AI Magazine Volume 20 Number 4 (1999) (© AAAI)

A Survey of Research in Distributed, Continual Planning

Marie E. desJardins, Edmund H. Durfee, Charles L. Ortiz, Jr., and Michael J. Wolverton

■ Complex, real-world domains require rethinking traditional approaches to AI planning. Planning and executing the resulting plans in a dynamic environment implies a continual approach in which planning and execution are interleaved, uncertainty in the current and projected world state is recognized and handled appropriately, and replanning can be performed when the situation changes or planned actions fail. Furthermore, complex planning and execution problems may require multiple computational agents and human planners to collaborate on a solution. In this article, we describe a new paradigm for planning in complex, dynamic environments, which we term distributed, continual planning (DCP). We argue that developing DCP systems will be necessary for planning applications to be successful in these environments. We give a historical overview of research leading to the current state of the art in DCP and describe research in distributed and continual planning.

The increasing emphasis on real-world applications has led AI planning researchers to develop algorithms and systems that more closely match realistic planning environments, in which planning activity is often distributed, and plan generation can happen concurrently with plan execution. These new research directions have increased the need for distributed systems of cooperating agents for continuous planning and decision support. Multiagent planning architectures, distributed planning, mixed-initiative planning, distributed scheduling, and work-flow management methods are currently active areas of research.

We argue that a new paradigm is needed to support this style of planning and that such a paradigm is beginning to emerge within the planning research community. We characterize this new approach to planning as distributed, continual planning (DCP). *Distributed planning* refers to an environment in which planning activity is distributed across multiple agents, processes, or sites. *Continual planning* refers to an ongoing, dynamic process in which planning and execution are interleaved.[1]

An agent should engage in distributed planning when planning knowledge or responsibility is distributed among agents or when the execution capabilities that must be employed to successfully achieve objectives are inherently distributed. An agent should plan continually when aspects of the world can change dynamically beyond the control of the agent, aspects of the world are revealed incrementally, time pressures require execution to begin before a complete plan can be generated, or objectives can evolve over time. Thus, DCP is important for an agent whose success and efficiency depends on how its current actions affect its future choices but that must operate in a complex, dynamic, and multiagent environment.

We first present a historical overview of planning research that emphasizes how the introduction of more realism into the problems faced by planning systems has led to current approaches that fit the DCP paradigm more closely. We then survey related research in distributed and continual planning. We conclude by outlining some important future directions that are needed for true DCP to become a reality.

Distributed, Continual Planning: A Historical Perspective

In this section, we present a framework for categorizing planning and execution systems that centers on how each system handles the context in which its planning and execution are carried out. We place a range of systems within this framework and discuss where continual planning, distributed planning, and distributed, continual planning arise.

The context-centered framework gives us a structure that we use to provide a historical perspective on research in this area. We divide this research into overlapping stages that are characterized by increasingly sophisticated treatment of planning and execution context. The earliest stage consists of systems that ignore context to simplify the planning problem. In the next stage, we begin to see systems that tolerate context by responding to or anticipating events that occur outside the planning and execution system. The third stage includes systems that exploit context, by taking advantage of their ability to control the world and interact with other agents. Finally, the most sophisticated systems take the initiative to establish context by actively interacting with and configuring the environment.

The context-centered framework gives us a structure that we use to provide a historical perspective on research in this area. We divide this research into overlapping stages that are characterized by increasingly sophisticated treatment of planning and execution context.

The characterizations we provide are a simplification of the issues and accomplishments being addressed in each of the stages. In this section, we refer to survey treatments and illustrative examples of work rather than provide a comprehensive bibliography of the rich planning literature. Our purpose is to illustrate the changing perspectives in the general planning community that have led to the current interest in DCP.

Ignore Planning and Execution Context

Planning is difficult. Even the simplest planning problem, that of determining how an agent can move from the current world state to a world state that satisfies its preferences, is intractable in the general case. For this reason, traditional AI planning research (Russell and Norvig 1995; Hendler, Tate, and Drummond 1990) has introduced assumptions and simplifications to make planning feasible. An agent is typically assumed to know everything that is relevant to the planning problem and to know exactly how its available actions can change the world from one state into another. The planning agent is assumed to be in control of the world, so that the only changes to the state are the result of the agent's deliberate actions. The agent's preferred world states are also constant throughout a planning episode (planning problem or set of objectives)—it will not "change its mind" about what goals to achieve in the midst of planning or while executing the plan.

These simplifications allow the planning problem to be serialized: A planning agent first formulates a plan and then executes it. It is assumed that the planning and execution for one episode have no bearing on the planning and execution done for previous or future episodes. The context in which planning and execution are done is effectively ignored during the planning and execution processes.

Tolerate Planning and Execution Context

In the real world, plans do not always proceed as expected. In dynamic, uncertain environments, an agent cannot in general make accurate predictions about the outcomes of its actions. One approach to handling uncertainty is to enumerate the possible states (contingencies) that might arise at execution time and plan for each of them, constructing a (possibly large) *conditional plan* that provides alternative courses of action for each contingency (Boutilier, Dean, and Hanks 1999; Schoppers 1987).

If the knowledge available to the agent is insufficient or suggests an intractably large set of contingencies, a better approach is *plan monitoring and repair* (Ambros-Ingerson and Steel 1988; Doyle, Atkinson, and Doshi 1986). In this approach, the agent formulates a nominal plan that looks best given what (little) the agent knows. As the plan is executed, progress is monitored to ensure that the predicted or assumed conditions actually hold. When deviations are detected, the agent halts execution and revisits its planning decisions, creating a revised plan. In effect, the episodic nature of "plan, then execute" is maintained, but episodes can be terminated prematurely (when a deviation is detected), and results from one episode can influence another (such as when the previous plan is repaired and reused). If no deviations occur, this approach is equivalent to traditional planning.

