

Comirem: An Intelligent Form for Resource Management

Stephen F. Smith, David W. Hildum, and David R. Crimm, *Carnegie Mellon University*

A significant historical limitation of planning and scheduling technologies is that they tend to be designed as batch-oriented solution generators, in which users can't directly guide solution development. Instead, they attempt to achieve desired results indirectly by manipulating system inputs. This user-interaction model and system design

perspective is at odds with the characteristics of most practical domains, where requirements, capabilities, and plans evolve incrementally and in parallel over time, and users invariably possess knowledge that should override aspects of system models. Furthermore, the need for reassessment and revision continues as the plan executes 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 awkward, indirect, and inherently inefficient problem-solving cycle. Comirem (*Continuous, Mixed-Initiative Resource Management*) is an "intelligent form" for resource management that's designed to more directly match the iterative, user-centered nature of planning and scheduling in practical domains. In this article, we provide an overview of the current Comirem system.

System fundamentals

Comirem implements a user-centered approach to scheduling and resource allocation on the basis of three basic principles:

- 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.
- Abstract domain models, coupled with graphical visualization, can provide an effective basis for

communicating decision impact and proposing decision options.

- 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.

Comirem promotes a graphical, spreadsheet-like model of user-system interaction. The user interacts with the system through Web-based forms that combine visual displays of the current plan and planning state with form-based editing capabilities so that the user can manipulate aspects of the solution and solution state. The user can opportunistically constrain, refine, and revise resource assignment decisions associated with various elements of the plan as information becomes available, as specific planning choices become apparent, as execution results dictate, or as circumstances otherwise warrant. When a user takes any action, the system computes consequences and updates interactive forms accordingly.

Comirem provides a range of support for user-driven resource management. At any point during planning, the system computes and maintains a basic set of resource allocation options for any task in the plan. As the user adjusts task requirements and resource capabilities, or as the user makes specific resource assignments, the system applies underlying

Comirem implements a user-centered, mixed-initiative approach to scheduling and resource allocation in continuous planning domains. Users interact with the system through a Web-based, spreadsheet-style GUI that reflects a structured-forms decision-making paradigm.

time and resource propagation techniques to update the feasible options for other tasks. This update identifies forced decisions and detects infeasible solutions. The user can invoke an automated scheduler to establish a plan's overall resource feasibility, and can couple this capability with an undo mechanism to explore the consequences of various requirement and resource availability changes. All interaction with the system is in terms of a high-level domain model, which shields the user from the details of the system's underlying search model.

We developed Comirem as part of DARPA's Active Templates Program. The system has been applied initially to the domain of Special Operations Forces planning, which is typified by very short advance notice. So, users must manage resources such as transportation assets, task forces, and equipment—each with nontrivial availability and usage constraints—under significantly stressful and time-pressured circumstances. Comirem represents a first attempt to provide an integrated framework for SOF resource management, which is a largely manual process. We've demonstrated and evaluated the current prototype using plans that SOF subject matter experts developed, and iterative feedback from these individuals has driven its evolution to date. Currently, the SOF-planning user community is further evaluating the prototype.

From a research perspective, we view SOF planning as a representative problem domain. Comirem is designed to allow input of other planning and scheduling domain models and is applicable to a broad class of resource management applications.

Representational primitives

In Comirem, the user specifies a domain model using the Ozone scheduling domain ontology, which allows a user-interpretable description of an application domain to be mapped to application system functionalities.¹ The Ozone ontology expresses domains in terms of *resources*, *activities*, and *constraints*. Semantically, an activity describes a particular task whose performance typically requires the use of one or more resources for some time period. 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. Using the ontology, Comirem provides predefined sets of activity types, resource types, and constraint types as representational primitives. A

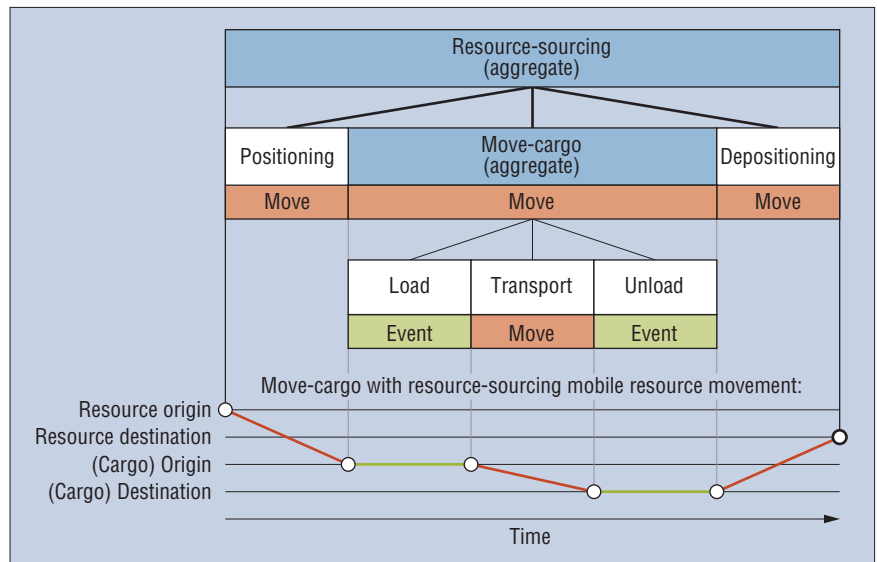


Figure 1. An aggregate activity type and its network of subactivities. Comirem provides an enhanced facility for defining new aggregate activity types.

user specifies a domain by specializing (as appropriate) these predefined sets and then specifying instances of each to formulate the set of available resources, the set of activities that make up the input plan, and associated constraints. The user then encodes these system inputs using an XML specification.

