

# Temporal Control Structures In Expert Critiquing Systems

Surya Ramachandran

pcOrder.com

6034 West Courtyard Drive, Suite 210

Austin, TX 78730

Surya.Ramachandran@pcorder.com

David C. Wilkins

Beckman Institute

University of Illinois

405 North Mathews Avenue

Urbana, IL 61801

dcw@uiuc.edu

## Abstract

*Critiquing in time critical and resource constrained domains requires special control structures that allow the expert critiquing system to compare user actions with that of its own. This is necessitated by the fact that the priorities of actions in such domains change over time and the system needs to be robust enough to reflect this in its critiques. This paper describes a model of expert critiquing where the use of blackboard architectures, temporal control structures and data manipulation and differential critiquing are effectively combined to tackle this issue. The proof-of-principal critiquing domain is novice experts training in the area of Ship Damage Control.*

## Introduction

Time has been a representational issue in some critiquing expert systems but its actual use has been more for an ordering of action sequences or plan operators. The lineage of such representations can be traced in planning systems like TPLAN [6] that use classical time intervals [1] for information storage and reasoning that is limited over the intervals. In this paper we focus on some of the forms of representation of time and the reasoning structures needed to generate critiques in time critical domains.

With the growing acceptance of the role of expert systems in the decision making process, there has been a definite need for more than simple explanation from such tools. The earliest references to systems that critique the human decision making process are the ATTENDING[11] critics. The development of expert critiquing systems can be traced from ONCOCIN [8] that was originally developed as an expert system and was later converted into a critiquing system, and HACKER [18] which was used to critique the output of other programs to the more recent efforts of plan generation and recognition [4].

The current domain that is being used to develop and test this model is the area of Ship Damage Control. The damage

control domain is characteristic of time critical and resource constrained decision making and is discussed in terms of the type of representations and reasoning structures that the domain demands. Further, there is a multimedia user environment (IDCTT) [10] that is currently operational. This paper describes the design of an expert critiquing system that integrates blackboard architectures, temporal control structures and differential expert critiquing (fig. 1). In the domain of crisis management, there are many actions that can be taken at any given instance of time. The difficulties often encountered are the identification of the various crisis and the next action that has to be performed for each of them. The Blackboard model of deliberation, scheduling and action forms the basis of the of critiquing in this domain by having the scheduler that numerically orders the actions at each problem-solving cycle. The critiquing module differentiates the user actions with respect to an equivalence class of next best actions to generate critique for the event. These critiques are displayed by the user interface during the scenario and are consolidated after the problem solving process for a more comprehensive critique.

## The Process of Expert Critiquing

Critiquing can be defined as the process of refining the users decision making ability by pointing out mistakes, suggesting alternate actions and being able to defend the goodness of a certain set of actions. The design of expert critiquing systems are either differential or analytic in nature. Differential critics [16] generate the next best action or operator that the user must initiate and then compare this to the actual decision taken by the user. If a difference is observed, these systems analyze this deviation (either in terms of a deviation from the optimal path [3] or as a possibility of gaining knowledge [17]) to generate a criticism of the users initiative. Components of these systems include, an expert system module to come up with the next best action to be performed, a differential analyzer that compares the user's action to the one generated by the expert module for the amount of deviation from what is considered optimal, and a dialog generation component that verbalizes these differences. Differential analysis techniques

in expert critiquing depend on accuracy of the solutions that are generated by the expert module by assuming that the expert knowledge always produces the optimal strategy. The incompleteness and inaccuracy in domain knowledge can be corrected by an interactive knowledge acquisition process aimed at capturing the missing or incorrect piece of information by analyzing the users rational.

Analytic critics rely more on the users proposed plan of action sequences (or focus of attention) as a basis for the generation of critiques. These systems do not have a need to generate any solutions on their own but instead analyze the proposed solution for errors and deficiencies. The critiques often tend to rely on the domain for general problem solving approaches. This allows them to function in domains that are relatively more incomplete as their knowledge is not directly solution dependent but rather on alternate approaches to handling a given problem.

Classification of critiquing expert systems can also be on the basis on the type of critique and the point of its generation. In the medical domain it is often found that the physician relies more on the support structures that are generated for explaining the action rather than the action itself [19]. Silverman [16] distinguishes the types of critics and the strategy of their generation and display.

