Negotiating User-Initiated Cancellation and Interruption Requests

Manuel A. Pérez-Quiñones
Human-Computer Interaction Laboratory
Naval Research Laboratory, Code 5512
Washington, DC 20375-5337
perez@aic.nrl.navy.mil
+1-202-767-9003

John L. Sibert
Department of EE and CS
The George Washington University
Washington, DC 20052
sibert@seas.gwu.edu
+1-202-994-4953

ABSTRACT
Interruptions and cancellations are important parts of a user interface, yet they are treated as special cases in user interface design and notations. In an effort to build a dialogue notation that allows for effective definition of these commands or user turns, we present a behavioral definition of interruptions and cancellations. We show several examples of how our definition accounts for different forms of behavior. The behavioral definitions provided here are a step towards providing better support for the definition and implementation of these turns.

Keywords
Cancellations, interruptions, human-computer dialogues.

INTRODUCTION
A desirable property of all human-computer dialogues is to put the user in complete control of the dialogue flow. To achieve this goal, the user must be able to interrupt the system at any time and be able to request a cancellation of any dialogue. We assume a synchronous dialogue between the user and the computer with each taking alternate turns, much as in human conversations. If interruptions and cancellations are allowed, the system response must be consistent with how humans deal with interruptions and cancellation in human conversations. These are rarely accepted without some form of negotiation between the participants. This negotiation helps the participants ground the conversation [1] and allows them to consider the possible implications. Not all interruptions and cancellations need to be negotiated. Some may be accepted without any negotiation because the possible "damage" done is minimal.

Cancellation requests, also referred to as the abort command, are difficult to define using dialogue notations and difficult to implement [3]. One reason for this is the poor understanding of what should happen, from a behavioral point of view, when users request cancellations. Also, the representation of these commands in dialogue notations is usually done as an afterthought. As a result, a complete definition of the abort semantics would make a "complete mess of the hierarchical structure of the dialogue description" [2] page 263. The notations are usually extended with global states, or special transitions to handle cancel requests. In addition, they provide no guidance or suggestions on what would be the appropriate way to implement these user turns.

In this paper we present a behavioral definition of interruptions and cancellations. This definition is based on a behavioral description of how humans negotiate interruptions and cancellations. A dialogue model [4] is being developed that considers the behavioral as well as the constructional issues related with the definition and implementation of interruptions and cancellations.

BEHAVIORAL DEFINITIONS

Cancellation Requests
A cancellation occurs when the user decides not to continue the current exchange and terminates the dialogue. A cancellation request can occur *only* at a user's turn*. But more importantly, it could occur at *any* user's turn. It is a request to terminate the dialogue in which the request is made. The system, as a response to the user request, has three options: (a) accept the request and terminate the dialogue, (b) reject the request and leave the dialogue unaffected, or (c) initiate a negotiation with the user on the consequences of accepting or rejecting the request. Furthermore, the results of the negotiation *can only be*: either a user acceptance of the consequences of the cancellation and thus a system acceptance of the request, or a user refusal of the consequences and thus a system rejection of the request. Figure 1 shows all the possible paths taken based on this definition.

The second user's turn state, labelled with an asterisk (*) in the figure, is a special state where the user's actions are limited to acceptance or rejection of the negotiation. This restriction prevents recursive definitions of cancellations, where each cancellation leads to negotiation over another cancellation (i.e. cancellation of a cancellation of a cancellation…). A cancellation request can only be voided by following the *reject* path out of the user's negotiation state.

Interruption Requests
An interruption occurs when the user desires to take a turn in the interaction while the system is in the middle of one of its

*. We are only concerned with user-initiated requests.

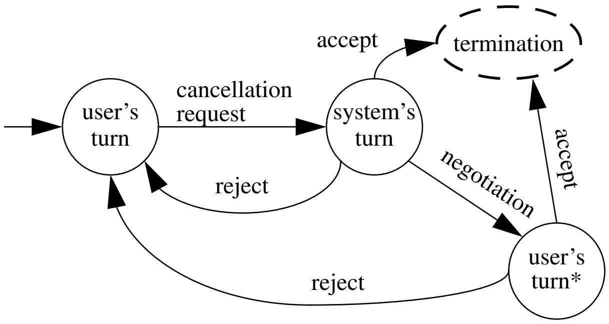

Figure 1. Cancellation Request Semantics

turns. For the user to take a turn at this point, he/she must request an interruption first. This results in an asynchronous event which must be handled by the system*. The possible outcomes of an interruption request are: (a) the system ignores the request, or (b) the system accepts the interruption request and thus the dialogue is interrupted. If the request is accepted, a negotiation with the user will determine when to resume the interrupted dialogue. Figure 2 shows the possible paths taken in an interruption request based on this definition.

