Title

Demon Timeouts Limiting The Life Span of Spontaneous Computations in Cognitive Models

Journal

Proceedings of the Annual Meeting of the Cognitive Science Society, 3(0)

Author

Small, Steven

Publication Date

1981

Peer reviewed

Demon Timeouts

*Limiting the Life Span of Spontaneous Computations in Cognitive Models**

Steven Small

Department of Computer Science
University of Rochester
Rochester, New York 14627

&

Departement d'Informatique
Universite Paris VIII Vincennes
93200 Saint Denis FRANCE

Introduction

The limitations in human short-term memory, of whatever extent they are determined to be, must be accounted for in constructing cognitive models. The view of this memory as a data (or knowledge) store leads to models with a fixed-length buffer of quickly accessible *short-term* data, and general procedures for replacing items in that buffer with other items (see Marcus [1979], for example). Another vantage point on this question comes from viewing the this memory in terms of the active problem solving processes of the model. This perspective suggests building models with time-limited processes that may not remain active indefinitely. Just as the size of an individual constituent in a fixed-length buffer may depend on the nature of that item, so the time limit of an active process (even the units of time employed) must be allowed to depend on the nature of its functioning.

Every active process in a cognitive model must be associated with another process, called a *timeout process*, which can force its premature termination (its *timeout* or *expiration*) on specific criteria. The timeout specifies how long the process (the *parent process*) may remain active, in terms of specific time units or model events, and in addition, what actions to take (or processes to initiate) on expiration. Note the enormous significance of this organization: The processing of individual actors in a model can depend on either the existence or the non-existence of relevant information. In terms of human problem-solving, this is analogous to our basing some decision on the explicit knowledge that we do not know something. Often the knowledge that we do not know something can be as useful to solving a problem as any other sort of knowledge.

*The initial research motivating the ideas described in this report was supported by grant NSG-7253 from the National Aeronautics and Space Administration to the University of Maryland. During the writing of this paper, the author was supported by a Fulbright-Hayes Lectureship in Artificial Intelligence at the Universite de Paris, and computer facilities were provided by the *Institut de Recherche et Coordination Acoustique/Musique*. The support of NASA, the Fulbright-Hayes program, and IRCAM are gratefully acknowledged.

Word Expert Parsing

This kind of approach has been taken in the modelling of human language comprehension by the *Word Expert Parser* (WEP). WEP views language comprehension as interactions among a large number of human cognitive processes, including word-based context analysis processes called *word experts*, and other mechanisms such as belief maintenance, general problem solving, and so forth. The system models this *understanding as memory interactions* through the message-passing behavior of its different constituent processes. Evaluation of the system ought not to be made through its output; rather, its important aspects are (a) the organization and representation of the model processes, (b) the order of execution of dependent processes and the concurrence of others, and (c) the messages exchanged among the processes.

An important aspect of WEP is that its processes are not permitted to wait indefinitely for information they might need to perform their functions. In understanding the meaning of a word in some sentence, for example, often a reader requires some knowledge of the context following it. How long would a reader wait for a piece of disambiguating information before choosing one of the competing interpretations of the word? Does this wait depend on general syntactic or semantic criteria, or on the particular words involved and their idiosyncratic couplings with other words? And a last question: How can a model of language understanding (or of any cognitive mechanism) account for this limitation in active short-term memory processing?

As a completely procedural distributed model, WEP does not contain some finite fixed length buffer to represent a short-term memory, but models the memory limitation with processes that have a strictly limited life span. The human understander does not wait indefinitely for information that could aid his/her understanding, and the model cannot either. When a process in WEP decides to await some piece of information from another process, a pattern-invoked *timeout process* is initiated to monitor the duration of the wait. In the ideal case, the desired information quickly becomes available in the model, and the awaiting process promptly receives it. If the information does not appear within a certain time, however, the timeout process must command the awaiting process to go on without it.

Timeouts

The general notion behind these *timeouts* is a simple one. Anytime a cognitive model would ordinarily create a demon, or any pattern-invoked process (a *spontaneous computation* [Rieger, 1977]), it should instead create a pair of competing demons, a *parent* demon and a *timeout* demon. The timeout demon should be guaranteed to trigger within a certain fixed amount of processing by the model as a whole. If the parent triggers first, the timeout disappears and the parent process performs its pattern-invoked operations. If the timeout triggers first, there are two possibilities; either (a) both the timeout and the parent disappear immediately, or (b) the timeout disappears immediately and the parent performs certain special timeout operations and then disappears. The timeout operations are process-specific and only execute in the case when their associated pattern-invoked process (demon) does not trigger within its allotted life span.

The result of this arrangement is to place a limitation on the duration of applicability of demons in the model, preventing them from waiting indefinitely for desired information. In this way, futile waits for information that will never arrive can be avoided, and excessively long waits -- far exceeding human memory limitations -- can be disallowed. Furthermore, by appropriate use of timeout operations, processes in a cognitive model can be triggered by either the invocation or the non-invocation of some pattern in the memory. By not knowing something within a certain time limit of model events, and by knowing about this lack, a system can make valuable inferences.

