This article appears as Chapter 17 of *Cognitive Technology: In Search of a Humane Interface* Edited by B. Gorayska and J.L.Mey (Editors). Elsever Science B.V., 1996.
(It is an edited version of an article that first appeared in the *Proceedings of the DARPA/RL Planning Initiative, February 1994* , Edited by Mark H. Burstein)

Issues in the Development of Human-Computer Mixed-Initiative Planning Systems

Mark H. Burstein
BBN Systems and Technologies, a division of
Bolt Beranek and Newman Inc.

Drew V. McDermott
Department of Computer Science
Yale University

> "A mixed-initiative system is one in which both humans and machines can make contributions to a problem solution, often without being asked explicitly."
> -- Jaime Carbonell, Sr.

Abstract

Mixed-initiative *planning* systems are systems in which humans and machines collaborate in the development and management of plans. The "initiative" in such systems is shared in that each can contribute to the formulation, development, management, refinement, analysis and repair of the plans developed "without being asked explicitly". Intuitively, the goal is to develop a style of interaction where both men and computers can further the state of an ongoing planning activity through contributions that include the many activities that surround the actual construction of plans, but are not normally part of what current Artificial Intelligence (AI) planning systems do. In this paper, we discuss some of the research areas that are likely to be important in transitioning prototype AI planning systems to a new role as collaborator in a mixed human/machine collective planning process. This paper is, in large part, the result of a series of discussions that took place in late 1993 and early 1994, among a group of AI researchers working on planning, supported by the ARPA/Rome Laboratory sponsored initiative in military planning and scheduling.

Introduction

The overall objective of research on *mixed-initiative planning* (MIP) is to explore productive syntheses of these strengths of both men and machines to build effective plans more quickly, and with greater reliability. Users need better, more intelligent, more active problem solving support than the current generation of plan authoring tools can provide, and AI planning systems need human support in such areas as problem definition, information interpretation, and spatial/perceptive reasoning if they are to be useful in real-world applications.

Through a series of discussions of both the electronic and face-to-face variety, a team of researchers in AI-based planning came to some substantial agreement on a set of issues that will need to be addressed in the development of mixed-initiative planning systems[1]. This chapter documents some of the conclusions reached in those discussions.

We were motivated to consider these questions by our work in the domain of military planning and scheduling, however, we believe that much of the discussion applies equally well to any system of people and software engaged in planning activities will be faced with much the same set of issues. We will define military planning for the purposes of this discussion as the organization of resources to carry out a military or humanitarian objective. For example, to plan for the evacuation of civilians from a region that is in turmoil, or the movement of men and materials to an area so as to provide relief after a natural disaster, planning activities include the identification of appropriate resources for carrying out the objectives, including the transportation of those resources to the area in which they will be used in advance of their planned time of use. A major part of military planning is the identification and planning for the movement of the resources to and from the region in question.

Our larger interest in mixed-initiative planning systems grows out of some observations of the strengths and weaknesses of both human and automated planning systems as they have been used (or considered for use) in the past. Humans are still better at formulating the planning tasks, collecting and circumscribing the relevant information, supplying estimates for uncertain factors, and various forms of visual or spatial reasoning that can be critical for many planning tasks. Machines are better at systematic searches of the spaces of possible plans for well-defined tasks, and in solving problems governed by large numbers of interacting constraints. Machines are also better at managing and communicating large amounts of data.

In addition to the potential for synergistic improvements in current planning processes by combining the strengths of these different kinds of planners, we must also recognize the currently burgeoning roles of electronic collaboration and electronic data access on tasks of all kinds. As network technology has matured and become widespread the notion that work on a shared task could be physically distributed has become a reality. Electronic conferencing and workflow tools are starting to reach the market place. On-line access to huge amounts of information via wide-area networks must be taken as a given. Multi-

---

[1] The face-to-face meetings took place in December, 1993 and January, 1994 at Yale University and BBN. Participants in these discussions included: James Allen (Rochester U.), Marie Bienkowski (SRI), Mark Burstein (BBN), Steve Cross (ARPA), David Day (MITRE), Gary Edwards (ISX), Nort Fowler (Rome Lab.), Matt Ginsberg (U. of Oregon), Jim Hendler (U. of Md.), Leslie Kaelbling (Brown U.), John Lemmer (Rome Lab.), Drew McDermott (Yale U.), Stephen Smith (CMU), Austin Tate (U. of Edinburgh), Craig Wier (ARPA), and David Wilkins (SRI).

agent distributed AI systems of many kinds are being explored, though much of the work is still in its early stages. We need to develop a clear vision of the role of these technologies in true mixed-initiative systems as well.

From a pragmatic standpoint, we took the major research question to be the following:

> How can human(s) and machine(s) best share information about and control of plan development?

That is how do we get positive synergy from interactions between human planners and automated planning and support software such that:

- Each works in areas where they perform best;
- "Agents" (using the term loosely to refer to both humans and software systems) are able to use appropriate (often concise or abstract) representations for communication of plans, constraints, assumptions and analyses to communicate with other agents that have different areas of expertise or functionality, and different kinds of communications skills; and
- "Agents" have means of acquiring and transferring authority for planning related tasks.

The remainder of this chapter elaborates on these questions and posits some directions to pursue in efforts to find answers to them. We begin by taking apart current notions of AI planning techniques to examine where they will need to change, perhaps radically, in order to fit into the world of collaborative problem solving. We then discuss some ideas about the near-term, focusing on ways that current generation AI-based planning systems might be adapted to support a more mixed-initiative style of interaction.

