

Mixed-Initiative Goal Manipulation

Michael T. Cox and Chen Zhang

■ Mixed-initiative planning systems attempt to integrate human and AI planners so that the synthesis results in high-quality plans. In the AI community, the dominant model of planning is search. In state-space planning, search consists of backward and forward chaining through the effects and preconditions of operator representations. Although search is an acceptable mechanism to use in performing automated planning, we present an alternative model to present to the user at the interface of a mixed-initiative planning assistant. That is, we propose to model planning as a goal-manipulation task. Here planning involves moving goals through a hyperspace in order to reach equilibrium between available resources and the constraints of a dynamic environment. The users can establish and “steer” goals through a visual representation of the planning domain. They can associate resources with particular goals and shift goals along various dimensions in response to changing conditions as well as change the structure of previous plans. Users need not know the details of the underlying technology, even when search is used within. This article empirically examines user performance under both the search and the goal-manipulation models of planning and shows that many users do better with the latter.

The idea to present planning as a goal-manipulation process rather than a search process has much appeal, at least intuitively. Goals have long been recognized as important to a full understanding of both human and machine problem-solving abilities (Newell and Simon 1963, 1972; Schank 1982; Schank and Abelson 1977). A goal provides a focus for inference and learning, and it makes explicit the objectives of a system, either

human or machine (Ram and Leake 1995). By having explicit representation of goals, a planning assistant can more easily reason not only about the domain of planning but also the direction of planning. Furthermore by including a means for explicitly representing user goals at the interface of a computer problem-solving assistant, users are encouraged to express their specific intentions, thereby forcing them to keep their “eyes on the prize.” In this article, we view goals somewhat differently than do most cognitive or engineering approaches to intelligence. Goals are mutable. They will change appropriately with user direction according to the current context; they will change inappropriately when left to the vagaries of a dynamic environment.

Overwhelming combinatorics face fully autonomous planners attempting to solve complex problems. A mixed-initiative approach inserts into the planning process an active human user whose knowledge is outside the transitive closure of more coarse-grained abstract domain representations used by planning assistants. However to perform such an insertion of human decision makers, it is not sufficient to simply allow a user to perform or override any decision the automated planner itself can make.

Planning is represented internally by most planning assistants as a search process. Formal representational schemas with preconditions, postconditions, and variable bindings represent actions. Such schemas are called planning operators and can change the world through the effects of their post conditions. Planning therefore consists of search from an initial state to a goal state by means of operator selection

and variable binding choices. But except for highly trained individuals, search is a reasonably difficult concept to grasp fully.

As an indication of the difficulties, consider how many operator variable-binding choices exist for a particular decision point in the search process. If N objects of type OBJECT exist and a state has two arguments of this type, then the user must select from N^2 choices. In general, for a predicate with j arguments, each type of which has cardinality $\text{card}(\text{arg}_j)$ instances, the maximum number of candidate choices can be as much as $\prod_{0 \leq i \leq j} \text{card}(\text{arg}_i)$. Thus a large number of choices exist in general, and the preponderance of similar choices may overwhelm a user.

In this article, instead of presenting the planning process to a user as a *search* through the effects and preconditions of operator representations, we hypothesize that the metaphor of *goal manipulation* will prove a more accessible means for humans to consider planning and therefore will result in better planning performance. The goal-transformations section describes the concept of a goal transformation upon which a mixed-initiative planning assistant can be implemented. The GRANS mixed-initiative planning assistant section introduces the GTrans planning interface to the PRODIGY planning and learning architecture. We compare and contrast this with PRODIGY's standard GUI. The planning for goal change section explains how a user can plan for goal change by introducing the Bridges Problem. The evaluation section summarizes a very small empirical study performed to compare the two different planning interfaces. Here the performance of users given a goal-manipulation metaphor generally exceeds that of users presented with the search metaphor. However we also note that expert users lose some flexibility under the goal-manipulation model that otherwise allows them to perform better under search in some circumstances. The related research section provides pointers into the literature for related research, and the final section closes the article with a brief discussion and directions for future research.

Goal Transformations

We start with the observation that goals often become obsolete given inherent uncertainties in a resource-limited world and a changing circumstance. For example the goal to have a postal package delivered may be abandoned because the recipient is no longer at the package address. Instead a goal to return the package to the sender may be substituted. Only

under the closed-world assumption can goals be defined as static input states to be achieved. Moreover, goals are often vague and ill-defined in the early stages of planning and require refinement throughout the planning process. Goals are not necessarily atomic; rather, they may be semiachievable.

For instance Williamson (1996) examined time constraints on goals. A goal may not be achieved by a certain date but may be achieved slightly later at a specific cost. In his conception, planning consists of maximizing a value function on goals. As Williamson further notes, a conjunctive goal is composed of multiple goals that all need to be satisfied. So satisfying some instead of all is better than satisfying none at all. We also prefer planners that do not simply fail under resource-poor circumstances. Instead planners should not only be able to revise their plans when resources are limited, but they should also be capable of revising their goals in order to achieve solutions that are acceptable. To address these problems we have created a theory of goal change formally represented as transformations.

A *goal transformation* represents a goal shift or change. Conceptually it is a change of position for the goal along a set of dimensions defined by some abstraction hyperspace (Cox 2000, Cox and Veloso 1998). The hyperspace is associated with two hierarchies. First the theory requires a standard conceptual type-hierarchy within which instances are categorized. Such hierarchies arise in classical planning formalisms. They are used to organize arguments to goal predicates and to place constraints on operator variables.

Goal-transformation theory also requires a unique second hierarchy. In normal circumstances the knowledge engineer and the subject-matter expert create arbitrary predicates when designing operator definitions. We require that these predicates be explicitly represented in a separate predicate abstraction hierarchy that allows goals to be designated along a varying level of specificity. For example consider a military domain. The domain-specific goal predicate *is-ineffective* takes an aggregate force unit as an argument (for example, *(is-ineffective enemy-brigade1)*). This predicate may have two children in the goal hierarchy such as *is-isolated* and *is-destroyed*. The achievement of either will then achieve the more general goal. Furthermore if the predicate *is-destroyed* had been chosen to achieve *is-ineffective*, the discovery of noncombatants in the battle area may necessitate a change to *is-isolated* in order to avoid unnecessary casualties.

Note also that to defer this decision, the goal movement may be to the more general is-ineffective predicate instead. Then when the opportunity warrants and further information exists, the goal can be reexpressed. In any case, movement of goals along a dimension may be upward, downward, or laterally to siblings.

Goal movement may also be performed by a change of arguments where the arguments exist as objects of or members of the standard object type-hierarchy. The goal represented as the type-generalized predicate (inside-truck Truck1 PACKAGE) is more general than the ground literal (inside-truck Truck1 PackageA). The former goal is to have some package inside a specific truck (thus existentially quantified), whereas the latter is to have a particular package inside the truck. Furthermore both of these are more specific than (inside-truck TRUCK PACKAGE).¹ Yet movement is not fully ordered, because (inside-truck Truck1 PACKAGE) is neither more or less general than (inside-truck TRUCK PackageA).

