An Agent-Based Architecture for a Multimodal Interface

Claudie FAURE - Luc JULIA
Télécom Paris - CNRS URA 820
46 rue Barrault
75634 Paris cedex 13
cfaure@sig.enst.fr - julia@sig.enst.fr

ABSTRACT

This paper concerns a multimodal interface for designing graphics. The human-machine interaction is achieved by using a pen computer connected to a speech recognition device. The user and the computer cooperate: the user sketches graphics with the pen, the computer understands the user's intention and displays an accurate version of the hand-sketched drawing. The final document is produced sequentially through a process of thinking and drawing. An agent-based architecture implements a multimodal interface with rapidly displayed feedback, supports a cooperative incremental design and adapts the interaction to the user style. At present, an initial application has been developed for drawing tables.

KEYWORDS: Pen & Voice Interaction - Agent-Based Architecture

INTRODUCTION

The domain of Human-Computer Interfaces (HCI) attests to an increasing interest for new input devices facilitating the interaction between the user and the computer. Multimodality is a way to improve human-computer interaction. The first multimodal interaction systems, such as the "Put-That-There" system [1], have spurred many studies in the field of multimodal HCI. Quite recently, several works have appeared concerning pen-based interfaces [2, 10], which offer a Computer Augmented Environment to the user with no equivalent in the classic pen and paper world and exploit the user's natural style of producing drawings. Our project combines the concepts of multimodal interaction [5] and of the electronic paper interface [7].

The desire to integrate a speech recognition device and a pen computer arises from the human tendency to combine gestures and speech, particularly obvious when manipulating graphic objects [6]. Such integration offers a wide range of "natural" modalities (speech, drawing, writing, and gestural commands) for the interaction.

A multiple channel HCI also presents the advantage of leaving users free to choose what they consider as the best interaction within the current context of the task.

The task here considered is the incremental design in some graphic domain (tables, diagrams, maps ...). The electronic paper is an input/output device. The graphics and gestural commands are hand-sketched on its surface, a visual version of the user's intentions are displayed on the same surface.

This paper will concentrate on the software architecture. See [4] for applications of agent-based architecture to multimodal interactive systems. It will be shown here that an agent-based architecture suitably implements both the requirements related to the multimodal interaction (such as rapid feedback, the possibility of blending the modalities and of adapting the style of the interaction to the situation), and those ensuring an effective collaboration between the user and the computer within the course of the design task.

The first application was aimed at designing tables and was demonstrated in the Summer of 1993 [8]. The present paper starts with an overview of the multimodal interaction system. The next two sections describe the peripheral agents and the interpretation of multimodal input, the final section concerns the behaviour of the user and of the machine at work.

THE MULTIMODAL INTERACTION SYSTEM

Figure 1 shows the input/output screen. It is divided into a "data space", where the user sketches and manipulates drawings using gestural commands and sees the results of the interpretive processes, and a "menu space" where the user can choose an item (either by pointing at it or saying its name).

The interaction may be roughly described as follows: the user sends input signals (pen entry and/or speech); these signals are interpreted by the system which then reacts by producing a visible feedback and by modifying its internal representations. The input signals are categorised as either command or data. Speech can only produce commands while the pen can be used to produce either data (writing and drawing) or gestural commands. Both categories necessitate signal interpretation and appropriate rapid feedback. The interpretation of graphics, which presently only works for tables, necessitates pattern recognition techniques, domain

Figure 1: A view of the interactive surface with a sketched table

knowledge and knowledge about visual communication. This paper neither presents the details concerning the interpretation of handsketched tables [3] nor those concerning recognition of gestural commands [9]. Here we emphasize multimodal interaction.

Several functions have to be implemented to achieve a multimodal interaction: a peripheral function ensuring the relation between the computer's external environment and internal representations; an interpretation leading to a meaningful representation from the input, and various actions which modify the internal representation and the displayed information.

These functions are implemented using an agent-based architecture. Agents may operate autonomously and simulate parallel processing. Consequently a multiple channel interaction —where each sensitive channel is associated to a specialised agent— becomes feasible.

The agents are organized heterarchically. The agent-based architecture can be described by specifying the various processing levels (from the agents involved in the peripheral function to the agents acting on the internal representations) although some agents have the possibility to communicate directly, regardless their level. This heterarchic structure is useful for distributing functions within the system with, as a consequence, a better cooperation between the agents.

THE PERIPHERAL AGENTS

The interaction involves five media: a microphone, a pen, a keyboard, a loudspeaker and a screen. These five media correspond to six modalities namely speech, writing, drawing, gestures for the input; visual display and sound for the output. Figure 2 shows the peripheral agents. Each media is associated to a media-agent which is directly connected to the set of modality-agents determined by the medium.

