Dialogue Expertise in Man-Machine Systems

George R. S. Weir

Scottish HCI Centre
George House
North Hanover Street
Glasgow G1 2AD
UK

Abstract

The present paper considers the nature of interaction in man-machine systems with special attention to the role of dialogue. Based upon this understanding, a concept of 'dialogue expertise' is derived as a specific facility for mediation. In particular, it is the ability to mediate between user and system so as to (one way) map operator objectives onto system functions, and (the other way) express system states in terms which are meaningful to the operator. This leads to the view that dialogue expertise is feasible in man-machine systems through the use of functional separation in the design of interface systems.

1 Introduction

The clearly felt need to improve the quality of interaction between man and machine in complex control systems rightly focusses attention on the system interface. Located between the operator and some application/ process to be controlled, the interface can be regarded as a mediator between the two. As such, it fills its primary role by relaying control actions from operator to process and by presenting process state information to the operator. Although this captures the gross detail of interaction, it does not do justice to the complexity of man-machine interaction.

The purpose of this paper is to provide a more detailed view of what is involved in interface mediation, i.e., the nature and sub-functions involved in human-computer interaction (HCI). In what follows, I detail the role of the system interface as such a mediator and propose a concept of 'dialogue expertise' which, if explicitly embodied in man-machine systems, promises considerable benefits for human-system interaction.

2 Interface Mediation

Interaction between man and machine can be viewed from a variety of standpoints. Some authors urge that the activity of human-computer interaction be viewed as a process comparable with human-human communication (e.g., [SR88], compare [BCM89]). The naturalness of communication, i.e., how closely it resembles human-human interaction, will certainly affect the attitude of the human participant, although not always positively.

Where a distinction can be drawn between available system functionality (the use for the system) and the nature of system interaction, a natural style of interaction runs the risk of users over estimating the intelligence and capabilities of the system [1]. Furthermore, some who advocate natural language interaction as a boost to the quality of HCI (cf. [SR88]) neglect the fact that most computer systems offer a very limited range of services, thereby requiring a quality of interaction which is inevitably restrictive in comparison to the multi-purpose interactive scope afforded by natural language.

Concern in the present paper is not with the nature of interaction in the above sense of 'style', but with the functional aspects of man-machine operations. In particular, we focus on the concept of dialogue and what this consists in for man-machine interaction.

[1] In examples of 'conversational systems', where there is no accessible functionality beyond the quality of system interaction, this may be less of a problem (e.g., ELIZA [Wei66].)

A simple sense of 'dialogue' covers all **overt** aspects of man-machine interaction, i.e., the input activity of the user/operator, and output, both visual and aural, from the system. This is the gross level of interaction, described above. Yet, there is clearly more involved in the process of man-machine interaction than such primary i/o. In any complex interactive system, we can identify three conceptually distinct components:

- presentation,
- application,
- mediation [2].

On the one hand, we have the actions of the user or operator. These actions (or omissions) constitute user input (or its absence). Of a similar order to such input, is the output from the computer system itself. This may take the form of text, graphical images, or some other variety of computer expression, such as annunciator signals, beeps or flashes. Such outputs are on a par with operator input because together they constitute a first level of interaction between user and computer system. This is the level of input and output (i/o), where the user *presents* his actions to the system and the system *presents* its responses to the user. For this reason we may call this the **presentation** layer[3].

On the other hand, somewhere conceptually behind this layer of presentation, the actions of the operator are directed toward some functionality which he is trying to manipulate. In a control system, the operator's actions are directed toward an effect on the process. In a database system, the user action would be directed towards retrieval of some information. Whatever the domain, the invocation of some functionality which lies 'behind' the presentation layer is the immediate objective of the user interaction.