A further way goals can change is to modify an argument representing a value rather than an instance. For example the domain of chess may use the predicate outcome that takes an argument from the ordered set of values {check-mate, draw, lose}. Chess players often opt for a draw according to the game's progress. Thus to achieve the outcome of draw rather than check-mate represents a change of a player's goal given a deteriorating situation in the game.

Generally, planning is to achieve a desired state by managing the world, the resources, and the intended outcomes one wishes. Although a change in the world is delivered by acting upon it, the selection of actions is but one choice a planning agent has at its disposal. The choice of goal states to include in a problem representation, the selection of a current goal to pursue, and the commitment of resources to each goal are equally as important as the action selections themselves. Thus planning is a context-dependent task of discovering, managing, and refining what one actually wants.

The GTrans Mixed-Initiative Planning Assistant

To directly support the goal-manipulation model, we implemented a mixed-initiative interface to a planning assistant through which the user manipulates goals, the arguments to the goals, and other properties. The interface hides many of the planning algorithms and knowledge structures from the user and instead emphasizes the goal-manipulation process

with a menu-driven and direct-manipulation mechanism. The assistant, called *GTrans*² (Cox 2000; Zhang 2002; Zhang, Cox, and Immaneni 2002), presents a direct manipulation interface to the user that consists of a graphical map with drag-and-drop capability for objects superimposed upon the map surface (see figure 1). *GTrans* helps the user create and maintain a problem file that is internally represented as follows. A planning problem consists of an initial state (a set of objects and a set of relations between these objects) and a goal state (set of goals to achieve). The general sequence is (1) to create a planning problem, (2) invoke the underlying planner to generate a plan, and then until satisfied given planning feedback either (3a) change the goals or other aspects of the problem and request a plan or (3b) request a different plan for the same problem.³ To support step 3a, *GTrans* currently implements three general classes of goal transformations: (1) goal-type transformation, (2) goal-argument transformation, and (3) valence transformation.

A *goal-type transformation* enables the user to manually transform a goal by moving the predicate of the goal along an abstraction hierarchy defined in the domain knowledge. The *goal-argument transformation* involves a movement through an abstraction hierarchy on one or more goal arguments. A *valence transformation* is performed by toggling between a positive or negative truth value of the goal (that is, whether to achieve the predicate or its negation).

The overall *GTrans* family of assistants incorporates a number of versions including a multi-user collaborative planning assistant (Immaneni and Cox 2004). This assistant configuration includes multiple networked copies for each user and a joint planning interaction mode along with information sharing capabilities. Another version integrates the mixed-initiative (human-directed) assistant with a multiagent (artificial agent) planning system called COMAS (Edwin 2001; Edwin and Cox 2001). COMAS (coordination in multiagent systems) also uses goal transformations for planning. Here we focus on a single-user version that interacts with one underlying planner.

The *GTrans User Interface Version 2.1* (Zhang 2002) is the implementation that hides the planning algorithms and representations from the human user by focusing on the goal-manipulation process (see again figure 1). It provides the user with a mechanism to directly manipulate the set of objects, initial state of the planning world, as well as the objectives of the planning problem. The user is able to

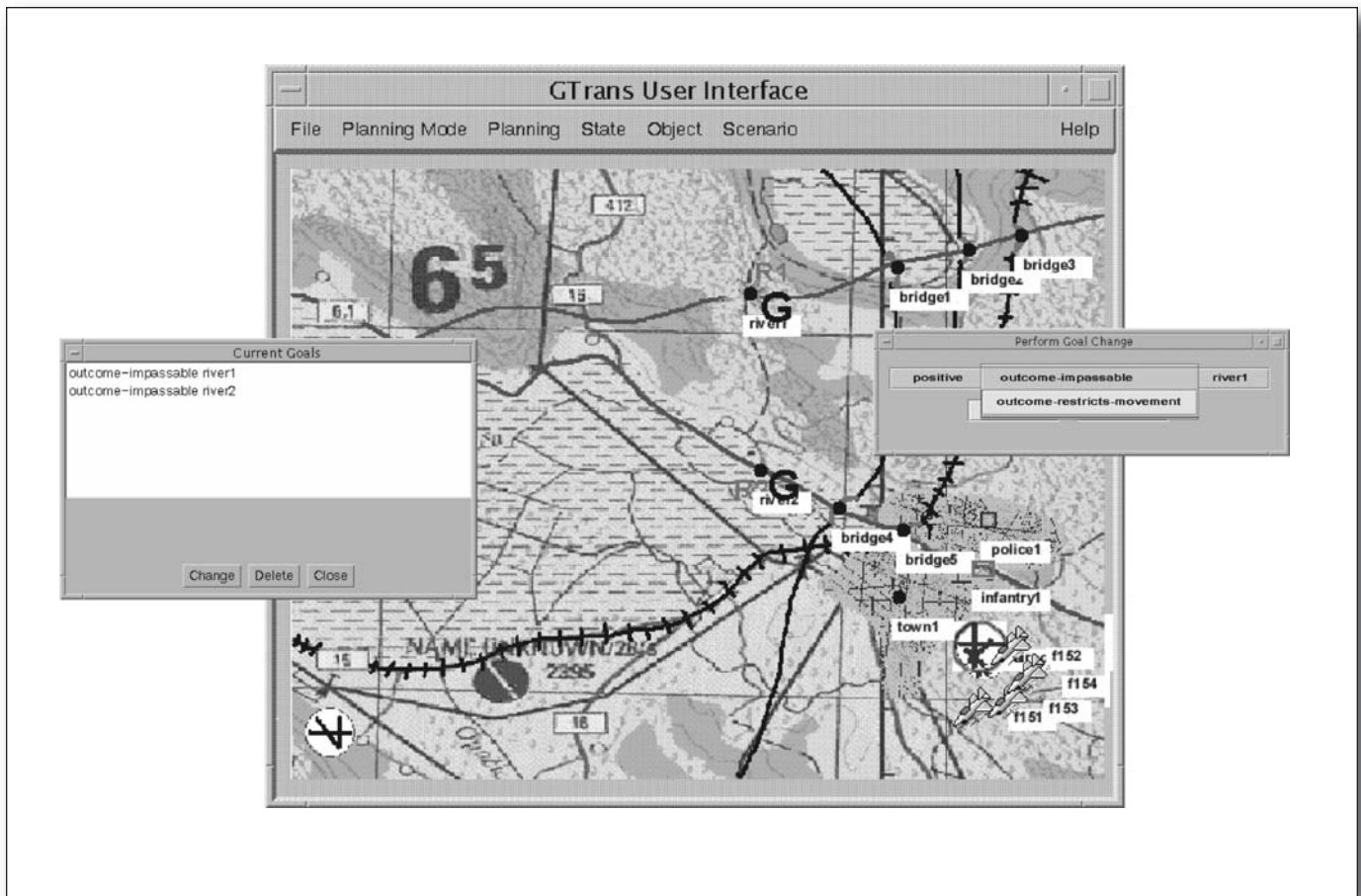


Figure 1. GTrans User Interface 2.1.

define the objectives or goals and to assign particular resources to achieve the goals. When the underlying planner fails to generate a plan because of insufficient resources or because the planning world changes, the user can asynchronously modify the goals and send them back to the planner for another round of planning, thus steering the planning process. Although figure 1 depicts the interface, discussion of the content of this window will follow in the next section.