Planning as a Collaborative Activity

As one moves from the current-day perspective taken by most AI planning researchers, who took it as their objective to develop "stand-alone" planning systems, to a model where issues of communication and collaboration are more central, a number of assumptions underlying research in this area must be questioned. We began with a model of planning activity in which we assumed that many "agents", of both the human and software variety, are actively cooperating, and from that perspective looked for aspects of current planning system theory and practice that would have to change.

The AI conception of planning is largely dominated by the notion of a single-threaded search through a space of possible (partial) plans for satisfactory solutions (e.g., Wilkins, 1988). This "classical" view is typically implemented in terms of some form of goal-refinement, back-tracking search algorithm. When the search for a plan is to be coordinated among many "agents", some of which are human, this model must be seriously questioned. Present-day, autonomous planning algorithms assume that plans are to be developed by systematic exploration of alternative plan refinements under programmatic control. The objective of these systems is usually to find a *satisfactory* plan, rather than an optimal one, although this distinction does not matter greatly here (yet). Human planners, on the other hand, do not search systematically in this fashion, but rather may jump around in the space of possible plans, perhaps based on preliminary analyses of what is "hard" about the problem, or more simply on pre-existing knowledge of particular solution models. If the plan is for an important objective, they will typically explore several approaches to some limited depth before choosing a path to pursue to a completely detailed plan. Therefore, in a mixed-initiative approach to search during planning and problem solving we must expect the human planners involved (hereafter "the users") may wish to dictate where and how much to search, while at other times, automated planning "agents" may be given reign to search problem spaces under their own control. Some redundancy in these collaborative efforts must be viewed as good, even essential, as the search techniques of these different kinds of agents are likely to be very different, and to the extent their results are comparable, each can serve to support and correct the other.

A second theme of this discussion is the need to support a variety of kinds of dialogue during planning. In an informal study of collaborative planning reported in (Allen, 1994), dialogues were collected of pairs of people working together to solve a planning problem in an artificial environment consisting of trains that could travel between cities, and needs for those conveyances. In each case, one of the subjects played the role of the "system" and the other the role of the "manager". The two could not see each other, and did not know each other. The only shared information they had was an initial map of the TRAINS world. Each interaction between the players was categorized according to its general purpose. Table 1 summarizes the relative frequency of these interactions, by category.

The kinds of interactions typically supported by current-day planning systems comprised less than 25% of the total. While this data is merely suggestive, it strongly suggests that effective collaboration in plan development must address as a central issue the question of how to manage and support the variety of kinds of dialog that were seen here as necessary to this kind of collaborative problem solving.

Evaluating & comparing options	25%
Suggesting courses of action	23%
Clarifying and establishing state	13.5%
Discussing problem solving strategy	10%
Summarizing courses of action	8%
Identifying problems and alternatives	7%

Table 1: Frequency of interactions by type

Issues for Mixed-Initiative AI-based Planning Systems

Given the disparate styles of problem solving found in people and machines, and the need for coordination during collaborative planning among these very different kinds of agents, each of the following areas of research related to AI-based planning and collaborative problem solving must be addressed to develop software capable of supporting a mixed-initiative model for planning:

Plan-Space Search Control Management addresses the question of how to coordinate various kinds of agents' exploration of potential solutions to a planning problem.

Representations of and Sharing of Plans by different kinds of agents for different (but related) purposes, and for communications among the collaborating agents.

Plan Revision Management is the problem of coordinating revisions to "the plan", especially if it is being revised during execution.

Planning and Reasoning under Uncertainty is an ongoing research area within the AI community. It addresses issues related to projecting into the future to anticipate the situation at the time that planned actions will occur, to enumerate their possible outcomes, and to estimate the likelihood of those outcomes. It is an area not been adequately addressed in most AI planning systems, and is one of the reasons for their lack of acceptance by user communities. It is hoped that this issue will benefit from a more synergistic man/machine approach.

Learning from past planning attempts, their results when executed, and from ones collaborators: While machine learning techniques have had some limited successes, it seems clear that there is far to go, and that this research area will have an important role to play in mixed-initiative environments generally. Agents are not "born" team-players. They must adapt to roles on a team based on the strengths and weaknesses of their teammates.

Inter-agent Communications and Coordination is not about planning per se, but is important here in recognition of the different ways human and machine team members might need to interact during planning. Issues related to this topic will be raised in each of the other topics as well. When viewed from the point of view of a collaboration between people and software systems, we expect that planning systems, taken as the union of all these participants, must be supported by tools that enable a variety of kinds of communications (graphical, language-based, audible) to take place. These communications are most certainly not just about the plans themselves, as the experiment cited above suggests.

We take each of these areas in turn:

Search Control Management

Control Dialogues to Establish Collaboration Patterns. There needs to be some amount of ongoing "dialogue" between the human and machine planning agents about how search will be organized, divided and conducted. It is most likely that the human planners will need to maintain control of the setting of major objectives, and cede control of some of the more mundane aspects of planning to the machine. This dialogue may not be, indeed is not likely to be, done with true natural language, but by a variety styles of interaction with a graphical user interface.

Assuming for the moment that a human "user" maintains control during these dialogues and is characterizing planning sub-tasks for his or her machine and human associates, then he or she must describe how to do search through a space of possible plans, and express to the other agents how their search is to be bounded. This dominant user must have the means to express search constraints of many kinds, including such things how to decompose the planning into partitionable subtasks, what assumptions to make in subtask planning about available resources, what assumptions to make about the world in which the plan will execute, and more general controls such as whether to be optimistic or pessimistic about the utilization of resources, and the "cooperativeness of the world" in which the plan will be executed.