In complex environments, however, the planning and execution context might change in ways that suggest a change in future plans without necessarily violating a currently executing plan. Unexpected changes in the world might provide opportunities to accomplish goals more efficiently or effectively. Alternatively, an agent's goals might themselves change, so that although the current plan could still be carried out, the motivation for doing so is lost (Cohen and Levesque 1990). When goals and aspects of the world can evolve continuously rather than be fixed throughout a planning episode, an agent should continually evaluate and revise its plans.

Continual planning recognizes that plan revision should be an ongoing process rather than one that is triggered only by failure of current plans. It also adopts the perspective of not planning too much detail too far into the future because evolving circumstances can render such details obsolete. Continual planning, therefore, tolerates the planning and execution context by maintaining flexibility and opportunism.

Exploit Planning and Execution Context

The approach to continual planning that we have outlined suggests an environment in which an agent is working toward accomplishing its objectives, despite the world dynamically changing in unpredictable ways that interfere with its plans. Such a pessimistic mind set need not be adopted, however, if the agent has knowledge, and the ability to use it, about its relationship to its environment and the other agents in the world. In this case, an agent should not simply tolerate the presence of other agents but should seek to exploit their presence by cooperating with them.

One particularly useful source of interaction is human participants in the planning process. Humans often have expertise and capabilities that computational agents lack. Rather than formulating weak plans based on its limited expertise and knowledge, therefore, an agent should, whenever possible, work with people to formulate more efficient and robust plans. A planning process that is distributed between computational and human agents in this way is referred to as *mixed-initiative planning* (Ferguson, Allen, and Miller 1996; Myers 1996; Veloso 1996).

A planning agent could also work with other computational agents. By bringing to bear complementary capabilities, a set of agents can collectively formulate a concerted plan of action for one or more of them to carry out (Wilkins and Myers 1998; Kambhampati et al. 1993). Similarly, by considering the capabilities of other agents, agents can accomplish their objectives by exploiting each other's strengths through cooperative execution.

Even if the agents do not explicitly cooperate by sharing expertise or capabilities, by exploiting the fact that changes to the world are often a result of other agents' planned actions, an agent can reduce its uncertainty, and increase the quality of its local plan, by influencing the plans that others adopt.

Establish Planning and Execution Context

The approaches to cooperative and negotiated distributed planning that were alluded to in the previous subsection are essentially distributed versions of traditional plan-then-act approaches. For a system of agents to engage in distributed, continual planning, however, each must be a continual planning agent, and more. As each agent elaborates its ongoing abstract plans into actions, the elaboration decisions should be compatible and preferably mutually supporting. The agents' distributed planning mechanisms should continually evaluate the relationships between refinement decisions and redirect decisions in cooperative (or at least compatible) ways.

The idea of allowing agents to exploit the larger multiagent context for planning (working together to plan) and execution (taking actions toward common goals) opens the door to purposely establishing such a context to improve what agents can accomplish (Stone and Veloso 1999; Tambe 1997; Shoham and Tennenholtz 1992). Agents that have formulated abstract plans can analyze potential relationships between their possible plans, commit to particular constraints on how they will realize these plans, and then incorporate these influences in their elaboration decisions in a decentralized way (Clement and Durfee 1999).

The approach to continual planning that we have outlined suggests an environment in which an agent is working toward accomplishing its objectives, despite the world dynamically changing in unpredictable ways that interfere with its plans.

Distributed Planning

The problem of constructing plans in a distributed environment has been approached from two different directions: One approach has begun with a focus on planning and how it can be extended into a distributed environment, where the process of formulating or executing a plan could involve actions and interactions of a number of participants. The other approach has begun with an emphasis on the problem of controlling and coordinating the actions of multiple agents in a shared environment and has adopted planning representations and algorithms as a means to an end. Although these two approaches have led to common ground in distributed planning, and indeed many researchers embrace both approaches (making unequivocal partitioning into one camp or the other impossible), we believe that the distinctions still help in understanding the state of the field.

We refer to the first approach as *cooperative distributed planning* (CDP). Because it places the problem of forming a competent (sometimes even optimal) plan as the ultimate objective, CDP is typically carried out by agents that have been endowed with shared objectives and representations by a single designer or team of designers. Although in some cases the purpose of the agents is to form a central plan, more generally the purpose is that the distributed parts of the developing plan will jointly execute in a coherent and effective manner. Thus, in CDP, agents typically exchange information about their plans, which they iteratively refine and revise until they fit together well.

The second approach, which we refer to as *negotiated distributed planning* (NDP) has a different focus. From the perspective of an individual agent engaged in NDP, the purpose of negotiating over planned activities is not to form good collective plans but to ensure that the agent's local objectives will be met by its plan when viewed in a global context. The emphasis is therefore not on cooperatively defining and searching the space of joint plans to find the best group plan, as in CDP, but on having an agent provide enough information to others to convince them to accommodate its preferences.

When individual preferences are aligned in ways where cooperation is in the self-interest of all concerned, CDP and NDP come together; thus, a hard-and-fast partitioning between the two approaches is impossible, and a number of efforts embrace both perspectives. For example, STEAM, which we discuss in Negotiated Distributed Planning, draws heavily on planning research; on the other hand, both DIPART and partial global planning, which are described in Cooperative Distributed Planning, have a strong agent flavor. Our purpose in making the loose division between CDP and NDP is not to pigeonhole any individual piece of research but to provide some structure to the research landscape, ranging from extreme CDP (where the purpose of distribution is simply to allow parallel computation of plans) to extreme NDP (where the purpose of planning is simply to find resource conflicts) with a rich space of systems in the middle, involving both sophisticated joint planning and the pursuit of self-interest on the parts of agents.

The problem of constructing plans in a distributed environment has been approached from two different directions ….

Cooperative Distributed Planning

In this subsection, we summarize past and current work in CDP. Most of the research we describe here involves two or more computer processes (agents) cooperating to build either a single plan or multiple interacting plans. However, we also mention other related research, for example, a single planner that has special capabilities for multiagent execution.

