Appears in the Proceedings of the Twentieth Annual Conference of the Cognitive Science Society

Evaluating Computational Assistance for Crisis Response

Wayne Iba Melinda Gervasio Pat Langley Stephanie Sage
([Iba,Gervasio,Langley,Sage]@Isle.Org)
Institute for the Study of Learning and Expertise
2164 Staunton Court, Palo Alto, California 94306

Abstract

In this paper we examine the behavior of a human-computer system for crisis response. As one instance of crisis management, we describe the task of responding to spills and fires involving hazardous materials. We then describe INCA, an intelligent assistant for planning and scheduling in this domain, and its relation to human users. We focus on INCA's strategy of retrieving a case from a case library, seeding the initial schedule, and then helping the user adapt this seed. We also present three hypotheses about the behavior of this mixed-initiative system and some experiments designed to test them. The results suggest that our approach leads to faster response development than user-generated or automatically-generated schedules but without sacrificing solution quality.

Introduction

Traditional cognitive science has focused on human cognition and intelligent artifacts, but it has devoted less attention to combined human-machine systems. Nevertheless, the same theoretical approach—description in terms of computational processes—and the same experimental method—studying the effect of processes on measures of performance—applies equally to such hybrid entities. Such research has been most prevalent in the area of intelligent tutoring systems (Sleeman & Brown, 1982), but it seems equally applicable to the study of human interaction with intelligent assistants.

In the following pages, we describe a prototype intelligent assistant for crisis response. In particular, we examine the task of responding to chemical spills and fires, which we describe in the following section. Crises exhibit three primary themes: threat, urgency, and uncertainty (Gervasio & Iba, 1997). In this paper, we are primarily interested in the element of urgency and the impact that a computational assistant can have on the effectiveness and timeliness of a response. The need for a rapid response to such situations suggests a case-based approach to computational support, in which the human-machine combination retrieves and adapts structures from a case library. We present INCA, an intelligent system that embodies this design constraint, and we evaluate this approach to crisis response through experimental studies with INCA and human subjects working together to construct a response to a crisis. As an interactive computational assistant, we claim that INCA supports the rapid development of high-quality responses. In closing, we discuss related research and describe some directions for future work.

The Hazardous Materials Domain

A hazardous materials incident occurs when a spill of some chemical with hazardous properties endangers humans, property, or the environment. Consider a situation involving a leak of a toxic, flammable liquid from some old, corroded containers in a warehouse. The leak might result in a build-up of noxious fumes that could prove fatal to any inhabitants. In addition, sparks from nearby electrical equipment pose the threat of an explosion. The leak may also seep into the ground and contaminate the water supply, thus endangering the resident flora and fauna as well.

To prevent these disastrous events, a response team must be able to effectively and efficiently eliminate the hazards posed by the incident. A response involves many kinds of actions, including containing and neutralizing the spilled material, extinguishing any fires, evacuating or isolating nearby populations, and cleaning up the involved area. A crisis response team must decide on the most appropriate course of action, based on factors such as the properties of the material involved, the size of the spill, and the available resources. Their job is complicated by the urgent nature of the situation—delays in responding to the situation will result in more negative environmental and economic consequences. The incident is also fraught with uncertainty—incomplete information about the material involved, imperfect information about the location of the containers, unpredictable durations of the different activities, etc. Large-scale hazmat incidents may require the participation of multiple agencies, introducing communication and coordination issues.

We have developed a computational assistant to aid human users in the construction of crisis responses. In order to evaluate the resulting mixed-initiative system, we developed HAZMAT, a simulated world involving hazardous materials incidents. We designed this synthetic domain according to information from the 1996 North American Emergency Response Guidebook (NAERG) (Transport Canada, the U.S. Department of Transportation, & the Secretariat of Communications and Transportation of Mexico, 1996), a handbook for first responders that describes the appropriate responses for different situations, providing information on the classification of hazardous materials and the different actions and resources involved in a response.