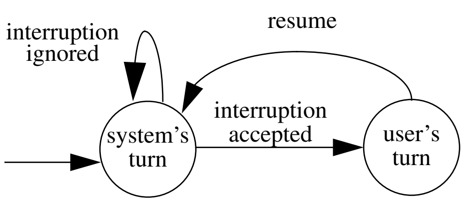

Figure 2. Interruption Semantics (partial)

Combination of Interruptions and Cancellations

Interruptions, as defined above, have very limited usefulness. But when combined with cancellations, all the power of interruptions is realized. Because a cancellation request is a user-initiated turn that can occur at any user's turn, a cancellation request can occur at the user's turn in the interrupted state of a user-initiated interruption. A combined diagram can be obtained by superimposing the initial user state of Figure 1 with the user's turn in the interrupted state of Figure 2. The resulting diagram, shown in Figure 3, contains the complete definition of an interruption request.

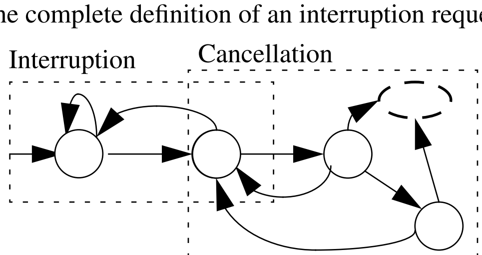

Figure 3. Interruption Semantics (full)

Examples

A quick study of interruptions and cancellations in human-computer interfaces reveals many different behaviors. In this section, we briefly describe some examples using the semantics presented here.

On Unix systems, Control-C and Control-Z are used to interrupt processes. Control-C is an interruption followed by an implicit cancellation. Its behavior is specified as an interruption request, which is followed by an automatic cancellation request that is accepted by default. Control-Z, on the other hand, is just an interruption request that is automatically accepted. The process is left interrupted until the commands *fg* (foreground) or *bg* (background) resume the process. A process cancellation request can be issued with the *kill* command, which is accepted without a negotiation.

On the Macintosh, a Command-. (period) while the computer is printing, is defined identically to a Control-C in Unix; the print job is terminated. A Cancel button in a dialogue box is defined as a cancellation-request that is, in most cases, automatically accepted.

Other systems provide the full behavior of the interruption request. For example, interrupting an ftp download in Netscape is defined as an interruption request with a negotiation (Are you sure you want to abort the download?) over a cancellation request. Accepting the consequences of the cancellation terminates the download; rejecting the consequences automatically resumes the interrupted download.

Use of Behavioral Definitions

The definitions presented work as templates for each instance of the handling of these user turns. Definition of new handlers require identifying which system transitions will be taken automatically and which will be done as a result of negotiation with the user. The examples above contain some transitions that were taken without user negotiation (e.g. cancellation request in Control-C). A second step is to determine which (if any,) user actions are to be collapsed into one. In the Netscape example, the user's interruption is also a cancellation request. Therefore, two transitions (interruption-accepted and cancellation-request) are made based on a single user action.

REFERENCES

1. Clark, H.H. and Brennan, S.E., Grounding in Communication, in *Shared Cognition: Thinking as Social Practice*, J. Levine, L.B. Resnick, and S.D. Behrend, Editor. 1991, APA Books: Washington, D. C.
2. Dix, A., *et al. Human-Computer Interaction*. Prentice-Hall, 1993.
3. Myers, B., ed. *Languages for Developing User Interfaces*. 1992, Jones and Bartlett Publishers: Boston, MA.
4. Pérez, M.A. Conversational Dialogue in Graphical User Interfaces: Interaction Technique Feedback and Dialogue Structure, in *Proceedings Companion of the ACM CHI'95 Conference on Human Factors in Computing Systems*, (Denver, Colorado, 1995), Addison-Wesley, pp. 71-72.

*. The event is asynchronous because it is generated by the user during the system's turn.