DOES THE HUMAN COMPONENT IN A NETWORK HAVE A PROTOCOL?

Mildred L. G. Shaw and Brian R. Gaines

Department of Computer Science, York University,
Downsview, Ont. M3J 1P3, and
Department of Industrial Engineering, University of Toronto,
Toronto, Ont. M5S 1A4.

Summary

The need to standardize protocols in the development of communication networks is generally recognized. It allows modules to be developed independently of one another that can be integrated together in a system simply and with a guarantee of effective operation. Hence networks can be set up in the short term which will be able to continue to function effectively in the long term through module change, enhancement, replacement and substitution. However, one important 'module' in most networks is the human operator interfaced to the network through a variety of terminal devices. This paper investigates the possibility of treating the human components in networks as modules with standardized protocols and hence gaining the same benefits for the total person-computer network as for the technical system alone.

Introduction

Communication networks are now both technically feasible and sufficiently low in cost to make them commercially viable in a wide variety of applications [1]. For the equipment the network protocols have been defined in great detail and, although there is no industry-wide agreement on standards, nevertheless the necessity for standards is accepted. In human factors this standardization and perception of needs is far less apparent. The people who have to use the systems are faced with a bewildering variety of protocols. The equipment designer for a network is able to work from detailed manuals giving exact protocols when interfacing to other network equipment but not when interfacing to people.

This difference in the status of equipment and people as network components is becoming even more anomalous as protocol conversion becomes increasingly important. Many conversions, such as that from text to speech, strongly involve human factors and cannot be developed without an adequate model of the human component.

In this paper we show that it is possible to establish a protocol for the human component in communication networks using the same framework as that adopted for local area networks. At the lower levels models of the properties of human sensation and perception are available. At higher levels models of human cognition enable practical predictions to be made. It is in the adaptive behavior of people, as they learn about the network, that novel problems arise and even at this level adequate models can be given.

People as Nodes in the Network

At some nodes in any network protocols are required for interaction with people. People-computer protocols have some similarities with computer-computer protocols and analogous rules apply to both at a systemic level. However, people-computer protocols need also to take into account characteristics of people that are not found in interaction with equipment -- notably that a person is adaptive and explores and models his or her environment. In this respect they are similar to people-people protocols and much of the psychological literature is applicable.

The ergonomics [2] of human interaction with terminals corresponds to the "physical" and "link" levels of the ISO model for open systems interconnection. In this paper we are concerned with the higher levels of the protocol where error detection and correction occurs and where information presentation is significant. We are particularly concerned with the adaptive processes of human learning that currently have few analogues in computer networks. The remainder of this paper presents rules for engineering effective dialog with people that have been developed through experience with a wide range of interactive systems [3], and compares and contrasts them with the protocols for communication between purely technical systems. It is shown that people-computer protocols can be developed and standardized and that this should be seen as an important part of network technology.

Basis for People-Computer Protocols

One interesting feature of people is that they generally do not have manuals which define the appropriate protocols to be used in operating with them. One way of obtaining this information is to ask them how they would like the interaction to be structured. However, many of them come to a network for the first time with no experience of interaction with a computer system and hence their answers may be misleading. The first rule in designing a protocol is to avoid this problem.

Rule 1 Introduce through experience: Interactive systems need to be experienced before their operation can be discussed meaningfully. Get prospective users onto a terminal on a related, or model, system before discussing their preferred protocol for interaction with their own system.

As well as mistrusting a naive user's view of a suitable protocol we also need to distrust a naive system analyst's view. The terminology, presentation and sequence of dialog should reflect the user's model of the activity.

Rule 2 Use the users' models: Base the protocol on the users' models of their activities so that the interactive dialog is similar to a conversation between two users mutually accepting this model.

People are continually changing their modes of operation, particularly in settling after the perturbation of the introduction of a new system. Hence the first two rules above need continuous application.

Rule 3 Design never ceases: Use the flexibility of the system through programming so as to close the adaptive loop between system and user through yourself as designer.

This can be done most effectively through the network's data collection capabilities.

Rule 4 Log activities: Use the network to maintain selective records of system and user activities and provide programs to analyse these in terms of, for example, errors, broken down by user and by dialog sequence.

Problems of User Adaption

The human ability to model and adapt, and their continuous application of it, causes new problems for the protocol designer.

Rule 5 The User will Model the System: Do not assume that the user is a passive static system to be controlled, modelled and directed by the computer. Evaluate the protocol in terms of its effect on an actively changing user who is attempting to comprehend the system.

One major problem for the user in modelling the system is random behavior on its part, such as delays in a timeshared system.