*Influencers* are preventive critics that divulge common errors and the kind of decisions that are expected before the user begins the problem solving process. *Debiasers* are more corrective strategies that are applied to the actions that have already been performed by the user. The suggestions are based on errors that the user has already committed. "Debiasers often check for Failure Modes (FM), explain the Effects (E) of the failures, explain their Causes (C), and then attempt to make repair actions (A). This is the FMECA strategy." *Directors* are stronger critics that transfer control from the user who is left to record data, to the system which tries to generate a deeper explanation by having the user analyze and compute a solution.

## Temporal Representation in Expert Systems

Temporal models can be classified according to their *primitives of time* [5]. In *point-based* theories the representation deals with individual points of time corresponding to a start and end times for events. In such theories reasoning is achieved over precedence, successions and simultaneity, and their negations and combinations. On the other hand, *segment-based* (or *interval-based*) theories represent intervals of time that correspond to the duration of the event. Allen [1] introduced a temporal logic which is represented as distinct time intervals and the various associations between them. An interval can be described as a discrete segment of time used to stipulate a period

of time over which an action or event holds. As there can be several intervals on a time line, Allen also introduced some common relationships and reasoning over them that we will represent in our own model.

In [2] the authors look at various components of temporal databases in terms of *time map maintenance*, partially specified constraints and resolution strategies for conflicts in the database. Shoham and Goyal [14] use time intervals and assertions on them that allow the significant objective of being able to interpret over intervals of *zero* length. Further, their representation uses a pair of end-points for intervals of time that are considered *linear* and *dense*, and they do not assume *homogeneity* in the truth of the assumptions over sub-intervals. Also, the automatic referencing over interval hierarchies in the TIMELOGIC [7] system has a strong correlation to the representational issues being tackled here.

## The Damage Control Domain

The damage control domain is one of complex and time-critical decision making. The domain concerns crisis management on Naval ships and has been further narrowed to involve critiquing the Damage Control Assistant (DCA) on the DDG class of destroyers. Damage control can be defined as the process of allocating resources and personnel to combat crises (e.g.: fire, smoke, flooding, rupture, etc.) in order to keep the ship in a maximum state of readiness. The system developed here is to be used as a damage control trainer in damage control training schools and is based on the DDG51 (Arleigh Burke) and the DDG53 (John Paul Jones) ships.

The main job of the DCA during a crisis can be represented as a hierarchical framework of goals and sub-goals. At the root of this tree (the main goal) is *Total Ship Survivability* (TSS) [13] which includes *Battle Readiness* (operability of weapon systems) and *Damage Control* (the identification, isolation and eradication of crises) which can be further broken-up into their individual constituents. The sub-goals at the lowest level can be associated to individual actions of the *DCA-world* or to a set of observed events. The DCA's actions at the lowest level of this hierarchy relate to crisis management using *identify*, *isolate* or *contain* and *fight* as a general principle to handle any crisis. *Identify* allows the DCA to confirm the existence of the crisis and any information that the investigation team can gather from the disaster site. The *isolation* of a crisis normally includes the setting of boundaries (smoke, fire or flood) to restrict the crisis within its space and electrically and mechanically isolating it. Finally the DCA will command a team of people to *fight* the crisis. This could be to fight the fire, de-flood the space, repair a rupture, etc.