In building a distributed planning system, some of the key questions to address include (1) How are plans or partial plans represented? (2) What is the basic plan-generation method? (3) How are tasks allocated to agents? (4) How do agents communicate with one another during planning? (5) How are the agents' actions coordinated? We discuss alternative approaches to these problems in the following subsections.

Plan Representation and Generation

Much of the research in CDP is built around notions of abstract plan decomposition, such as hierarchical task network (HTN) planning (Erol, Nau, and Hendler 1994). An *abstraction-based plan representation* allows a distributed planning agent to successively refine its planning decisions as it learns more about other agents' plans.

Corkill's (1979) distributed version of Sacerdoti's (1977) NOAH planner was one of the earliest efforts in distributing an HTN planning algorithm. The basic planning procedure and plan representation used by Corkill are the same as those of NOAH: Planning proceeds through a hierarchy of plan levels, where at any plan level, a partial plan is a partial order of goals and primitive actions. Each distributed agent solves its goals in the same way as NOAH, at each level expanding each unplanned goal by finding an applicable operator (called a SOUP procedure) that solves it. NOAH's plan representation is extended to include a special node type representing a placeholder for another agent's plan as well as special primitive actions for synchronizing multiagent execution.

Just as the SIPE-2 planner (Wilkins 1988) is conceptually descended from NOAH, DSIPE (distributed SIPE-2) (see the desJardins and Wolverton article, also in this issue) is conceptually descended from Corkill's distributed NOAH. Both are distributed versions of HTN planners, and the two systems use similar approaches to some of the key representational problems in distributed planning, especially in how the multiple agents maintain a consistent (albeit incomplete) picture of each other's plans. In DSIPE, the local view of the other agents' subplans is called a *skeletal plan*.

Similarly, in Durfee and Lesser's (1991) *partial global planning* (PGP), each agent maintains a partial global plan, which stores its own partial picture of the plans of all the members of the group. PGP focuses on distributed execution and run-time planning and uses a specialized plan representation, where a single agent's plan includes a set of objectives, a long-term strategy (ordered lists of planned actions) for achieving these objectives, a set of short-term details (primitive problem-solving operations), and a number of predictions and ratings. Planning—both the expansion of objectives into planned actions and the ordering of these actions—is based on a set of heuristics. Partial global planning (PGP) and its descendants, including generic PGP (Decker and Lesser 1992), have been applied to problems that include distributed acoustic monitoring (Durfee and

Lesser 1991), cooperative information gathering (Decker et al. 1995), and cooperative robotics (Clement and Durfee 1999; Durfee and Montgomery 1991).

Task Allocation In cooperative planning systems, typically an explicit, often centralized, mechanism assigns subgoals to the agents that will be responsible for planning them.

Distributed NOAH and DSIPE have no automated methods for allocating tasks among the planning agents; the user specifies which agent solves which top-level goals. PGP allows agents to send potential group plans to each other, essentially allowing agents to propose (and counterpropose) an assignment of activities (tasks) to each other (Durfee and Lesser 1989). Because agents can choose to participate or not in such plans, such assignments involve mutual selection between agents' contract nets.

DIPART (Pollack 1996) is an experimental platform for analyzing a wide range of planning and execution mechanisms, including approaches to distributed planning. One of the key areas studied in DIPART is *load balancing* (that is, task allocation) among multiple planning agents. The load-balancing methods developed are high-level approaches that are independent of any particular planning representation or algorithm. Initial allocation of tasks to agents is specified by the user, but tasks can automatically be reallocated at run time according to load-balancing considerations.

Lansky's COLLAGE (Lansky and Getoor 1995; Lansky 1994) is not a multiagent planning system as such, but it represents one promising way of approaching the task-assignment problem in distributed planning. COLLAGE uses a technique called *localization* to decompose a planning problem into subproblems called *regions*. Localizations in COLLAGE can be generated automatically based on abstraction levels or scope. A set of heuristics are used to find regions that minimize the number of interactions between regions, thus permitting COLLAGE to solve the subproblems in each region (relatively) independently. Partitioning subgoals among distributed planning agents according to localization would likewise improve the overall efficiency of the multiagent planning system by minimizing conflicts and interactions among agents (and reducing the need to check for them).

Communication Cooperative distributed planners use a variety of techniques for agents to share information. At the conclusion of each planning level, distributed NOAH applies an extended version of NOAH's planning critics to coordinate the planning agents. Each agent sends the other agents a record of the important effects of the newly expanded portions of its plan (although Corkill does not make it clear which effects are considered to be important). Each agent inserts a record of all new effects it receives into its local plan.

DSIPE extends distributed NOAH's approach in a number of areas, most notably by expanding the types of planning information that are shared among the planning agents and developing irrelevance-based filtering methods for reducing communication requirements (Wolverton and desJardins 1998).

In PGP, communication goes through the channels of a metalevel organization, which specifies the roles and responsibilities of planning agents in maintaining plan coordination (for example, which agents have authority to reconcile conflicts among agents' plans), much like a regular organization specifies how the agents should be organized to solve a domain-level problem. The work of Jennings (1995) codifies such responsibilities into communication conventions that all agents follow to ensure sufficient awareness about the pursuit of collective plans.

Communication among planning agents to support load balancing in DIPART is generally done by multicasting to all members of the agent's (user-assigned) subgroup. However, in heavy load situations, agents can trade off complete knowledge of system load in exchange for preservation of bandwidth by selective unicasting (called *focused addressing* in contract nets). In this case, they must rely on estimates of other nodes' loads based on their histories.

Coordination For a group of cooperative distributed planners to reach a state in which the individual subplans jointly achieve the common objectives, the agents must coordinate their decision making.

In distributed NOAH, when an agent's critics detect a conflict between one of its own actions and those of another agent, instead of adding an ordering constraint as it would if were not distributed, it inserts a special execution-time coordination action: either *Signal*, which informs another agent of the completion of an action, or *Wait*, which waits for a signal from another agent.