One style of cooperation that might be selected by a user would have that user retaining control of and directing search at the higher, more abstract levels of plan development, while ceding to automated planning agents responsibility for pointing out critical areas to address carefully (such as potential resource shortages), or charging them to explore in detail specific issues or plans for known subtasks. Further communications between these various automated agents and the human planner(s) would involve summarizing the results of plan analyses and the presentation of potential options for doing specific subtasks.

On the other hand, it seems clear that this is just one of a range of collaboration models, and that different users will wish to vary the form of search control and collaborate more or less closely in plan development, depending on how "cut and dry" the problem to be solved is, *and their understanding and faith in* the capabilities of their electronic collaborators.

Variable speed and resolution response. In the end, we want our collaborative planning tools to produce detailed plans. But at earlier stages of the process, we want them to assist in generating cruder plans quickly, so that preliminary analyses can be performed. In essence, collaboration at the early stages of plan development may be a time of consideration of and dialog about abstract alternatives, leading to the more precise formulation of objectives. This characterization of the preliminary phase of planning suggests that planning problems need to be viewed as solvable at different levels of resolution (or "abstraction"), which, in effect, means having multiple different representations of what "plans" or "solutions" are.

Decoupling and recombining plans The user should have the ability to isolate sets of subgoals that are only loosely coupled to the rest of the overall set of goals, in the sense that plans for those subgoals can be developed in parallel and later be combined. The techniques required to identify such goal sets and in combine the plans developed into an overall solution are still open research areas.

Context registration. There will need to be means for constantly conveying where in the problem solving the *team of humans and software agents* is currently working, and who is performing what tasks. We will refer to this the maintenance of a shared problem solving context as *context registration.*. When a software agent completes a task, the human-computer interface, itself a kind of coordinating and mediating "agent" must be able to give the user a succinct, coherent picture of the current "state of play" in the planning process, and a summary of the conclusions reached, which can be interrogated for more detail.

Users may also need to convey preferences for levels of "communications volume" or communications bandwidth in their dialogues with the other planning agents. Plans are often too large to view in a single picture or short text. Many different perspectives and styles of visualization are useful at different times, and these typically rely on abstractions and approximations that help to convey the gist of a plan or subplans under discussion. Dialogues must be supported addressing the issue of the amount and presentation style of information to be conveyed (e.g., to the user) at different stages of planning. These dialogues may tend to occur at landmark points in the search for an effective plan, when the locus of planning activity is about to shift to a different level or perspective. Good techniques for graphical and other forms of summarization will be critical, and all communications techniques for conveying perspectives on the plans under consideration must readily support requests for elaborations and explanations as needed.

Intent recognition. Oftentimes, users will not explicitly convey all of the constraints they know to apply in the context of a planning problem. This is a problem for automated, "autonomous" planning tools. When people collaborate to solve planning problems, it is assumed that each participating planner will understand the context in which the plan will be executed well enough to make new planning assumptions as they elaborate their part of the plan, and seek out any additional information needed to make their plans effective. They will also identify problems with their own assumptions and those of the agent who tasked them in order to "shore up", if possible, or reject, if necessary, possible plans that they might have otherwise produced. For example, if they cannot make a workable plan with the resources assumed available they will communicate this back to the agent that tasked them. This shared "world context" extends also to shared knowledge of the many standard components of plans they might use to achieve particular goals.

Consider, for example, a request made by one agent that another develop a plan for a conference in another city. The agent tasked could infer from what was stated (city, dates, attendees) not only that subplans would be needed for reserving the meeting space, and alerting the participants, but also that plane reservations would be needed to transport the participants if the city was distant, and that hotel reservations would be needed if the meeting was longer than would permit traveling on the same day. The agent developing this plan might discover, on researching the problem, that the hotels available were not close to the meeting space, and so need to generate plans to reserve cars for local travel as well. The agent who requested the plan be developed might, even though he or she did not ask for a rental car, be explicit in requesting that particular kinds of sound and projection equipment be made available at the meeting, because that was a piece of information not inferable from the general task.

Clearly constraints can be left implicit in this kind of communication of an abstract plan to be elaborated by another agent, when tasking another agent to refine that plan. Indeed it is almost necessarily the case that details will be left out, if the communication is to be succinct enough to make it worth defining the task for another to carry out. Mixed initiative planning systems must be capable of using prior knowledge of the particular domain of planning (and the preferences of the agents specifying the abstract plan) to fill in such details, in addition to making reasonable assumptions about the environment that the plan will be carried out in, so that the specification and communication of a planning subtask is not overburdened with a large volume of "common-sense" details. On the other hand, the requesting agent must know what details cannot be left implicit if the resulting plan is to be satisfactory. Each must have at least a partial model of what knowledge the other has and what assumptions must be made explicit in their communications during collaboration.

It will be important, if such collaborations are to be successful, that the automated systems do not unnecessarily impede plan development by requiring too many details be filled in advance or asking too many questions when the information can be inferred. On the other hand, it is equally important that these systems be able to ask *refinement questions* when important details are omitted. This is one of the many kinds of dialogue that must be supported. We also discuss some of the problems raised by this issue in a later section of this chapter devoted to initiative.

Plan analysis. The computer must provide the user with a set of tools for analyzing fragments of plans, and comparing versions of plans and plan fragments that were constructed under different assumptions. These tools should include plan displayers that highlight different information, statistical packages for analyses of uncertain outcomes, and sensitivity analyzers that check whether actions are to take place under conditions leading to higher than normal failure rates. Means must be developed for describing, requesting, and/or attaining automatically the information required for use of these tools.