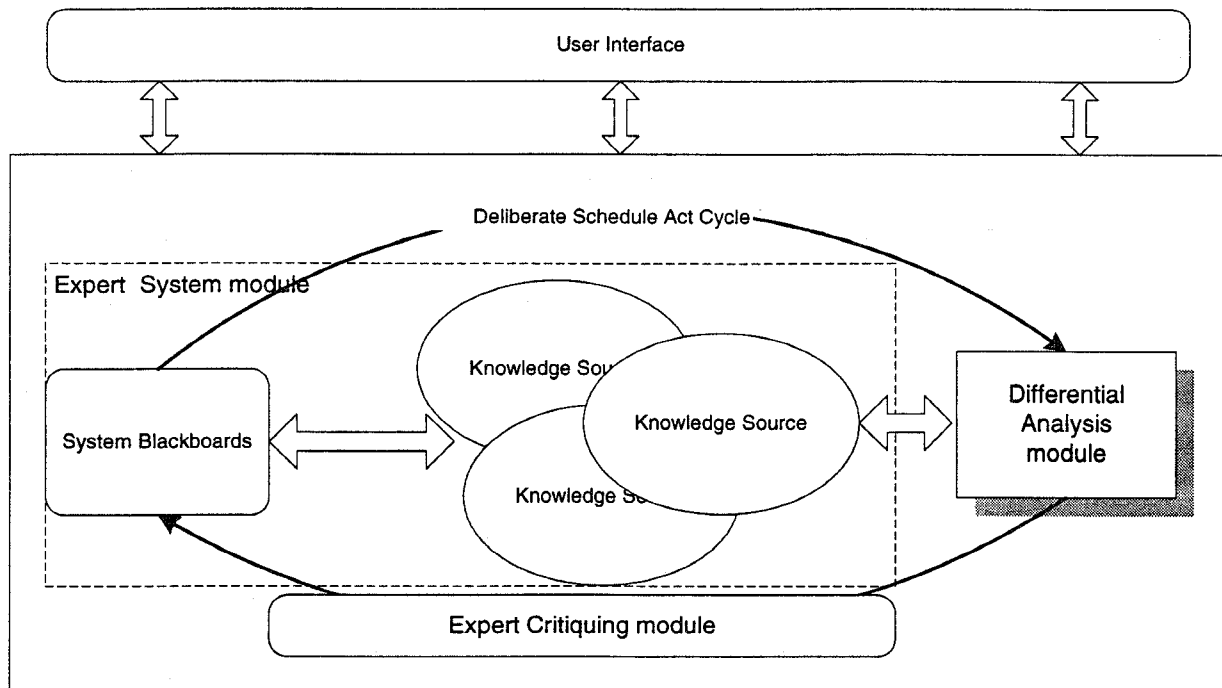


Figure 1: System Overview

## Project Overview

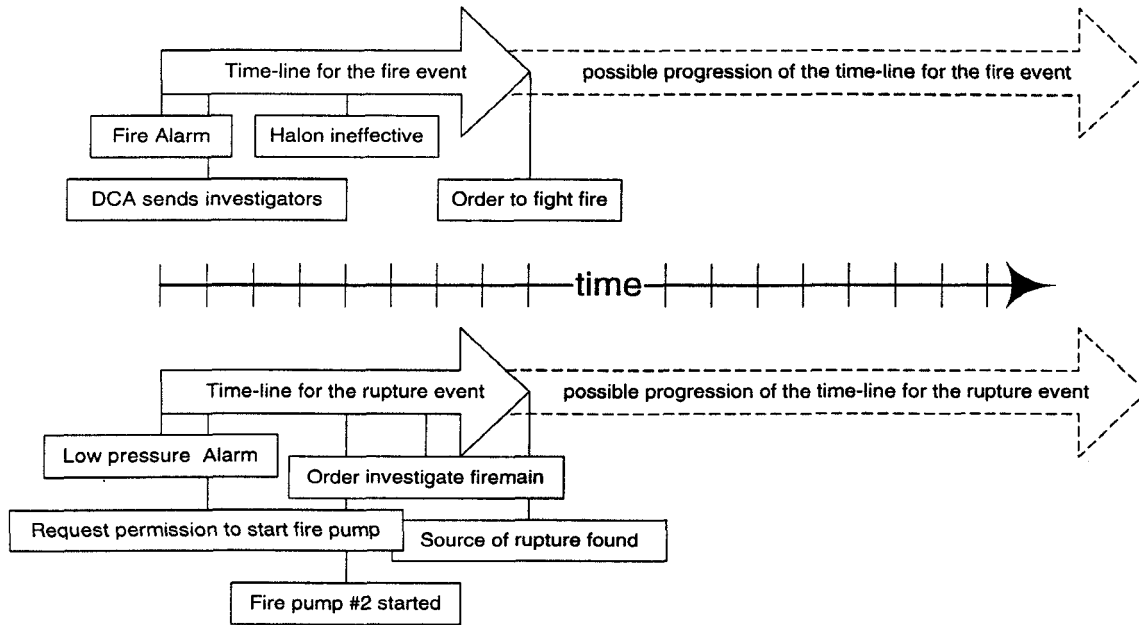
The model presented in this paper for critiquing in time critical and resource constrained domains takes a differential analysis approach. In such architectures the system is decomposed into three major components as shown in figure 1: the expert system module, the differential analysis module and the expert critiquing module.

The *expert system module* is responsible for the generation of the next best actions that can be taken by the user based on the current state of the environment. This module is based on a classical expert systems architecture and contains all the domain specific information that is needed to solve the problem. The general framework of the expert system consists of the integration of a blackboard architecture, a temporal representation and control module, and a temporal resource manager for its reasoning capability. The Blackboard is used to deliberate over the various possible actions [20] that the knowledge sources produce based on the current information on the blackboard and the events that signify a change in the environment. This change and the users actions will trigger different parts of the knowledge sources which will in-turn lead to the posting, modification or deletion of actions on the blackboard. Each action on the blackboard has a value that signifies its priority. This value is updated during each problem-solving cycle and is discussed in detail under

*Temporal Control*. These priorities (or urgencies) also form the basis of segregation of actions into equivalence classes.