PGP achieves coordination and task assignment by a process of *negotiation*. When agent *A* informs another agent about some portion of its local plan, *B* merges the new information about *A* into its partial global plan. It then searches for ways to improve the global plan, for example, by eliminating redundancy or better utilizing the group's computational resources. It proposes these im-

One approach has begun with a focus on planning and how it can be extended into a distributed environment ….

The other approach has begun with an emphasis on the problem of controlling and coordinating the actions of multiple agents in a shared environment ….

provements to some subset of the other agents. Each other agent can accept the new global plan and incorporate the changes made to its own local plan, it can reject the new global plan, or it can respond with a counterproposal. Throughout the process, each agent continues to act on its current best assessment of the plan it should pursue rather than wait to converge on a plan (which, because of dynamics, might be quite ephemeral anyway). Thus, this approach is designed for domains where temporary incoherence among agents is tolerated as information about unanticipated changes to plans is propagated. It is not designed for domains with irreversible actions, where conflicts among agents' plans should be resolved before any step is taken that could cause other agents' plans (and, thus, the system as a whole) to fail.

Coordination between planning agents in DIPART is accomplished in two ways: (1) incremental merging of individual agent subplans (Ephrati, Pollack, and Rosenschein 1995) and (2) *multiagent filtering* (Ephrati, Pollack, and Ur 1995), in which each agent commits to the goals it has already adopted but bypasses new options that are incompatible with any other agent's known or presumed goals.

An agent working in a world where unexpected hindrances or opportunities could arise should continually be on the lookout for changes that could render its planned activities obsolete. Because its plans can undergo continual evaluation and revision, such an agent will continually be planning, interleaving planning with execution.

Negotiated Distributed Planning

Much of the research into the design of autonomous agents has had as its goal a formal account of mental state that would serve to explain the manner in which the attitudes of belief, desire, and intention (BDI) engender behavior, both at the individual agent level and at the group level. A *BDI theory* could be viewed as explanatory to the extent that one felt comfortable in ascribing some sense of rationality to the behaviors predicted by the theory that were causally connected to individual mental states; in this respect, the term *rational balance,* as applied to an agent's mental state and as coined by Nils Nilsson, is an apt one. For example, an intention can be viewed as reflecting an agent's commitment to perform an action; such a commitment is assumed to remain in force until it is satisfied, or the agent no longer believes the action is possible (Cohen and Levesque 1990; Shoham 1990; Bratman 1987). By virtue of its persistence, an intention will thereby constrain the options an agent will be disposed to consider; a *rationally balanced agent* would then be one that was not continuously reevaluating each possible option in light of new intentions or goals. Within such models, plans are derivative structures built up from elements of the agent's mental state (Pollack 1990).

The BDI decomposition strategy can also serve as an implementation-neutral way to architect an agent and also analyze both individual and collective behavior. In contrast to the previous research, which has largely been formal in nature, the design of rational-agent architectures (Haddadi and Sundermeyer 1996; Bratman, Israel, and Pollack 1988) has been guided primarily by computational concerns centered on an agent's inherent resource limitations (Russell and Subramanian 1995; Simon 1983). Agent architectures augment an agent's mental world with a set of processes, each associated with a particular architectural module; processes can either supply input for other modules or modify the agent's repository of beliefs, desires, and intentions. As an example, an architecture might include a deliberation module whose role is to weigh alternatives in light of existing commitments (intentions) and new observations (beliefs). A connection between formal theories of mental state and agent architectures (or implementations) can be drawn by reifying mental actions, each representing the behavior of an architectural module (Ortiz 1999).

Collaboration Theories of mental state for individual agents must be extended in important ways to model collaborations between agents. The problem is that a group's plan to collaborate on some task is not simply the sum of the agents' individual plans (Grosz and Kraus 1996; Bratman 1992; Searle 1990). In particular, some notion of an individual's commitment to (parts of) the group activity must be captured. The idea of a joint intention can be introduced to represent the group's commitment toward some group goal (Sonenberg 1992; Levesque, Cohen, and Nunes 1990). One can also augment an agent architecture to include processes that capture behavioral conventions for modifying commitments and supporting coordinated behavior (Jennings 1995). One problem with the joint-intentions model, however, is that communication points must be built in and cannot be derived automatically.

The theory of SHAREDPLANS is an alternative approach that builds group plans from ordinary beliefs and intentions, together with a new attitude called an *intention-that* (Grosz and Kraus 1996; see the Grosz, Hunsberger, and Kraus article, also in this issue). Intentions-that model commitments to states of the world instead of actions. Such intentions can represent, among other things, an individual agent's commitments to aspects of the world that will ensure the success of the group activity; various stages of partiality of a shared plan can thereby be modeled (corresponding to, for example, partial beliefs about how to perform an action, the identity of appropriate agents for a group task, and the knowledge preconditions for an action). Theories based on this idea grew from work in natural language understanding (Grosz and Sidner 1986) and have been applied to discourse understanding and the development of collaborative user interfaces (Lochbaum 1998). Unlike the irreducible joint-intentions model, communications are dynamically triggered as a consequence of failed intentions-that.

Some systems have been developed that combine both approaches; for example, although the STEAM system (Tambe 1997) is based on the joint-intentions model, it also borrows some ideas from SHAREDPLANS for representing partial plans and integrates ideas about teamwork capabilities into the design of individual agent architectures.

Decision theory provides a computational framework for making decisions about coordination and communication in a distributed planning context. Both STEAM and the recursive modeling method (RMM) approach (Gmytrasiewicz, Durfee, and Wehe 1991) have used models of the costs and benefits of communication and applied decision-theoretic methods to determine what and when to communicate. Boutilier (see the article, also in this volume) describes approaches for integrating decision-theoretic planning and multiagent systems.