The HAZMAT World

A HAZMAT incident is a spill, and possibly a fire, involving one of 50 different classes of hazardous materials, varying in form (solid, liquid, gas) and in hazardous properties (e.g., toxic, corrosive, and flammable). Incidents are categorized as being large or small (involving no more than 50 gallons of hazardous material), and they may occur indoors or outdoors. There are four types of spills, varying in the amount already spilled and the rate of spillage, and there are five types of fires, varying in the amount of spilled material on fire and if there is a fire, the rate of fire growth. HAZMAT thus involves a space of 4000 different incident classes. Incidents also have associated fire and health hazards that respectively measure the probability of a fire starting (if there isn't one already) and the level of danger to one's health. These secondary problem features are functions of the material, spill, and fire comprising an incident.

HAZMAT currently includes 49 different actions for addressing a spill or a fire (e.g., stop the leak, extinguish with alcohol-resistant foam) as well as the hazards presented by the spill or fire (e.g., absorb with dry sand, eliminate ignition sources, knock down vapors with water from a hydrant). Each action requires some subset of the 25 types of resources currently provided in HAZMAT. These resources include crew members, water sources such as pumpers (fire engines) and hydrants, different kinds of extinguishers, and absorbent material like sand and soda ash.

We can evaluate the effects of different actions on a situation using the HAZMAT simulator, which maintains processes for tracking and updating the dynamic characteristics of the domain for a given incident. Specifically, the state of the world is simulated with numeric variables, corresponding to the nominal-valued features of a HAZMAT incident. These include: the size and rate of the spill, the size and rate of any fire, and the sizes and rates of the fire and health hazard. The values of these variables are determined by the simulated processes such as the spill rate and fire growth rate and each variable may also be influenced by particular actions initiated by the crisis responder.

The HAZMAT Response Task

Given a particular type of hazardous material, NAERG defines a subset of actions (which we call the *legal* actions) to be used in developing a response. For example, a fire involving a flammable, toxic solid may be extinguished using a CO_2 or dry chemical extinguisher, but not a water or foam extinguisher. Each HAZMAT problem involves some number, possibly zero, of each type of resource, and since each action requires some minimum set of resources, some legal actions may not be applicable to a problem either. In addition, each resource is associated with a capacity and a quantity, the *capacity* being the maximum number of actions that may use the resource simultaneously and the *quantity* being the amount of that resource available for consumption. The actions that form a response must not violate the capacity or quantity constraint of any resource.

The task faced by a HAZMAT crisis responder is to choose a subset of the legal actions for a problem and to schedule them on the available resources, without violating any resource constraints. In our case, the crisis responder is a hybrid system consisting of a human user and our intelligent computational assistant.

Using HAZMAT, we can vary the severity of hazmat crisis problems, and monitor and evaluate the effects of different responses through the simulator. We can also introduce various types of assistant mechanisms tailored to specific aspects of the response task and evaluate their utility with respect to the overall response. In our experiments, we used HAZMAT to randomly generate problems with varied characteristics, and to evaluate human performance in crisis response with a computational assistant under different conditions.

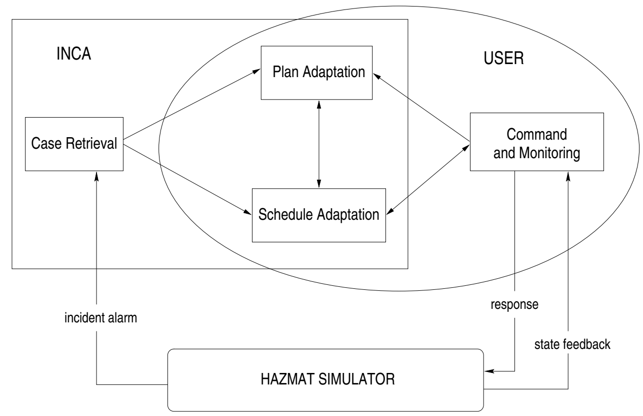

Figure 1: Mixed-initiative response to HAZMAT crises involving INCA (the INteractive Crisis Assistant) a human user.