The *differential analysis module* compares the ranked action-set that is produced by the expert system component to the current action that the user has chosen. If the users action falls within the highest equivalence class of the ranked action-set then it is considered proper and the system updates its state of the world. If the users action is either not in the highest class of the ranked action-set or not in the set at all the differential analysis component alerts the critiquing component of possible discrepancies by generating critique tags.

The third major component is the *expert critiquing module*. Expert critiquing can be in the form of suggestions, pointing out errors or a mixture of both. The difference between the users actions and the contents of the blackboard lead to the generation of critique tags. The critique tags which are grouped into errors of commission (i.e.: those that are considered improper) and errors of omission (i.e.: those that are ignored), are the main components of the on-line critique display. They are also the core of the off-line critique which is a more in-depth analysis of the users actions. The main difference between the on-line and the off-line critique generation module is the depth of the analysis that is performed.



**Fig 2: Time-lines for the fire and rupture events**

The critique tags reflect the results of an improperly taken user actions and the consequences of actions that are ignored by the user. The off-line critiquing module combines the critique tags that are generated throughout the problem solving along with the AND/OR fields of the actions on the Action Blackboard to generate a more in-depth analysis of the students mistakes.

The main structure of the off-line critique is based on a break-down of the students performance according to the grouping of the knowledge sources. The student performance in each area is analyzed according to the mistakes made reflected by the critique tags in that particular area and the strengths of the student based on the correct (or pertinent) actions taken and the amount of time saved on each. The student is also given a score in these individual areas and is combined for an overall performance score.

### Temporal Control

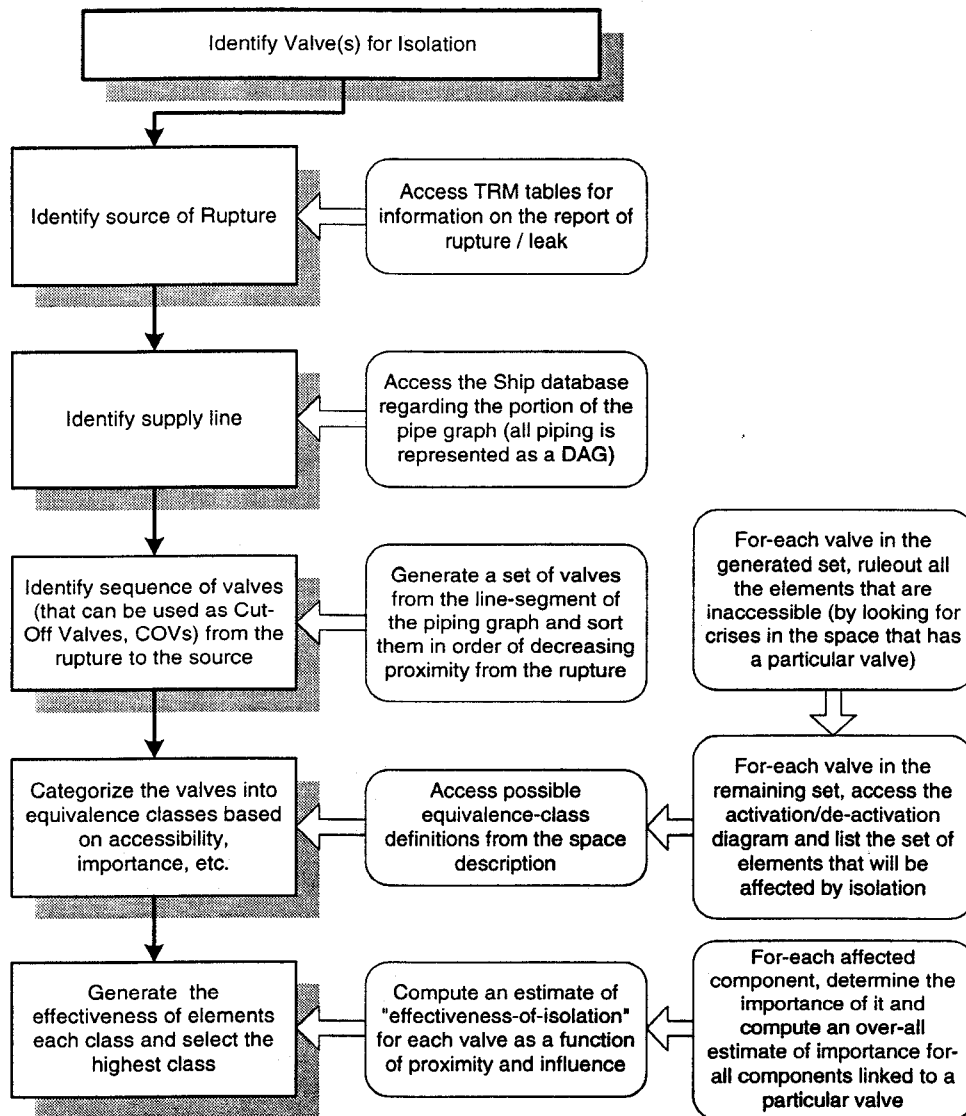
The input from the environment system is converted to a format that is common to all the sub-systems involved with the DCA training system for homogeneous communication of information. An event communication language (ECL) serves as the standard for all data that is generated or transferred between the modules of the expert critiquing system. The parsed input is used to trigger knowledge sources like the structure function diagram of the ship, that reason over this event. This process leads to the modification of the blackboards that contain all the actions that can be performed. Knowledge sources can modify