The user interface is implemented in Java Version 1.2. The overall architecture includes two principle components: (1) Prodigy/Agent (written in Allegro Common Lisp) and (2) the user-interface single-agent component. The interface presents a graphical map display and a set of pull-down cascading menus to the human planner. The human can create objects, define initial states, set or transform goals, create problems, and solve problems. The *Prodigy/Agent* assistant⁴ (Cox et al. 2001; Elahi 2003; Elahi and Cox 2005) allows the underlying PRODIGY planner to communicate with the interface through a Knowledge Query Manipu-

lation Language (Finin, Labrou, and Mayfield 1997), or KQML, protocol. The two programs use sockets to pass KQML performatives.

PRODIGY (Carbonell et al. 1992; Veloso et al. 1995) is a domain-independent, nonlinear, state-space planner implemented at Carnegie Mellon University (CMU). It searches for a sequence of actions that transform the environment from an initial state into a final state containing the goal state. Like all state-space planners, its problem specification includes a set of objects existing in the planning environment, the initial state of the environment, and the goal state that needs to be achieved by the plan. Prodigy/Agent consists of a wrapper program and the PRODIGY planner. The wrapper program serves as a software interface between the PRODIGY planner and external agents.

PRODIGY has a standard user-interface that comes with the CMU system (see figure 2). The *Prodigy 4.0 User Interface 2.0* (Cox and Veloso 1997) provides access to the underlying data structures such as the search tree, the goal-sub-goal graph (left canvas in figure 2), and the emerging plan (right canvas in figure 2). It also

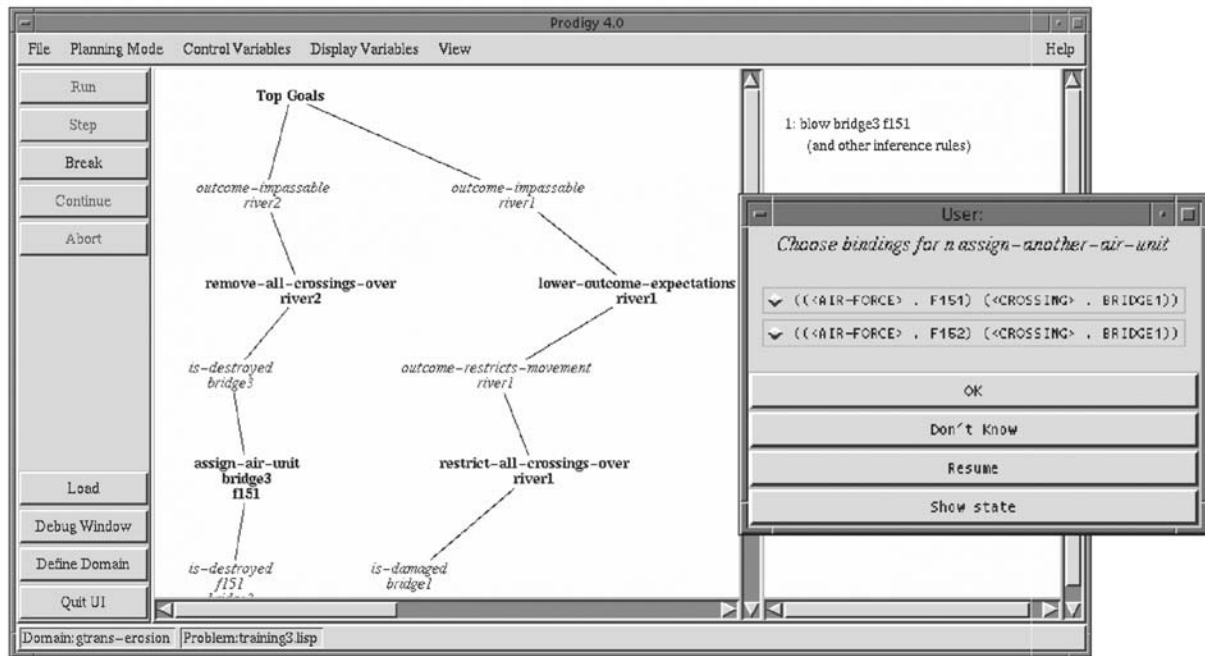


Figure 2. Prodigy 4.0 User Interface 2.0

includes various pull-down menus and buttons to control planning and interface parameters, planning mode, and other facets. The interface displays the progress of the planning process in terms of search. Cox and Veloso claimed that the interface provides a mechanism that improves user performance with the system, but this is a weak claim, because the previous interface was simply a LISP interpreter. Here we will compare the standard PRODIGY interface to the alternative GTrans interface.

When it loads a domain, the GTrans user interface obtains the domain information from Prodigy/Agent through four types of requests. In response to an obj-request, Prodigy/Agent sends back to the GTrans user interface the types of objects that exist in the domain. After receiving a goal-request, Prodigy/Agent replies with all the possible goals that can be achieved in the domain. To support goal-classification information, PRODIGY reads each operator from a domain file and sends to GTrans a list of positive effects from each operator. The effects are represented as (predicate arg1-type arg2-type... argN-type). GTrans then can associate objects defined on the map with goals relevant to each. A tree-goal-request asks for all of the

goal hierarchies in the domain, and a state-request seeks all of the possible initial states in the domain.

Unlike domain information that remains static for a given domain, problem information (including object information, state information, and goal information) may change over time as the planning world changes. The communication of problem information between the planning user interface and Prodigy/Agent occurs after the human planner creates a set of objects, specifies the initial state, sets the goal state, and gets ready to run the problem. After receiving the request to search for a plan as well as the problem information, Prodigy/Agent runs the PRODIGY planner and sends the resultant plan, if any, back to the requesting planning user interface. In case of failure in generating a plan, Prodigy/Agent simply sends back a failure message indicating no plan can achieve the goal state of the problem.

Planning for Goal Change

Here we introduce a problem that illustrates the trade-offs between resource allocation decisions in a military air campaign planning

domain. The Bridges Problem is simply to make rivers impassable by destroying all bridges across them. This task universally quantifies the variable *<crossing>* with the relation (enables-movement-over *<crossing>* river) and requires a separate air unit for each crossing in order to achieve the goal. We simplify the problem by assuming that an air resource can destroy one bridge and damage an arbitrary number of others. Therefore if a new crossing is discovered or an old resource becomes unavailable,⁵ the constraints of the problem change, thereby forcing dynamic replanning. When a goal state is composed of conjunctive goals for multiple rivers, an interesting trade-off exists for the user. To maximize the goal satisfaction, the optimal user will allocate resources to rivers with fewer crossings (Edwin and Cox 2001).

The Bridges Problem is an instance of a class of problems with the following form:

$$(\forall x \mid \text{goal}(x) \forall y \mid \text{subgoal}(y, x) \exists r \mid \\ (\text{resource}(r) \wedge p(r, y)) \vee p'(y))$$

where $\text{value}(p'(y)) = \alpha(\text{value}(p(r, y)))$ and $0 \leq \alpha \leq 1$.

