Interactive Resource Management in the COMIREM Planner

Stephen F. Smith and **David W. Hildum** and **David R. Crimm**
The Robotics Institute, Carnegie Mellon University
5000 Forbes Avenue, Pittsburgh PA 15213
{sfs, hildum, dcrimm}@cs.cmu.edu

Abstract

In this paper, we describe Comirem, a light-weight, interactive tool for resource management in continuous planning domains. Comirem is designed for domains where complex, heterogeneous sets of resources are required to execute planned activities and usage must be synchronized to satisfy complex temporal and spatial constraints. Comirem promotes an opportunistic planning paradigm, where resource allocation decisions are made and revised incrementally as plans and availability constraints become known and refined during the planning process, and as execution deviates from planned behavior. To this end, resources are assigned (and re-assigned) via a least-commitment, constraint-posting scheduling procedure.

The Comirem system design follows three basic mixed-initiative principles: (1) that users will want to make planning and resource allocation decisions at different levels of detail in different circumstances and correspondingly delegate more or less decision-making responsibility to system processes in different contexts, (2) that abstract domain models, coupled with graphical visualization can provide an effective basis for communicating decision impact and proposing decision options, and (3) that incremental, adaptive problem-solving capabilities, which attempt to localize change whenever possible and appropriate, are central to maintaining continuity in the planning and resource management process. These principles are used to provide a variety of tools for mixed-initiative resource allocation, feasibility checking, resource tracking and conflict resolution.

1 Introduction

Resource management is a fundamental aspect of planning in many practical domains. Most generally, it is concerned with mapping system capabilities to input requirements over time. A given requirement typically requires the execution of a specific set of tasks, which in turn depends on the availability of specific resources. The system has some finite set of resources available for accomplishing various tasks, and allocation of these resources to competing tasks in an effective way is the crux of the resource management problem.

Historically, two system design factors have limited the effectiveness of planning tools in resource-constrained domains. First, resource management has generally been treated as a separable, post-process to plan generation. In some cases this division has followed from organizational structure (e.g., the "wall between engineering and production" in manufacturing domains); in others it has simply been an attempt to simplify the overall problem. There are a couple of reasons why this is usually a bad idea.

- First, planning decisions are often driven by what capabilities (or resources) are present. It is not uncommon for input requirements to over-reach system capabilities (particularly at early stages of the planning process) and early consideration of resource feasibility is essential to effective requirements analysis and efficient determination of feasible courses of action.
- A second problem is that plan generation and resource allocation decisions are generally quite inter-dependent. As resources are assigned to tasks, various usage constraints will frequently imply the generation and insertion of plan fragments that accomplish necessary support tasks.

A second historical limitation of planning technologies is that they tend to be designed as batch-oriented solution generators, where users attempt to manipulate inputs to achieve desired results. This user-interaction model and system design perspective is at direct odds with the characteristics of most practical planning domains, where requirements, capabilities and plans evolve incrementally and in parallel over time, and users invariably possess planning knowledge that should over-ride aspects of system models. Furthermore, the need for re-assessment and revision continues as the plan is executed and results deviate from expectations. In practice, planning and resource management is fundamentally an *iterative* process, and attempting to support this process with a batch-oriented solver results in an awkard, indirect and inherently inefficient problem solving cycle.

In this paper, we describe Comirem, an interactive planning tool designed to embed resource management more centrally into the planning process, and organized to more directly match the iterative nature of planning and scheduling in practical domains. Comirem promotes a graphical, spreadsheet-like model of user-system interaction, wherein resource assignment decisions can be constrained, refined and revised opportunistically as information becomes available, as specific planning choices become apparent, as execution results dictate, or as circumstances otherwise warrant. The user can invoke an automated scheduling capability to establish overall resource feasibility of a plan, and this capability can be coupled with an undo mechanism to explore the consequences of various requirement and/or resource availability changes. Similarly, the user can re-prioritize various resource allocation goals and preferences (e.g., prefer execution speed over resource vulnerability) and evalute alternative resource assignments. The system is currently being applied to the domain of Special Operations Forces (SOF) planning, where the use of transportation assets, task forces and equipment must be managed over time.