Resources

Comirem supports the definition of both stationary and mobile resources. Use of stationary resources, such as bases and airstrips, is strictly a function of their capability to accommodate various tasks. Mobile resources are more complex. In the case of a transport vehicle, use depends not only on the resource's capability to support a given task but also on such additional factors as capacity, speed, and location. In other cases, such as personnel, use can occur at different locations but might require transport in between. Resources can also be grouped into higher-level *aggregate* resources, such as military task forces, to perform activities as a single entity for some period of time.

Activities

Comirem provides two basic activity types for specifying plan elements: *moves* from one location to another and *events* occurring at a single location. Both types of activities can require resources to execute; moves typically require mobile resources, whereas events can require any type of resource. Additionally, activities impose a number of resource allocation constraints:

- *Capability/resource requirements* designate the types of resources that might alternatively be used to support the activity, expressed either in terms of high-level *capabilities* (for example, light transport and close air support) or specific resource types (for example, rotor, fixed-wing, MH-47, and C-130).
- *Duration* is 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* is specified in terms of common cargo and passenger types and quantities, indicating the activity's capacity requirement.

The user specifies a plan for input into Comirem as a set of activities together with an associated set of temporal constraints, which we'll discuss in a moment. These user-level input activities might imply more complex hierarchies of subactivities. For example, a transport activity involving an aircraft might consist of positioning and depositing flights, load and unload steps, and the actual cargo movement between the origin and destination.

Comirem provides a facility for defining new aggregate activity types that specify how to expand an activity of this type into a network of subactivities, with each subactivity possibly specifying its own requirements and imposing its own resource use constraints (see Figure 1). Once a set of aggregate types has been specified for a given domain, the

user interacts strictly with the plan's user-level elements, and Comirem manages their corresponding expansions.

Within a plan, a sequence of activities might require the same set of resources. For example, a military task force might be configured with a specific set of vehicles to perform a specific multistage operation. Such activity sequences can be specified collectively as *threads* in Comirem. Threads define a local context in which a group of activities can share a set of resources. When the user assigns resources to a thread, they're available exclusively to the activities that constitute the thread.

Constraints

Comirem provides the ability to specify temporal relationships between activities (and by extension, threads) using three kinds of constraints. To support relative sequencing of activities, Comirem implements a set of canonical binary temporal relations, such as *before* and *after*, *same-start* and *same-finish*, *overlaps*, and *contains*. These relations enforce sequencing rules while attempting to preserve flexibility in the time bounds of constrained activities.

To achieve tighter synchronization of activities, Comirem provides *reference hour* (ref-hour) constraints for linking sets of activities to a relative date, which the user can assign to a particular point in time and shift as necessary. This capability is ideal for defining the *N-Hour* (notification hour) and *H-Hour* (zero hour) synchronization points typically used in military plans. The user can link an activity to a ref-hour by either its start or finish time, or both. These links define what we call the *preferred* start and finish times for the activity.

Each ref-hour maintains its own flexible time window (defined by upper- and lower-bound offsets from the ref-hour's time point), which establishes a "level of precision" with respect to the synchronization of activities (the default is three minutes). If the preferred start and finish times of linked activities occur within this level of precision as execution unfolds, a plan is considered to be executing as planned.

Anchoring constraints introduce lower and upper bounds on the start and finish times of activities, for defining not-earlier-than (that is, NET or EST/EFT) and not-later-than (NLT or LST/LFT) temporal relations. Release dates and due dates are typical anchors: a release date imposes an EST constraint on

the initial activities in a plan, while a due date imposes an LFT constraint on the final activities in a plan.

Constraint management capabilities

Comirem takes a *flexible-times* approach to allocating resources to activities over time. Activities in the plan may float within the time bounds that current problem constraints permit, and the feasibility of various resource assignments is ensured by sequencing any pair of activities that's assigned the same resource. Contrast this approach with that of most current scheduling tools, which ensure resource feasibility by assigning fixed start and end times when allocating resources to

Comirem takes a flexible-times approach to allocating resources to an activity over time. Fixed-time scheduling, used by most current scheduling tools, generally implies overcommitment.

activities.² Fixed-time scheduling can simplify the enforcement of various resource-use constraints, but it also generally implies considerable overcommitment. Flexible-times scheduling, on the other hand, retains greater flexibility. At both the plan development and plan execution stages, this minimizes the need for change.

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* constraint network³ and applying an incremental STP constraint network solver. Briefly, temporal constraints are represented in an underlying graph $G \langle N, E \rangle$, 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 (edges) between relevant start- and end-time points. As the system enforces planning and scheduling decisions, generally either new constraints are introduced into the network (for example, two activities are sequenced as a result of being assigned the same resource) or existing constraints are adjusted (for example, when an activity's duration changes as the result of assigning a different type of resource, or an anchor is modified). In either case, constraint propagation updates the bounds of the affected nodes and checks for cycles in the resulting network. The lack of a cycle ensures continued temporal feasibility. Otherwise, the system has detected a conflict, and backtracking or some amount of constraint relaxation is necessary.