Rule 6 Avoid acausality: Make the activity of the system a clear consequence of the user's actions.

One important feature of a system to aid a user modelling it is consistency.

Rule 7 Consistency: Ensure that all terminology and operational procedures are consistently applied throughout the protocol.

Another important feature to aid modelling is uniformity.

Rule 8 Uniformity: Ensure that all terminology and operational procedures are uniformly available throughout the protocol.

Person or Computer in Control

There is a difficult choice to be made in many situations as to who should have primary control over the interaction. It is tempting to try to analyse the user characteristics and take them into account automatically, e.g. by switching to a more supportive protocol when naive user behavior is detected. However, the information to do this accurately is not easy to obtain.

Rule 9 User should dominate computer: Either the computer or the user should dominate the interaction or there will be instability. If the computer is to dominate it must be programmed to, and have sufficient information to, model the user. If the user is to dominate then the computer system must be simple to understand. At the present state of the art the user should dominate the system.

A simple way of enabling the skilled user to adapt the system himself is to allow grouping of responses.

Rule 10 Parallel-sequential tradeoff: Allow the user maximum flexibility to make his responses holistically (in parallel) or serially (in sequence) according to his wishes.

In enabling the user to control the interaction it is important that he is kept aware of the state of the protocol.

Rule 11 Make the state of the dialog observable: give the user feedback as to the state of the dialog by making an immediate unambiguous response to any of his inputs which may cause the dialog to branch - the response should be sufficient to identify the type of activity taking place.

Another important feature in enabling the user to cope with the system is to give him access to its protocols where they apply.

Rule 12 Query in context: Distribute information and tutorial material appropriately throughout the protocol to be accessed by the user through a simple uniform mechanism, e.g. responding with a "?".

Responses to queries should not be verbose, but additional information should be available if required.

Rule 13 Query in depth: Organize the response to queries so that the user accesses brief memory aids first at any point but has further access to more detailed explanations.

The documentation of the protocol should enable the user to put himself in the context of interacting with the system.

Rule 14 User manuals should be based on actual user dialog: Illustrate the use of the protocol by showing actual dialog sequences that achieve specific objectives. Illustrate the structure of the system through dialog sequences where queries are used to elicit information about the protocol.

A protocol complying with the rules given here should be its own best training aid.

Rule 15 Train through experience: get users interacting with an actual system as their initial training and introduce them to the system facilities such as query-in-depth.

Error Detection and Recovery

Error detection and recovery procedures are as important with people-computer protocols as with computer-computer protocols. They should take into account the person's need to explore the system characteristics and to be able, and to feel able, to recover readily from any consequences of this.

Rule 16 Validate data on entry: Check syntax, and values against norms.

However, norms that filter errors also may reject actual data.

Rule 17 Be flexible in validation: Beware of rejecting data or querying too much as being outside norms. Allow the user to over-ride the error checking where appropriate.

Always allow also for undetected errors and give the user back control at events with major consequences.

Rule 18 Have the user himself revalidate major updates: Show the consequences of major actions clearly before acting upon them and ask for user confirmation.

When a user detects errors he should be able to exit without problems.

Rule 19 Provide a reset command: that cleanly aborts the current activity back to a convenient checkpoint. The user should be able at any stage in a transaction to abort it cleanly with a system command that takes him back to a well-defined checkpoint as if the transaction had never been initiated.

For minor entry errors the abort and restart is wasteful and a simpler mechanism is desirable.

Rule 20 Provide a backtrack facility: that allows a user to return through the dialog sequence in reverse.

Errors in entry are minimized by providing default values representing the most likely entry.

Rule 20 Provide default values: that are accepted by the user by, for example, returning a null message.

This is also useful for record update and correction.

Rule 21 Make corrections through re-entry: use the entry dialog with default field printouts from a record as a means of correcting the record.

Conclusions

The human component in a network has many properties in common with the computer components and there is much in common between the ergonomics of the person-computer interface design and the protocols of the computer-computer interface design. The human component does have some unusual characteristics of modelling the interaction and adapting to the system and these require separate consideration in establishing person-computer protocols. However, when this is done a set of rules can be established that provide a basis for a protocol that provides effective person-computer **dialog engineering.**

References

1. Davis, G.R. (ed.) **The Local Area Network Handbook.** New York: McGraw-Hill, 1982.

2. **Ergonomic Principles in Office Automation.** Stockholm: Ericsson Information Systems, 1983.

3. Gaines, B.R. & Shaw, M.L.G. "Dialog Engineering" in Sime, M.E. & Coombs, M.J. (eds.) **Designing for Human-Computer Communication.** London: Academic Press, 1983.