The remainder of this paper is organized as follows. In section 2 we first introduce the basic representational elements of Comirem planning models, followed in section 3 by a summary of basic procedures for managing the time and resource constraints that are definable in these models. In Section 4, we describe the mixed-initiative resource management capabilities that have been constructed using these primitives and illustate their usage in a specific planning context. Next, in Section 5, we identify the basic mixed-initiative problem-solving principles that underpin Comirem's resource management capabilities. We then summarize implementation status and briefly point up related work.

2 Representational Primitives

Models are expressed in Comirem in terms of *activities*, *resources* and *constraints*. An activity requires the use of one or more resources for some period of time to perform a designated task. Both activities and resources are governed by a set of constraints, which impose restrictions on when activities can execute, what resources can be assigned, and when resources are available for assignment. In the following subsections, we present these basic representational entities in further detail.

2.1 Resources

Comirem supports the definition of both stationary and mobile resources. Usage of stationary resources is strictly a function of their capability to accommodate various tasks. Mobile resources, alternatively, are a bit more complex. In some cases (e.g., transport vehicles) usage depends not only on their capability to support a given task, but also on such additional factors as capacity, speed, range and location. In other cases (e.g., personnel), usage can occur at different locations but may require transport between locations. Resources can also be grouped into higher-level *aggregate* resources (e.g., a military task force), to perform activities as a single entity for some period of time.

2.2 Activities

Comirem provides two basic activity types, one for representing *moves* from one location to another (typically achieved using one or more mobile resources), and another for representing *events* taking place at a single location (and possibly requiring one or more stationary resources). Activities impose a number of resource allocation constraints:

- *capability/resource requirements* designate the set of resources that might alternatively be used to support the activity, expressed either in terms of high-level *capabilities* (e.g., light-transport, close air support) or specific resource types
- a *duration*, specified either as a range, a single fixed time value, or as a function of speed and distance (in the case of moves) that dictates how long the activity will take
- a *manifest*, specified in terms of common cargo and passenger types and quantities, indicating the capacity requirement of the activity

A given input activity may expand into a hierarchy of sub-activities. For example, a move involving an aircraft may be comprised of positioning and depositioning legs, load and unload steps, and the actual flight leg between the origin and destination. Comirem provides the means for defining activity *types* that specify the expansion of an activity into a network of sub-activities, with each sub-activity possibly specifying its own requirements and imposing its own resource usage constraints.

Within a plan, it is possible that a set of activities may require that the same resource or collection of resources be used by each activity in the set, as in the case where a military task force is assigned to perform a series of missions using its own equipment. These sets of activities are typically contiguous sequences that define a series of moves from one location to another, which can be identified collectively as *threads* in Comirem. Threads define a local context within which a set of resources can be shared by a group of activities. When resources are assigned to a thread, they are available exclusively to the activities that comprise the thread.

2.3 Temporal Constraints

In addition to the previously mentioned constraints on activities and resources, Comirem provides the ability to synchronize activities (and by extension, threads) using two different kinds of constraints. To support the relative sequencing of activities, Comirem implements a set of canonical binary temporal relations, such as *before* and *after*, *same-start* and *same-finish*, and *overlaps* and *contains*. These relations enforce sequencing rules while attempting to preserve flexibility within the time bounds of constrained activities.

To achieve a possibly tighter degree of synchronization of activities, Comirem also allows specification of temporal *anchors*, which tie activities (by either start or finish times, or both) to a particular reference point along the timeline. For example, a release date (i.e., a *no-earlier-than* start constraint) is enforced by anchoring the start times of all source activities in the plan. Anchors can also be used to define special reference points within a plan, like the *N Hour* and *H Hour* typically defined in military plans. Anchors may be shifted in time or associated with a temporal interval, in which case the time bounds of anchored activities are updated accordingly.