Managing resource constraints

A temporally feasible plan isn't necessarily resource feasible. As we indicated previously, basic resource feasibility is maintained by enforcing a disjunctive constraint on the execution of any two activities using the same resource. If two such activities would otherwise have potentially overlapping execution windows, preservation of feasibility requires that an explicit sequencing constraint be introduced between them. For an activity requiring several resources, this can imply several distinct links. Comirem provides many more specialized resource constraint management techniques, aimed at satisfying additional resource use 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 resource's location prior to its allocation to this activity, a supporting positioning activity might be necessary, and depending on its subsequent obligations, depositioning might likewise be necessary. As discussed earlier, a move activity expansion typically includes provisions for such derivative activities. As the system executes a given resource assignment, it automatically enforces location constraints by appropriately adjusting relevant duration constraints.

Resource carrying capacity. Mobile resources have a carrying capacity that limits how

much cargo they can transport on any given trip. So, a given move activity might require multiple resources (and multiple trips) to accommodate the quantity its manifest designates. When the user or the system allocates resources of a given type to a given activity, *sibling* activity networks are automatically created and instantiated to correspond with each required trip. Depending on the number of resources actually available, these trips might occur in multiple waves, with some resources being reused.

Mixed-initiative resource management

The Comirem constraint management and scheduling infrastructure we described in the previous section provides the basis for a range of interactive and semiautomated resource-management capabilities. As we mentioned, we designed Comirem to provide a graphical, spreadsheet-style model of user-system interaction in which the user opportunistically manipulates various planning and plan data and the consequences of user changes (for example, updated sets of decision options, and instantiation of implied derivative-activity networks) are made apparent. The user can selectively employ more automated capabilities, such as requesting resource feasibility checking and autoscheduling for larger fragments of the developing plan. In such a case, the resulting “propagation of effects” of a user action can be significant and might 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.

Comirem’s current planning focus is assigning resources to a (possibly evolving) plan and then managing these assignments as the plan’s execution moves forward. As we indicated earlier, input to the system takes the form of a set of user-level activities and constraints, which collectively specify the actions needed to accomplish certain end goals for a given scenario. A typical SOF plan can involve upwards of 100 or so activities comprising multiple synchronized mission threads to achieve a set of military objectives. Such objectives include reconnaissance and surveillance, covert and overt offensive actions on targets and forces in enemy-held or controlled environments, and capture and recovery of personnel and material.

Figure 2 gives a simple example of the type of problem of interest, graphically depicting a

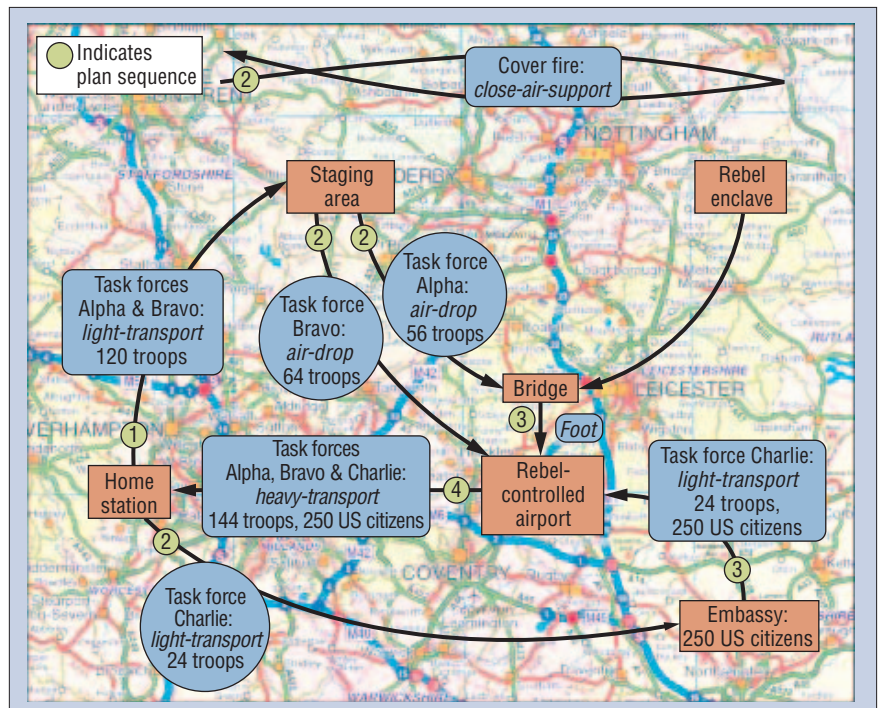


Figure 2. A foreign embassy rescue plan (Embassy-Rescue).