Negotiation Given a representation for group plans, the question remains of how agents should negotiate to reach an agreement over distribution of tasks and resources. One way to ensure coordination of agent activities is to embed them in environments in which certain social laws must be adhered to (such as the traffic laws in Shoham and Tennenholz [1992]). This approach builds coordination into the environment by focusing on public behavior rather than on private preferences. Other approaches make use of ideas from game theory, such as notions of solution stability and equilibrium, and assume that agents are developed independently and are self-interested; that is, they maximize individual utility. Because agents are not necessarily assumed to be cooperative, much of the effort is directed toward developing protocols that prevent lying. One attractive strategy is to characterize different domains and identify appropriate protocols for each type of domain (Rosenschein and Zlotkin 1994).

Other ideas that have been adapted from the field of economics include *negotiation strategies* that take time into account, explicitly handling the resource-constrained nature of agents (Kraus, Wilkenfield, and Zlotkin 1992; Osborne and Rubinstein 1990). Agents are assumed to have a common metagoal to minimize the amount of work that each agent must do. In the case of agents who do not have a common, fixed goal during negotiation, an alternative is for agents to construct arguments that they can use to influence the decision making of other agents (Sycara 1990).

Voting schemes, also borrowed from the field of economics, have been applied to achieving consensus. For example, in one multiagent scheduling system, agents first express their individual preferences, and then a voting mechanism is used to determine group choice (Sen, Hayes, and Arora 1997). When the assumption of cooperation is dropped, systems can become susceptible to lying. Many approaches around this problem have been proposed: One, for example, requires that agents pay a tax (such as the CLARKE tax [Ephrati and Rosenschein 1991]) that, in effect, transfers utility outside the system. Agents are still assumed to be self-motivated and maximize their own utility, but it can be shown that there is only one dominant strategy: telling the truth.

Contract nets (Davis and Smith 1983) represent a market-based approach to negotiation. Contract nets provide a high-level communication protocol in which tasks are distributed among nodes in a system. Nodes are classified as either contractor or manager nodes, and contracts to perform tasks are established through a bidding process. A similar approach is the *consumers and producers approach* (Wellman 1992) in which agents are classified as either consumers or producers, and each type of agent tries to maximize its own utility. Goods have auctions associated with them, and agents can acquire goods by submitting bids in the auction for the commodity. A more dialogue-oriented approach is found in the collaborative negotiation system based on argumentation (CONSA) (see the Tambe and Jung article, also in this issue), which uses negotiation methods based on argumentation structures (Toulmin 1958) to resolve conflicts.

A major challenge that remains is the development of models that encompass aspects of the partiality of both mental state and the planning process itself. Methods must be developed for adapting the various approaches in a way that is consistent with the resource-constrained nature of planning agents: Planning should be a continuous, incremental process at both the individual and group levels.

Continual Planning

An agent working in a world where unexpected hindrances or opportunities could arise should continually be on the lookout for changes that could render its planned activities obsolete. Because its plans can undergo continual evaluation and revision, such an agent will continually be planning, interleaving planning with execution.

Reactive planning systems (Agre and Chapman 1987) can be viewed as a special case of continual planning in which planning looks ahead only to the next action. Because of their near-term focus, such systems generally do not handle problems that require complex orderings of actions to accomplish tasks. The assumption in these systems is that such orderings should not be pursued anyway because the later actions are inappropriate by the time they are reached.

Flexible plan-execution systems such as PRS (Georgeff and Lansky 1986) and RAPS (Firby 1987) strike a compromise between looking ahead to sequences of actions and avoiding commitments to specific future actions by exploiting hierarchical plan spaces. Rather than refine abstract plan operators into a detailed end-to-end plan, these systems interleave refinement with execution. Plan refinement is delayed as long as possible, so that detailed decisions are made with as much information as possible. However, this delay can mean that refinement decisions at abstract levels are made, and acted on, before all the detailed refinements have been made. If these abstract refinements introduce unresolvable conflicts at lower levels, the plan can fail in the middle of execution. It is therefore critical that the specifications of abstract plan operators be rich enough to summarize all (or at least most) of the relevant refinements to anticipate and avoid such conflicts (Clement and Durfee 1999).

The continuous planning and execution framework (CPEF) (see the Myers article, also in this issue) continually constructs and revises plans that evolve in response to a dynamically changing environment. CPEF integrates HTN planning techniques, plan execution and monitoring, and dynamic plan-repair methods. Users can interact closely with the system through the advisable planner (AP). The execution component can execute plans at multiple levels of abstraction, permitting an open-ended planning process in which the level of plan detail can vary depending on the timing and nature of the particular planning task being considered.

The costs of execution monitoring and opportunistic replanning must be considered by continual planning systems. In many respects, execution monitoring is simpler: As an agent steps through its planned actions, it periodically checks to make sure that the state of the world that it has reached is consistent with the state that was predicted by the plan. The challenge is to keep the costs low enough so that monitoring does not consume too much time and effort on the part of the agent, either by devising efficient perceptual strategies (Musliner, Durfee, and Shin 1995; Doyle, Atkinson, and Doshi 1986) or by performing the checks less frequently. Looking for new opportunities is more problematic: Unlike simple monitoring, where the agent knows what to look for (which conditions are necessary for the rest of the plan to succeed), finding opportunities is generally a more open-ended problem. Researchers have explored trade-offs between having bold agents that seldom reconsider their plans and cautious agents that frequently reevaluate their plans (Kinny and Georgeff 1991).

Different continual planning approaches implement ongoing monitoring and replanning tasks differently. For example, UM-PRS (Lee et al. 1994) maintains the hierarchy of plans currently in progress and, before each action, traverses this hierarchy from the top down to ensure that the motivations for the next action are still in force. If during this process it encounters a plan abstraction that is now deemed inappropriate, it considers alternative ways of achieving the goal it had been working on and looks for opportunities for directing its attention elsewhere. In contrast, the SOAR architecture (Laird, Newell, and Rosenbloom 1987) can also perform continual planning but does not explicitly reason over a hierarchy of intentions. Instead, it has been extended to generate rules that explicitly monitor the current plan's context (Wray and Laird 1998). Because these rules are woven into SOAR's rule network, they do not require explicit periodic polling, instead triggering interrupts to SOAR when appropriate.