The problem is to maximize the value of the quantified expression above. That is, for all goals, x , and for all subgoals of x , y , the problem is to maximize the value of the conjunction of disjunctions $p(r, y) \vee p'(y)$ using some resources r where the value of the predicate p' is some fraction between 0 and 1 of the value of p . In the case of the Bridges Problem, the goals are to make impassable rivers x by the predicate is-destroyed applied to bridges y over the rivers or by the predicate is-damaged where a damaged bridge is considered 50 percent destroyed ($\alpha = 0.5$).⁶ Thus given enough resources and the goal to make a river impassable, it is enough to destroy all bridges across it. Without sufficient resources, a plan to destroy most bridges and damage the rest results in maximally restricting the movement across it.

Figure 1 in the previous section illustrates a scenario with four air units (resources) and two rivers, the first with three bridges and the second with two. With insufficient resources available, one of the goals must change. The user should notice that two resource units allocated to the two-bridge river and two to the three-bridge one is best. In such a case, the goal to make the second river impassable is achieved fully, whereas two of the three bridges can be destroyed over the first river and one damaged. Note that this suggests a change to the first goal rather than the second. By transforming the goal (outcome-impassable river2) to (outcome-restricts-movement river2) the planner is able to achieve greater partial success (5/6 versus 3/4).

GTrans provides a unique facility to transform or *steer* goals in such situations. When the user receives feedback from the underlying Prodigy/Agent planner (either a successful plan that may not be acceptable by the user, a failure to generate any plan, or an apparent problem due to the planner taking too long), the user can asynchronously modify the goals and send them back to the planner for another round of planning. The user does this by graphically manipulating the goals by selecting the Change Goals choice on the Planning pull-down menu.

Figure 1 shows two dialogue boxes that allow the user to change the goal of making river1 impassable. By highlighting the goal the user wishes to change and then clicking on the Change button from the Current Goals pop-up menu, the goal can be modified by any of the general classes of transformations mentioned earlier. Because the goal-type transformation moves the goal predicate itself down a level of abstraction, the goal changes to achieving the state of outcome-restricts-movement rather than the state of outcome-impassable.

Figure 2 above illustrates the standard PRODIGY interface with the same problem as that in figure 1. Instead of manipulating a representation of the goals, the user manages the problem in terms of the search technology that underlies PRODIGY. The figure shows the user a choice of variable bindings that determines which air unit will be used against bridge1. In the goal-manipulation model, the user effects change directly to a representation of the goal. In the search model, the change is represented by a PRODIGY inference rule. This rule is called lower-outcome-expectations and is visible in the goal-subgoal graph of figure 2. It asserts that, if the goal to restrict movement over river1 is achieved, then the goal to have the river impassable is also achieved. The user must select this rule from among operator choices during planning. Having made this change in the step prior to the step shown in the figure, the user must now decide which air unit to use for damaging river2 by specifying a binding for variable *<AIR-FORCE>* in figure 2.

Evaluation

Cox and Veloso (1997) made the claim that the Prodigy 4.0 User Interface 2.0 was an effective interface for mixed-initiative planning because of three characteristics. Firstly the interface allows both generative and case-based planning and as such is better suited to human cognitive constraints, because case-based reasoning represents and utilizes experience in the

form of cases. Secondly the interface displays the planning process as well as the plan itself. This process is a run-time animation of the PRODIGY planning algorithm as shown in the goal-subgoal graph. Thirdly the interface has the potential to support both experts and novices in the planning technology. However these purported characteristics were never substantiated empirically.

This study examines the first and third characteristics, although we do not test the user performance under the case-based mode in PRODIGY (that is, Prodigy/Analogy; see Veloso [1994]). As noted by Cox and Veloso, however, the interface was primarily designed for the expert technologist. Some suggestions were made as to how the interface might be improved for novices. But instead of minor changes to the existing interface, we argue here that a completely new approach is needed as incorporated in the GTrans interface. Indeed, we expect that the goal-subgoal graph is of little help to the naive user who does not understand the search process. Instead it may actually interfere with decisions.

Design

An experiment was performed with human subjects to compare and contrast the cognitive models each interface implements. The experiment is designed to evaluate the differences of the two models under differing amounts of task complexity using both expert and novices. This experiment uses 18 variations on the Bridges Problem in the military domain as test problems. In these problems, insufficient resources exist with which to solve the problem completely. Choices can be made, however, so that a solution is produced that achieves a partial goal satisfaction represented as a ratio of the subject's partial solution to the optimal partial solution. Given the objective of the experiment, we identify three independent variables or factors and one dependent variable as follows.

The cognitive model of planning. This variable is either the search model or the goal-manipulation model. Seven subjects were assigned to the search model, whereas six subjects were assigned to the goal-manipulation model. The model is determined by which interface is presented to the subject from which to perform the experiment. The search model is presented by the Prodigy 4.0 User Interface 2.0, whereas the goal model is presented by the GTrans user interface.

The subject expertise. The conjecture of Cox and Veloso (1997) is that the effect of the interface on user performance may be dependent

on the user's expertise level. Subjects were categorized as either experts or novices. We define experts as those subjects who have some knowledge of AI planning, while novices are those without AI planning knowledge. If a participant has taken CS409/CS609 (Principles of Artificial Intelligence), CS712 (Multiagent Systems and Mixed-Initiative Planning), or CS714 (Machine Learning) and acquired some knowledge of AI planning (specifically of Prodigy) before the experiment, he or she is categorized as an expert. Otherwise, he or she is considered as a novice. Through a random assignment, the six expert subjects were divided four to the goal-manipulation model and two to the search model; the novice subjects were divided two to the goal model and five to search.

The problem complexity. We also predict that the effect of the planning model on the user performance partially depends on the difficulty level of a given problem. Because we normally consider a problem with a longer solution to be more complex than another problem with a shorter solution, the complexity of the problem is defined to be proportional to the number of steps in the optimal plan for that problem. Out of the 18 problems administered to subjects, the first 6 are easy problems, the second 6 are medium, and the third 6 are hard. Although we divided complexity categories solely based on solution length, most easy problems have fewer goals to solve than do harder problems. The averages are 2.33 goals per easy problem, 2.5 for medium, and 3.5 for hard. Furthermore problems labeled easier require fewer changes to their goals to generate a solution. Easy problems average 86.95 in terms of the percentage of goal satisfaction an optimal plan can achieve. Medium problems average 85.06, and hard problems average 84.80.

The measured dependent variable is as follows. The *goal-satisfaction ratio* is the ratio of the actual goal satisfaction achieved by the participant's plan to the goal satisfaction achieved by an optimal plan. In all problems, the percent achievement for an optimal plan is less than 100 due to resource constraints. If the subject generates the optimal plan, however, the goal-satisfaction ratio will be 1. As such this measure represents the quality of the subjects' planning process.

Procedure

We developed a training package that helps the participants familiarize themselves with the planning domain and the mixed-initiative planning interfaces. The training package consists of a brief introduction to the experiment,

a set of rules for the military domain, and four training problems with various complexities in the military domain accompanied with step-by-step instructions and explanations (see the appendices of Zhang [2002] for full training details and the raw subject data). Each training problem consists of a number of steps. For each step a screen shot is displayed on the left side, and the instructions together with the explanations are given in text format on the right side.