Representations of plans and plan-related information sharing

For collaboration over plans to work, we assume that there must be shared representations of those plans, and means of extracting and reformulating those representations into forms convenient to the various collaborators. This does not *necessarily* mean that there is a single place where the full representation of a plan is stored, but that collaborators can get efficient access to pieces of the plan, as needed. Visualizations of plan representations must be intelligible to human users, and extractions/reformulations of planning constraints and other plan-related information must be possible to provide information for specific automated planning and/or analysis processes. In reality, this is perhaps the biggest barrier to collaboration. Dialogues about the plan under development must be framed in terms of consideration of alternatives and their justifications, almost in the style of an argument (Allen and Ferguson, 1994).

Most planning done by people is at best reduced to raw text and graphics, rather than encoded in electronic forms amenable to manipulation by computer systems. Another role of interfaces to planning systems must be to make it as convenient as possible to maintain plans in electronic forms, rather than more exclusively human readable forms.

Shared representations. It is generally assumed that if the planning process is distributed, there must be a representation of "the plan" that is shared among the collaborators. It should support a variety of visualizations, abstractions and translations into more specialized forms for specific purposes.

Abstractions. It will be necessary to represent plans at different levels of detail. Even after a plan has been elaborated, the user must be able to see a "low-resolution" version highlighting particular aspects of the plan.

Visualizations. If the user is to have a chance of understanding the current state of a plan that is only partially specified, then there must be many ways for users to view and edit any part of the plan, as well as its justifications and ramifications. For example, it should be possible to display the state of affairs expected at any point in a schedule. It should be possible to run a "movie" that shows possible unfoldings of the plan over time. Visualizations should support a variety of perspectives and "filters" on such views, highlighting such things as resource utilization, workload, transportation of materials, etc.

Uncertainty. The user must not be misled into thinking that nominal plan values are certain. The system must help to disabuse him or her of such illusions. Uncertain information is likely to be handled in several different ways in representations of plans. Where possible, sources of uncertainty should be recorded, along with planning decisions dependent on those uncertainties, so that plan revision can be done more automatically. The quality of information and of its source should be available wherever it is likely to be suspect. In addition, tools such as decision theoretic models that explicitly reason with probabilistic information should be supported, where applicable. These tools require more detailed estimates of probabilities than just discrete alternatives.

Versioning, author tracking, change authority. As part of the support for interactive, collaborative plan development there will be a need for better mechanisms for maintaining versions of partially developed plans, both so that collaborators can explore options in parallel without global commitments, and so that plans can be compared, contrasted, combined, and, in general, referred to without confusion. Information associated with different versions that will be important in the collaborative dialogue includes information about authorship, who has authority to change particular aspects of that plan version, what views of the plan are most useful, etc.

Plan Revision Management

There is a serious sense in which one is never planning "from scratch", and in which planning is never completed. Planning should be viewed as a continuous, ongoing process involving alternatives exploration, refinement, diagnosis, repair and recombination, in the face of constantly changing information. Even before execution has begun, human planners are constantly striving to improve the quality of the information used for planning, and that is as likely to cause replanning to occur as run-time contingencies.

Maintaining continuity between plan versions. As execution time grows imminent, there is a need to alter the patterns of plan change, and one's preferences among alternatives, toward those that maintain continuity, or minimize execution-time replanning at lower levels of detail. Activities involving advance preparation should not be changed once they have begun, unless those changes are consistent with the preparations made. Activities in progress incur even greater costs if changed in an incompatible way. Future planning systems need to be able to deal with this range of continuity-maintenance constraints due to the potentially varying need to minimize disruption of ongoing activities.

Run-time replanning. True execution-time replanning raises another set of issues beyond continuity. As the time available for planning and the age of one's information about the current state of the world diminishes, your team's own activities in executing "the plan" must be considered as part of the process. Once execution starts, parts of the plan become historical and what matters is the relationship of the outcome of those parts of the plan to the remainder of the unexecuted plan. Indeed, during execution, predicting a future state of affairs and its impact on the remainder of the plan may be based on observing *unexpected* changes occurring in the present.

Coordinating multi-agent planning tasks. The whole situation is complicated by the fact that multiple agents may be attempting to modify an ongoing plan at essentially the same time, and that different agents may have responsibility for, indeed be the only ones capable of, revising particular portions of the plan. There are a number of issues to be addressed here: *coordination of plan update authority*, either through a central manager, or through distributed authority management; *information update notification* , to ensure that the proper agents are made aware of the information that may lead them to revise the portions of the plan they control; *information and plan consistency management* , which is needed in the face of the potential acquisition of inconsistent information and the possibility of contradictory plan changes being made by different agents simply because some take longer to complete their work and update the plan; and *resource coordination* , such as between different plans that may be executed at the same time.

Planning under Uncertainty.

Planning, perhaps more than other kinds of reasoning, is fundamentally based on uncertain information. There is the uncertainty in the timing of availability of resources, uncertainty in one's own information sources, uncertainty in the actions of other agents operating in the environment, and uncertainty in one's ability to estimate the outcome and time required to complete planned actions. All of these different sources of uncertainty can sometime be modeled as discrete alternatives, as is currently done in AI planning systems, and at other times it might be described probabilistically.

The key point is that when planning is to be done collaboratively, and the goal is not necessarily to get down to small atomic actions that can be executed by single agents in the world, managing uncertainty typically needs to be done more explicitly.