Note that such manipulaion is not the immediate effect of interaction. The immediate effect usually lies at the presentation level, and is likely to be some acknowledgement of user input, perhaps by echoing the input to the operator's display screen. The primary role of the presentation layer is to service the user by giving access to the 'remote' facility (process, database, or whatever) so that he can manipulate it in some way and be appraised of any result or effects of his action(s). The functionality which the operator accesses through the layer of i/o at the presentation level, we call the **application**. Ordinarily, this application is not itself a part of the interface. More appropriately, manipulation of the application is the reason that the user requires the interface.

The concepts of presentation and application are adequate for an understanding of the nature of many interactive computer-based systems, particularly where there is a simple mapping from items in presentation to application functions. On the other hand, if access to more elaborate combinations of application functions are sought this requires an additional conceptual realm for the interaction. To presentation and application we must add the realm of **mediation**. This intermediate realm of interaction affords a more complex mapping between presentation and functions in the application [4]. The best way to understand these concepts is through examples. Several illustrations can be drawn from state-of-the-art video cassette recorders (VCRs), which are complex man-machine systems in their own right.

2.1 Example 1: Index search

The first aspect I wish to highlight is a VCR facility called 'indexing'. On my home system, this allows the operator to locate the start point of any recording made previously. In itself, this is rather a high level control concept, which is effected by translating this user requirement into lower level machine functions. The way that this is achieved ably illustrates the separate realms of presentation, application and dialogue.

[2] This classification corresponds to that used by exponents of User Interface Management Systems (UIMS). See [Pfa85] and [Low88] for examples. I use the less opaque concept of mediation in preference to that of dialogue.

[3] Although this layer corresponds to the simple sense of 'dialogue' mentioned above, we shall see that a richer sense of dialogue covers more than presentation.

[4] Hartson [Har89] describes the roles of presentation and mediation as 'external dialogue' and 'internal dialogue'.

