Appears in the AIPS-98 Workshop on Interactive and Collaborative Planning

Interactive Adaptation for Crisis Response

Melinda Gervasio Wayne Iba Pat Langley Stephanie Sage
Institute for the Study of Learning and Expertise
2164 Staunton Court, Palo Alto, California 94306
{gervasio, iba, langley, sage}@isle.org
http://www.isle.org

Abstract

Crisis domains present the challenge of developing good responses in a timely manner. In this paper, we present an interactive, case-based approach to crisis response that provides users with the ability to rapidly develop good responses while leaving ultimate decision-making control to the users. We introduce INCA, the INteractive Crisis Assistant we have implemented for planning and scheduling in crisis domains. We also present HAZMAT, the artificial domain involving hazardous material incidents that we developed for the purpose of evaluating different responses and various assistant mechanisms. We then discuss two preliminary studies that we conducted to evaluate scheduling assistance in INCA. Results from the first set of experiments indicate that INCA's case-based scheduling assistance provides users with initial candidate solutions that enable users to develop high quality responses more quickly. The second set of experiments demonstrates the potential of machine learning methods to further facilitate interactive scheduling by accurately predicting preferred user adaptations. Based on these encouraging results, we close with directions for future work and a brief discussion of related research.

Introduction

Responding to crises like natural disasters and military invasions is a complex activity that stands to benefit from computational aids. Crisis response has a number of characteristics, which we review elsewhere (Gervasio & Iba, 1997), that distinguish it from other problems involving the generation of plans and schedules. In this paper we focus on two features: urgency, which indicates the need for *rapid* response, and the combination of human planners' cognitive limitations in crisis and their final decision making responsibility, which indicates an *interactive* approach to support.

Previous systems for crisis response have been predominantly autonomous in nature. More recent systems often provide interactive modes that let human users directly control the plan development process (e.g., OPLAN-2 (Tate et al., 1994), SOCAP (Bienkowski, 1996)). Like these systems, the INteractive Crisis Assistant (INCA) that we present in this paper provides various forms of computational support while allowing crisis managers to control the development of a crisis response. In this paper, we focus on the evaluation of the assistant mechanisms in INCA to determine their utility in crisis response.

Organizations that respond to crises typically adopt standard operating procedures that are situation specific, suggesting a *case-based* approach to computational support. These procedures and the regular training exercises such organizations perform can serve as a case library from which users can select the plans and schedules most relevant to a new crisis. Typically, a retrieved case will require some adaptation before it is appropriate to a new situation. INCA retrieves plans and schedules from previous cases, and then lets the user interactively adapt them using repair-space operators.

Interactive systems present the opportunity to gather traces of user behavior and the potential to learn user models. An *adaptive user interface* (Langley, 1997) can capitalize on regularities in users' behavior by presenting preferred options as default selections, or by more carefully exploring the search space of problem states that the user is likely to traverse. INCA exploits learned user models by recommending the most probable adaptation operations during scheduling.

We will explore these issues in the context of a domain involving hazardous material incidents, which we describe in the following section. Then we present INCA, the crisis assistant that we developed for the interactive construction of responses. We maintain that our approach helps users develop responses to crises more rapidly than they could from scratch. We also predict that it will let them produce higher quality solutions than a purely generative approach. As an adaptive interface, INCA can acquire user models from user interaction, and we expect that INCA can utilize these models to anticipate users' repair operations, further reducing the time required to generate an appropriate response to a crisis. These hypotheses are subject to empirical test, which led us to design and execute the experiments that we report in the fourth and fifth sections. Our results support these hypotheses and thus encourage us to continue exploring our case-based approach to crisis response. In the final section, we discuss related work and outline some areas for future work.

The Hazardous Materials Domain

The hazardous materials domain exhibits the three primary themes of crisis: threat, urgency, and uncertainty (Gervasio & Iba, 1997). A hazardous materials incident occurs when a spill of some material with hazardous properties poses a threat to humans, property, or the environment. This entails a sense of urgency in that delays in responding to the situation typically result in more negative environmental and economic consequences. It also has uncertainty in various forms: incomplete information about the material involved, imperfect information about the location of the containers, unpredictable durations of the different activities, etc.