3 Constraint Management Capabilities

Comirem takes a *flexible-times* approach to allocating resources to activities over time. Activities in the plan are allowed to float within the time bounds that are permitted by current problem constraints, and the feasibility of various resource assignments is ensured by sequencing any pair of activities that are assigned the same resource. This approach can be contrasted with most current scheduling tools, which ensure resource feasibility by assigning *fixed* start and end times when allocating resources to activities. Fixed-time scheduling can simplify the enforcement of various resource usage constraints but it also generally implies considerable decision over-commitment. Under a flexible-times scheduling approach, alternatively, greater decision flexibility is retained. At both plan development and plan execution stages, this minimizes the need for change.

3.1 Managing Temporal Constraints

Within this flexible-times framework, temporal constraint propagation and consistency enforcement is achieved by encoding various elements of the plan as a Simple Temporal Problem (STP) constraint network [Dechter *et al.*, 1991], and applying an incremental STP constraint network solver (in our case based on Dijkstra's shortest path algorithm). Briefly, temporal constraints are represented in an underlying graph $G < N,E >$, where nodes in N represent time points, and edges in E are distances (labeled as $<$ *lower bound, upper bound* $>$ pairs) between the time points in N. A special time point, called "calendar zero," grounds the network and has the value 0. The network maintains lower and upper bounds on the time points by propagating the bounds on the distances of the edges. Activities, anchors and temporal sequencing constraints are uniformly represented as temporal constraints (i.e., edges) between relevant start and finish time points. Planning/scheduling decisions generally correspond to the introduction of new constraints into the network (e.g., sequencing two activities which have been assigned the same resource) or the adjustment of existing constraints (e.g., refining the duration of an activity, modifying an anchor). In either case, constraint propagation updates the bounds of affected nodes and checks for cycles in the resulting network. The lack of any such cycle ensures continued temporal feasibility of the plan. Otherwise a conflict has been detected, and backtracking (or some amount of constraint relaxation) is necessary.

3.2 Managing Resource Constraints

As implied above, basic resource feasibility is maintained by enforcing a disjunctive constraint on the execution of any two activities utilizing the same resource. If two such activities would otherwise have potentially overlapping execution windows then, establishment of feasibility requires that an explicit sequencing constraint be introduced. Note that for an activity requiring several resources, this can imply several distinct links.

Comirem provides a number of more specialized resource constraint management techniques, aimed at satisfaction of additional resource usage constraints:

- resource location - To support a given move activity, a resource must be at the activity's origin location at its start and will be at the destination location at its finish. Depending on the location of the resource prior to its allocation to this activity, a supporting positioning activity might be necessary and likewise, depending on its subsequent obligations, de-positioning might be necessary. As discussed earlier, a move activity expansion typically includes provisions for such *derivative* activities. As a given resource is assigned, location constraints are enforced by appropriately adjusting relevant duration constaints.
- resource carrying capacity - A given move activity specifies a manifest quantity which, together with the carrying capacity of available resources, will ultimately determine how many trips will be required and how long the overall move will take. When resources of a given type are allocated to a given activity, *sibling* activity networks are created and instantiated to correspond with each required trip. Depending on the number of resources actually available, these trips may occur in multiple waves, with some resources being reused.

4 Mixed-Initiative Resource Management

The above constraint management and scheduling procedures are configured to provide a range of interactive and semi-automated resource management capabilities within Comirem. Generally speaking, Comirem is designed to provide a graphical spreadsheet-style model of user-system interaction, where the user opportunistically manipulates various planning and plan data (e.g., requirements and capabilities, resource allocation decisions) and the consequences of user changes (e.g., updated sets of decision options, instantiation of implied derivative activity networks) are made apparent. The user can selectively employ more automated capabilities (e.g., requesting resource feasiblity checking and auto-scheduling for larger fragments of the developing plan), in which case the resulting "propagation of effects" of a user action can be significant (and may involve heuristic choice). Also following the spreadsheet metaphor, the user can change her mind at any point, retract one or more prior actions, and resume the planning process in another direction.