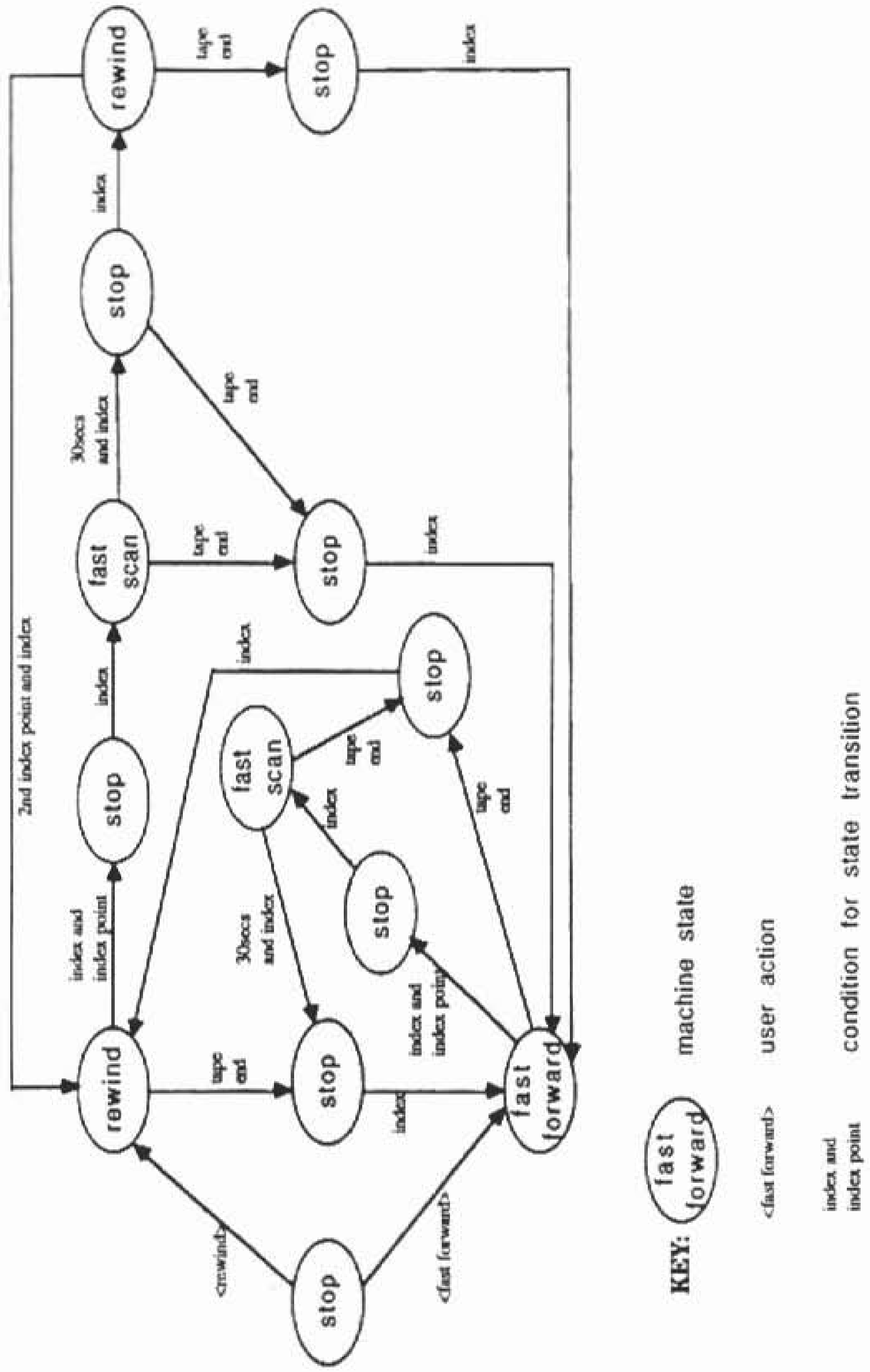

Figure 1: State Transition Diagram for 'index search'.

In this VCR scenario, the indexing facility is switched on or off through a remote control handset. When the appropriate button is pressed, a display light marked 'index' is toggled. If the index facility is enabled and the user switches the VCR to rewind, the machine rewinds the tape until the next previous start-of-recording point is located [5]. Upon reaching the first previous recording point, the VCR switches to play, then immediately to fast scan forward. If the user does not switch the system to play or to stop within thirty seconds the VCR switches from fast scan back to rewind, until it finds a start-of-recording point prior to that first located. Thereafter, the above sequence repeats. If the user does not interrupt this searching, before the system reaches the beginning of the tape, the search sequence resumes in fast forward mode. Details of this 'range of behaviour' are illustrated in Figure 1.

2.2 The nature of mediation

What lessons can we learn from this sophisticated machine behaviour? Following our earlier definition of presentation, we have the operator's input combined with the system output. For the present example, this consists in the user pressing the index button followed by the rewind control. In addition, we include the illumination of the index light on the VCR display panel, followed by the normal display illumination for rewind, play and fast scan modes.

The application side for this example consists in the set of discrete VCR operations which are sequentially executed during the elaborate search procedure, viz., *rewind*, *stop*, *play*, *fast scan*, *stop*, *rewind*, etc. If not interrupted by the operator, this sequence of 'primitive' application functions will change to include *fast forward*, in place of *rewind*.

The true value in this example is to illustrate where mediation takes place. Clearly, the contents of the presentation layer are distinct from those in the application. Yet, if we combine the scope for i/o at the presentation level with the range of available functions at the application level, we do not have all that we require to provide the afore described index search facility. Once the operator has pressed the appropriate two buttons the VCR must execute a complex sequence of primitive functions. How does it know which operations to perform and in which order? Of course this is pre-defined. But the representation of the required sequence of primitive functions does not exist at the same level as these functions themselves. This is the realm of mediation.

The term 'mediation' is used to describe the intermediary role performed at this level of interaction [6]. Our 'index search' example illustrates this role of mediation, whereby a mapping from the level of i/o (presentation) to the level of machine functions (application) is afforded. In the VCR, mediation is not merely a conceptual level but is implemented distinctly from application and presentation. This is apparent from the presence of 'knowledge' which will be found neither in the application functions nor in attributes of the presentation. In particular, the sequencing information required to embody the index search procedure resides at this intermediate level, as does the facility to limit the duration of fast scan to thirty seconds. Furthermore, communication is required between the application level and that of mediation in order to signal the locating of a start-of-recording point on the tape. In response, mediation returns a switching command which changes the application state from rewind (or fast-forward) to fast scan (see Figure 1).