Figure 2: The peripheral agents

Modality-agents perform some processes and activate reflex actions. The processes are aimed at accessing valuable information from an input which will be useful for higher level of processing. A modality agent performs a process without knowing what the other peripheral agents are doing. Reflex actions are mainly activated in order to rapidly display feedback, they do not involve higher levels of processing: they result from a direct connection between modality-agents.

Processes and reflex actions imply that the modality agents have a view of the internal representation of the application and knowledge of the menu space which makes them able to match menu item names with their position on the screen.

We give two examples to illustrate the usefulness of peripheral processes and reflex action. Gestures are used to select a part of the drawing or a position (such as in the case of /put it here/; by pointing the user selects the graphic object /it/ and the location /here/). The gesture-agent matches the (x,y) coordinates of the pen on the screen with the internal representation of the displayed drawing. This process results in an early discrimination between object and position which provides valuable information for the interpreter. A reflex action is activated by sending this information to the display-agent which displays appropriate feedback (a selected object is displayed in doted lines, a position is visualised with a cross). The user realizes what the computer has understood. The feedback appears before the completion of the command, the user can then change the selection. Figure 3 shows examples of feedback resulting from the interpretation of different kinds of object selection gestures.

Another example of the interest offered by reflex action concerns the equivalence between speech and gestural input. An item on the menus can be pen-pointed or selected by saying its name. Any kind of selection "switches on" the selected item (its colour changes). This is achieved by the

Figure 3: Examples of visible feedback after objet selection

Figure 4. Media and multimodal interaction

display-agent after it has received (from the speech- or gesture-agent) information about the item's position.

The processes that are performed at the peripheral level unload the upper levels of the system where the commands and the data are completely interpreted. Thus, in this fashion, the understanding function is distributed within the whole system. The display function is also distributed. The peripheral display-agent is activated by other peripheral agents, but the application contains specialised display-agents which are able to change the displayed information without any communication with the agents outside the application. These specialised display agents display the interpreted version of the hand-sketched drawing, the result of an action (erase, move, add new parts) and "switch off" the menu item when it is deactivated (after an action has been completed).

UNDERSTANDING MULTIMODAL INPUT

Above the peripheral agents, the interpreter-agent supports the multimodal interaction. This agent transforms the messages it receives from the peripheral level into an a-modal symbolic representation defined as a triplet: Verb - Object(s) - Attribute (<VO*A>). The interpreter instantiates the VO*A triplet with the information provided by the peripheral agents, according to its knowledge about the verbs. An event is obtained when VO*A has been correctly instantiated. Figure 4 shows the path from the pen and microphone media to an event.

The vocabulary contains 15 verb classes. A class of verbs contains several synonyms. The total comprises 35 (French) verbs. The verbs are associated to rules specifying which symbols of the VO*A triplet must be instantiated to obtain an event. For example, if V is instantiated by /erase/, the symbol A need not be associated to any physical input signal. The VO*A triplet represents an event if O is already instantiated, if not the interpreter-agent scans the peripheral agents until it has received the missing information.

The interpreter has no access to the internal representation of the application, nor does it to the displayed information. Gestural commands (as pen pointing) have already been interpreted at the peripheral level as object(s) or location. At this level, the early discrimination will be represented by O* and A.

The same event can be obtained using different input signals. For example: The user can say /put/, /put this/, /put here/, /put here this/, /put this here/, /this, this, this, put here/, or can say nothing and pen-points the item menu Put. The objet(s) (/this/) and the position (/here/) are selected with gestural commands in the case concerning the tables. Verbal object selection (e.g.: the square(s), the red circle, ...) are studied for other applications (such as network diagrams).

The pen is used to produce data (drawing or writing) which must be interpreted and then re-displayed accurately. Likewise the pen is used to produce gestural commands, these gestures must disappear once they have been interpreted. Ambiguity may arise: drawing a circle, writing the letter O or selecting parts of the graphics by a surrounding circle produce graphically identical data. This problem is overcome with the state commands: Draw, Write, Correction (for gestural commands). A state command gives an a priori modality label to the forthcoming pen entries.

The system will react to each event. Once the interpreter-agent has detected an event, this event is sent to the action-agent which establishes the link with the application. The symbolic event representation is interpreted in order to activate the appropriate processes within the application. The information conveyed (but not used) by the interpreter, as the identity of the selected objects, is used by the action-agent to send parameters to the activated application procedures. The interpreter need only know that one object or one position has been selected while the action-agent must know which object or which position.

The events modify the internal representation of the application. The drawing representation is changed when the user manipulates the drawings, either by adding, moving or erasing lines. These changes are not the passive record of the user's action on the drawing : the application updates the graphics according to the domain knowledge (for example, erasing a line in a table results in displaying the table without this line but also other lines may be transformed in order to fit the conventions of table structure). The state of the application is changed after each state command, afterwards the application awaits a specified pen-input modality (drawing, writing or gesture). The state of the application is visible from the corresponding peripheral agents which know if they have to react to the pen-entry by comparing their own modality label with the current state of the application.