The current planning focus in Comirem is on the problem of assigning resources to a (possibly evolving) plan and then managing these assignments as execution of the plan moves forward. Figure 1 gives a concrete example of the type of problem of interest. It graphically depicts a plan for evacuating citizens from a foreign embassy. Following an initial staging activity, the plan consists basically of three parallel threads: one aimed at securing a local airport for eventual extraction, one concerned at blocking the advance of hostile forces to the airport and a final thread aimed at transfering citizens from the embassy (where they have congregated) to the airport. Once at the airport, everyone flies out. Several of the activities in this plan, specifically the moves, require various transport and combat capabilities, and input constraints also specify initial assumptions concerning the locations and availability of various air and land assets. Additionally, each

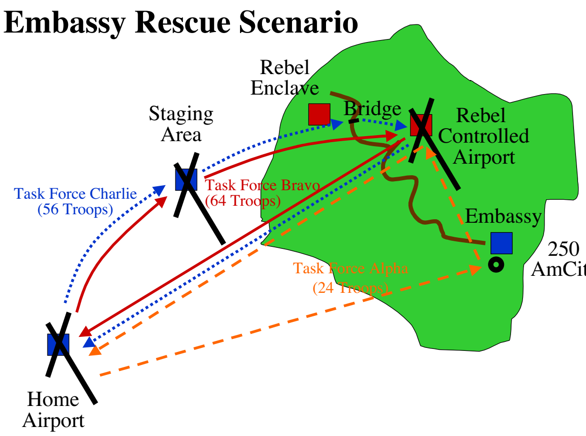

Figure 1: Embassy Rescue Scenario

thread of the plan presumes an independent task force, which must be configured from available personnel.

In the following subsections we summarize the functional capabilities provided in Comirem for addressing such resource management problems.

4.1 Interactive Planning and Resource Allocation

The problem at earlier stages of the planning process is one of reconciling an input concept of operations, represented as a set of inter-related activities, with a heterogeneous set of available resources. Activities specify required capabilities, which may be provided by one or more types of available resources. Figure 2 shows the Embassy Rescue plan just discussed within the Comirem GUI. Here we are looking at a “vector” display of the plan, where horizontal bars represent locations over time and diagonal lines represent movement activities from place to place. Currently unassigned activities are designated in yellow.[1]

Given this sort of input state, Comirem provides the following support for refining and sourcing a given input plan:

- *option generation* - For each unassigned activity in the plan, Comirem maintains the current set of feasible allocation options. Figure 3 shows the situation for the initial `position-for-insertion` activity of the plan. There are two feasible choices: 3 *MH-47* helicopters or 9 *MH-60* helicopters. The numbers required reflect the size of the manifest (task force) to be transported and the respective carrying capacities of the assets; the durations given reflect the quantities available, their speed, and their locations.

- *visualization of decision impact* - As the user elects to commit to a given option, impact of this assignment is reflected in the plan. Figure 4 (top), for example, shows the MH-47 assignment (where 3 helicopters will fly in tandem). Since these assets are initially based at *Home-Station*, there is no need for a positioning flight; but a depositioning flight has been added. In Figure 4 (bottom) the impact of alternatively choosing the MH-60’s is shown. In this case availability constraints force the activity to be performed in two waves of 7 and 2 trips (and hence the significantly longer duration).

  Other visual cues provide information about the constrainedness of the current planning search space. The vector display can be re-colored to indicate decision criticality (a function of how many resource assignment options remain). Similarly, the display can be toggled to show the degree of temporal flexibility associated with different activities; Figure 4 actually shows the earliest execution interval of `position-for-insertion`.