Future Research Directions

Although distributed planning has been explored by a few researchers for many years, the recognition of its importance in current and future applications involving networked agents has triggered a more broad-based, concerted investigation into this still maturing field.

Reasoning and negotiation techniques are needed that permit distributed planning agents to understand the aspects of the distributed plan that are relevant for their decisions, identify and pursue opportunities for coordination, asynchronously refine their plans as other agents' plans are evolving, and modify their plans in response to information that arrives from other agents.

Open-ended planning, in which plans can continually be refined to varying levels of abstraction as the planning horizon is extended, will be a key capability for continual planning. Plan-repair methods in current systems are typically at either the reactive level or the generative level; methods for smoothly integrating plan-repair techniques at multiple levels of abstraction and varying time scales are needed.

Several articles in this issue describe ongoing research efforts that are focused on true DCP, that is, methods for distributed and continual planning in dynamic, uncertain domains. Pollack and Harty describe a plan-management system that goes beyond simply interleaving decisions about planning and execution by incorporating techniques for monitoring the environment, assessing alternative plans, iteratively elaborating abstract partial plans, controlling the metalevel planning process, and coordinating distributed planning agents. Durfee describes methods for continual, cooperative elaboration and revision of distributed partial plans in a realistic application domain of semiautonomous ground vehicles. Myers presents a framework for continuous planning and execution that is partially distributed (that is, the planning and execution capabilities are distributed, although the planning process itself is not).

However, most of this research is still in its early stages, and many research challenges remain. Both distributed planning and continual planning need to be better understood for DCP systems to be built, yet simply combining distributed and continual planning methods that have been independently developed might not be sufficient. The synchronization problem in distributed planning becomes even greater when the distributed agents' plans are being executed concurrently, and better models of the overall planning and execution process are needed. Similarly, execution of plans is complicated by the presence of other agents, requiring not only methods for distributed plan execution but distributed plan repair.

Note

1. The term *continuous planning* is also used to describe this style of planning, but

because this term is also sometimes used to refer to planning for continuous problem spaces (for example, using continuous models of time and space), we use the less ambiguous term *continual planning.*

References

Agre, P. E., and Chapman, D. 1987. PENGI: An Implementation of a Theory of Activity. In Proceedings of the Tenth International Joint Conference on Artificial Intelligence, 268–272. Menlo Park, Calif.: International Joint Conferences on Artificial Intelligence.

Ambros-Ingerson, J. A., and Steel, S. 1988. Integrating Planning, Execution, and Monitoring. In Proceedings of the Seventh National Conference on Artificial Intelligence, 81–88. Menlo Park, Calif.: American Association for Artificial Intelligence.

Boutilier, C.; Dean, T.; and Hanks, S. 1999. Decision-Theoretic Planning: Structural Assumptions and Computational Leverage. *Journal of Artificial Intelligence Research* 11:1–94.

Bratman, M. 1987. *Intentions, Plans, and Practical Reason.* Cambridge, Mass.: Harvard University Press.

Bratman, M. E. 1992. Shared Cooperative Activity. *The Philosophical Review* 101(2): 327–341.

Bratman, M. E.; Israel, D. J.; and Pollack, M. E. 1988. Plans and Resource-Bounded Practical Reasoning. *Computational Intelligence* 4(4): 349–355.

Clement, B. J., and Durfee, E. H. 1999. Top-Down Search for Coordinating the Hierarchical Plans of Multiple Agents. In Proceedings of the Third International Conference on Autonomous Agents, 252–299. New York: Association of Computing Machinery.

Cohen, P., and Levesque, H. 1990. Intention Is Choice with Commitment. *Artificial Intelligence* 42(2–3): 213–261.

Corkill, D. D. 1979. Hierarchical Planning in a Distributed Environment. In Proceedings of the Sixth International Joint Conference on Artificial Intelligence, 168–175. Menlo Park, Calif.: International Joint Conferences on Artificial Intelligence.

Davis, R., and Smith, R. 1983. Negotiation as a Metaphor for Distributed Problem Solving. *Artificial Intelligence* 20(1): 63–109.

Decker, K. S., and Lesser, V. R. 1992. Generalizing the Partial Global Planning Algorithm. *International Journal of Intelligent and Cooperative Information Systems* 1(2): 319–346.

Decker, K.; Lesser, V.; Prasad, M. N.; and Wagner, T. 1995. MACRON: An Architecture for Multiagent Cooperative Information Gathering. In Proceedings of the Conference on Information and Knowledge Management (CIKM) Workshop on Intelligent Information Agents. New York: Association of Computing Machinery.

Doyle, R. J.; Atkinson, D. J.; and Doshi, R. S. 1986. Generating Perception Requests and Expectations to Verify the Execution of Plans. In Proceedings of the Fifth National Conference on Artificial Intelligence, 81–88. Menlo Park, Calif.: American Association for Artificial Intelligence.

Durfee, E. H., and Lesser, V. R. 1991. Partial Global Planning: A Coordination Framework for Distributed Hypothesis Formation. *IEEE Transactions on Systems, Man, and Cybernetics* 1(1): 63–83.

Durfee, E. H., and Lesser, V. R. 1989. Negotiating Task Decomposition and Allocation Using Partial Global Planning. In *Distributed Artificial Intelligence, Volume 2,* eds. M. Huhns and L. Gasser. San Francisco, Calif.: Morgan Kaufmann.

Durfee, E. H., and Montgomery, T. A. 1991. Coordination as Distributed Search in a Hierarchical Behavior Space. *IEEE Transactions on Systems, Man, and Cybernetics* 21(6): 1363–1378.