We also observe that:

- People can't deal with too many (slightly varying) alternative plans or scenarios. One can't overload a user with a million alternative scenarios whose probabilities, if known, would sum to 1. Identifying and analyzing *qualitatively distinct* plans should be stressed.
- There are a variety of current tools designed to help humans analyze (often implicit) uncertainty in their current plan representations. We anticipate continued frequent appeals to *sensitivity analyses* that reveals how the projected effectiveness of plans change with changes in key resources.
- In many environments, the emphasis should be on finding robust plans, as opposed to ones that will be optimal if no assumptions are violated. The system should point out which resources are most likely to be under stress (e.g., waypoints in a transportation plan that are projected to operate at capacity).
- Getting users to assign probabilities to events is hard. Getting users to provide (even qualitative estimates of) probabilities for every uncertain fact is nearly impossible.

Learning

Teams don't start out working well together, they must "grow into it" by learning the most useful ways of contributing, and the times not to contribute (too much). Since people are already fairly adaptive (within limits), the issue is one of finding opportunities for the automated systems to do useful learning to make them better team members. Some near-term objectives here would be learning of:

User preferences: If the user repeatedly asks for a particular type of statistical analysis, visualization, node expansion, constraint handling preference (conservative or liberal), or problem decoupling, the system could begin to anticipate such requests and automatically do them or inquire if they should be done. A recent example of this kind of learning is (Sycara et al, 1994), for an adaptive case-based scheduling system.

Prior plans and their effects: Users may want to generate new plans by modifying old ones, in whole or in parts. Case-based reasoning techniques are a potentially easy way to get plan-level learning into MIP systems. The system could help by indexing and retrieving stored plans as similar goals are stated for new problems, and by recording failures and the conditions that led to them, so that they can be brought to the attention of users if similar plans are constructed.

General and domain-specific planning knowledge or heuristics: If the automated planning components of a mixed-initiative system are to keep pace with change, or improve on their initial capabilities as provided by the system designers, there will be a continuing need to develop and refine the heuristics for the automated planning tasks that the system provides. It is desirable that at least some of this knowledge updating and maintenance come about as a result of interactions with the human users of the system. This may motivate some additional (possibly off-line) clarification dialogues so that the system can learn from user directives about such things as searching through the planning space, operator preferences under different conditions, etc.

Inter-agent communications and coordination

Given the highly distributed nature of military planning, it is going to be important that mixed-initiative planning systems of the future be open systems where multiple humans and multiple machines are collaborating in an open architecture. While this adds a number of complications to the study of mixed-initiative systems, many of the issues need to be addressed equally well in distributed AI systems research, in improved technologies for electronic collaboration between humans, or in distributed systems support generally. Nonetheless, there are a number of issues that are unique to person-machine collaboration, and to large-scale distributed planning systems that involve both human and machine agents.

As a first pass, it seems useful, until we see more artificially intelligent software agents running around, to break down the issues along the lines of whether the agents communicating with each other are human or machine. Many of the issues related to electronic collaboration between people are now being addressed in distributed groupware and workflow systems that are available commercially. The issues that relate to inter-software-agent collaboration are largely being addressed by the Distributed AI community and the AI knowledge sharing research community. However, the issue of knowledge sharing is ubiquitous, and there are some specific things to be said about this with respect to planning systems:

Distributed information management will need to be coordinated among many disparate knowledge and information sources, with varying amounts of sophistication, varying capabilities for query processing, and with varying levels of accuracy and timeliness in the data provided. Where to store shared data, what kinds of transactional mechanisms are required, how to make access fast enough, how to make sure information is disseminated in a timely fashion to those who need it, and how to control access to it are all ongoing concerns.

Maintenance of and timely access to shared plans is a related but more specific issue: it seems inevitable, given near term hardware technologies and the large amounts of information that is required for large-scale, distributed planning that plan-related data is passed around between computers, reformulated for use by different agents, and cached in those new forms. This potentially makes the maintenance of plan version consistency and information access support *every agents' problem* .

The central issue for to mixed-initiative planning systems is communication between humans and software systems. Echoing some of our earlier comments, we see the important research areas here as:

Dialogue-based task management for interactively controlling search, communications bandwidth, asynchronous interruption management, and delegation.

Context registration using many kinds of clarification dialogues, summarization, elaboration and explanation techniques.

Flexible, interactive visualizations of plans and support information from different perspectives as a means of conveying information to users, and providing graphical contexts for communications to the machine.

Information acquisition and management, which often dominates the planning process, also means the transformation of information into usable electronic forms, so that it can be related to electronic versions of the plans under development. There are a number of potential opportunities for greater machine "initiative" to provide assistance in this area, in the world of the

Internet, as well as many potential stumbling blocks related to the interpretation of text and graphics into representational forms.

A key point here is that there must be increased effort put into ensuring that useful representations of plans are captured in *usable* electronic forms, where the uses are by both humans and software systems. Constraints that are left implicit in the head of users or in raw text cannot be part of the cooperatively agreed upon plans that are developed, and this means lost opportunities for automation, especially in replanning.

Initiative Is In The Eye Of The Beholder

If initiative is viewed as acting to achieve shared goals, *without being asked explicitly*, it seems fair to ask what it means to ask a computer program to do something. A program will be "triggered" under certain conditions; the end user may classify some of these conditions as "having been explicitly asked," but for the person who wrote the code the distinction is not terribly significant. After a human user has used such a program for a while, he or she may well come to expect the computer's actions as predictable responses to their own. For example, users of spreadsheets do not perceive the programs' calculations (which result from changes in data values in cells) as being the result of system "initiative". In the same degenerate sense, programs that complain when you enter invalid responses to their prompts are not seen as engaging in "clarification dialogues".