the contents of the blackboard in the following ways : the creation of one or more new action chains, the modification of existing chains and the deletion of one or more of the chains. The scheduler gathers evidence for and against each chain and computes an ordering. The amount of time that the user can wait before performing an action is a function of its importance of the action in terms of the functionality and effect. The maximum amount of time allowed for the action is *Critical + allowable\_margin* after which, the action is considered as delayed or missing. In any time critical environment, the priority of each action changes over time at different rates (the urgency to combat a fire in a magazine room grows differently than the need to give the C.O. a status report). The urgency of any action in the DCA domain increases monotonically with time and can be expressed as a function of its temporal priority and a weighted related events. The functions that generate the temporal priority for an action are linear, quadratic or exponential.

$$\begin{aligned} (P_{now})_{Linear} &= [T_E * (P_{MAX} - P_{MIN}) / T_F] + P_{MIN} \\ (P_{now})_{Quadratic} &= T_E^{[\log(P_{MAX} - P_{MIN}) / \log(T_F)]} + P_{MIN} \\ (P_{now})_{Exponential} &= T_E^{[T_E * \log(P_{MAX} - P_{MIN}) / (T_F * \log(T_F))]} + P_{MIN} \end{aligned}$$

Where :

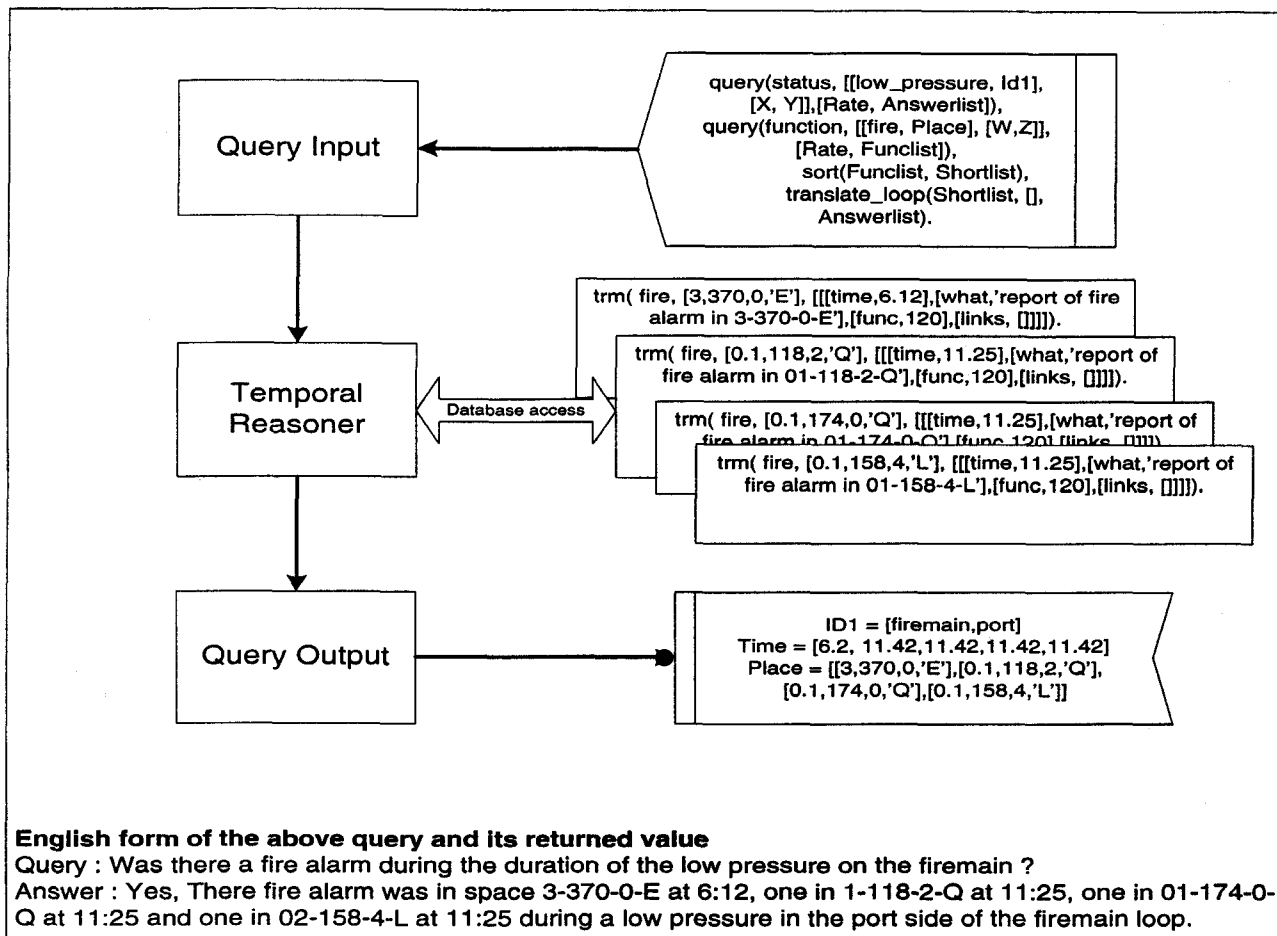
$(P_{now})_x$  = The current Temporal Priority (governed by x)  
 $T_E$  = Is the Elapsed Time (time since action was conceived)  
 $T_F$  = Is the Fire Time ( max time that the action can wait)  
 $P_{MAX}$  &  $P_{MIN}$  are the Priority-Scale bound



**Figure 3: Chill-Water / Fire-Main Valve Isolation**

There are two main reasons for such a priority generation function. One is to provide the action with a numerical value that it can be sorted and grouped on and the second is the generation of a score that depicts the goodness of the action. The numerical ordering of relationships between events in the DCA domain are based