HAZMAT Response with INCA

Crisis response teams often rely on standard operating procedures to guide their decisions. They also undergo numerous training exercises which let them hone their skills as well as refine their practices. Together, these suggested the *case-based* approach to crisis response that we decided to implement in INCA, our INteractive Crisis Assistant. Recent computational approaches to crisis have also revealed the importance of maintaining human input in problem solving, hence we have taken a mixed-initiative approach to crisis response. Figure 1 depicts such a hybrid system for HAZMAT response involving INCA and a human user.

To develop a solution, INCA first retrieves a case for a similar problem from a library of previous solutions. INCA then performs some initial adaptation of the solution, which involves a plan and a schedule, and presents this candidate solution to the user, who can perform additional adaptation as desired. Responding to a HAZMAT incident thus involves close interaction between INCA and the human user, who together must decide on the actions to include in a response and assign them to be executed by specific available resources.

Case Retrieval and Initial Adaptation

INCA is responsible for finding a similar, previous case from its library and performing an initial adaptation of the retrieved solution. A *case* consists of a problem, a set of resources, a set of legal actions, a plan, and a schedule. Matching is performed on the first three components, which are represented by a feature vector, and the case with the greatest number of features in common with the current situation is retrieved; ties are broken arbitrarily.[1]

[1]In our experiments, we used a simpler matching function that considered only the legal actions, which we found to result in comparable retrieval performance.

After the most similar previous case is retrieved, INCA performs an initial adaptation of the case's plan and schedule. Adapting the plan involves two operations: deleting actions that were legal for the case problem but are illegal for the current problem, and adding actions that were illegal for the case problem but are legal for the current problem. In this way, INCA prevents the user from considering any illegal actions and ensures that the user is aware of all the legal actions. Adapting the schedule involves two steps: matching one-to-one the case's resource list to the resources available in the current problem, and removing previously scheduled actions that have no corresponding resources in the current problem. Any actions without corresponding resources and the new legal actions are left unscheduled.

Interactive Adaptation

After INCA retrieves a case and completes its initial adaptation of the case plan and schedule, it presents the candidate solution to the human user for additional modifications that are desired. In our experiments, we considered only schedule adaptation, so we will focus on that process here. However, the plan adaptation process is also interactive—the user may modify the hierarchical plan by expanding or deleting nodes, thereby affecting the actions or jobs available for scheduling.

As discussed earlier, every action has a minimum resource requirement and every resource has associated capacity and quantity constraints. Allocating resources to an action involves choosing some number of multiples of its minimum requirement and choosing the specific resources themselves, subject to the constraints imposed by the resources. In addition, the scheduler must choose a duration based on the number of resources allocated to the action and whether there are any simultaneous actions that also have an effect on the world state variables affected by the action. For example, the duration for the action of extinguishing a fire with water from a hydrant would be shorter with more men and more hoses or if another extinguishment action—say, extinguishment with dry sand—will also take place.

A scheduled action thus corresponds to four decisions: the number of resources allocated to the job, the specific resources chosen, the start time, and the duration. The initial candidate solution that INCA adapts from a retrieved case and presents to the user will typically contain some scheduled jobs and some unscheduled jobs. By interacting with INCA, the user can modify or repair this schedule in five ways. The user can add jobs to the schedule, delete jobs from the schedule, shift the start time of a scheduled job, change the duration of a job, or switch a job from one resource to another.

INCA interacts with the user through a menu-driven graphical user interface and provides assistance during the schedule adaptation process in various ways. After the user chooses a particular repair operator, INCA takes the user through the necessary set of decisions. For example, if the user chooses to shift the start time of an action, INCA first asks whether the user wishes to shift the job earlier or later, and then it asks for the amount by which to shift the start time. INCA may also suggest default values, which the user may accept or ignore as desired. The user can recognize a schedule that violates a capacity or quantity constraint by its graphical layout as well as the textual information provided in the display. However, INCA also checks for this and prevents the execution of such illegal schedules.