The Idealized HAZMAT World

We have developed an artificial hazardous materials world, HAZMAT, for the purpose of evaluating problem solvers' responses to hazardous materials incidents. In developing HAZMAT, we consulted the 1996 North American Emergency Response Guidebook (NAERG) (Transport Canada et al., 1996), a handbook for first responders that describes the appropriate responses for different hazardous materials situations. It provides information on the classification of hazardous materials and the different actions and resources involved in a response.

There are 50 classes of hazardous materials, varying in form (solid, liquid, gas) and in hazardous properties (e.g., toxic, corrosive, flammable). A HAZMAT incident is a spill, and possibly a fire, involving one of these hazardous materials. 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 the rate of fire growth. These parameters combine to form a space of 4000 different incident classes. In addition, these incidents are associated with fire and health hazards that 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.

In HAZMAT, there are 25 types of resources, such as hydrants, dry chemical extinguishers, soda ash, and crew members. These resources are used by the 49 different actions available for responding to an incident, with different actions requiring different resources. The actions address a spill or 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).

Task Description

A HAZMAT problem consists of one or more incidents and some number (possibly zero) of each type of resource. 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.

The HAZMAT response task is to choose a subset of the legal actions for a problem and to schedule them on the available resources so that they can be executed to deal with the incident. Unlike traditional planning and scheduling, there is no clear delineation between planning and scheduling responsibilities here. Determining the set of actions to schedule (i.e., planning) is interleaved with assigning resources to those actions (i.e., scheduling). We will elaborate on this matter in the section describing the crisis assistant.

Simulation

HAZMAT includes a simulator for evaluating various responses (including no response) to an incident. The simulator maintains processes that track and update the dynamic characteristics of the domain for a given incident. The state of the world is defined by eight numeric variables, corresponding to the nominal-valued features of a HAZMAT incident: the size and rate of the spill, the size and rate of any fire, and the sizes and rates of the fire and health hazards. The values of these variables are influenced by the specifics of the given incident and the actions initiated by the problem solver. Each action has the potential to impact some subset of these parameters. For example, extinguishment reduces the size and rate of a fire, while knocking down vapors reduces the fire and health hazards.

Using HAZMAT, we can vary the severity of crisis problems, as well as 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 this initial implementation of HAZMAT, we focused on individual problem solvers, and we excluded the evacuation and first aid operations that are part of the complete response.

INCA: A Crisis Response Assistant

Now we can describe the system we have developed to assist users respond to HAZMAT incidents. INCA (the INteractive Crisis Assistant) takes an interactive, case-based approach to crisis response. Given an incident, INCA retrieves a solution for a similar problem from a case library of previous solutions and performs some initial adaptation. INCA presents this candidate solution to the user, who performs additional adaptation as needed. We first discuss case retrieval and initial adaptation followed by the interactive adaptation of plans and schedules. After this, we briefly describe the graphical user interface, and, finally, we present additional aspects of INCA that set the stage for our empirical evaluation.

Case Retrieval and Initial Adaptation

The case retrieval component is responsible for finding a similar, previous case from the case library. A *case* consists of a problem, a set of resources, a set of legal actions, a plan, and a schedule. Each case is encoded as a feature vector, with similarity measured by a simple count of matching features.

After INCA retrieves the most similar case, it performs some initial adaptation. Modifying the case plan involves two operations: removing actions that were legal in the case problem but are illegal in the current problem, and adding actions that were illegal in the case problem but are legal in the current problem. INCA thus ensures that no illegal actions are executed and that no legal actions are excluded without the user's knowledge. Modifying the case schedule also involves two steps: matching one-to-one the resources of the case and the currently available resources, and removing previously scheduled actions that have no corresponding resources in the current problem. INCA leaves unscheduled the new legal actions and those actions with no corresponding resources.

This approach to adaptation takes advantage of two aspects of the domain. First, there are no causal supports between the actions of a problem—that is, an action does not establish preconditions for any other action. Thus, an action can be scheduled independently of other actions. Second, the resources are naturally grouped into pools, the members of which are completely substitutable. Thus, a resource of a particular type in one problem is just as good as another resource of the same type in another problem.