on the constraints that are propagated between sub-intervals of the same event and other events. The *urgency* of the various actions on the blackboard allow the creation of equivalence classes. The critiquing component of the model uses a differential analysis technique to compare the users action to the contents of the action blackboard to check relevance. The action is then checked for differences in

unification (the DCA can specify incorrect values for the parameters of the actions). When the action performed is preceded by a class of actions that are of greater urgency, the action causes a critique tag to be generated which serves as a pointer to the location at which the DCA faltered. The action is however not removed from the blackboard as the user may decide to take the action in some point in the future. Now after the scenario ends the critique generation module reasons over the accumulated critique tags that have been generated during the scenario to deliver a complete student critique.



**Figure 4: Example of the TRM query process**

We will now look at an example of the reasoning in the knowledge sources of the model with respect to sub-task of valve isolation, and the complexity of interacting sub-goals. One of the reasons that the DCA would need to isolate a valve is that when there is a rupture in a pipe the valve controlling the flow would need to be shut-off before the pipe can be fixed so that the repairs can progress quickly. The procedure, even though simple in itself, becomes complex when it is *overlapped* with other interacting events (fig.2). Let us consider a scenario where the pipe in question is carrying fire-fighting water to some team trying to combat a blaze on the ship and the rupture in the firemain piping is somewhere between the source of the water and the point of intended usage. The DCA now has a choice to either ignore the fire, isolate a valve up-stream of the rupture and then fix it, or to try and fix the rupture in-place without valve isolation.