- *requirements and capabilities editing* - Both activity and resource attributes can also be edited to alter the constraints and requirements of the problem. In Figure 3, the `position-for-insertion` activity is shown to require a *light-transport* capability (shown in red), which maps to either of the helicopter types mentioned earlier. Through selection in this display, the user is free to adjust these requirements, either specifying alternative or additional capabilities or platforms that might be viable to consider. Any adjustments result in a corresponding update of feasible options. Other timing constraints on activities (e.g., deadlines or synchronization points) can similarly be manipulated to open up (or restrict) the set of options.

  Resource attributes can similarly be manipulated to vary basic problem assumptions. For example, the numbers of resources available and their initial locations can be revised to evaluate the consequences of an alternative deployment. In Figure 5 we show an editing action to introduce the constraint that only one aircraft can be on the ground at a time at location *Staging-Area* (whereas it had previously been specified as *unconstrained*). Figure 6 shows the conseqence on the `position-for-`

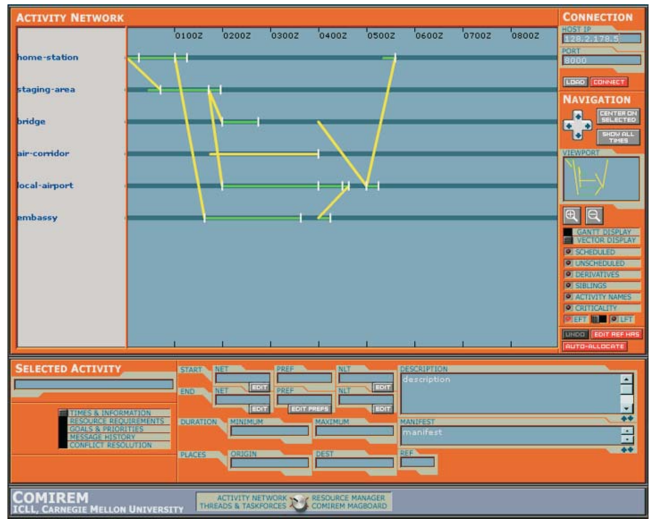

Figure 2: Comirem GUI

[1] For convenience, the figures in this paper have been posted in color at the following URL: `http://www.cs.cmu.edu/afs/cs/project/ozone/www/public/IJCAI-03-MIIS-WS-figures.html`

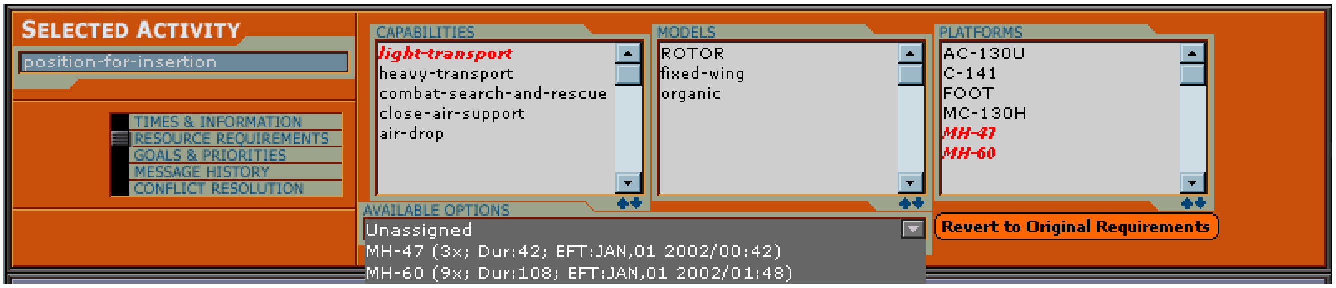

Figure 3: Option Generation

insertion activity; the 3 *MH-47* flights now must unload in a staggered fashion.