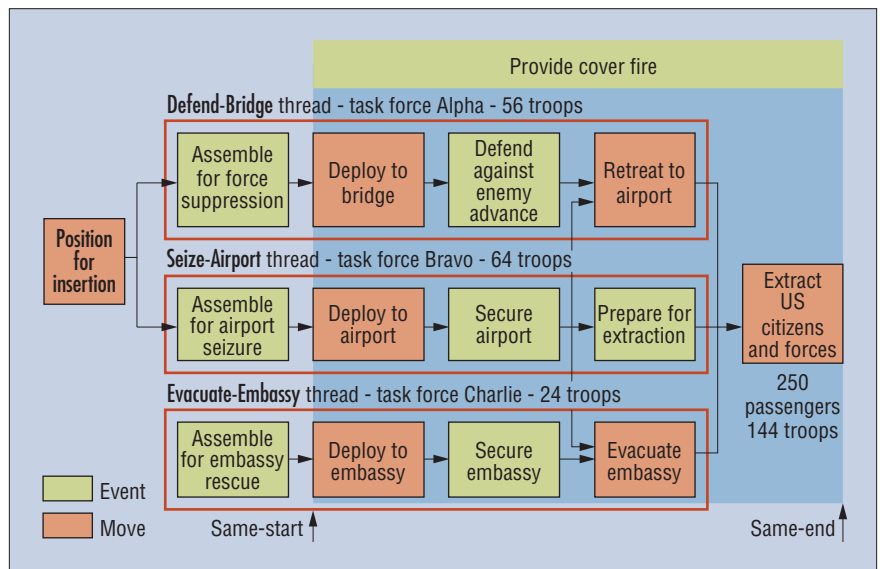


Figure 3. Embassy-Rescue from an activity-centered perspective.

plan for evacuating citizens from an embassy in a foreign capital. Figure 3 shows the same plan from an activity-centered perspective. Following an initial staging activity, the Embassy-Rescue plan consists basically of three parallel threads: one aimed at securing a local airport for eventual extraction (Seize-Airport), one concerned with blocking the advance of hostile forces to the airport (Defend-Bridge), and

one aimed at transferring citizens from the embassy, where they’ve congregated, to the airport (Evacuate-Embassy). Once at the airport, everyone is flown home.

Several activities in this plan, specifically the moves, require various transport and combat capabilities. Input constraints also specify initial assumptions concerning the locations and availability of various air and

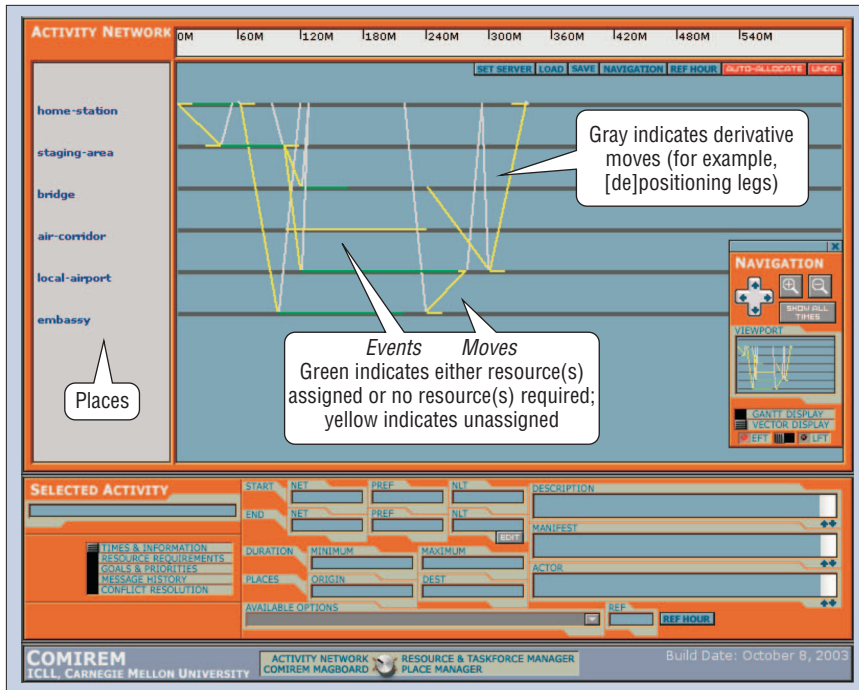


Figure 4. An annotated Comirem GUI “vector” display of the Embassy-Rescue plan showing an empty activity-editing form at the bottom.

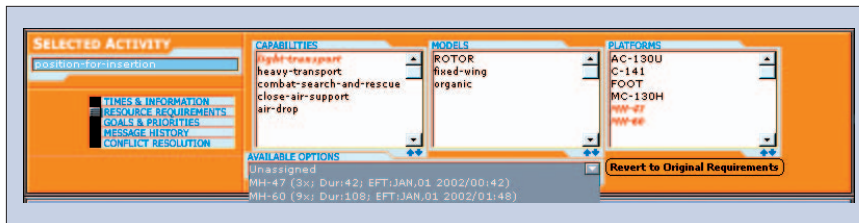


Figure 5. Option generation and selection for the position-for-insertion activity.

land assets. Additionally, each thread presumes an independent task force, which must be configured from available personnel. Starting from the initial plan, Comirem helps the user allocate resources to input activities. In cases where feasible assignment entails the expansion of user-level activities and establishment of appropriate enabling conditions (such as for positioning an aircraft where it’s needed), Comirem solves such planning subproblems dynamically and automatically as specific resource assignments are considered. If the overall process is successful, the result is a detailed plan in which each activity is assigned the resources it requires and is designated to execute in a specified finite time interval.

Using the Embassy-Rescue plan as a reference, we’ll summarize Comirem’s key functional capabilities for intelligently helping the user address such resource allocation problems.