From this perspective, we should be careful with the use of the term "initiative," and focus on revising our models of automated systems' interactions with users, with the goal of improving their utility in collaborative endeavors. By focusing too much on initiative for initiative's sake, we risk burdening ourselves with the impractical task of producing modules that impress us with their intelligence.

Taking that point of view, it seems clear that the problem is not initiative, but mixing. Various agents will be involved in working on a problem. Let's imagine that some of them are humans and some are computer programs. Each one gets triggered under certain circumstances, and must make a modification to an evolving plan. For example:

> A scheduling program might be run to produce a timetable for activities and shipments.
>
> A route-planning program might fill in the details of shipment routes.
>
> An inventory planning program might project the availability of raw materials.
>
> A person might prioritize the major activities involved, such as selecting which items on sales orders to manufacture first.
>
> A probabilistic plan checker might look for places where the plan is likely to fail, and introduce "risk-reduction strategies".

We might start out assuming that there is a user who has final authority to accept the plans produced by the system. (There may be grades of users, with different authority.) We could further assume that this user has a reasonably accurate mental model of the capabilities of the machine. For example, he or she might know that a transportation planning program can fill in routes and produce schedules and timetables, but would make a mistake if not told that some critical vehicle was in the shop for repairs.

Given these other assumptions, it also seems reasonable to assume that there will always be a way to issue explicit commands to trigger automatic capabilities. Hopefully, there will be many circumstances where issuing such explicit directives will not be necessary, because the user will find it convenient to let the system to "just do it" without being told. Characterizing the circumstances under which the phrase "just do it" might apply is certainly one way of increasing the "mixing" of the activity performed by the agents in this shared environment, and reducing the burden of the user to directly state every step required to complete his or her task.

This simple-minded notion of initiative is easily sought, and is usually not `AI', but rather the province of all engineers and programmers that seek to automate support functions for users. If, however, the context shared by the user(s) and the system(s) is sufficiently rich, in terms of their models of the task to be performed and the inputs and outputs of the planning process, then better, more sophisticated context-specific "triggers" can be implemented. This seems straightforward, if unglamorous. It also seems clear that one will want it to be easy to retract anything done automatically, if the user decides it isn't right, and, better yet, be able to fix whatever was done, without completely redoing it, to take advantage of whatever was done correctly.

This style of interaction is fairly easy to produce programatically if the size of the task performed by the system is small, or the user acknowledges that there is a standard process to get an acceptable answer, and the system has an encoding of that procedure. It may be equally OK (to the particular user) if the system has a fallible method of doing it, and the user doesn't care how it gets done, either because it is a quick and dirty hypothetical plan, or because it can be fixed later.

Unfortunately, as we stated at the outset, there are several related problems with this model, as it might be applied to present-day AI-based planning and scheduling tools:

- The user is not typically "in control" of current-day automated planning processes, except at the very beginning and the very end, and the product is typically a plan or schedule of substantial size and complexity.
- Questions that these systems ask of users tend to be at the system's discretion, without much attempt at staying "in synch" with the user's way of approaching and solving the problem (or building the plan).
- "Backing up" in a search space is not usually an option that the user has, or, if it is an option, its consequences are not easily understood.

These problems might be manageable if all such modifications satisfied the *top-down refinement* constraint: which can be characterized as the requirement that plan modifications always take a plan from a more abstract to a more instantiated state; or return from an instantiated state to a previous abstract state. If a set of planning modules obeys this constraint, then mixing initiative becomes fairly simple. Each participant sees an abstract plan, to which it can add information. No one ever changes information once it is added, except to return to a previous state, discarding all subsequent changes (and noting that they should not be tried again)[2].

Unfortunately, it is safe to assume that this constraint will never hold in practice, because users generally require the authority and capacity to change any part of the plan they can see. If humans are to be in the loop at all, they just won't tolerate a system that puts them in a straitjacket. This is partially because users are often better at saying what they don't like about a plan than what they do like. It's also partially because it is at present difficult to imagine maintaining the knowledge bases of the AI planning systems such that they have proper representations of every constraint that the human planner is

---

[2] Even when this simple model works, some styles of planning search may be more suitable than others. For example, it is probably better if the subsequent changes were ones that were related to the change that is discarded (i.e., justification-based backtracking is more reasonable here than chronological backtracking).

operating under in a new planning situation. If the planning system doesn't have a full representation of the problem, then "arbitrary" modifications of the solutions produced may be necessary, even when a solution is "right" as far as the system can tell.

An issue, then, for a mixed-initiative planner is to help the user characterize what "don't like it" means, or what to do about it in as useful a way as possible. Responses to a user stating "I don't like X" might include any of:

- Just drop X (and perhaps any goals it supported);
- Reduce the priority of the goals and constraints that caused the introduction of X and try to replan with those altered assumptions;
- Explain why it is critical that X is there and ask if the user still wants to remove it;
- Ask for a reason for removing that element, so as to be able to incorporate the criticism as a rational constraint with a basis.