HAZMAT incident response is a real-time problem—the situation continues to develop even as the crisis responder is constructing a response.[2] At any point during problem solving, the user may decide to *post* the schedule, which begins the execution of the scheduled actions at their respective times. The user may also request situation updates and continue to interact with INCA to modify the solution according to changes in the world revealed by the updates. The crisis response cycle ends when the crisis situation reaches a stable point—either when the execution of the scheduled actions successfully stops the spill and any fire, or all the material spills and any fire burns out.

Empirical Studies

In designing INCA, we decided that a case-based approach to crisis planning and scheduling would support more rapid response than a generative approach. This decision embodies our primary hypothesis, which we can test experimentally, along with a secondary intuition that this scheme would also produce higher quality responses. In this section we report preliminary studies that examine both of these claims.

Experimental Setting and Dependent Measures

We have already stated our basic hypotheses, but to test them we must move beyond the intuitive level to operational claims. We used the HAZMAT domain and its associated simulator as our testbed, with each problem consisting of a single incident (a spill and possibly a fire). In each situation, we gave the subjects a description of the incident and they used the graphical interface to produce a schedule that addresses the problem. However, we also needed dependent measures, independent variables, and reasonable control conditions.

Naturally, a case-based approach requires a case library. We decided to utilize the library construction process as our control condition. We presented a sequence of fifty problems to the subjects who were required to develop response schedules entirely from scratch. Subjects selected unscheduled jobs and assigned them to specific resources, also choosing start times and durations.When a subject was satisfied that their schedule adequately addressed the current incident or could not be further improved, the subject quit that problem and began the next one. The final solution for each problem was stored, as were the adaptations made by the user to generate the response as well as the time taken for each adaptation; this data was used for later evaluation. We collected the solutions and stored them as cases, thereby forming the respective case libraries used in conditions three and four.

Our concern with rapid response to crises suggested speed as an obvious dependent variable. More precisely, we tracked the time a subject took to transform an initial schedule provided by the system into one they found acceptable enough to execute. The second issue, quality, posed more challenges, since there is no right or wrong response to a HAZMAT inci-

[2]However, in an attempt to prevent a confound, we did not count the time taken to generate a response when evaluating the effectiveness of that response. That is, we artificially removed the real-time nature of the crisis response task for the purposes of our experiments.

dent and different subjects may judge schedules according to different subjective standards.

As a means of measuring quality, we used the simulator to compute an improvement metric reflecting the benefit of executing the user's response compared to letting the incident proceed without intervention. Toward this end, we ran a given test problem through the HAZMAT simulator without any response—the spill and any fire were allowed to proceed unabated until all the material was spilled and/or burned. During this process, we recorded particular world-state parameters (e.g., the amount of material spilled, the spill rate, and the health hazard level). We then simulated the same test problem together with the response constructed by the subject, collecting the data on the same state variables. We measured the percent improvement for each state variable as the difference between the variables' unabated values and their values with the response generated by the user, divided by the original unabated value. Although users may differ in the relative importance they place on respective state variables (and the corresponding improvements), for the sake of uniformity we counted each variable equally by taking a simple average over these variables. Our overall percent improvement measure reflects the average reduction in amount of material spilled and burned and how much the hazards were reduced as a result of the user's response.

Case Seeding vs. Manual Generation

Our basic prediction is that INCA's seeding of schedules with retrieved cases will improve the overall behavior of the human-machine system. Testing this claim requires us to compare the standard version of INCA, in which users interactively repaired a schedule that the system retrieved from its case library, with the control condition, where the seed schedule was empty and the user was required to construct the entire schedule from scratch.[3] The intuition is that retrieved cases, to the extent that they are appropriate to the problem, give users a head start compared to starting with an empty schedule; therefore, subjects could finish sooner and spend more time improving quality.

The first and second rows of Table 1 show the experimental results for these two conditions on both dependent measures, based on two subjects, each of whom dealt with thirty problems in the seeding condition and fifty in the control situation. First, we see a strong effect in the time taken to generate a response where the case-seeded trials (row 1) required significantly less time than the control condition (row 2). Comparing quality for these rows also shows a slight advantage for the case-seeded condition but this difference is not statistically significant. We believe that one explanation for the absence of an effect in quality is that the experimental design did not put subjects under time pressure during response generation. Consequently, the extra time spent in the control condition may have been used to bring the level of quality to that of the case-seed condition. This suggests future experiments where we strictly control response time so that we can compare quality across conditions at corresponding times during response generation.