- *automated assignment and feasibility* - The user can selectively rely more on system decision procedures. For any selected set of unassigned activites in the plan, the user can invoke an underlying automatic scheduler to generate a feasible assignment. This action constitutes a check of the overall resource feasibility of the plan (assuming whatever decisions have been previously fixed by the user). The allocation goals and preferences that focus the automated scheduler can be adjusted by the user to produce different sets of resource assignments, in scenarios where overall resource capabilities contain this flexibility.
- *what-if analysis* - As implied by much of the discussion above, most constraints and planning decisions are pliable through the interface to allow general exploration of the problem and solution space. The system provides a general ability to "undo" any user-level action (or sequence of actions) previously taken. Coupled with the above described capabilities, this mechanism provides a flexible, open-ended framework for what-if analysis.

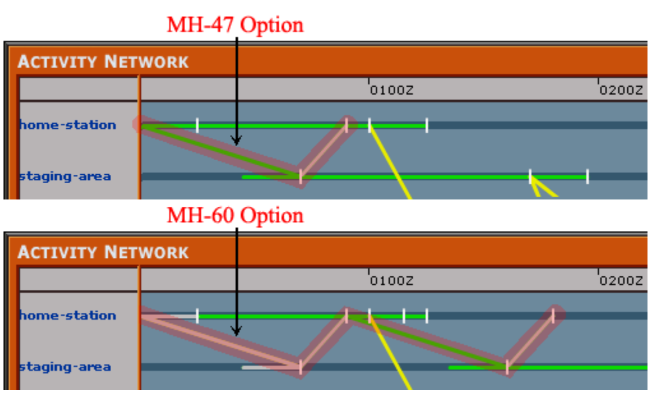

Figure 4: MH-47 and MH-60 assignment options

4.2 Resource Configuration

Complementary capabilities for interactively configuring aggregate resources (task forces in our embassy rescue scenario) from more primitive resource objects (in this case, individual

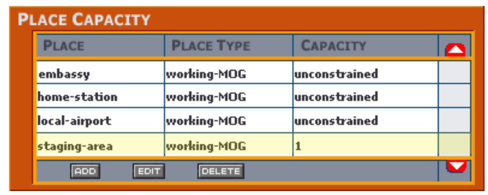

Figure 5: Imposing a maximum-on-ground constraint

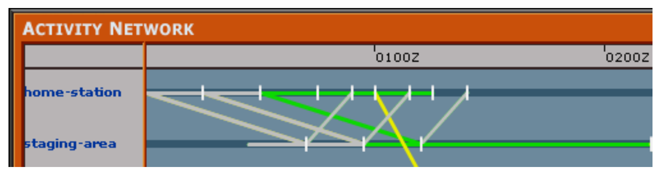

Figure 6: Revised MH-47 assignment

personnel) are provided through a drag and drop interface. As a given resource is added to an aggregate under development, the system checks basic availability constraints. When an aggregate resource is assigned to an activity (or to a larger thread of activities in the plan), this resource and all of its constituents become otherwise unavailable for allocation over the period that the activity or thread executes. Hence, just as in the case where primitive resources are allocated, the potential addition of a resource to an aggregate resource or the potential assignment of an aggregate resource to a thread may be infeasible due to prior commitments, or likewise, may require additional sequencing of competing activities in the plan.

4.3 Execution Management

Comirem provides some additional tools for managing resources and resource assignments as execution unfolds:

- *resource tracking* - A "mag-board" UI display provides a geographical visualization of the location of all resources at any given point in time. The display distinguishes those resources that are (1) in-transit from one location to another, (2) at a location but engaged in some activity (e.g., providing air cover support, securing an airport), or (3) at a location and available for use.
- *plan tracking* - Both the vector display of the plan (and an additional Gantt based view) can also be used to visualize the current state of execution. As information

Figure 7: Conflict resolution options

about the actual start and end of activities is received, the executing frontier of the plan is highlighted and any conflicts between the actual and expected course of events is signaled (involved activities turn red).