Hence, a key problem in building mixed-initiative planning systems, especially using as components many of the kinds of problem solvers around today, most of which were originally designed to operate autonomously, is the problem of getting those modules to correctly determine which criticisms by other agents are "boundary conditions", and which are best treated merely as preferences. Consider a couple of examples of where this issue comes up, some of which are discussed in (Smith and Lassila, 1994):

- The user looks at a transportation schedule produced by a program, and notices that a certain high priority package is being shipped by rail rather than by air. The user edits the schedule (graphically or by altering text fields) so that unit now come flies into the nearest airport to its destination. The scheduler is rerun. It must treat the new shipment plan for that package as a constraint on what it produces.
- The user looks at a plan generated by the program, and notices that an airplane is flying almost completely empty, and switches its passengers to another flight to save the trip. It turns out that the automated planner had inserted flight so that the plane would be available for another flight from the destination later that day. Later, the planner is rerun, and has to schedule yet another plane to go pick up the passengers of that later flight. (If the planner is not rerun, the schedule will no longer accurately reflect the plan.) The user may not notice this, but if he or she does notice, he or she will want to know what's going on. It would have been helpful if the system had been able to tell the user as soon as the edit was proposed that the edit is impossible without making other changes.
- Some modules produce only plan assessments, without making any other changes. It might be desirable for the user to edit these assessments, when it thinks that the automated system is overlooking something. After further plan revision, the user will have to re-edit the assessment, because the automated system has no way of knowing how much of the revisionist assessment should be preserved. The user might then fall into a rhythm of blindly upgrading a plan even when it has been revised to a point where it really is bad.

One can imagine lots of examples of this sort. They may not have much in common in detail; that is, there may not be a general theory of intent recognition that will let the person "explain" to the computer what to preserve about the edits made in each case. We should at least hope that the user will be able to receive enough information to validate their own actions.

A more modest research agenda, then, might be one based on the idea of anticipating types of edits to plan structures. One basic principle is that anything a person can see on the screen ought to be editable. Another basic principle is that what's on the screen ought to be what people find natural to think about. At a very early stage in the process, we need to track possible edits of the plan display. There are two sources of information about this. One is just the raw syntactic *possibility* of edits. Whatever can be displayed (location, time, ordering, etc.) can in principle be changeable. Another source of information is what changes the human planners currently make. For each possible change to a plan, we ask, What boundary condition might this change imply? If there is a choice of boundary conditions, a dialogue should take place in which the choices are described to the user in his or her own terms and he or she is asked to clarify which is meant. Any tool that works on a plan must be prepared to obey notes about such boundary conditions.

Non-proscriptive Forms of Initiative are the Most Easily Accepted

It should be a goal of systems built for person-machine mixed-initiative planning tasks to be as helpful as possible to the human user without being counter-productive in the sense of making more work than they provide. One way to maximize this is to do as many of the "little things" that make effective use of the computer's capabilities and resources in ways that do not automatically modify the plans under development. There are some tasks that are naturally non-destructive in this sense, and others that can be made so by putting their products in the forms of "suggestions" that can be considered and either accepted or rejected.

Whether these things are or are not due to "initiative" on the part of the system is likely to be based on the conditions under which they occur. That they are valid, useful contributions made a timely fashion is all users should really care about. That they support more effective distribution of the labor is most important.

Some things in the naturally **non-interfering** category are:

- **Initiation of information retrieval requests** to other agents; these can be triggered by such things as the "named" phases of a planning process, and so require minimal information about the specific planning problem to be triggered. One can think of this kind of activity as analogous to a nurse asking you to fill out a medical questionnaire before being seen by the doctor. In an on-line system, it might mean automatically sending out email requests to other users or "off-line" databases to collect information before beginning a phase of planning.
- **Automatic "highlighting" of information requirements** for which the user is the primary source; Examples include flagging of unexpanded goals which need to be worked on to complete a phase of planning, assumptions under which a plan or plan element will remain valid or achieve its mission, etc. Many kinds of visual cues may be used to identify pending or incomplete tasks, new information which must be taken into account, etc.
- **Promoting the reuse of experience** by automatic retrieval of relevant prior plans or fragments (cases), that are similar to the ones under development, either to provide additional "cut and

paste" opportunities, or to provide examples of potential problems that might come up when executing the plan under consideration.

- **Automatically sending inputs to plan evaluation tools**, including constraint checkers, resource analyzers, simulations. On completion of such evaluations, the user should be notified only of important issues that were uncovered, and even then perhaps only a notice that the results are available to be considered whenever the user is ready to look at them, unless the problems are of sufficient priority.

Some things that can be most useful if done on a **"not to interfere"** basis. The issue with this kind of activity is the potential for harm, either in generating work for agents that is subsequently thrown away, or in causing too many interruptions of the user's activities. Beyond endeavoring to find the most appropriate triggering conditions for these activities, all of the usual ways of hedging by getting confirmation that the action is desirable should be considered. Also, means of canceling these behaviors individually and globally (in the sense that a user never wants that feature) should be provided. This means support for issuing retraction messages for automatically triggered activity requests (which could also be triggered if the goal they served is abandoned). Some example activities in this category are:

- **Eager elaboration of trivial subgoals** when no alternative choices need to be considered.
- **Limited-depth subgoal search** to find a few qualitatively different feasible elaborations to present to the user as options.
- **Automatic notifications to collaborators** that their participation (based on plan content) will be required for some planning task, especially when some advance preparation is required on their part, or scheduling of their time is an issue.
- **Notification that planning assumptions have changed** and the planner must consider redoing part of his plan. This could be either an execution-time issue (the world changed), or a planning time one (the head planner changed a critical decision, or new information arrived).
- **Reconciliation of local and more global objectives**, especially where different human agents are responsible for these different perspectives. This is really a specialized form of plan consistency checking.

Task Decomposition Models Provide Useful Triggers for System Initiative

As the prior discussion suggests, when several agents collaborate in a mixed-initiative fashion on a planning task, each agent must be able to recognize *when* they can perform useful work in service of the overall task objectives, and *when and how* to convey the results of their work to the other cooperating agents. For the team as a whole to succeed, each agent that initiates activity must share *compatible* models (though perhaps implicit ones) of the tasks to be performed, in addition to knowledge of the specific goals to be achieved by the plans they produce.