Interactive planning and resource allocation

The problem at the planning process’s earlier stages is reconciling an input concept of operations, represented as a set of interrelated activities, with a heterogeneous set of available resources. Activities specify required capabilities, which one or more types of available resources might provide.

Figure 4 shows Embassy-Rescue as presented by the Web-browser-based Comirem GUI. It presents a “vector” display of the plan, in which the background horizontal bars represent locations over time, diagonal lines represent moves from place to place, and horizontal lines represent events at a location. Given this input state, Comirem provides the following support for refining and sourcing a given plan.

Option generation. For each unassigned

activity in the plan, Comirem maintains the current set of feasible allocation options. Figure 5 shows the situation for the initial position-for-insertion activity. The two feasible choices are three MH-47 helicopters or nine MH-60 helicopters. The numbers required reflect the number of troops in assigned task forces Alpha and Bravo that must be transported and the respective assets’ carrying capacities; the durations given reflect the number of resources available, their speed, and their locations.

Visualization of decision impact. As the user elects to commit to a given option, the plan reflects the assignment’s impact. Figure 6a, for example, shows the MH-47 assignment—three helicopters will fly in tandem. Because these assets are initially based at home-station, a positioning flight is unnecessary; however, a depositioning flight is necessary (recall Figure 1). Figure 6b shows the impact of alternatively choosing the MH-60s. In this case, availability constraints force the activity to be performed in two waves of seven and two trips (hence the significantly longer duration). The utilization displays for both options (Figure 6c) make this clear. Other visual cues provide information about the constrainedness of the current planning search space. The user can recolor the vector display to indicate decision criticality (a function of how many resource assignment options remain). Similarly, the user can toggle the display to show the degree of temporal flexibility associated with different activities; Figure 6 shows the earliest execution interval of position-for-insertion.

Requirements and capabilities editing.

The user can edit activity attributes to alter the constraints. Figure 5 shows the resource requirements for position-for-insertion. The activity requires a light-transport capability, which maps to either of the aforementioned helicopter types. Through selection in this display, the user is free to adjust these requirements, specifying either alternative or additional capabilities or platforms that might be viable. Any adjustments result in a corresponding update of feasible options. Other timing constraints on activities, such as deadlines or synchronization points, can similarly be manipulated to expand or restrict the set of options.

The user can similarly manipulate resource attributes to vary basic problem assumptions. For example, the numbers of resources available and their initial locations can be revised

to evaluate an alternative deployment's consequences. Figure 7a shows an editing action to introduce the constraint that only one aircraft can be on the ground at a time at the **staging-area** location—the maximum-on-ground constraint; it had previously been specified as unconstrained. Figure 7b shows the consequence on **position-for-insertion**: the three 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 activities in the plan, the user can invoke an underlying automatic scheduler to generate a feasible assignment. This action constitutes a check of the plan's overall resource feasibility, respecting whatever decisions have previously been made.

What-if analysis. As implied by much of the previous discussion, most constraints and planning decisions are pliable through the interface to allow general exploration of the problem and solution space. Comirem provides a general ability to undo any user-level action or sequence of actions previously taken. Coupled with the capabilities described in this section, this mechanism provides a flexible, open-ended framework for what-if analysis.

Resource configuration

A drag-and-drop interface (see Figure 8) provides complementary capabilities for interactively configuring aggregate resources (task forces in our *Embassy-Rescue* scenario) from more primitive resource objects (in this case, individual personnel). As the user adds a resource to an aggregate under development, the system checks basic availability constraints. When the user assigns an aggregate resource to an activity (or to a larger thread of activities), this resource and all of its constituents become otherwise unavailable for allocation over the period that the activity or thread executes. So, just as when the user or the system allocates primitive resources, the potential addition of a resource to an aggregate resource or the potential assignment of an aggregate resource to a thread might be infeasible owing to prior commitments or, likewise, might require additional sequencing of competing activities in the plan.