[3]This control condition consisted of the same runs used to construct the case libraries.

Table 1: Scheduling time and schedule quality with 95% confidence intervals for each experimental condition.

	time	quality
personal case seeded	127.35 ± 19.91	34.33 ± 4.64
user generated	168.98 ± 17.07	33.67 ± 4.06
system generated	203.27 ± 30.88	29.52 ± 4.42
other case seeded	126.58 ± 15.82	31.83 ± 4.53

Case Seeding vs. Automatic Generation

Although the above comparison provided a clear test of our hypotheses, the dual facets of the INCA-user collaboration suggest another test involving a separate control condition, in which the intelligent assistant rather than the user generates a schedule from scratch. The intuition here is that even though both the schedule from the retrieved case and the one from the autonomous scheduler are complete, the latter will tend to require more adaptation because it is less cognitively compatible with solutions expected and desired by the user. Therefore, responses to problems in this condition should take longer to complete and be ultimately less effective.

To conduct this test, we developed an autonomous scheduler that uses a variety of heuristics to choose jobs, resource multiples, resources, durations, and start times. Jobs are chosen arbitrarily from the list of unscheduled jobs, and resources are chosen based on minimum requirements and earliest availability. The duration is chosen to be the average expected duration given the problem and the action.[4] Provided there are sufficient quantities of the chosen resources for the chosen duration, the jobs are then scheduled as early as possible on the chosen resources, subject to capacity constraints. With these heuristics, the autonomous scheduler tends to schedule as many actions as early as possible, using the least number and amount of resources as possible.

The comparison between the case-seed condition and the system generated seed (rows 1 & 3 of Table 1, respectively) shows an even stronger effect than we see between case-seeding and generation from scratch. The results in the table reveal that the users spent much more time modifying the schedule generated by the system than they did on the schedule from the retrieved case. However, once again the apparent differences in response effectiveness between rows 1 and 3 are not significant; we believe the same explanation applies as before.

The table also shows that even generating the schedule entirely from scratch (row 2) required much less time than repairing an autonomously generated one, although this difference was not significant. This finding surprised us but tends to support our intuition that solutions generated autonomously are cognitively incompatible with those desired by users. It also suggests the need for improved heuristics to guide the autonomous scheduler's search for good solutions.

[4]The expected duration was computed using only the equivalent gross ranges for the problem feature values, and not through any projection or simulation mechanism.

Individual Differences

The above results suggest that INCA provides an appropriate mixture of human and computer initiative for crisis responses, at least in the HAZMAT domain. Case seeding combined with user repair fared better, at least in speed, than either human or system generated schedules. But the results say nothing about the source of the retrieved schedules. This suggests another hypothesis: that users benefit more (again, in both speed and quality) from cases they developed themselves than they do from cases constructed by someone else. We tested this prediction by running subjects in a fourth condition where cases were retrieved from the other subject's case library instead of their own.

The results from this study do not support either of our hypotheses. The differences shown in Table 1 between personal case seed (row 1) and other case seed (row 4) are not significant for either speed or quality. Separate analyses of the data for the two subjects reveal a main effect on the case library that is used rather than the predicted interaction between case library and subject. That is, one user's case library is better in terms of scheduling time for both subjects. (Again, there were no significant differences in quality.) There are at least two explanations for this, the first being that one subject's cases were simply better and required less revision for either subject. An alternative explanation is that we did not sufficiently eradicate learning effects and one of the subjects was still improving across conditions. Because we could not properly mix the control or case construction condition, we did not mix any of the conditions; now that we have several case libraries, future experiments will properly randomize presentation of problems from each condition.