If we are looking for mixed-initiative systems to do more than systems that perform functions when a command is issued by the user, then we need to find more sophisticated "triggering" criteria, that can be based on any or all of: the user's inputs, the currently represented state of the plan under construction, background data and other potentially "live" information about the world in which the plan is to be executed. This situation is simplified if there are some quickly recognizable cues as to where in the process of plan construction the agent who is "in charge" (typically the user) is focused, so as to avoid the impossible chore of scanning and interpreting all of the available data to discover what tasks are likely to be appropriate ones at any given time. There are now working examples of "plan editing tools" that break complex planning activities down into recognizable phases with different displays appropriate to each phase of the task. These displays provide a crude form of "context registration", in addition to providing specialized task support. In planning tasks where such explicit models of the structure of the task are appropriate, such phase shifts can serve as triggering cues for planning support activities by other agents.

People engaged in complex tasks break them down into stages or phases so that they can organize the work into activities with specific methods of achievement, limit the depth of their own problem solving, simplify the applicability conditions for the subactivities involved, delegate subtasks to others, and in order to *learn* how and when to perform subtasks in the first place. Newell's (1981) "problem spaces" is a formalization of this notion. As different phases of planning may or may not use the same problem solving styles, the representations of the problem and the vocabulary used in solutions may change as one changes phases. As a result, it may be a lot easier to say when one is "done" with a phase of planning, (e.g., that *all subgoals have been expanded* into a set of activities defined in advance to represent a *consistent level of description* in the domain; or that one has *validated the plan* at that level of description by some set of methods such as simulations and constraint consistency checks) than it is to say that one has found the *best* plan in some absolute sense.

For example, the different kinds of planning and evaluation activities that go into the overall military planning process have very different information requirements. They use completely different software tools and have different ways of interpreting and applying results. For example transportation schedulers and simulators serving similar functions require classes of inputs in terms of consumers (e.g., things to move) and resources or producers (e.g., ways to move them). Information about the resources must be of consistent kinds, and matched to the consumers. The consumers in the transportation scheduling phase, the people and equipment to be moved, are the resources in the employment planning phase.

This relationship between phases of planning makes transportation schedulers or simulators useful as resource constraint checkers during employment planning, since the overall plan will not work if one cannot marshal the needed resources. However, when using a transportation scheduling tool as a resource analysis tool, different criteria, in terms of detail and accuracy, are used to gather inputs to the program. Far less detail is needed to apply a scheduler in this second role, which means that it can be used earlier in the planning process if appropriate, approximate data can be supplied. A very useful form of system initiative would be to automatically assist in developing inputs to a scheduler so that such preliminary analyses can take place in a timely fashion. Also, since use of a scheduler in this formative phase of the planning process is for such a different purpose, the interpretation of the result is also radically different. An acceptable result might be as simple as "doesn't use any more resources than allowed", or an indication of a critical resource shortage, instead of a detailed schedule. The system must know this context in order to present the appropriate conclusions to the users or other software agents.

Another reason for working with explicit models of planning tasks is that it may be critical for effective, adaptive learning by the system and by the user. Since effective collaboration means *learning* to act in a coordinated fashion, user's need to have cues to guide their expectations of the each others' capabilities in different contexts, and learning takes place more effectively if the context of the activity being learned or refined is a localized task.

The Road Ahead

We have tried in this chapter to "raise the ante" for research and development of AI-based planning systems. The history of planning in AI comes very much from the tradition of robotics; that is, providing autonomous entities with a capability to move and act in the world. But the worlds in which these planners worked tended not to change much, "fight back" at all, and certainly not actively collaborate. This world view has colored much of the last 30 years of planning research. Our goal was to "open up the box" and reconsider these assumptions in the light of recently raised opportunities for virtually global electronic collaboration, and the new emphasis on cognitive technologies for interactions between people and machines. We hope and expect that many of the research areas we touched upon will become heavily discussed issues in the next few years.

References

Allen, James F. and Ferguson, George (1994) "Arguing About Plans: Plan Representation and Reasoning for Mixed-Initiative Planning," In Burstein, M. (Ed.) *Proceedings of the 1994 ARPA-Rome Laboratory Planning Initiative Workshop* , Morgan Kaufman, San Mateo, CA. pp. 123-132.

Allen, James F. (1994) "A Perspective on Mixed-Initiative Planning" In Burstein, M. (Ed.) *Proceedings of the 1994 ARPA-Rome Laboratory Planning Initiative Workshop* , Morgan Kaufman, San Mateo, CA. pp. 486-495.

Smith, S. F. and Lassila, O., (1994) "Toward the Development of Flexible Mixed-Initiative Scheduling Tools," In Burstein, M. (Ed.) *Proceedings of the 1994 ARPA-Rome Laboratory Planning Initiative Workshop* , Morgan Kaufman, San Mateo, CA. pp. 145-154.

Sycara, K. and Miyashita, K (1994) "Evaluation and Improvement of Schedules According to Interactively Acquired User-defined Criteria" In Burstein, M. (Ed.) *Proceedings of the 1994 ARPA-Rome Laboratory Planning Initiative Workshop* , Morgan Kaufman, San Mateo, CA. pp. 155-164.

Newell. A., (1990) *Unified Theories of Cognition*, Harvard University Press, Cambridge, MA.

Wilkins, D.E., *Practical Planning: Extending the Classical AI Planning Paradigm*. Morgan Kaufmann Publishers, Inc., San Mateo, CA. August 1988.