In this instance, the mediation amounts to a plan for organising the low-level behaviour of the VCR into an operation which has higher-level semantics, i.e., index search. The mediation performs an **interpretation** of the operator's high-level goal (index search) in terms of the lower-level machine functions. In this example, the elaboration (interpretation) is **system-directed**, and the mapping from presentation to application is few-to-many. In other instances, the mediation is more user-directed, with the presentation to application mapping being many-to-few. The latter is apparent in a second VCR example.

2.3 Example 2: VCR programming

Programming a VCR to record a particular television channel at a specific time is accomplished through an elaborate interactive procedure, requiring the user to select program mode and then

[5] The VCR automatically marks a record point on the tape whenever a recording is begun.

[6] This equates to the dialogue controller component in UIMS.

enter key items of information (minimally: day, channel number, recording start time, and recording stop time). This interaction, including error handling (values out of range) and correction facilities (for incorrect entries), is controlled at the level of mediation without reference or recourse to application functionality. Activity at the presentation level is rather intensive, whereas the eventual result of all of this interaction is only two application functions i.e., the VCR switches to *record* and, sometime later, to *stop*. There are two phases to VCR programming:

- program elicitation
- program execution

Firstly, in the programming phase, the user is engaged in an interactive procedure with its preset sequence and error handling. The course of this interaction is controlled at the level of mediation and is clearly user-directed (see Figure 2). During this process, mediation attends solely to the user's input[7]. Secondly, in the phase of program execution, mediation handles the switching of VCR functions in accordance with the values given by the user.

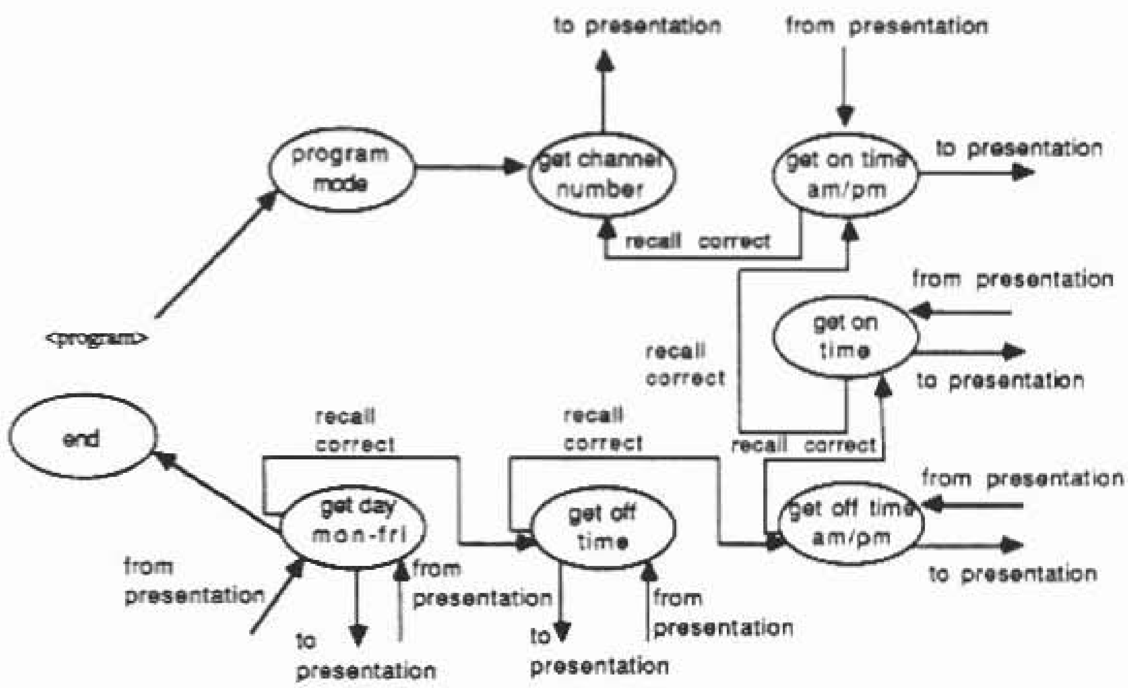

Figure 2: Mediation Level of VCR 'program mode'.