Interactive Plan Adaptation

A plan for HAZMAT response is a special form of a hierarchical task network. The root node of every plan is the abstract action `handle-incident`. A node at one level expands to a set of nodes at the next lower level. These nodes are neither conjunctive nor disjunctive in that any subset of an expansion may eventually be executed. Thus, both a null plan and a plan including all legal actions are valid solutions; the difference is in their impact on the incident. The leaves of the plan constitute the actions to be scheduled on the available resources. Only *primitive* actions (i.e., actions with no further expansions) may be scheduled; thus, unexpanded nodes cannot be scheduled.

Interactive adaptation allows the user to modify the initially adapted case plan in two ways. First, the user may expand any unexpanded nodes to explore additional courses of action. This has the effect of adding to the set of jobs for scheduling. Second, the user may delete any subtree of the network. This has the effect of removing both scheduled and unscheduled jobs.

Although we discuss planning and scheduling separately, there is not a clear division of responsibilities. In contrast to the traditional planning and scheduling framework, the actions or jobs selected in the planning phase do not all have to be allocated to resources in the scheduling phase. Decisions about which actions to schedule (i.e., planning) are also made during scheduling. However, the planning component can delete large groups of actions, thus limiting the size of the set of jobs to be considered by the scheduler. For example, if there is already a fire, removing the high-level node `prevent-fire` limits the scheduler's attention to the more relevant `handle-fire` actions. We expect that future applications of this system, including an expanded HAZMAT domain, will require more of the traditional planning operations such as the determination of binding and ordering constraints.

Interactive Schedule Adaptation

The resources associated with a problem are organized into pools of identical resources. Each resource is associated with a capacity and a quantity, the *capacity* being the maximum number of jobs that may be simultaneously scheduled on that resource and the *quantity* being the amount of that resource available for consumption. For example, a pumper may have capacity 4 and quantity 1000 (units of water).

Every action is associated with a minimum resource requirement constraint. For example, the action of extinguishing with water using a pumper requires a minimum of two crew members, one hose, and one pumper, and it will consume ten units of water per minute. Allocating resources to this action involves choosing some number of multiples of this minimum set and choosing the specific resources themselves, subject to the capacity and quantity constraints imposed by the resources.

The scheduling process is complicated by the fact that, in this domain, jobs have variable duration. To illustrate, consider the action of extinguishing a fire with water from a hydrant. The appropriate duration of this action will depend not only on the size and growth rate of the fire but also on how many resources are allocated to the action and on any simultaneous actions that also address the fire. Extinguishment will be faster with more crew members and more hoses and if extinguishment with dry sand is also taking place. Scheduling a job thus involves four decisions: the number of resources to allocate to the job, the specific resources to allocate, the duration, and the start time.

Interactively adapting the case schedule involves one of five operations: adding jobs to the schedule, deleting jobs from the schedule, shifting the start time of a scheduled job, changing the duration of a scheduled job, or switching a job from one resource to another.[1] A schedule is *infeasible* if it violates any capacity or quantity constraint; the scheduling assistant prevents the execution of infeasible schedules.

[1] There is also an operator to change the number of resources assigned to a job, and we are currently exploring other similar higher level operators (i.e., scheduling macros). However, the scheduling component in the experiments had only the five simple operators.

HazMat incident response is a real-time problem—the situation continues to evolve even as the user (crisis responder) is developing a response. Thus, at any point during problem solving, the user may *post* the schedule to begin execution of the scheduled actions. The user may also request situation updates and continue to modify both the plan and the schedule accordingly. The crisis response cycle ends either when the execution of the scheduled actions successfully stops the spill and extinguishes the fire, or all the material is spilled and burned.

Graphical User Interface

Inca includes a graphical front-end that provides the user with a point-and-click interface for making plan and schedule modifications. The interface is organized into two screens, one for plan adaptation and one for schedule adaptation, and the user may switch between the two arbitrarily. The planning screen displays the current problem description, the available resources, and the current plan in the form of a nested list of actions reflecting the hierarchical structure of the plan. The user adapts the plan by clicking on an action (node) and selecting a command (e.g., expand, delete) from the pop-up menu that appears next to it.

The scheduling screen displays the current problem description, the scheduled and unscheduled jobs, and the assignment of jobs to resources (i.e., the schedule). Each scheduled job is color coded, and appears as a set of colored *job-resource blocks* in the schedule, one for each resource used by the job. The user adapts the schedule by clicking on a job-resource block and selecting a command (e.g., add a job, switch to another resource) from a pop-up menu. Figure 1 shows the scheduling screen with the user about to perform an adaptation operation. Inca's graphical interface also serves as an interface to the HazMat simulator, allowing the user to request problem updates and post schedules for execution by the simulator.