Timeout Criteria

The question of how long to spend looking for helpful information before going on without it, whether in problem-solving or cognitive modelling, is clearly not a new one. How far ought one to search a promising path in a chess tree, or how many associations in a semantic memory ought to be examined, before available knowledge is used to come to some conclusion, even if not the right one? Various answers to this problem have been studied, although from a different perspective than the one suggested here. From our vantage point, there are two interrelated questions: (a) What objects or events might be used to trigger demon timeout in a particular cognitive model? And similarly, (b) What sequence of events in the model can be used to measure the duration of the wait for desired information?

The progression of events in a cognitive model depends fundamentally on the nature of the particular activity being modelled. In natural language understanding, a person reads words and sequences of words of various lengths, creates conceptual conglomerates, follows character and plot development (in reading a story) or speaker intentions (in participating in a dialogue), makes hypotheses and has some hypotheses fulfilled and others rejected, and so forth. Each of these facets of language understanding represents an event that might be used to trigger demon timeout in an understanding model. Furthermore, the events could be counted and treated as units of time for measuring durations in the model.

For a model of government decision-making or administrative behavior, perhaps the number of memoranda reaching a particular department would be a good gauge on the progress of the problem-solving activity. Another measure might be the number of responses from departments participating in a decision, or from sub-departments of a modelled one. Recent history suggests that in modelling government processing, the number of adverse newspaper articles or adverse decisions in other branches of government could be useful. Problem-solving systems must not wait indefinitely for data needed to make inferences; better make a wrong inference than come to no decision at all. The impatience of people in such decision-making situations makes this all the more important in cognitive modelling.

Timeouts in Word Expert Parsing

The processes in the Word Expert Parser limit their wait for relevant information by employing several of the yardsticks mentioned above; i.e., the system maintains counts on certain model actions intended to correspond to relevant events of natural language understanding. At present, these include the total number of words read by the system, the number of conceptual conglomerates created by the model, and the number of sentence breaks encountered. While these are fairly straightforward measures, our current research will give WEP two new timeout criteria. In both cases, the timeout processes might be considered meta-processes, as their actions are based on the functioning of other processes (rather than on their results or on messages from them).

The first new waiting criterion causes a timeout on the termination of some model process. Thus, a process might wait for some information just as long as some other process were still active (and could provide it, for example). The moment the specified process terminated, the demon for the awaiting process would expire, and perhaps carry out its special timeout actions before terminating itself. This timeout criterion has been simple to implement. The other new waiting criterion causes the timeout of some demon on the creation of an identical demon (i.e., one awaiting the same information) by another process. To put that another way, the system may permit a process to wait for some datum or event just so long as no other process initiates a wait for the same datum or event.

This particular strategy aids in modelling a phenomenon that occurs frequently in natural language comprehension. The problem of anaphoric reference illustrates the issue. When a computer model such as WEP begins reading a definite noun group, it ought immediately to initiate a reference process to search for the referent of that group. How long should this reference process be permitted to search before using available information to conclude its processing? One answer might be that when the understanding of further discourse required knowledge of the referent, the process should come to some decisions immediately. The timeout actions of a reference process might provide some default substitute for the desired referent, for example, while at the same time initiating some new process to proceed along a different path based on the (known) situation at hand.

The timeout of a process allows it to come to some conclusion without waiting indefinitely for data that will insure the correct decision. In this example, the timeout of the reference process takes place when the understanding of subsequent text requires information about the searched-for item. In this way, the reference process can be forced to make some inference (e.g., the definite noun group has no referent) when the overall understanding process requires it, but not before. Ideally, of course, the correct referent would be found before then, and none of this would be necessary. Timeout demons exist, however, solely for those situations when the ideal context for making correct inferences does not present itself. They are a means for deciding when to make an inference without all the facts being present.

Summary

An important facet of general problem-solving, and likewise cognitive modelling, is how long to spend looking for helpful information before going on without it. In a computer model based on spontaneous pattern-invoked processes (demons), there must be a means for limiting the duration (in terms of some measure of model convergence) of their applicability. This paper suggests ***timeout processes*** as a means of limiting the life span of pattern-invoked demons in a cognitive model. Every process in a model must be associated with a timeout process, which monitors various specific model events and prevents the process from remaining active indefinitely. If the process does not finish its activity before being stopped by its timeout, it may nonetheless perform certain timeout actions before disappearing. These actions can make use of the (possibly valuable) information that the trigger pattern for the process never arrived. Timeouts thus enable processes to use knowledge about not having certain knowledge to help make important inferences in cognitive models.

References

Marcus, Mitchell P. (1979), ***An Overview of a Theory of Syntactic Recognition for Natural Language***, AI Memo #531, MIT Artificial Inteligence Laboratory.

Rieger, Chuck (1977), ***Spontaneous Computation in Cognitive Models***, Cognitive Science, vol. 1, no. 3.

Small, Steven (1980), ***Word Expert Parsing: A Theory of Distributed Word-Based Natural Language Understanding***, TR #954, Department of Computer Science, University of Maryland.

Small, Steven (1981), ***Toward a Cognitive Mechanics for Distributed Modelling***, Technical Report, Department of Computer Science, University of Rochester (to appear).