The main criterion that enables the DCA to choose one of the above two options (we assume that the DCA is going to do something about the rupture) is the fact that, *is the isolation procedure going to affect some system on the ship that is depending on the flow through the pipe?* (fig.3). In this case there is a fire that is being fought with what little water that is flowing through that pipe. Also, the longer the DCA takes to decide the course of action to take, greater the probability of the fire spreading. So it is evident in this domain that the DCA needs to concentrate not only on the quality of their decision making but also on the amount of time it takes them. The representation of this problem can be viewed in the following manner; the initial report of a rupture (say at time  $t$ ) creates a time interval  $I_{ij}$  which denotes the events related to the rupture (here the end of the interval is left unspecified since it may ultimately lead to the sinking of the ship). The interval can be viewed as a bound for certain activities that can be associated to repairing the rupture. The individual events that constitute this time interval can either

be specified as a total order of events/actions that have to occur within the interval frame or partially ordered or as an incomplete order. Now another interval  $I_{kl}$  corresponding to the fire was asserted with a fire alarm at time  $k$ . The main criterion that enables the DCA to choose one of the above two options (we assume that the DCA is going to do something about the rupture) is the fact that, *is the isolation procedure going to affect some system on the ship that is depending on the flow through the pipe?*. In this case there is a fire that is being fought with what little water that is flowing through that pipe. Also, the longer the DCA takes to decide the course of action to take, greater the probability of the fire spreading. So it is evident in this domain that the DCA needs to concentrate not only on the quality of their decision making but also on the amount of time it takes them. The representation of this problem can be viewed in the following manner; the initial report of a rupture (say at time  $i$ ) creates a time interval  $I_{ij}$  which denotes the events related to the rupture (here the end of the interval is left unspecified since it may ultimately lead to the sinking of the ship). The interval can be viewed as a bound for certain activities that can be associated to repairing the rupture. The individual events that constitute this time interval can either be specified as a total order of events/actions that have to occur within the interval frame or partially ordered or as an incomplete order. Now another interval  $I_{kl}$  corresponding to the fire was asserted with a fire alarm at time  $k$  (the end of this interval is unknown at this time). Further let us assume that the two intervals can be associated by the relationship *starts* ( $I_{ij}, I_{kl}$ ).

$\forall I_{xy} \sim \exists I_{ab} [$   
 $(\text{type}(I_{xy}, \text{rupture\_alarm}) \wedge \text{affects}(I_{xy}, \langle \text{sub\_systemlist} \rangle) \wedge$   
 $\text{intersection}(\text{sub\_system}(I_{ab}, \text{system}), \langle \text{sub\_system list} \rangle) \wedge$   
 $\text{greater\_than}(\text{time\_point}(y), \text{time\_point}(a) \wedge$   
 $\text{below\_threshold}(\text{system}, \text{now}) ) \Rightarrow$   
 $\text{action}(I_{xy}, (\text{isolate\_valve}, \text{ok}), \text{now}) ]$   
 (Valve isolation is an acceptable procedure iff it does not cutoff supply to a sub\_system that is *vital* or it cuts off supply to a sub\_system that is *not vital*).

The intervals that are used by this knowledge source to reason over the intervals and their relationships is achieved by querying the temporal resource manager (TRM). The TRM for the DC domain is based on the principles of a temporal database [2, 9, 15]. Information is stored in the database according to time intervals based on Allen's time interval work, as well as categorized into a classification of what aspects of the ship the particular piece of information affects. The TRM can be broken down into a series of tables. Each table contains information about its specific domain on the ship. For each user action (command input) into the TRM, an entry is made into one of these tables. Table entries are not disjoint, one command often updates three or more tables. Table domains often overlap, and many commands depending on its variables relate to many

different domains and could be a subset of a larger table. The redundancy in tables is used to aid in consistency and speed when retrieving information. There are currently over 120 different ECL commands that exist in our version of the DC domain. Each individual command contains an unique variable list to specify the exact command. Each table entry contains the command, the time at which the information was reported, and any variables associated with the command. The variables are internally stored in a tuple format with each tuple consisting of an identifier for the variable and the value. In this way, the order of the variables need not be consistent, or even the variables associated with each command. The table entries are stored chronologically thus making the task of finding a specific *crisis interval* (fig. 4) trivial just by iterating through the table entries, as not all entries in a table may specifically refer to a given interval. The tables are broken up into various categories, each of which is concerned with a specific division of the ship. Every major piece of machinery on the ship has a table associated with it. Examples of crisis tables are those associated with a fire in a specific compartment, a rupture in part of the firemain, etc. Crisis tables are created dynamically. Each different TRM command has a data rule associated with it. This rule specifies which set of tables will be updated and what specific information will be stored in those tables. For different variants of a command, there may be multiple rule associations. When multiple rules are present, the correct rule is chosen by a meta-rule type associated with a command.