Preparation for Evaluation

We designed Inca to be an interactive, case-based planning and scheduling assistant for crisis response. The experiments discussed in the following sections focus on the interactive schedule adaptation aspect of Inca. To this end, we developed assistant technology for the autonomous generation and adaptation of HazMat responses. The autonomous planner uses heuristics to prune out all illegal and some undesirable actions. We used this planning assistant to determine the set of jobs to be scheduled during case development, and to automatically expand newly legal actions and prune out undesirable actions during initial plan adaptation.

The autonomous scheduler uses a variety of heuristics to choose jobs, resource multiples, resources, durations, and start times. We used this scheduling assistant in one of the experimental conditions to provide an initial starting schedule in lieu of retrieving and adapting a previous schedule from a case. We also used

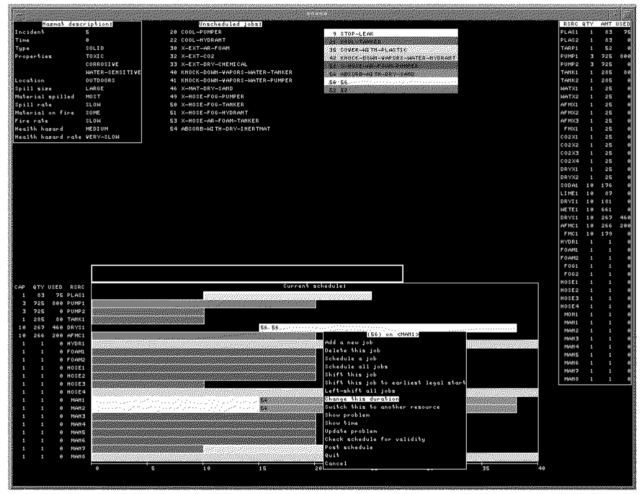

Figure 1: Inca scheduling screen showing the user changing the duration of an action to correct a resource overallocation.

the heuristics to suggest default values to the user during interactive scheduling, although no evaluation was performed on the usefulness of these heuristics. The scheduling heuristics used in the experiments tended toward scheduling as many jobs as early as possible, using the fewest resources for each job.

Case-Based Seeding

We predicted that the tools provided by Inca would let users *more efficiently* construct crisis responses of *higher quality*. This claim formed the basis for our experimental hypotheses and motivated the design of the pilot study discussed in this section. It also pointed to the dominating dependent measures we considered—efficiency and quality. We first describe the experimental design and setup and introduce the independent and the dependent measures before presenting our hypotheses and results.

Experimental Design and Setup

In this study, subjects were given a description of a HazMat problem and instructed to develop a response through Inca's interactive graphical interface. Each problem consisted of a single incident, and the effectiveness of the solutions was determined using the HazMat simulator. To prevent a confound between response generation time and response effectiveness, we removed the real-time nature of the response problem. Thus, no constraints were placed on the speed of response generation, and the subjects did not receive runtime feedback regarding the effectiveness of their response.

Our goal in this study was to evaluate the utility of the case-based retrieval and adaptation mechanisms that provide the initial schedule, or *seed*, for the user's response. The experimental design was thus geared towards evaluating the efficiency and quality of solutions resulting from alternative initialization strategies. We tested subjects over a number of trials in four experimental conditions, with each trial corresponding to the development of a schedule for one problem.

In the first condition (*no seed*, the control condition), subjects had to develop a schedule entirely from scratch. The solutions generated under this condition were also used to construct personal case libraries. In the second condition (*auto seed*), INCA initialized the scheduling process for each incident using the autonomous scheduler. The third and fourth conditions relied on INCA's case retrieval and adaptation mechanisms to provide the initial schedule. INCA used the subject's personal case library in the third condition (*own seed*), and another subject's case library in the fourth condition (*foreign seed*). A subject ended the scheduling process and moved on to the next trial when he was satisfied that the current schedule adequately addressed the incident or could not be improved further.