[7]On sophisticated systems the mediation may be capable of concurrent interaction, e.g., allowing the user to program whilst also running the index search.

During programming, the mediation is user-directed. During execution, the mediation interprets the user input in terms of a sequence of system functions, and its operation is system-directed. Unlike the example of index search, VCR programming involves a many-to-few mapping from presentation to application.

3 Dialogue Expertise

Example 2 illustrates that the VCR system has the capacity to 'interpret' the user's high-level goal in terms of a two phase procedure (program elicitation and program execution). Program elicitation involves a sequence of user-directed actions (via presentation) which elicits values for the required variables. Thereafter, these values are used in the execution phase to 'drive' the application level of VCR functions.

Note, however, that the VCR program elicitation could use a superior form of user-directed mediation which allowed entry of variables in ant order, i.e., user selected ordering. The distinction between presentation, application and mediation make it apparent that, for this case, the strict ordering of user interaction is not essential for the system-directed phase of operation. Hence, this is an unnecessary constraint on the user.

Further, the possibility of user confusion may be ameliorated through additional interactive prompting or help. This can readily be built into the user-directed mediation once the scope for separation between the three system dimensions is appreciated.

From this example, we begin to appreciate the significance of mediation, i.e. it gives us the distinction between system- and user-directed activity within a man-machine system. These notions clearly support a concept of dialogue expertise, which sheds light on desirable features in man-machine interaction.

Specifically, the pressing need in man-machine systems is for increased flexibility for operator interaction and greater support for operator activity from the interface system. Progress in meeting these requirements will come through the design of interface systems which embody the functionally separate roles of presentation, application and mediation. Greater flexibility in operator interaction requires that the interface to the man-machine system have more capacity for interpreting operator's objectives.

This aspect of 'dialogue expertise' is embodied in the design by establishing the mapping between high-level control objectives and the manipulation of system functions (system-directed interpretation). The form of user-directed elicitation phase is a separate design issue. Through this appreciation comes greater likelihood of designs for viable and supportive interaction. This aspect of dialogue expertise is achievable through emphasis on user-directed mediation.

Clearly, flexibility and power in these two aspects of dialogue expertise are a major step toward improved man-machine interaction. Limitations in the scope of these capacities shows itself as restrictions on man-machine dialogue. The following example helps to make this clear.

Looking again at the VCR domain, we readily find instances where desirable opportunities for interaction are not satisfied through limitations in system implementation. Thus, although my home VCR affords the desirable possibility of programming the system whilst it is playing a tape it does not admit the equally desirable facility of programming the system whilst it is recording. Such a limitation is less likely to arise in a system designed with an eye to separation of interaction levels, particularly, the role of dialogue mediation.

Dialogue expertise in man-machine systems lies in the sophistication of system- and user-directed mediation. Two features have been identified: the degree to which operator's high-level objectives can be mapped to control of system functions, and the flexibility of user-directed activity in the elicitation of control variables. Flexible design of this combination allows the system to exhibit a degree of 'intelligence' in its interaction.

Further factors may be added to dialogue expertise. For example, the degree to which user-directed mediation employs qualities of the presentation level to reflect individual operator's comprehension and information processing abilities.

