. Negroponte, Nicholar1995. Being Digital.oronetBooks.

5. Nigay L., Coutaz J., 1995, A Generic Platform for Addressing the Multimodal Challenge,CHI '95 Proceedings.
6. Page, J.H., and Breen A. P. 1996, The Laureate Text-to-Speech system: Architecture and Applications. BT Technology Journal 14(1), 84-99.
7. Schomaker l., Nijtmans J., Camurri A., Lavagetto F., Morasso P., Benoit C., Guiard-Marigny A., Le Goff B., Robert-Ribes J., Adjoudani A., Defee I., Munch S., Hartung K. and Blauert J., 1997 *'A Taxonomy of Multimodal Interaction in the Human Information Processing System'*, Report of the ESPRIT PROJECT 8579 MIAMI, February.
8. Sullivan, J. W., and S. W. Tyler, eds, 1991. Intelligent User Interfaces. Frontier Series. New York: ACM Press.
9. Tsui K.C., Azvine B., Djian D., Voudouris C., and Xu L.Q., 1998, Intelligent Multimodal Systems, BTTJ Jounal, Vol.16, No. 3, July.