In determining the dependent measures for comparing subjects' performance over the different experimental conditions, we focused on two specific characteristics of performance: efficiency and quality. We measured *efficiency* by the total time required by the user to develop a solution, and we measured *quality* as the improvement realized through the application of the actions specified in the constructed schedule.

To compute this improvement, we simulated each problem twice: first, without any intervention, allowing the spill and any fire to proceed unabated, and then, with the user's scheduled response. During each simulation, we recorded the parameters characterizing the state of the world (e.g. amount spilled, burn rate, health hazard level). Then we calculated the percentage improvement using the differences between corresponding parameter values in the two simulations. We report this as a reduction percentage, with higher numbers representing increasing quality (i.e., 0% means no improvement and 100% means infinite improvement). The quality measure thus reflects how much less material was spilled and burned and how much the hazards were reduced through the intervention of the user.

To summarize the setup, there were two primary independent measures: 1) whether an initial seed was provided or not, and 2) what particular type of seeding was used—heuristic scheduling, personal case library, or foreign case library. There were also two primary dependent measures: 1) the time taken to construct a response (efficiency), and 2) the performance improvement due to the response (quality). Each subject solved fifty problems in the no seed condition and thirty problems in each of the remaining conditions. For further details on these experiments, see (Iba et al., 1998).

Experimental Hypotheses and Results

We expected seeding to facilitate the development of a response, since it starts the scheduling process with a partial solution. In particular, we expected to see advantages for the case-seeded conditions because the cases were generated by the users in the first place and we propose that schedules based on these cases constrain the space of solutions to those that are "cognitively compatible" with a user.

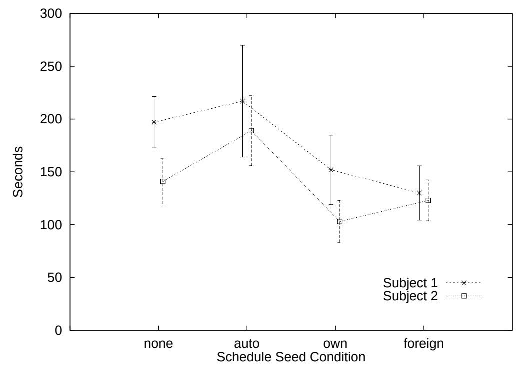

Figure 2: Mean scheduling time (with 95% confidence intervals) for each subject under the various experimental conditions.

Hypothesis 1: Response time (scheduling time) will be lower with schedule seeding than with no seeding.

Specifically, we expected a progression of faster response times, with auto seed being faster than no seed, foreign seed faster still than auto seed, and own seed being the fastest of all.

Figure 2 shows the average time taken by each user to develop a response in each of the four experimental conditions. The results partially support our hypothesis. For both subjects, scheduling was most efficient with case seeding. However, scheduling in the auto seed condition seemed to be even slower than scheduling with no seed, particularly for the second subject. Also, while the own seed condition resulted in slightly better efficiency than the foreign seed condition for the second subject, the reverse was true for the first subject.

A possible explanation for the negative results regarding the auto seed condition is that the heuristics used by the autonomous scheduler may have resulted in schedules that were highly incompatible with the users' preferences. Indeed, interviews with the users revealed that they often spent a considerable amount of time clearing schedules under the auto seed condition. The mixed results regarding the case-seeded conditions may be explained by the second subject having stronger individual preferences than the first subject. That is, the second subject exhibited a stronger bias for his own schedules, while the first subject exhibited a bias only against the schedules of the heuristic scheduler.

We also expected seeding to yield higher quality solutions. The intuition is that users will have more opportunity to improve the quality of a response if they do not have to spend time generating it in the first place.

Hypothesis 2: Solutions derived from seeded schedules will be more effective than responses generated from scratch.

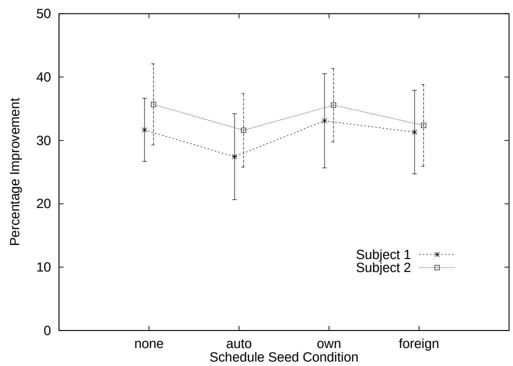

Figure 3: Mean performance improvement (with 95% confidence intervals) for each subject under the various experimental conditions.