Ephrati, E., and Rosenschein, J. 1991. The CLARKE Tax as a Consensus Mechanism among Automated Agents. In Proceedings of the Ninth National Conference on Artificial Intelligence, 173–178. Menlo Park, Calif.: American Association for Artificial Intelligence.

Ephrati, E.; Pollack, M. E.; and Rosenschein, J. S. 1995. A Tractable Heuristic That Maximizes Global Utility through Local Plan Combination. In Proceedings of the First International Conference on Multiagent Systems, 94–101. Menlo Park, Calif.: American Association for Artificial Intelligence.

Ephrati, E.; Pollack, M. E.; and Ur, S. 1995. Deriving Multiagent Coordination through Filtering Strategies. In Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence, 679–685. Menlo Park, Calif.: International Joint Conferences on Artificial Intelligence.

Erol, K.; Nau, D.; and Hendler, J. 1994. HTN Planning: Complexity and Expressivity. In Proceedings of the Twelfth National Conference on Artificial Intelligence, 1123–1128. Menlo Park, Calif.: American Association for Artificial Intelligence.

Ferguson, G.; Allen, J.; and Miller, B. 1996. TRAINS-95: Toward a Mixed Initiative Planning Assistant. In Proceedings of the Third International Conference on Artificial Intelligence Planning Systems, 70–77. Menlo Park, Calif.: American Association for Artificial Intelligence.

Firby, R. J. 1987. An Investigation into Reactive Planning in Complex Domains. In Proceedings of the Sixth National Conference on Artificial Intelligence, 202–206. Menlo Park, Calif.: American Association for Artificial Intelligence.

Georgeff, M. P., and Lansky, A. L. 1986. Procedural Knowledge. *Proceedings of the IEEE* 74: 1383–1398.

Gmytrasiewicz, P. J.; Durfee, E. H.; and Wehe, D. K. 1991. The Utility of Communication in Coordinating Intelligent Agents. In Proceedings of the Ninth National Conference on Artificial Intelligence, 166–172. Menlo Park, Calif.: American Association for Artificial Intelligence.

Grosz, B. J., and Kraus, S. 1996. Collaborative Plans for Complex Group Action. *Artificial Intelligence* 86(1): 269–357.

Grosz, B. J., and Sidner, C. 1986. Attention, Intentions, and the Structure of Discourse. *Computational Linguistics* 12(3): 175–204.

Haddadi, A., and Sundermeyer, K. 1996. Belief-Desire-Intention Agent Architectures. In *Foundations of Distributed Artificial Intelligence,* eds. G. O'Hare and N. Jennings, 169–186. New York: Wiley.

Hendler, J.; Tate, A.; and Drummond, M. 1990. AI Planning: Systems and Techniques. *AI Magazine* 11(2): 61–77.

Jennings, N. 1995. Controlling Cooperative Problems in Industrial Multiagent Systems Using Joint Intentions. *Artificial Intelligence* 75(2): 195–240.

Kambhampati, S.; Cutkosky, M.; Tenenbaum, J.; and Lee, S. 1993. Integrating General-Purpose Planners and Specialized Reasoners: Case Study of a Hybrid Planning Architecture. *IEEE Transactions on Systems, Man, and Cybernetics* 23(6): 1503–1518.

Kinny, D. N., and Georgeff, M. P. 1991. Commitment and Effectiveness of Situated Agents. In Proceedings of the Twelfth International Joint Conference on Artificial Intelligence, 82–88. Menlo Park, Calif.: International Joint Conferences on Artificial Intelligence.

Kraus, S.; Wilkenfeld, J.; and Zlotkin, G. 1992. Multiagent Negotiation under Time Constraints. Technical Report, CS-TR-2975, Computer Science Department, University of Maryland at College Park.

Laird, J. E.; Newell, A.; and Rosenbloom, P. S. 1987. SOAR: An Architecture for General Intelligence. *Artificial Intelligence* 33(1): 1–64.

Lansky, A. L. 1994. Located Planning with Diverse Plan Construction Methods. Technical Report FIA-TR-9405, NASA Ames Research Center, Mountain View, California.

Lansky, A. L., and Getoor, L. C. 1995. Scope and Abstraction: Two Criteria for Localized Planning. In Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence, 1612–1619. Menlo Park, Calif.: International Joint Conferences on Artificial Intelligence.

Lee, J.; Huber, M. J.; Durfee, E. H.; and Kenny, P. G. 1994. UM-PRS: An Implementation of the Procedural Reasoning System for Multirobot Applications. In Proceedings of the AIAA/NASA Conference on Intelligent Robotics in Field, Factory, Service, and Space, 842–849. Houston, Tex.: NASA Lyndon B. Johnson Space Center.

Levesque, H. J.; Cohen, P. R.; and Nunes, J. H. T. 1990. On Acting Together. In Proceedings of the Eighth National Conference on Artificial Intelligence, 94–99. Menlo Park, Calif.: American Association for Artificial Intelligence.

Lochbaum, K. E. 1998. A Collaborative Planning Model of Intentional Structure. *Computational Linguistics* 34(4): 525–572.

Musliner, D. J.; Durfee, E. H.; and Shin, K. G. 1995. World Modeling for the Dynamic Construction of Real-Time Control Plans. *Artificial Intelligence* 74:83–127.

Myers, K. L. 1996. Advisable Planning Systems. In *Advanced Planning Technology: Technological Achievements of the ARPA/Rome Laboratory Planning Initiative,* ed. A. Tate, 206–209. Menlo Park, Calif.: AAAI Press.

Ortiz, C. L. 1999. Introspective and Elaborative Processes in Rational Agents. *Annals of Artificial Intelligence and Mathematics* 25(1–2): 1–34.

Osborne, M. J., and Rubinstein, A. 1990. *Bargaining and Markets.* San Diego, Calif.: Academic.