Conflict analysis and user-oriented explanation

User acceptance of system decisions in

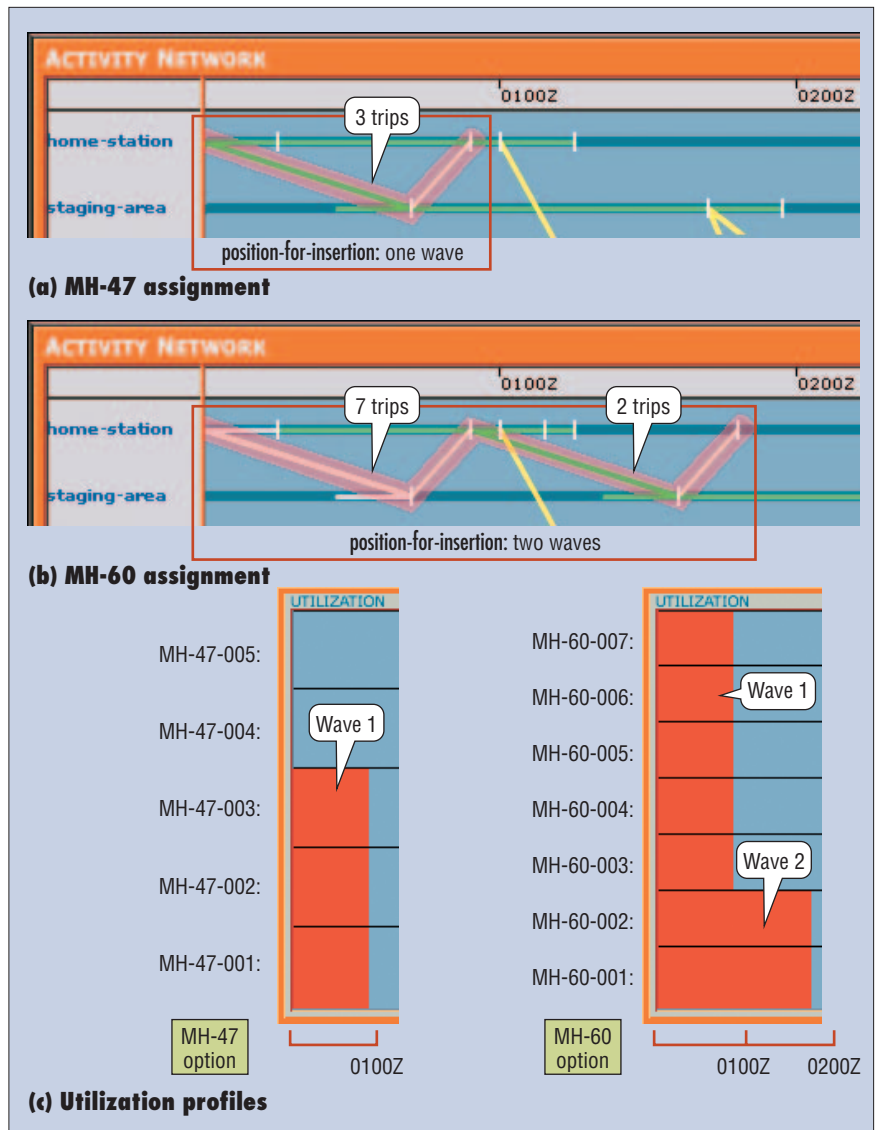


Figure 6. Two options for the position-for-insertion activity: (a) three helicopters make three trips or (b) seven helicopters make nine trips. (c) The utilization displays show the MH-60s' longer duration.

successful problem-solving episodes, and effective user involvement in circumstances where a system reaches a problem-solving impasse, both require that a system be comprehensible. This requirement, in turn, implies that a system be capable of explaining its decisions in user-understandable terms. Our previous work has argued for the use of a scheduling domain ontology as a basis for translating user problem specifications into internal system models.¹ Recent work with Comirem has focused on a complementary use of a domain ontology to compute user-oriented explanations of system decisions and conflicts.⁴ The central idea

is to use Comirem domain ontology knowledge to

- compute the set of user-relevant and manipulable constraints that form an explanation,
- identify various constraint-relaxation options that the user might consider to resolve a conflict, and
- provide content for generating user-understandable explanations of conflicts and possible resolution actions.

Figure 9 illustrates a duration conflict and shows Comirem's response in a situation

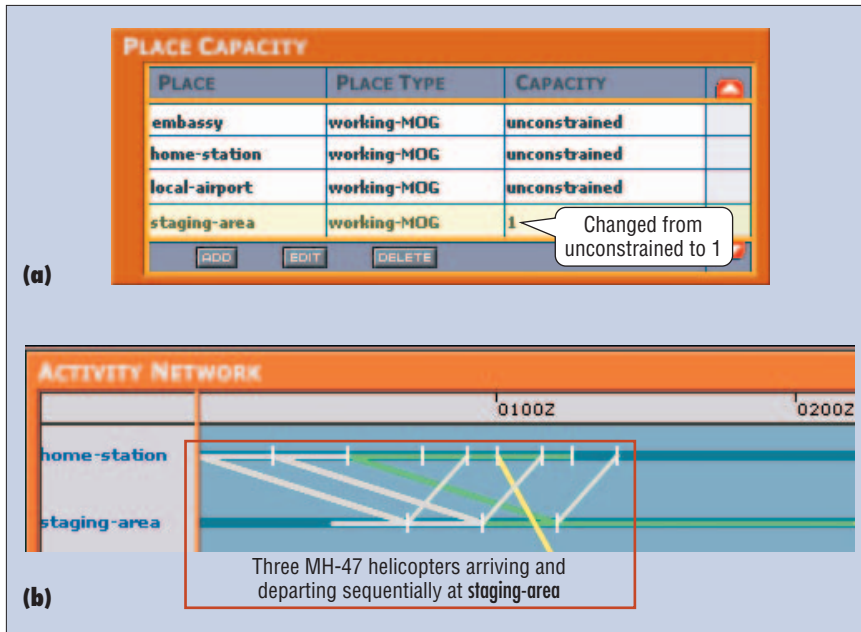


Figure 7. (a) Imposition of a maximum-on-ground capacity constraint on the staging-area location and (b) the resulting revised MH-47 assignment for the position-for-insertion activity.

where it has detected that completing an activity by its imposed deadline is impossible if the assigned resource is used. In this case, knowledge that relates temporal constraints in the underlying STP constraint network to various domain-level constructs (for example, activity and relation types) is used to filter the set of constraints the system finds to be in conflict by the temporal constraint propagator and to identify the subset that's user manipulable. Once the system identifies this subset of constraints, it uses knowledge associated with corresponding domain entities to propose possible constraint relaxation options. The user can then choose from this set of proposed options.