- *conflict analysis* - In the event that one or more activities in the plan are detected as infeasible (either due to unexpected execution results or as a result of a temporally inconsistent input plan), the system provides support resolving the underlying constraint conflicts. Through identification and analysis of the set of constraints involved (i.e., the cycle returned by the temporal constraint propagator), the system generates a set of basic conflict resolution options. For example, Figure 7 shows options for resolving a conflict where with current resource assignments it is not possible to complete an activity by its imposed deadline.

5 Mixed-Initiative Planning Principles

The design of the Comirem planner reflects several basic principles that we believe are fundamental to collaborative, human-computer interaction in practical planning domains:

- *Adjustable decision-making autonomy* - One key to effective collaborative planning is a decision-making process that allows the degree of automation to vary according to problem solving context and user preference. The Comirem planner allows the user to inject herself into the decision-making process at different levels of granularity. At the lowest level, the system takes responsibility only for proposing and implementing resource assignments for individual elements (or activities) in the plan, and the user retains control over resource assignment decisions. However, even in this (largely manual) mode of operation, the system is frequently playing a quite substantial role, in determining the consequences of all relevant constraints on user allocation decisions (e.g., the overall duration of a move, given that multiple trips are required, that aircrews must rest during the course of the move, etc.). The user is free to make specific adjustments to an activity's resource requirements (e.g., adding another aircraft type to the set of possible allocation options) or to over-ride specific usage constraints (e.g., reduce computed duration constraints due to knowledge of a strong tail wind) without carrying along and anticipating interactions with all other constraints. In other decision contexts the user can delegate increased decision-making scope to the system. The user may choose to manipulate constraints associated with larger plan fragments (e.g., plan threads), and request feasiblity checking and/or scheduling in response to assess consequences. At the end of the automation continuum, the user can request the construction of a fully sourced plan, in which case the system constructs a schedule according to user specified goals and preferences. Through iterative generation, retraction and adjustment of (sets of) resource assignments, system planning and scheduling processes support collaborative reconciliation of planning objectives and desired actions with available resources.

- *Translation of system models and decisions* - A second key to effective collaborative planning is an ability on the part of the system to communicate elements of its internal models and solutions in user comprehensible terms. In Comirem, this is accomplished through a combination of graphical visualization and model-based explanation techniques. As is illustrated in Figures 4 and 6 visual displays are used to compactly convey various decision implications (e.g., how many trips are required if this type of transportation resource is used, where various resources will be sourced from, overall makespan of the "move" activity) as well as potentially non-obvious allocation constraints (e.g., MOG limitations, etc.). In other advice giving contexts, internal system representations are mapped to user level actions through the use of abstract domain models. In Figure 7, for example, Comirem's higher-level "activity-resource" model allows the system to impose structure on the distance constraints involved in a "cycle" that has been detected in the temporal constraint network (i.e., characterizing them as inter-dependent sets of activity durations, inter-activity precedence relations, etc.). This structure then provides the basis for identifying and proposing relevant, user-level constraint relaxation actions that make sense for resolving the conflict.

- *Incremental problem-solving procedures* - A third key to effective collaborative planning is a problem-solving process that promotes incremental decision making. In complex, ill-structured planning domains, all relevant constraints are generally not known at the outset, and those that are can frequently change as planning and execution progresses over time. Hence, planning proceeds with whatever information is available, decisions are made on the basis of current information, and as additional information becomes known, conflicting decisions are appropriately revised. Within such a continuous planning process, the ability to maintain a sense of problem-solving continuity to the user is crucial. In arriving at a given plan or schedule (either partial or complete), the user has invested time and energy in understanding, assessing and validating various component decisions. The introduction of new or changed constraints should not arbitrarily result in wholesale changes to the plan. Instead the system should act to preserve prior decisions wherever possible and meaningful. The Comirem planning framework is designed generally with an incremental, change-based problem-solving process in mind, and furthermore allows the user to explicitly manage the scope of allowable change in specific decision contexts.