Pollack, M. 1990. Plans as Complex Mental Attitudes. In *Intentions in Communication,* eds. P. R. Cohen, J. Morgan, and M. E. Pollack, 77–103. Cambridge, Mass.: MIT Press.

Pollack, M. E. 1996. Planning in Dynamic Environments: The DIPART System. In *Advanced Planning Technology: Technology Achievements of the ARPA/Rome Laboratory Planning Initiative,* ed. A. Tate, 218–225. Menlo Park, Calif.: AAAI Press.

Rosenschein, J. S., and Zlotkin, G. 1994. *Rules of Encounter.* Cambridge, Mass.: MIT Press.

Russell, S. J., and Norvig, P. 1995. *Artificial Intelligence: A Modern Approach.* Upper Saddle River, N.J.: Prentice Hall.

Russell, S. J., and Subramanian, D. 1995. Provably Bounded-Optimal Agents. *Journal of Artificial Intelligence Research* 2:575–609.

Sacerdoti, E. D. 1977. *A Structure for Plans and Behavior.* New York: American Elsevier.

Schoppers, M. J. 1987. Universal Plans for Reactive Robots in Unpredictable Environments. In Proceedings of the Tenth International Joint Conference on Artificial Intelligence, 1039–1046. Menlo Park, Calif.: International Joint Conferences on Artificial Intelligence.

Searle, J. 1990. Collective Intentions and Actions. In *Intentions in Communication,* eds. P. Cohen, J. Morgan, and M. Pollack, 401–415. Cambridge, Mass.: MIT Press.

Sen, S.; Haynes, T.; and Arora, N. 1997. Satisfying User Preferences While Negotiating Meetings. *International Journal of Human-Computer Studies* 47:407–427.

Shoham, Y. 1990. Agent-Oriented Programming. Technical Report, STAN-CS-90-1335, Computer Science Department, Stanford University.

Shoham, Y., and Tennenholtz, M. 1992. On the Synthesis of Useful Social Laws for Artificial Agent Societies. In Proceedings of the Tenth National Conference on Artificial Intelligence, 276–281. Menlo Park, Calif.: American Association for Artificial Intelligence.

Simon, H. 1983. *Reason in Human Affairs.* Stanford, Calif.: Stanford University Press.

Sonenberg, E.; Tidhar, G.; Werner, E.; Kinny, D.; Ljungberg, M.; and Rao, A. 1994. Planned Team Activity. Technical Report, 26, Australian Artificial Intelligence Institute at Melbourne.

Stone, P., and Veloso, M. 1999. Task Decomposition, Dynamic Role Assignment, and Low-Bandwidth Communication for Real-Time Strategic Teamwork. *Artificial Intelligence* 111(1): 241–273.

Sycara, K. 1990. Persuasive Argumentation in Negotiation. *Theory Decision* 28(3): 203–242.

Tambe, M. 1997. Toward Flexible Teamwork. *Journal of Artificial Intelligence Research* 7:83–124.

Toulmin, S. 1958. *The Uses of Argument.* London: Cambridge University Press.

Veloso, M. M. 1996. Toward Mixed-Initiative Rationale-Supported Planning. In *Advanced Planning Technology: Technological Achievements of the ARPA/Rome Laboratory Planning Initiative,* ed. A. Tate, 277–282. Menlo Park, Calif.: AAAI Press.

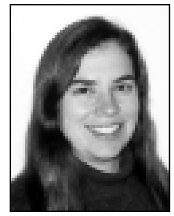
Wellman, M. 1992. A General-Equilibrium Approach to Distributed Transportation Planning. In Proceedings of the Tenth National Conference on Artificial Intelligence, 282–289. Menlo Park, Calif.: American Association for Artificial Intelligence.

Wilkins, D. E. 1988. *Practical Planning: Extending the Classical AI Planning Paradigm.* San Francisco, Calif.: Morgan Kaufmann.

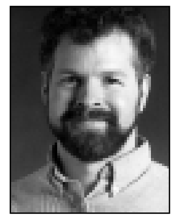
Wilkins, D. E., and Myers, K. L. 1998. A Multiagent Planning Architecture. In *Proceedings of the Fourth International Conference on Artificial Intelligence Planning Systems,* 154–162. Menlo Park, Calif.: AAAI Press.

Wolverton, M. J., and desJardins, M. 1998. Controlling Communication in Distributed Planning Using Irrelevance Reasoning. In Proceedings of the Fifteenth National Conference on Artificial Intelligence, 868–874. Menlo Park, Calif.: American Association for Artificial Intelligence.

Wray, I, R. E., and Laird, J. 1998. Maintaining Consistency in Hierarchical Reasoning. In Proceedings of the Fifteenth National Conference on Artificial Intelligence, 928–935. Menlo Park, Calif.: American Association for Artificial Intelligence.



Marie desJardins is a senior computer scientist in the Artificial Intelligence Center at SRI International. Her current research projects focus on distributed planning and negotiation, machine learning, and information management. Other research interests include probabilistic reasoning, decision theory, and intelligent tutoring systems. Her e-mail address is marie@ai.sri.com.



Edmund H. Durfee is an associate professor of electrical engineering and computer science at the University of Michigan, where he also directs the Artificial Intelligence Lab and holds a joint appointment in the School of Information. His research interests are in distributed AI, multiagent systems, planning, and real-time problem solving, applied to problems ranging from digital libraries to cooperative robotics, from assistance technologies to electronic commerce. His e-mail address is durfee@umich.edu.

Charles Ortiz is a senior computer scientist in the Artificial Intelligence Center at SRI International. He is also the past director of the Applied AI Technology Program at SRI. His e-mail address is ortiz@ai.sri.com.

Michael Wolverton is a computer scientist in the Artificial Intelligence Center at SRI International. His current research focuses on planning, distributed AI, information management, and case-based reasoning. His e-mail address is mjw@ai.sri.com.