The given examples of man-machine dialogues stress the role of specific information concerning operator goals and their mapping to system functions. Such data is the basis for flexibility in mediation, which is the bulk of dialogue expertise. Thereby, the design of man-machine interfaces to exhibit this quality mat readily embrace knowledge-based and expert systems techniques. Such approaches, either for implementation or design prototyping, provide a ready means of explicitly representing the required information and of manipulating its relationships. In turn, the use of explicit and dynamic models which maintain such knowledge of the operators and the application offer the final possibility of dynamic tailoring of dialogue to meet the needs of individual operators and the current domain contexts.

The benefits which accrue from such a design strategy include the possibility of separate design for much of the individual levels of the MMS (i.e., presentation, application and mediation). Further, the individual components may be more readily analysed and tested prior to final implementation (cf. [AR85]). Work towards such systems is underway within the ESPRIT project #857 ('Graphics and Knowledge Based Dialogue for Dynamic Systems'). Therein, the central role of dialogue mediation is made explicit through use of a knowledge-based system which controls both user- and system-directed mediation[8]. This approach addresses improvement in support for operator interaction through explicit representation for the roles of dialogue in man-machine systems.

Acknowledgements

My work in HCI and man-machine systems is funded through the CEC ESPRIT programme within project #857, 'Graphics and Knowledge Based Dialogue for Dynamic Systems'. Partners on this project are AXION A/S, Birkerød, Denmark (prime contractor), Asea Brown Boveri, Heidelberg, FRG., Laboratory for Man-Machine Systems, University of Kassel, FRG., Department of Chemical Engineering, University of Leuven, Belgium, and the Scottish HCI Centre, Glasgow, UK. The present paper draws upon a background of work on dialogue design within the Scottish HCI Centre, especially contributions from J.L. Alty and J. Mullin. My thanks, also, to Erik Hollnagel for judicious comments on preliminary ideas for this paper.

References

[AM87] J. L. Alty and J. Mullin. The role of the dialogue system in a user interface management system. In H. J. Bullinger and B. Shackel, editors, *Human-computer interaction: Interact'87: proceedings of the 2nd IFIP Conference on human-computer interaction*, pages 1007–1012. North Holland, Amsterdam, 1987.

[AR85] J. L. Alty and R. A. Ritchie. A path algebra support facility for interactive dialogue designers. In P. Johnson and S. Cook, editors, *People and Computers: Designing the Interface*, pages 296–315. Cambridge University Press, 1985.

[AW87] J. L. Alty and G. R. S. Weir. Dialogue for dynamic systems. In *ESPRIT '87: achievements and impact*, pages 819–825. North Holland, Amsterdam, 1987. Commission of the European Communities.

[BCM89] T. J. M. Bench-Capon and A. M. McEnery. People interact through computers not with them. *Interacting with Computers*, 1(1):31–38, 1989.

[Har89] H. R. Hartson. User-interface management control and communication. *IEEE Software*, 6(1):62–70, January 1989.

[HW88] E. Hollnagel and G. Weir. Principles for dialogue design in man-machine systems. In *Proceedings of the IFAC/IFIP/IFORS/IEA Conference on Man-Machine Systems*, Oulu, Finland, 1988.

[8] The complete system, known as 'GRADIENT', is detailed in [Wei88]. Our approach to dialogue design is described by [AM87] and [AW87]. See also [HW88].

[Low88] J. Lowgren. History, state and future of user interface management systems. *Sigchi bulletin*, 20(1):32–44, July 1988.

[Pfa85] G. E. Pfaff, editor. *User Interface Management Systems*. Springer Verlag, Heidelberg, 1985.

[SR88] D. L. Sanford and J. W. Roach. A theory of dialogue structures to help manage human-computer interaction. *IEEE Transactions on Systems, Man, and Cybernetics*, 18(4):567–574, 1988.

[Wei66] J. Weizenbaum. ELIZA- a computer program for the study of natural language communication between man and machine. *Communications of the ACM*, 9:36–45, 1966.

[Wei88] G. R. S. Weir. Graphical dialogue support for process supervision and control. In P. V. S. Rao and P. Sadanandan, editors, *SEARCC'88: modern trends in information technology*, pages 417–422, New Delhi, 1988. Tata McGraw-Hill.