Each subject is randomly assigned to either the search condition or the goal-manipulation condition. Next, subjects are administered the training package. Guided by the detailed instructions and explanations, subjects solve the training problems under either the search model or the goal-manipulation model. Subjects are allowed to go through the training package only once. This normally lasts 30 minutes. Subjects then start to solve 18 test problems with various complexities in the military domain until they succeed or fail. If the subject succeeds and obtains a plan for a problem, the plan together with the problem information (including object information, state information, and goal information) is saved on the disk; the goal-satisfaction ratio achieved by the plan is also calculated and recorded. If the subject fails by getting into an infinite loop, the experimenter will terminate the current problem for the subject so that the subject can start on the next problem. The goal-satisfaction ratio he or she achieves for the current problem is recorded as zero. After each problem, a goal-satisfaction calculator computes the goal satisfaction achieved by the subject. Then the experimenter gives the subject the feedback on the maximum goal satisfaction as well as the actual goal satisfaction achieved by the subject.

Note that all subjects collaborate with Prodigy/Agent to generate a solution regardless of the interface, so the experiment is a test of mixed-initiative performance. Under the goal model, the user examines state conditions and manipulates the goals. The subject then invokes Prodigy/Agent for a plan. Under the search model, the user examines similar conditions (and the visual scene map) plus the goal-subgoal graph and then chooses from alternatives presented by Prodigy/Agent at each decision point where alternatives exist. If only one choice exists at any decision point, Prodigy/Agent makes the decision and continues until a search choice exists for the subject. In all cases the search tree depth-bound for Prodigy/Agent is set to 200.

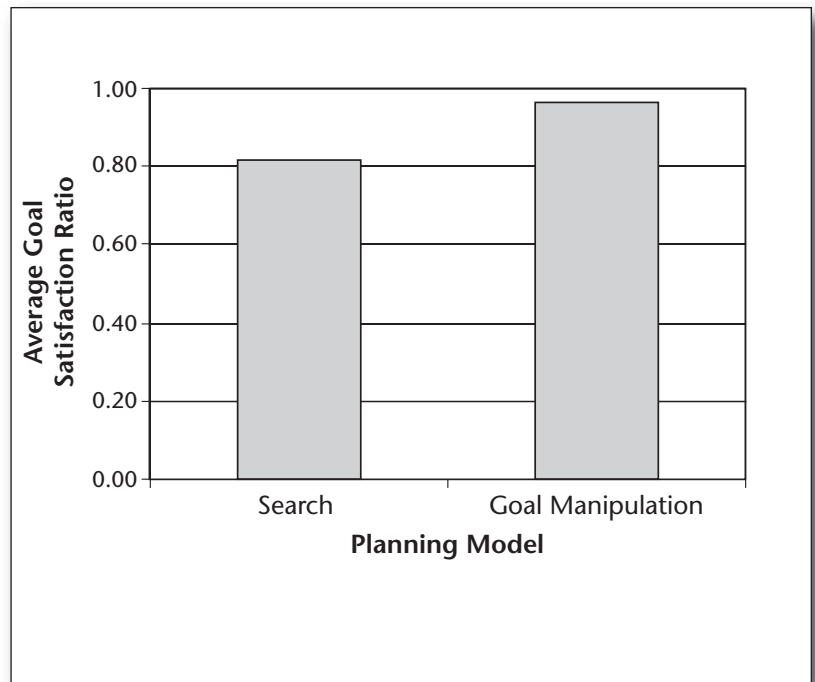


Figure 3. Goal Satisfaction as a Function of Cognitive Model.

Results

The graph of figure 3 shows the mean of the goal-satisfaction ratio under the goal-manipulation model and the search model. When presented with the goal-manipulation model, subjects achieve over 95 percent goal satisfaction on average. When presented with the search model, subjects achieve about 80 percent goal satisfaction on average.

Given that the cognitive model itself is an important factor as concluded in previous analysis, we next examine the possible relationships among three independent variables: planning model, problem complexity, and subject expertise. Figure 4 plots the average goal-satisfaction ratio for each combination of the planning model and the problem complexity. As can be observed from the graph, when the goal-manipulation model is presented to the user, the goal-satisfaction ratio generally remains the same with increasing problem complexity; but when the search model is presented to the user, the goal-satisfaction ratio decreases as the problem complexity increases. It is possible that the effect of the planning model on the user performance depends on the problem complexity.

The next step of our analysis is to examine the possible interaction effects between the planning model and the user expertise level. Figure 5 shows the average goal-satisfaction

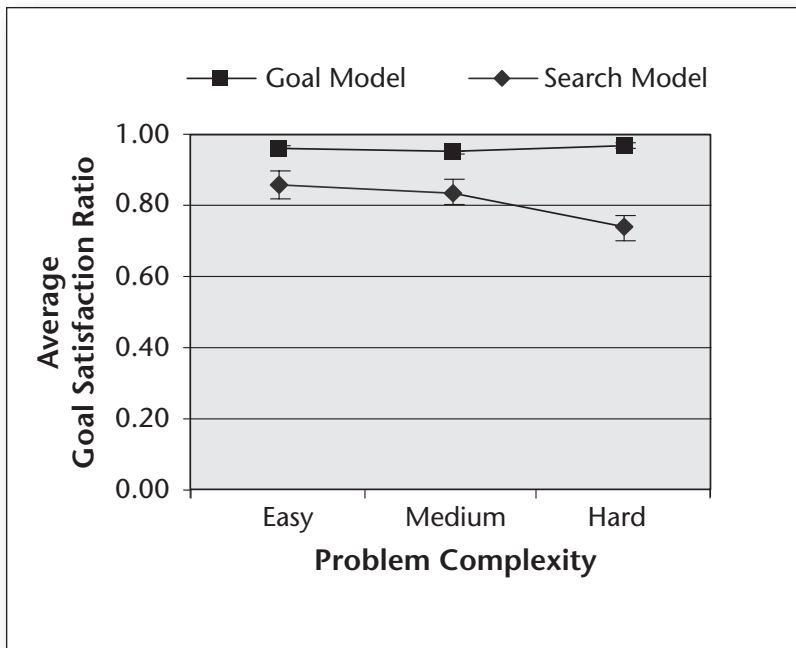


Figure 4. Goal Satisfaction as a Function of Problem Complexity.