Discussion and Conclusions

In the previous sections, we described our basic hypotheses and analyzed data from preliminary experiments using the INCA system. The empirical results supported one of our primary hypotheses—case-based retrieval and adaptation mechanisms initialize a response schedule such that it can be adapted by a crisis response manager more quickly than if the schedule had been seeded by another mechanism. However, we did not find statistically significant support for our second hypothesis—case-based initialization methods should yield more effective responses than alternative initializations. At this time, the most we can confidently claim is that case-based seeding allows faster response without sacrificing quality. Based on this, we also claim that INCA provides an appropriate, effective and efficient approach to crisis response deserving of further investigation.

Although we found mixed results for our main hypotheses, we are not yet ready to abandon our prediction of improved quality through case-seeding. First, we note that in crisis response, responses must be both effective and timely and that these issues are inextricably entwined. If we discount improvements in outcome by the time taken to generate the response, the results do support significant differences between the two case-seed conditions vs. the user generated condition. More importantly, our experimental design failed to control for total response time and consequently our quality measure may be revealing a ceiling effect. An additional problem for our pilot study was that differences between problem incidents were greater than differences across conditions. That is, the differences in response generation time between difficult and easy problems was greater than that for a single problem under varied conditions. This observation was even more apparent when considering response effectiveness or quality. If we give subjects a sufficiently limited amount of time across all conditions and control for problem difficulty, we are confident that we will find the differences in response effectiveness that we predicted.

Finally, there are a few additional problems that we believe may be confounding our results in other ways. First, characteristics of the HAZMAT crisis response domain make it very difficult to develop heuristics for effective autonomous scheduling and our comparison to the system generated response may be too weak of a straw man. Second, INCA, and in particular INCA's graphical interface, places limitations on the user's ability to easily make certain schedule modifications, potentially increasing the time required to recover from poor seeds. Third, interviews with the subjects suggest the need for exploratory mechanisms that facilitate a user's exploration and discovery of solutions. With respect to quality, if we are seeing an artificially lowered ceiling effect due to insufficient feedback during response generation, such tools could raise the achievable level of performance and reveal differences between conditions. Addressing these issues is one part of our future work.

Related Work

While early approaches to crisis response planning were predominantly autonomous in nature, more recent systems provide interactive modes that, like INCA, let humans directly control the plan development process (e.g., OPLAN-2 (Tate, Drabble, & Kirby, 1994) and SOCAP (Bienkowski, 1996)). Unlike INCA, however, these systems aid users in developing solutions from scratch rather than help them in adapting solutions from previous cases.

CLAVIER (Hinkle & Toomey, 1994), a case-based system for autoclave loading, is an early example of a system that interactively adapts previous solutions. In the context of case-based systems for crises, JMCAP (desJardins, Francis, & Wolverton, 1998) uses a hybrid planner for the development of maritime evacuation operations, and CHARADE (Perini & Ricci, 1995) determines initial intervention plans to control forest fires. Like INCA, these systems also use a case-seeding mechanism to initialize the development of a response. INCA differs in that it is an *adaptive user interface* (Langley, 1997) that can acquire user models to alter its behavior to provide personalized assistance (Gervasio, Iba, & Langley, 1998). This paper also focused on an explicit evaluation of the benefits of case seeding in a mixed-initiative setting.

DIAL (Leake, 1995) is a case-based disaster response planner that can also learn from user interaction. In contrast to INCA, DIAL learns *adaptation cases* instead of *solution cases*; DIAL also takes a predominantly automated approach in that it resorts to interactive adaptation only if it does not already have an applicable adaptation case. Thus, DIAL serves as a learning apprentice while INCA is more of an adaptive assistant. CABINS (Miyashita & Sycara, 1995) is an interactive case-based assistant for job-shop scheduling that, like INCA, learns user preferences for the purpose of tailoring its behavior to individual users. CABINS uses case-based methods to learn preferences in the form of *repair cases* but INCA uses case-based reasoning to seed the response process and employs other inductive learning techniques to acquire user preferences.

Future Work

The results from this pilot study have been encouraging but considerable work still remains. The most straightforward task involves refining our experimental design and re-running the revised experiments with additional subjects, including experts in the hazardous materials domain, to replicate our findings regarding the superiority of case-seeded crisis responses over solutions generated from scratch or initialized by other means. An interesting issue is the degree to which different experts prefer distinct solutions, which will shed light on the importance of personalization in this domain.