6 Implementation and Status

The Comirem planner is implemented in Common Lisp and runs as an HTTP server on multiple platforms (Solaris, MacOS and Windows). The GUI runs as a Shockwave application in a standard web browser (Netscape, Internet Explorer). The GUI sends URL-encoded requests to the Comirem server and the server responds with XML-formatted text. This client/server paradigm allows for any client to access and use any server residing on the same network (including the internet), regardless of platform. Also, the client requires only that its browser have a (freely available) Shockwave plug-in installed (and a network connection).

Comirem is currently being integrated with a collection of other tools developed for the SOF domain, including SOFTools and SOFPlan (both interactive plan authoring tools) and C2PC (an interactive mapping program). Comirem accepts XML-formatted plan data files from SOFTools and SOFPlan, performs resource feasibility checking and exports "magboard"-style output data for display in C2PC. One current research effort involves the attempt to extract detailed routing information from SOFPlan for use as input to the duration-calculation process for aircraft movements in Comirem.

7 Related Work

The Comirem effort descends from earlier work with the AMC Barrel Allocator [Becker and Smith, 2000], a tool for day-to-day allocation of aircraft and aircrews for the USAF that is now part of the operational planning system in the Tanker/Airlift Contol Center at the USAF Air Mobility Command. Whereas this application focuses on large-scale problems with more narrrowly circumscribed types of allocation constraints, Comirem focuses on smaller-scale but more ill-stuctured types of allocation problems.

A number of other recent efforts in interactive planning have also started to break the mold of trying to integrate with batch-oriented, systematic planning technologies. TRIPS [Ferguson and Allen, 1998] adopts a similar opportunistic view of plan development and is designed for use in similar transportation related planning domains. Comirem appears to address a richer class of resource usage constraints, while TRIPS provides greater generality with respect to construction of activity networks. PASSAT [Myers *et al.*, 2002] has recently introduced the concept of plan sketches, which similarly enforces no strong commitment on the order in which planning decisions must be made and allows the user to defer and/or relax various constraints as the plan is being assembled. PASSAT however, has no explicit model of time and hence deals with a quite different class of planning problems. OPLAN [Drabble and Tate, 1995] is a final system that has long emphasized an opportunistic user-system interaction model.

8 Current Directions

Our current work with Comirem is focused in several directions. One thrust aims at expanding the set of resource usage constraints that can be effectively handled, specifically the treatment of producer-consumer relations. For example, in transshipment contexts, where cargos are being moved to an intermediate location and then reconstituted for subsequent movement elsewhere, the relative carrying capacities of the resources involved at each stage may allow execution of respective activities to be overlapped. A second research thrust is directed at providing direct interactive support for creating and manipulating activities and activity networks. Finally, we are interested generally in the design of user-directed strategies for flexible-times scheduling.

9 Acknowledgements

The research reported in this article was sponsored in part by the Department of Defense Advanced Research Projects Agency (DARPA) under contract F30602-00-2-0503 and by the CMU Robotics Institute.

References

[Becker and Smith, 2000] M.A Becker and S.F Smith. Mixed-initiative resource management: The amc barrel allocator. In *Proc. 5th Int. Conference on AI Planning and Scheduling*, Breckenridge CO, April 2000.

[Dechter *et al.*, 1991] R. Dechter, I. Meiri, and J. Pearl. Temporal constraint networks. *Artificial Intelligence*, 49, 1991.

[Drabble and Tate, 1995] B. Drabble and A. Tate. O-plan mixed-initiative planning capabilities and protocols. Technical report, Univ. of Edinburgh, 1995.

[Ferguson and Allen, 1998] G. Ferguson and J. Allen. Trips: An integrated intelligent problem-solving assistant. In *Proc. AAAI-98*, Madison, WI, 1998.

[Myers *et al.*, 2002] M.A Myers, W.M. Tyson, M.J. Wolverton, P.A. Jarvis, T.J. Lee, and M. desjardins. Passat: A user-centric planning framework. In *Proc. 3rd Int. Workshop on Planning and Scheduling for Space*, Houston, TX, Oct 2002.