Figure 3 shows the percentage improvement realized by the solutions of each user under the different experimental conditions. While the mean improvement under the auto seed condition is slightly worse than the other conditions for both subjects, the large variances do not support any clear differences between the conditions. One possible explanation for this result is that the scale of differences between problems may have been greater than the differences between conditions. That is, easy problems led to large improvements while difficult problems allowed only minimal improvements. Closer examination of the data revealed that problems often did cluster into ranges of performance improvement across all conditions. Another possible explanation is that the subjects' experience with the HazMat domain may have been insufficient for them to properly evaluate schedule quality. Thus, they would have been unable to focus on the most crucial adaptations giving the greatest gains.

The measure of quality for Hypothesis 2 evaluates schedules only in terms of their effects in the simulation. In a crisis context, however, it is important not only to come up with good solutions but also to come up with them quickly. Thus, a better measure of quality would also account for the time taken to produce the schedule. Given the results on scheduling time, which favor case seeding, and the results on performance improvement, which show no differences between the conditions, we can expect to see greater benefits from case seeding if we discount the percentage improvement by the scheduling time.

Predicting User Adaptations

The second set of experiments was motivated by our long-term objective for Inca to become an adaptive user interface (Langley, 1997). We claim that machine learning can be used to acquire user models that can be used to adapt system behavior to better assist users. We begin by presenting our formulation of the learning problem before discussing the experiments and results.

Experimental Design and Setup

We defined Inca's performance task as: Given the current scheduling state, as characterized by the problem, available resources, legal actions, and current schedule, predict the adaptation operation the user will perform next. This can be translated into a standard classification task, with the class being the scheduling operation and the instance being the scheduling state for which the prediction is being made.

The prediction task can be formulated at various levels, from the highest level predicting a general scheduling operator such as ADD *any* action, to the lowest level predicting a specific operator instantiation such as ADD the `cover-with-tarp` action on `crew members #1` and `#4` and `tarp #2` starting at time 34 for a duration of 8 time units. For our initial formulation, we chose an intermediate level, requiring the specification of a particular action but not specific resources or amounts. We also combined the remove action, change duration, shift action, and switch resource operators into a single REPAIR operator. This resulted in ADD and REPAIR operations for each of the 49 actions, for a total of 98 different classes. We used 86 attributes to describe the scheduling state: 12 nominal attributes to represent the problem, 25 Boolean features to denote whether or not resources of each type were available, and 49 attributes to indicate the schedule status of each action (i.e., whether it was legal and whether it was feasibly scheduled).

We characterized Inca's learning task as: Given a set of training examples, learn a classifier that makes correct predictions on new examples. The information we gathered during the schedule seeding experiments provided the data for our learning experiments. Specifically, each schedule adaptation performed while solving a problem corresponds to a training or test instance. We extracted 935 examples (data set 1) and 1049 examples (data set 2) from the problem-solving traces of the first and second subjects respectively. The results reported in the next section were obtained by running ID3 (Quinlan, 1986) over ten trials on each data set, with each trial using 800 randomly chosen examples for training and the rest for testing. See (Gervasio et al., 1998) for more details on these learning experiments.

A baseline study (Figure 4, base) showed that predictive accuracy after learning was significantly better than guessing randomly (1.02%) and guessing the most frequent class (8.57%). However, even in the best case (26.87% for data set 2), there were more than twice as many misclassifications as correct classifications. Our analysis of the results suggested our formulation of the problem as the prime suspect for this poor performance, so we investigated various problem reformulations with the goal of improving performance. Thus, in these experiments, the primary independent measure was problem formulation, and the primary dependent measure was predictive accuracy.