SITUATED (INTER)ACTION

The advantages of speech- and pen-based interaction have been exploited in two ways:

- pen and speech (PAS), where speech and gestures are combined to produce an event (as saying /erase this/ and pen-pointing an object)
- pen or speech (POS), where several ways to do the same thing are offered to the user (as saying /move/ or pen-pointing the visualised menu item Move).

PAS brings some of the behaviour of the natural communication into the HCI world and is often considered as the most advanced style of multimodal interaction. POS also presents certain advantages. POS offers a flexible interaction, users can either choose to speak or use the pen without informing the system of their choice.

The interaction may be seen as a task aimed at performing an incremental design which is the main task. Proper collaboration implies that the interaction does not disturb the ongoing main task. For example, designers may experience a situation where shifting their attention from the drawing to the visualised menu will trouble the course of their thought process. Moreover, the menus contain a limited number of items in order to ease their reading and item access, not all of the available items are visualised. A spoken menu allows users to focus their attention on the drawing and to rapidly access the invisible item menu (a pen-pointed choice may be used to select it in one or two steps).

The availability of several modalities is also of great interest in replacing a modality by another in case of failure in the signal recognition, this may be the case when speech entry is used in noisy environments then the user can skip to pen entry and carry on the dialogue.

The way the understanding function has been implemented allows the user to pronounce words which are not in the vocabulary. Only the informative components of the input are considered, for example /this/ or /here/ do not contain information about the objects or location. The system works more as a word spotter than an interpreter of the whole sentences as a NL parser would do. The advantage is to leave the user free to add verbal productions to the informative spoken or gestural commands.

Graphics interpretation has not been described here, but it should be noted that the application adapts to the situation. The values of the parameters involved in the graphic interpretation process are automatically adapted to the "quality" of the displayed drawing. For example, the tolerance thresholds are higher for an hastily hand-sketched draft than for one carefully drawn.

CONCLUSION

A multimodal interface system is developed with a speech device (DATAVOX a product of VECSYS) and a pen-computer (NCR NotePad 3130). The agent-based architecture was chosen to implement the system. Agents are autonomous, they are able to perform local processes. They can cooperate by direct communication or indirect communication through another agent. Representations or functions can be distributed within the agent-based system. The first application on tables is now used as a tool for testing human behaviour in order to acquire knowledge about the multimodal interaction in the context of an HCI. Other fields of application are in progress, mainly networks diagrams and architectural drawings.

REFERENCES

1. Bolt, R.A. Put-That There: Voice and Gesture at the Graphics Interface. *ACM Computer Graphics, 14(3)*, 1980, pp. 262-270.
2. Endo Y., Akimichi S., Milne M. The Context-Based Graphic Input System: T-Board. In *Proc. HCI*

International'93, Orlando. Elsevier, 1993, pp. 1004-1009.

3. Faure, C., Julia, L. Interaction homme-machine par la parole et le geste pour l'édition de documents : TAPAGE. In *Proc. L'interface des mondes réels et virtuels,* Montpellier, 1993, pp. 171-180.
4. Gourdol, A., Nigay, L., Salber, D., Coutaz, J. Two case Studies of Software Architecture for Multimodal Interactive Systems : VoicePaint and a Voice-enable Graphical Notebook. In *Proc. Engineering for human-computer interaction.,* Ellivuori. North-Holland, 1992, pp. 271-284.
5. Hanne, K.-H., Bullinger, H.-J. Multimodal Communication: Integrating Text and Gestures. *Multimedia Interface Design*. ACM Press, 1992, pp.. 127-138.
6. Hauptmann, A.-G., McAvinney, P. Gestures with speech for graphic manipulation. *International Journal of Man-Machine Studies, 38,* 1993, pp. 231-249.
7. Higgins, C.-A., Ford, D.-M. Stylus Driven Interface - The Electronic Paper Concept. In *Proc. ICDAR'91,* Saint-Malo, 1991, pp. 853-862.
8. Julia, L., Faure, C. A multimodal Interface for Incremental Graphic Document Design. In *Poster Proc. HCI International'93* Orlando, 1993, p. 237.
9. Poirier, F., Julia, L., Rossignol, S., Faure, C. TAPAGE : édition de tableaux sur ordinateur à stylo vers une désignation naturelle. In *Proc. IHM'93* , Lyon, 1993
10 Zhao, R. Gesture Specification and Structure Recognition in Handsketch-Based Diagram Editors. In *Proc. HCI International'93* (Orlando), Elsevier,1993, pp. 1052-1057.