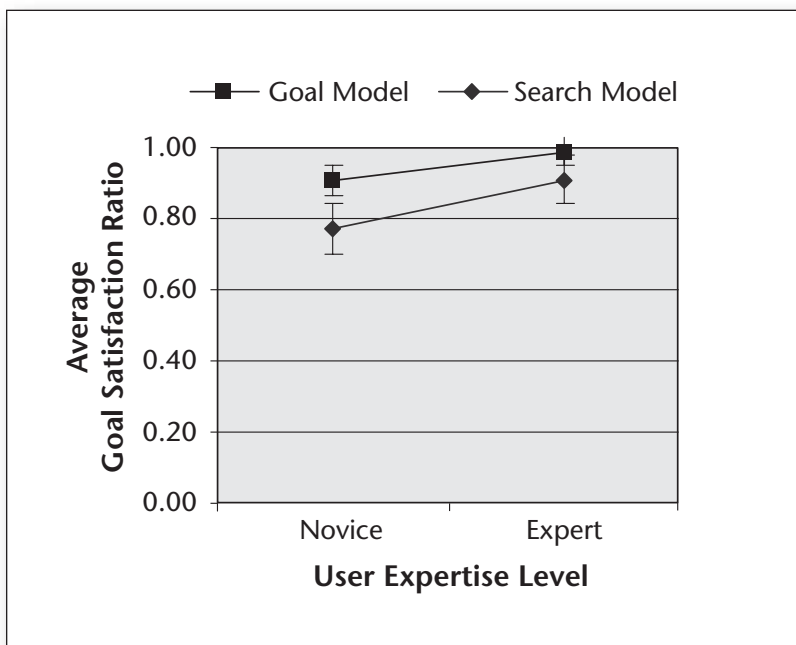


Figure 5. Goal Satisfaction as a Function of Expertise.

ratio for each combination of the planning model and the user expertise level. It is apparent that experts perform better than novices under both planning models.

These results are as yet preliminary because of the small number of subjects examined in this pilot study and, as such, are mainly suggestive. However they do support the idea that the kind of metaphor presented to a human user does affect planning performance, and the study does reinforce our intuitions about the nature of human cognition involved with planning tasks. However we do not wish to overstate the analysis. Some experts using the search model were able to outperform other experts using the goal-manipulation model on identical problems. For users that understand search well, the standard interface provides more information pertaining to system performance. Moreover because the goal-manipulation model omits a representation of the search algorithm, a user cannot modify parameters, such as search depth-bound, that significantly affect performance. Further conclusions await a larger-scale investigation planned for the future.

Related Research

We are not the first to criticize the search-driven approach to planning. Ferguson and Allen (1998) have long described planning as essentially a dialogue process between agents, both human and synthetic. More recently Myers describes planning as a hybrid goal-directed and data-driven sketching process by humans (Myers et al. [2003]; see also Forbus, Usher, and Chapman [2003]). Indeed most in the mixed-initiative planning community (see Aha [2002], Burstein [1994]) stress the need for human-in-the-loop control of planning.

A number of researchers are recognizing additional deficiencies with traditional planning applications developed by the AI community. Chien et al. (1996) make the claim that because of tractability problems with large real-world planning problems, full automation is not realistic. Consequently plans must be understandable and easy to modify by users. They further argue that the pragmatics of planning environments require tools to support the life cycle of plans. Thus planning aids such as those that support knowledge acquisition and plan verification are required. Finally, they and others (for example, Pollack and Horty [1999]) argue that there is more to planning than just plan generation. Plan management consists of replanning in the face of exogenous change, considering the goals of other agents, and

managing constraints such as time, plan quality, and flexibility.

Moreover, many other researchers have performed experiments with human users to assess mixed-initiative assistants of various sorts (for example, Sycara and Lewis [2004]; Thompson, Goker, and Langley [2004]). Furthermore, others have designed and evaluated mixed-initiative planners that represent the resource assignment part of the problem differently than we do (see Becker and Smith [2000], for example). Yet our theory and implementation explicitly represent the aspects of planning that focus on goal change. Constraint relaxation and optimization techniques represent such change implicitly. Some of the research that addresses goal change explicitly (that is, Shapiro, Lesperance, and Levesque [2005]) does so with limited scope by only considering dynamic goal additions and deletions to the current goal specification for agent systems. The only other research that explicitly treats goal transformation is that of Vukovic and Robinson (2005). They have based their work upon the theory presented here and have implemented a robust extension in the domain of web service composition.

Finally, Das (Das, Kar, and Parilla 1996; Das and Naglieri 2001) reports psychological data from human subjects and factor analysis to support his theory that human planning can be classified into three distinct types. At the lowest level is what he calls *operations planning* that chooses actions to carry out immediate goals. As plan generation, this level is most similar to the model of planning as search, although he does not claim that search processes implement the cognitive mechanism underneath. At the highest level is *activity planning*. This is a general orientation for broad life themes such as planning for retirement. Between the two is a kind of planning Das calls *action planning*. This level is more abstract than operations planning and includes plan management and environment monitoring. This theory is consistent with our view of planning as a process that manages goals, assigns resources to goals, and adjusts goal priorities over time and as the environment changes.

Alternatively, search is still the dominant model from which to study planning behavior. For example Ratterman et al. (2001) examine human planning abilities and ask whether artificial intelligence algorithms using search are accurate descriptions of human planning processes. In particular they differentiate between partial and total-order planning as alternative models of the planning process. Given their data, they argue that partial-order

planning corresponds to planning in normal adults and older children. They also argue that full-order planning exemplifies young children and adults with prefrontal cortex damage. However our results start to call into question their basic assumption that some kind of search model constitutes the most effective representation and that the main question is only which search model covers the data best. Actual human planning processes may be much more complex, perhaps more in line with the theory of Das and associates.

Conclusion

In the experiment presented here, subjects are indirectly learning the Bridges Problem evaluation function specified by Edwin and Cox (2001) and implemented in the COMAS system. This function automatically calculates the optimal resource allocation policy for planning problems given resource limited environments. What is unique is that this resource allocation task can be conceptualized as a goal-change task instead. The problem for the subjects in the goal-manipulation model is to decide which goal most needs changing rather than to decide which resources are directly allocated to which subgoal. The underlying planner will manage these details for the subject. In the search model, however, the user is confronted with a more direct and overly detailed allocation decision. See figure 2 for an example of the allocation choice presented to the user under the search model.

The results from the experiment suggest that an effect exists on human planning performance and efficiency of performance due to the GTrans interface in contrast to the standard PRODIGY interface. We argue that these effects are due to the cognitive model of planning presented to the user. Because both software systems present a mixed-initiative interface to the same Prodigy/Agent assistant underneath, the effects are not due to the autonomous planner itself. We claim that the metaphor of planning as goal manipulation is easier to understand than is search.

The differences between the two approaches can be categorized along four of the dimensions put forward in this special issue (Tecuci, Boicu, and Cox 2007). In terms of *task*, both the search approach and the goal-manipulation approach assign the intelligent assistant the task of generating the plan that solves the given problem. In the former approach, however, the user chooses search-specific decision alternatives, whereas in the latter the user chooses a goal alternative to the problem specification.

In terms of *control*, the two are very dissimilar. In the search approach, the Prodigy/Agent assistant controls the flow of processing, and the user responds to the dialogue prompts. In the goal manipulation, the user sets the goals and then invokes the planner when ready. The effect of this control division is that, in the Prodigy 4.0 UI, the user must focus on what the assistant wants. In the GTrans interface, the focus is on what the user wants.

In terms of *awareness*, another asymmetry exists. In both models of planning, the artificial assistant has no explicit representation of the user. Because of the way we designed the wrapper for Prodigy/Agent, the assistant performs the same way it would in a stand-alone mode. However, in the search approach the user is made very aware of the planning assistant in the interaction. Indeed the user is forced to plan in terms of the search tree data structure internal to Prodigy/Agent. In the goal-manipulation approach, the underlying planning assistant is less exposed to the user.