## Future Directions and Acknowledgments

The research described in this paper on temporal control structures for expert critiquing is using ship crisis control as the proof-of-principle domain. Current focus is on the modifications that have to be made to the existing blackboard architectures in order to accommodate forms of inference that go beyond opportunistic scheduling [12]. From preliminary observations about the representational adequacies of the critiquing model, it is evident that this principle can be applied over domains that involve time critical and resource constrained decision making.

The authors are grateful to the members of the Knowledge Based Systems Group at the University of Illinois, who provided helpful feedback on this research. We also thank John Viene, who implemented the Temporal Resource Manager modules and Vicki Fairchild who programmed the Temporal Resource Manager Query module. This research was funded in part by grants N00014-95-1-0934 and N00014-95-1-0749 from the Office of Naval Research.

## References

- [1] Allen, J.F., Maintaining Knowledge about Temporal Intervals. *Comm of the ACM*. Vol 26, pp 832-843. 1983.
- [2] Dean, T.L., McDermott, D.V., Temporal Database Management. *Artificial Intelligence*. Vol. 32, pp 1-55. 1987.
- [3] Fischer, G., McCall, R., Morch, A., JANUS: Integrating Hypertext with a Knowledge-Based Design Environment. *Proc. Hypertext '89*, ACM, pp 105-117. 1989.
- [4] Gertner, Abigail S., Real-Time Critiquing of Integrated Diagnosis/Therapy Plans. *Proc. Of Workshop on Expert Critiquing Systems*. AAAI, pp 6-13. 1993.
- [5] Hayes, Patrick J., A Catalog of Temporal Theories. Tech Report No. UIUC-BI-AI-96-01. Beckman Institute for Advanced Science and Technology, Univ. of Illinois at Urbana-Champaign. 1996.
- [6] Hogge, John C., TPLAN : A Temporal Interval-Based Planner with Novel Extensions. *Tech Report No. UIUCDCS-R-87-1367*. CS Dpt., Univ. of Illinois at Urbana-Champaign. 1987.
- [7] Koomen, Johannes, A.G.M., The TIMELOGIC Temporal Reasoning System. *Tech Report No. 231*. CS Dpt. Univ. of Rochester. March 1989.
- [8] Langlotz, C.P., Shortliffe, E.H., Adapting a Consultation System to Critique User Plans. *Int. Journal of Man-Machine Studies*. Vol. 19, pp 479-496. 1983.
- [9] Maiocchi, R., Pernici, B., Temporal Data Management in Real-Time Systems: A Comparative view. *Internal Report No. 88-046*. Dpt. di Elettronica, Politecnico di Milano, 1988.
- [10] McLean, Mark, Integrated Damage Control Training Technology. Student Guide. NAVSEA 03K12. October 1994.
- [11] Miller, P.L., A Critiquing Approach to Expert Computer Advice : *ATTENDING*. London : Pittman Press, 1984.
- [12] Nii, P., Blackboard Systems: The Blackboard Model of Problem Solving and the Evolution of Blackboard Architectures. *AI Magazine*. Summer 86. pp 38-53. 1986.
- [13] NWP 62-1 (Rev. D), Surface Ship Survivability. *Naval Warfare Publication*. 1993.
- [14] Shoham, Y., Goyal, N., Temporal Reasoning in Artificial Intelligence. *Exploring Artificial Intelligence*. Morgan Kaufmann. Chapter 11, pp 419-438. 1987.
- [15] Shoham, Y., *Artificial Intelligence: Techniques in Prolog*. Chapter 8: Planning and Temporal Reasoning. Morgan Kaufmann. pp 199-256. 1994.
- [16] Silverman Barry G., Critiquing Human Error. Academic Press. 1992.
- [17] Silverman Barry G., Expert Critics: Operationalizing the judgment/decision making literature as a theory of "bugs" and repair strategies. *Knowledge Acquisition*. Vol 3, pp 175-214. 1991.
- [18] Sussman, G.J., A Computer Model of Skill Acquisition. New York: American Elsevier. 1975.
- [19] Teach, R.L., Shortliffe E.H., An analysis of physician attitudes regarding computer-based clinical consultation systems. *Computers and Biomedical Research*. Vol. 14, pp542-558. 1981.
- [20] Wilkins D.C., Clancey, W.J., Buchanan, B.G., Using and evaluating differential modeling in intelligent tutoring and apprentice learning system. *Intelligent Tutoring Systems*. Hillsdale: Lawrence Erlbaum Associates. pp 257-279. 1988.