We also plan to extend INCA in various directions. We intend to involve the user in the case retrieval process—which is currently INCA's sole responsibility—by employing interactive dialogues, as in the adaptation process, that let the user direct retrieval to appropriate, preferred cases. We have also developed and are testing advisory mechanisms for recommending repairs/cases that users can accept or override as they deem necessary. Naturally, we plan to design and execute experiments to evaluate the rate at which human users accept the recommendations, as well as recommendations' objective utility in terms of quality of the resulting solutions. Our main interest is in advisors that will make recommendations based on models learned from an individual's previous interactions with INCA. We anticipate that advisory mechanisms will further improve efficiency, particularly in the case of learned models, where we expect a greater likelihood of suggesting repairs that the user will find desirable.

In the longer term, we hope to expand our software to support coordination among multiple crisis managers. This will involve detecting resource conflicts among different users' schedules and recommending steps to resolve those conflicts while still meeting each user's goals. Traces of such conflicts and their resolutions will again provide data for learning, which should let the system improve its ability to recommend resolutions that are likely to work for particular sets of users. Such adaptive models of user interaction are a natural extension to the approach we have taken with individual crisis response.

The results from this pilot study and other ongoing work with INCA indicate that we have developed an exceptionally fertile framework for exploring issues of interactive crisis response. They also suggest that we have a promising candidate for computational assistance in crisis settings that merits additional attention.

Acknowledgements

This work was supported by the Office of Naval Research under Grant N000014-96-1-1221. We would also like to thank Greg Trafton, Michael Fehling, Gregg Courand, Mark Maloof, and three anonymous reviewers for insightful and beneficial comments.

References

Bienkowski, M. (1996). SOCAP: System for operations crisis action planning. In Tate, A. (Ed.), *Advanced Planning Technology* (pp. 70–76). Menlo Park: AAAI Press.

desJardins, M., Francis, A., & Wolverton, M. (1998). Hybrid planning: An approach to integrating generative and case-based planning. *Working Notes of the AAAI-98 Workshop on Case-Based Reasoning Integrations*.

Gervasio, M. T., & Iba, W. (1997). Crisis response planning: A task analysis. *Proceedings of the Nineteenth Annual Conference of the Cognitive Science Society* (p. 929).

Gervasio, M. T., Iba, W., & Langley, P. (1998). Learning to predict user operations for adaptive scheduling. *Proceedings of the Fifteenth National Conference on Artificial Intelligence*.

Hinkle, D., & Toomey, C. (1994). CLAVIER: Applying case-based reasoning to composite part fabrication. *Proceedings of the Sixth Conference on Innovative Applications of Artificial Intelligence* (pp. 55–62).

Langley, P. (1997). Machine learning for adaptive user interfaces. *Proceedings of the Twenty-First German Annual Conference on Artificial Intelligence* (pp. 53–62).

Sleeman,, D., & Brown, J. S. (Eds.) (1982). *Intelligent tutoring systems*. New York: Academic Press.

Leake, D. B. (1995). Combining rules and cases to learn case adaptation. *Proceedings of the Seventeenth Annual Conference of the Cognitive Science Society*.

Miyashita, K., & Sycara, K. (1995). CABINS: A framework of knowledge acquisition and iterative revision for schedule improvement and reactive repair. *Artificial Intelligence*, *76*, 337–426.

Perini, A., & Ricci, F. (1995). An interactive planning architecture *Proceedings of the European Workshop on Scheduling and Planning*.

Tate, A., Drabble, B., & Kirby, R. B. (1994). O-Plan2: An open architecture for command planning and control. In Zweben, M. and Fox, M. S. (Eds.), *Intelligent Scheduling*. San Francisco: Morgan Kaufmann.

Transport Canada, the U.S. Department of Transportation, & the Secretariat of Communications and Transportation of Mexico. *1996 North American emergency response guidebook*.