Finally the differences along the dimension of *communication* is an important one. The GTrans assistant allows the user to express choices using direct manipulation and drag-and-drop events in a map-based graphical environment. Although the search condition of the experiment placed the same map showing resources and locations on the screen for experimental control, the map was not interactive and the user interacted with the system through simple dialogue boxes.

In summary this article has contrasted two models of planning as a conflict of metaphors, but as mentioned above, the models are not mutually exclusive. Search is still an important computational technique that can be used efficiently in mixed-initiative assistants as illustrated in this article. However, we argue that search is not an appropriate model to present to the user, especially a naive one. A stronger view is that human planning processes themselves are not strictly search but rather composed of process levels in which goal manipulation is a prime process. Certainly,

however, more evidence is necessary to clarify the exact nature and composition of human cognitive planning abilities.

Acknowledgements

This research was supported by the Dayton Area Graduate Studies Institute and Air Force Research Laboratories under grant number HE-WSU-99-09 and by the Air Force Office of Scientific Research (AFOSR) under grant number F49620-99-1-0244. We thank Andrew Richardson for his assistance in collecting subject data. Much credit is also due to suggestions by anonymous reviewers. Finally Manuela Veloso and Pat Langley have been instrumental over time with respect to this research.

Notes

1. A type generalized predicate ($p \text{ TYPE}$) is equivalent to the existentially quantified expression $\exists x \mid (\text{TYPE})(x) \wedge p(x)$.
2. The GTrans home page is www.mcox.org/GTrans.
3. Prodigy 4.0 has the ability to iterate the planning process for a given problem generating a different solution depending upon the *:multi-sols* keyword parameter (Carbonell et al. 1992). GTrans incorporated this into the interface. See Zhang (2002).
4. The Prodigy/Agent home is www.mcox.org/Prodigy-Agent.
5. For example, an air unit may break down during planning. In this article we will not include a dynamic environment that changes. Rather we concentrate on planning under limited resources and leave environmental change for further study. Dynamic changes are discussed elsewhere.
6. An example analog in the logistics domain is to use truck resources to deliver most priority packages in all towns to their destinations in the morning and the remaining packages in the afternoon. We implemented the logistics domain along with an emergency management domain. Zhang (2002) reports results from a skill-transfer experiment from the military domain described here to the logistics domain.

References

- Aha, D., ed. 2002. Papers presented at the Workshop on Mixed-Initiative Case-based Reasoning, held during the Sixth European Conference on Case-Based Reasoning, ECCBR, Aberdeen, Scotland, 4 September.
- Becker, M. A., and Smith, S. F. 2000. Mixed-Initiative Resource Management: The AMC

Narrel Allocator. In *Proceedings of the Fifth International Conference on Artificial Intelligence Planning System*, ed. S. Chien, S. Kambhampati, and C. A. Knoblock, 32–41. Menlo Park, CA: The AAAI Press.

Burstein, M., ed. 1994. *Proceedings of the ARPA/Rome Laboratory Knowledge-based Planning and Scheduling Initiative Workshop*. San Francisco: Morgan Kaufmann.

Carbonell, J. G.; Blythe, J.; Etzioni, O.; Gil, Y.; Joseph, R.; Kahn, D.; Knoblock, C.; Minton, S.; Perez, A.; Reilly, S.; Veloso, M. M.; and Wang, X. 1992. Prodigy 4.0: The Manual and Tutorial. Technical Report CMU-CS-92-150. Computer Science Department, Carnegie Mellon University, Pittsburgh, PA.

Chien, S.; Hill, R.; Wang, X.; Estlin, T.; Fayyad, K.; & Mortenson, H. 1996. Why Real-world Planning Is Difficult: A Tale of Two Applications. In *New Directions in AI Planning*, ed. M. Ghallab and A. Milani, 287–298. Amsterdam: IOS Press.

Cox, M. T. 2000. A Conflict of Metaphors: Modeling the Planning Process. In *Proceedings of the 2000 Summer Computer Simulation Conference*, 666–671. San Diego: The Society for Computer Simulation International.

Cox, M. T., and Veloso, M. M. 1998. Goal Transformations in Continuous Planning. Unpublished paper presented at the AAAI Fall Symposium on Distributed Continual Planning.

Cox, M. T., and Veloso, M. M. 1997. Supporting Combined Human and Machine Planning: An Interface for Planning by Analogical Reasoning. In *Proceedings of the Second International Conference on Case-Based Reasoning*, ed. D. Leake and E. Plaza, 531–540. Berlin: Springer.

Cox, M. T.; Edwin, G.; Balasubramanian, K.; and Elahi, M. 2001. Multiagent Goal Transformation and Mixed-Initiative Planning Using Prodigy/Agent. In *Proceedings of the 4th International Multiconference on Systemics, Cybernetics and Informatics*, Vol. 7 (1–6). Orlando, FL: International Institute of Informatics and Systemics.

Das, J. P., and Naglieri, J. A. 2001. The Das-Naglieri Cognitive Assessment System in Theory and Practice. In *Handbook of Psychological Assessment: Ability, Achievement, and Behavior in Children*, ed. J. J. W. Andrews, D. H. Saklofske, and H. L. Janzen, 34–63. San Diego: Academic Press.

Das, J. P.; Kar, B. C.; and Parilla, R. K. 1996. *Cognitive Planning: The Psychological Basis of Intelligent Behavior*. London: Sage Publications.

Edwin, G. 2001. COMAS: Coordination in Multiagent Systems. Masters dissertation, Wright State University, Computer Science and Engineering Department, Dayton, OH.

Edwin, G., and Cox, M. T. 2001. Resource Coordination in Single Agent and Multiagent Systems. In *Proceedings of the 13th IEEE International Conference on Tools with Artificial Intelligence*, 18–24. Los Alamitos, CA: IEEE Computer Society.

Elahi, M. M. 2003. A Distributed Planning Approach Using Multiagent Goal Transformations. Masters dissertation, Wright State University, Computer Science and Engineering Department, Dayton, OH.

Elahi, M., and Cox, M. T. 2005. A Multiagent Approach to AI Planning. In *Proceedings of the 8th International Conference on Computer and Information Technology*. Dhaka, Bangladesh: Islamic University of Technology (IUT).

Ferguson, G., and Allen, J. 1998. TRIPS: An Intelligent Integrated Problem-Solving Assistant. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence*, 567–573. Menlo Park, CA: AAAI Press.

Finin, T.; Labrou, Y.; and Mayfield, J. 1997. KQML as an Agent Communication Language. In *Software Agents*, ed. J. Bradshaw, 291–316. Cambridge, MA: AAAI Press/The MIT Press.

Forbus, K.; Usher, J.; and Chapman, V. 2003. Sketching for Military Courses of Action Diagrams. In *Proceedings of the Intelligent User Interfaces Conference (IUI-03)*. New York: ACM Press.

Immaneni, T., and Cox, M. T. 2004. GTrans: An Application for Mixed-Initiative Collaborative Planning during Emergency Response Situations. In *Proceedings of the 2004 International Symposium on Collaborative Technologies and Systems*, ed. W. W. Smari and W. McQuay, 121–126. San Diego: Society of Modeling and Simulation International.