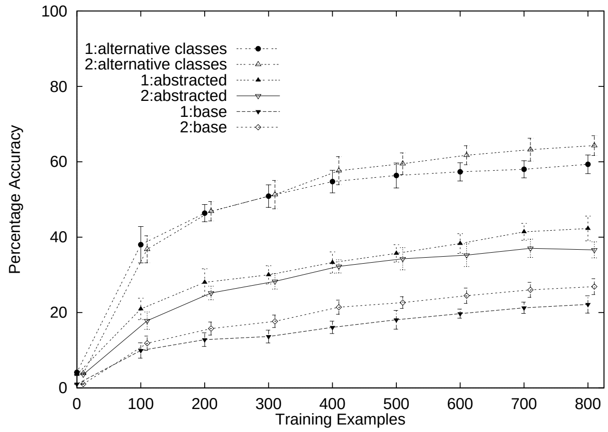

Figure 4: Learning curves for the various experimental conditions, showing predictive accuracy (with 95% confidence intervals) under different problem formulations.

Experimental Hypotheses and Results

In the first experiment, we wanted to investigate the effects of effects of simplifying the problem via class abstraction. As stated earlier, the prediction task can be formulated at different levels, and for the baseline study we had chosen an intermediate level requiring the specification of an ADD or REPAIR of a particular action.

Hypothesis 1: Predictive accuracy will increase on a more abstract prediction task.

To test our hypothesis, we abstracted the task to require the prediction of an ADD/REPAIR on a class of actions rather than on specific actions (e.g., "extinguish with hose" rather than "extinguish with hose using water from a hydrant").

The results support our hypothesis: accuracy increased significantly for both data sets with class abstraction (Figure 4, abstracted). This result may not be surprising since the abstraction results in a simpler prediction task. However, it raises the issue of determining an appropriate task level in mixed-initiative settings such as interactive scheduling. By predicting a class rather than a particular action, the system requires the user to make an additional decision. However, this transfer of decision-making responsibility to the human user is not necessarily undesirable as it may lead to more effective interaction overall.

In the second experiment, we wanted to investigate the effects of allowing the prediction of alternative operations. The formulation of the performance task as the problem of predicting the user's next operation was a natural fit to the data provided by the user traces. However, it is unnecessarily difficult in that it requires the system to predict a user's immediate next adaptation operation. A user may arrive at the same schedule through different sequences of operations, many of which may be equally acceptable to the user.

Hypothesis 2: Predictive accuracy will increase on the modified task of predicting any subsequent adaptation operation.

To test this hypothesis, we first modified the original examples to include a set of alternative classes, consisting of all the subsequent operations the user invoked to solve the HazMat incident, minus those that were illegal in the current state. Training proceeded as before, but, during testing, we judged a prediction to be correct if it matched the instance's label or one of its alternative classes.

The results support our hypothesis: significantly greater accuracy was achieved on the reformulated task (Figure 4, alternative classes). The results might seem obvious since allowing alternative classes simplifies the prediction task. However, as stated earlier, predicting the user's immediate next operation may be an unnecessarily difficult task. We claim that the modified task of predicting any of the user's subsequent operations is in fact more appropriate in that it better captures what concerns users during the scheduling process.

Discussion and Conclusions

The results from our pilot study on schedule seeding support our primary hypothesis and claim that Inca's case-based retrieval and adaptation mechanisms provide an initial schedule that allows a crisis responder to develop a solution more quickly than constructing a response from scratch or from an initial schedule provided by another mechanism. In addition, the results show that solution quality was maintained with this increase in efficiency. These results support our claim that Inca provides an appropriate, effective, and efficient approach to crisis response.

Preliminary results with learned user models also support our intuition that regularities in user response behavior can be extracted and exploited through an adaptive user interface. Users can recognize a recommended operation more quickly than they can generate it themselves if it is one they would have selected anyway. Although we have not yet collected timing data in the condition where the model's recommendations are presented to subjects for acceptance or rejection, we anticipate additional speed gains to be revealed.

Related Work

As discussed earlier, many crisis planning and scheduling systems today, including OPLAN-2 and SOCAP, have interactive modes that let users to maintain control of the problem-solving process. With the importance of the human-computer interface in mixed-initiative systems, there has also been interest in developing more natural interfaces, such as the speech interface in TRAINS-95 (Ferguson et al., 1996), a planning assistant for train routing. Most of these interactive systems aid users in developing solutions from scratch. In contrast, Inca aids users in adapting solutions from previous cases. This paper also focused on the evaluation of the scheduling assistance in INCA.