Principles of mixed-initiative planning and scheduling

The Comirem planner's design reflects several basic principles that we believe are fundamental to effective human-computer interaction in practical planning domains. (See the sidebar for related work in interactive planning.)

Adjustable decision-making autonomy

One key to effective practical planning is a decision-making process that allows the degree of automation to vary according to the problem-solving context and user preference. The Comirem planner lets the user 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; the user retains control over resource-assignment decisions. However, even in this largely manual mode of operation, the system frequently plays a quite substantial role in determining the consequences of all relevant constraints on user allocation decisions (for example, a move's overall duration, given that multiple trips are required and that aircrews must rest during the move). The user is free to make specific adjustments to an activity's resource requirements (such as adding another aircraft type to the set of possible allocation options) or to override specific use constraints (such as reducing computed duration constraints on the basis of knowledge of a strong tail wind) without anticipating interactions with all other constraints.

In other decision contexts, the user can delegate increased decision-making scope to the system. The user might choose to manip-

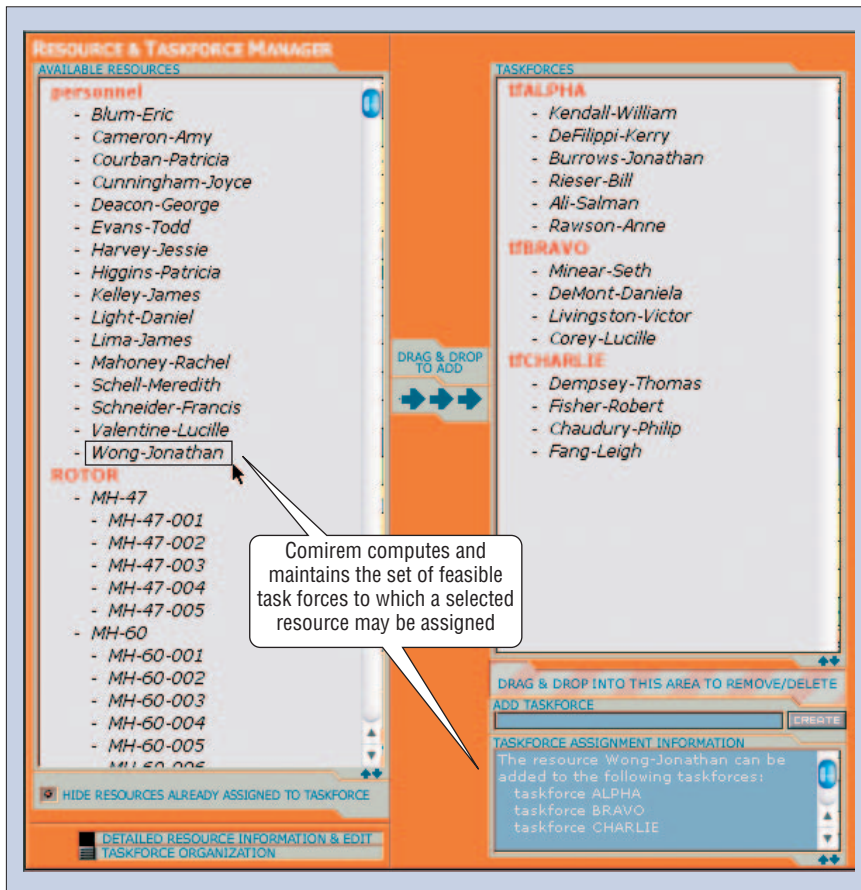


Figure 8. Users can configure aggregate resources through a drag-and-drop task force composition form.

Related Work in Interactive Planning

The Comirem effort descends from earlier work with the AMC Barrel Allocator, a tool for day-to-day allocation of aircraft and aircrews for the US Air Force that is now part of the operational-planning system in the Tanker/Airlift Control Center at the USAF Air Mobility Command.¹ Whereas this application focuses on large-scale problems (thousands of activities) with more narrowly circumscribed types of allocation constraints, Comirem focuses on smaller-scale problems (hundreds of activities) with more ill-structured types of allocation problems. From a system-design viewpoint, this focus has required a richer domain modeling and constraint management infrastructure.

Many other recent efforts in interactive planning have also started to break the mold of trying to integrate with batch-oriented, systematic planning technologies. TRIPS (Thinking, Reasoning, and Intelligent Problem Solving) adopts a similar opportunistic view of plan development and is designed for use in similar transportation-related planning domains.² Comirem appears to address a richer class of resource usage constraints, while TRIPS provides greater generality with respect to construction of activity networks. PASSAT (Plan Authoring System Based on Sketches, Advice, and Templates) has recently introduced the concept of plan sketches, which similarly enforce no strong commitment on the order in which planning

decisions must be made and let the user defer and relax various constraints as the plan is being assembled.³ PASSAT however, has no explicit model of time, and hence deals with a quite different class of planning problems. O-Plan2 is another system that has long emphasized an opportunistic user-system interaction model.⁴