Myers, K. L.; Jarvis, P. A.; Tyson, W. M.; and Wolverson, M. J. 2003. A Mixed-Initiative Framework for Robust Plan Sketching. In *Proceedings of the Thirteenth International Conference on Automated Planning and Scheduling*, ed. Enrico Giunchiglia, Nicola Muscettola, and Dana Nau, 256–265. Menlo Park, CA: AAAI Press.

Newell, A., and Simon, H. A. 1972. *Human Problem Solving*. Englewood Cliffs, NJ: Prentice-Hall.

Newell, A., and Simon, H. A. 1963. GPS, A Program That Simulates Human Thought. In *Computers and Thought*, ed. E. A. Feigenbaum and J. Feldman, 279–293. New York: McGraw Hill.

Pollack, M. E., and Horty, J. F. 1999. There's More to Life than Making Plans. *AI Magazine* 20(4): 71–83.

Ram, A., and Leake, D. 1995. Learning, Goals, and Learning Goal. In *Goal-Driven Learning*, ed. A. Ram and D. Leake, 1–37.

Cambridge, MA: The MIT Press.

Ratterman, M. J.; Spector, L.; Grafman, J.; Levin, H.; and Harward, H. 2001. Partial and Total-Order Planning: Evidence from Normal and Prefrontally Damaged Populations. *Cognitive Science*, 25(6): 941–975.

Schank, R. C. 1982. *Dynamic Memory: A Theory of Reminding and Learning in Computers and People*. New York: Cambridge University Press.

Schank, R. C., and Abelson, R. P. 1977. *Scripts, Plans, Goals and Understanding: An Inquiry into Human Knowledge Structures*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Shapiro, S.; Lesperance, Y.; and Levesque, H. 2005. Goal Change. In *Proceedings of the International Joint Conference on Artificial Intelligence*, 582–588. Detroit, MI: International Joint Conferences on Artificial Intelligence.

Sycara, K., and Lewis, M. 2004. Integrating Intelligent Agents into Human Teams. In *Team Cognition: Understanding the Factors That Drive Process and Performance*, ed. E. Salas and S. Fiore, 203–231. Washington, DC: American Psychological Association.

Tecuci, G.; Boicu, M.; and Cox, M. T. 2007. Seven Aspects of Mixed-Initiative Reasoning. *AI Magazine* (28)1.

Thompson, C. A.; Goker, M. H.; and Langley, P. 2004. A Personalized System for Conversational Recommendations. *Journal of Artificial Intelligence Research* 21: 393–428.

Veloso, M. M. 1994. *Planning and Learning by Analogical Reasoning*. Berlin: Springer.

Veloso, M. M.; Carbonell, J. G.; Perez, A.; Borrajo, D.; Fink, E.; and Blythe, J. 1995. Integrating Planning and Learning: The PRODIGY Architecture. *Journal of Theoretical and Experimental Artificial Intelligence* 7(1): 81–120.

Vukovic, M., and Robinson, P. 2005. Goal Morph: Partial Goal Satisfaction for Flexible Service Composition. *International Journal of Web Services Practices* 1(1–2): 40–56.

Williamson, M. 1996. A Value-Directed Approach to Planning. Doctoral dissertation, University of Washington, Computer Science Department, Seattle, WA.

Zhang, C. 2002. Cognitive Models for Mixed-Initiative Planning. Masters dissertation, Wright State University, Computer Science and Engineering Department, Dayton, OH.

Zhang, C.; Cox, M. T.; and Immaneni, T. 2002. GTrans Version 2.1 User Manual and Reference. Technical Report No. WSU-CS-02-02. Dayton, OH: Wright State University, Department of Computer Science and Engineering.

AAAI Annual Business Meeting

The annual business meeting of the Association for the Advancement of Artificial Intelligence will be held at 12:45 PM, Monday, July 23, 2007 in the Hyatt Regency Vancouver in Vancouver, British Columbia, Canada.



Michael T. Cox is a senior scientist in the Intelligent Distributing Computing Department of BBN Technologies, Cambridge, Massachusetts. Previous to this position, Cox was an assistant professor in

the Department of Computer Science and Engineering at Wright State University, Dayton, Ohio, where he was the director of Wright State's Collaboration and Cognition Laboratory. His research interests include case-based reasoning, collaborative mixed-initiative planning, understanding (situation assessment), introspection, and learning. More specifically, he is interested in how goals interact with and influence these broader cognitive processes.



Chen Zhang is a Ph.D. candidate in management information systems at the Krannert Graduate School of Management at Purdue University. She earned her MBA degree from Bowling Green State University

and her M.S. degree in computer science from Wright State University in 2000 and 2002, respectively. Her current research interests include open source software development teams and online community dynamics.

New & Recent Books from AAAI Press

Twenty-Second AAAI Conference on Artificial Intelligence

Adele Howe and Robert C. Holte,

Program Chairs

AVAILABLE IN JULY

International Conference on Automated Planning and Scheduling 2007

Mark Boddy, Maria Fox, and Sylvie Thiebaux,

Program Chairs

AVAILABLE IN SEPTEMBER

Twentieth International FLAIRS Conference

David Wilson and Geoff Sutcliffe,

Program Chairs

AVAILABLE IN JULY

Tenth International Conference on Principles of Knowledge Representation and Reasoning

Patrick Doherty and John Mylopoulos,

Program Chairs

NOW SHIPPING!

Thinking about Android Epistemology

Kenneth Ford, Clark Glymour,

and Patrick J. Hayes

www.aaai.org/Press/Books/ford.php

Smart Machines in Education

Kenneth Forbus

and Paul Feltoich

www.aaai.org/Press/Books/forbus.php

Data Mining: Next Generation Challenges and Future Directions

Hillol Kargupta, Anupam Joshi,

Krishnamoorthy Sivakumar, and Yelena Yesha

www.aaai.org/Press/Books/kargupta2.php

New Directions in Question Answering

Mark Maybury

www.aaai.org/Press/Books/maybury3.php

K-CAP 2007



October 28-31 2007
Whistler, Canada

Fourth International Conference on
Knowledge Capture

- Advice taking systems
- Knowledge acquisition tools
- Knowledge extraction systems
- Knowledge-based markup techniques
- Acquisition of problem-solving knowledge
- Knowledge Engineering & the Semantic Web
- Mixed-initiative planning & decision-support tools
- Knowledge Engineering & modelling methodologies

K-CAP 2007 Organising Committee

Conference Chair Derek Sleeman (Univ. of Aberdeen)

Program Chair Ken Barker (Univ. of Texas)

Workshop & Tutorials Chair John Gennari (Univ. of Washington)

Local Arrangements Rob Kremer (Univ. of Calgary)

Publicity Jeff Pan (Univ. of Aberdeen)

Treasurer John Thompson (Boeing Company)

Sponsorship (Co-Chairs) Yolanda Gil (USC),
David Leake (Indiana Univ.)

Webmaster David Corsar (Univ. of Aberdeen)

Workshop/Tutorial proposals April 7, 2007

Technical papers May 6, 2007

Posters July 1, 2007

<http://www.csd.abdn.ac.uk/kcap07>

K-CAP 2007 is sponsored by ACM SIGART and held in cooperation with AAAI