CLAVIER (Hinkle & Toomey, 1994) is an advisory system for autoclave loading that, like INCA, retrieves previous cases for users who can then interact with the system to perform additional modifications. INCA differs in its domain and consequent focus on planning and scheduling. DIAL (Leake, 1996) is a case-based planner for disaster response, whose adaptation process is predominantly automated, in contrast to the interactive case adaptation process in INCA. DIAL also learns *adaptation cases*, in contrast to the *solution cases* in INCA. CABINS (Miyashita & Sycara, 1995) is an interactive assistant that uses case-based methods to learn user preferences for job-shop scheduling. Like CABINS, INCA incorporates learning mechanisms that let it adapt its assistant behavior to different individuals. However, while CABINS employs case-based methods to learn individual preferences in the form of *repair cases*, INCA uses case-based reasoning primarily as a seeding mechanism, and it employs inductive learning techniques to acquire user preferences.

Future Work

Our preliminary experimental results have been encouraging, but considerable work remains. Some of the unanticipated findings in our experiments on schedule seeding have led us to revise our experimental design to control for confounding factors such as problem difficulty and unlimited scheduling time. In the near future, we intend to run similar experiments with more subjects to replicate our findings. We are also planning experiments to test the utility of recommendations from learned user models in the context of HAZMAT crises of varied severity. We hope to conduct studies exploring differences in expert and novice performance as well. An interesting issue involves the degree to which different experts prefer distinct solutions, which will reveal the importance of personalization in this domain.

We also plan on extending INCA in various directions. Based on the experimental results and feedback from the subjects, we plan on modifying INCA's graphical interface in various ways to better facilitate interaction with the human user. We are currently extending the interactive nature of INCA to case retrieval, which to this point has been entirely automated. This would allow the user to direct case retrieval to the most appropriate, preferred case.

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.

Acknowledgments This work was partially supported by the Office of Naval Research under Grant N000014-96-1-1221. We would also like to thank Michael Fehling, Gregg Courand, and other members of the Organizational Dynamics Center at Stanford University for many useful discussions on crisis; and Mark Maloof for his helpful comments on the paper.

References

Bienkowski, M. 1996. SOCAP: System for Operations Crisis Action Planning. In *Advanced Planning Technology*, Tate, A. (ed.), 70–76. Menlo Park: AAAI Press.

Ferguson, G.; Allen, J.; and Miller, B. 1996. TRAINS-95: Towards a Mixed-Initiative Planning Assistant. In *Proceedings of the Third International Conference on Artificial Intelligence Planning Systems*, 70–77.

Gervasio, M. T. and Iba, W. 1997. Crisis Response Planning: A Task Analysis. In *Proceedings of the Nineteenth Annual Conference of the Cognitive Science Society*, 929.

Gervasio, M. T.; Iba, W.; and Langley, P. 1998. Learning to Predict User Operations for Adaptive Scheduling. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence.*

Hinkle, D. and Toomey, C. 1994. CLAVIER: Applying Case-Based Reasoning to Composite Part Fabrication. In *Proceedings of the Sixth Conference on Innovative Applications of Artificial Intelligence*, 55–62.

Iba, W.; Gervasio, M.; Langley, P.; and Sage, S. 1998. Evaluating Computational Assistance for Crisis Response. In *Proceedings of the Twentieth Annual Conference of the Cognitive Science Society.*

Langley, P. 1997. Machine Learning for Adaptive User Interfaces. In *Proceedings of the Twenty-First German Annual Conference on Artificial Intelligence)*, 53–62.

Leake, D. B.; Kinley, A.; and Wilson, D. 1996. Acquiring Case Adaptation Knowledge: A Hybrid Approach. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, 684–689.

Quinlan, R. 1986. Induction of Decision Trees. *Machine Learning* 1:81–106.

Transport Canada; the U.S. Department of Transportation; and the Secretariat of Communications and Transportation of Mexico. *1996 North American Emergency Response Guidebook.*

Miyashita, K. and Sycara, K. 1995. CABINS: A Framework of Knowledge Acquisition and Iterative Revision for Schedule Improvement and Reactive Repair. *Artificial Intelligence* 76: 337–426.

Tate, A.; Drabble, B.; and Kirby, R. B. 1994. O-Plan2: An Open Architecture for Command Planning and Control. In *Intelligent Scheduling*, Zweben, M. and Fox, M. S. (eds.), 213–239. San Francisco: Morgan Kaufmann.