References

1. M.A. Becker and S.F. Smith, "Mixed-Initiative Resource Management: The AMC Barrel Allocator," *Proc. 5th Int'l Conf. Artificial Intelligence Planning Systems (AIPS 2000)*, 2000, pp. 32–41.
2. G.M. Ferguson and J.F. Allen, "TRIPS: An Integrated Intelligent Problem-Solving Assistant," *Proc. 15th Nat'l Conf. Artificial Intelligence (AAAI 98)*, AAAI Press, 1998, pp. 567–573.
3. K.L. Myers et al., "PASSAT: A User-Centric Planning Framework," *Proc. 3rd Int'l NASA Workshop Planning and Scheduling for Space*, 2002; www.ai.sri.com/~passat/passat-nasa-wkshp-2002.pdf.
4. A. Tate, B. Drabble, and R. Kirby, "O-Plan2: An Open Architecture for Command, Planning and Control," *Intelligent Scheduling*, M. Zweben and M.S. Fox, eds., Morgan Kaufmann, 1994, pp. 213–240.

ulate constraints associated with larger plan fragments, such as threads, and request feasibility checking or autoscheduling in response to assess consequences. At the end of the automation continuum, the user can request the construction of a fully sourced plan. 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 practical planning is the system's ability to communicate elements of its internal models and solutions in user-comprehensible terms. Comirem accomplishes this through a combination of graphical visualization and model-based explanation.

As Figures 6 and 7 illustrate, visual displays compactly convey various decision implications, such as how many trips are required if this type of transportation resource is used, from what location will various resources be sourced, and what will be the overall duration of a particular move activity. They also convey potentially nonobvious allocation constraints, such as maximum-on-ground limitations.

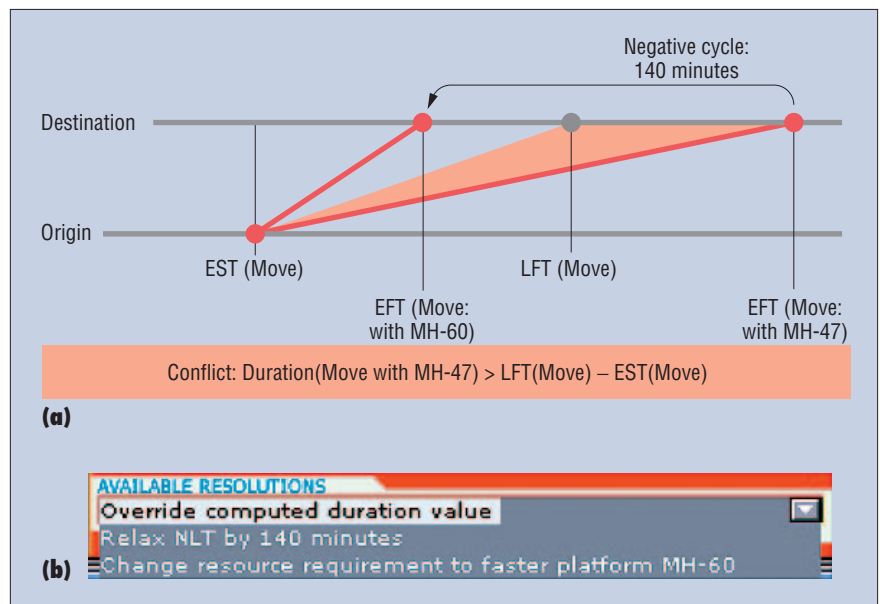


Figure 9. (a) A duration conflict summary, noting earliest and latest start and finish times, and (b) its corresponding Comirem-generated resolution options. The user can override the domain constraints on the basis of deeper strategic information, such as favorable weather; extend LFT(Move) by 140 minutes to account for the MH-47 duration constraint; or assign the faster MH-60 resource to Move.

In other advice-giving contexts, internal system representations are mapped to user-level actions using abstract domain models. In Figure 9, for example, Comirem's higher-level activity resource model lets the system

impose structure on the distance constraints involved in a cycle that's been detected in the temporal-constraint network. That is, the system characterizes the constraints as interdependent sets of activity durations, interac-

The Authors

Stephen F. Smith is a Research Professor at Carnegie Mellon University's Robotics Institute and the director of the institute's Intelligent Coordination and Logistics Laboratory. His research interests include planning and scheduling, constraint reasoning, mixed-initiative problem solving, and multi-agent systems. He received his PhD in computer science from the University of Pittsburgh. Contact him at the Robotics Inst., Carnegie Mellon Univ., 5000 Forbes Ave., Pittsburgh, PA 15213; sfs@cs.cmu.edu; www.cs.cmu.edu/~sfs.

David W. Hildum is a Project Scientist at Carnegie Mellon University's Robotics Institute and a member of the institute's Intelligent Coordination and Logistics Laboratory. His research interests include knowledge-based scheduling, configurable systems, and cooperative distributed problem solving. He received his PhD in computer science from the University of Massachusetts at Amherst and is a member of the AAAI. Contact him at the Robotics Inst., Carnegie Mellon Univ., 5000 Forbes Ave., Pittsburgh, PA 15213; hildum@cs.cmu.edu; www.ri.cmu.edu/people/hildum_david.html.

David R. Crimm is a Research Programmer at Carnegie Mellon University's Robotics Institute and a member of the institute's Intelligent Coordination and Logistics Laboratory. His research interests include planning and scheduling, mixed-initiative intelligent systems, and human-computer interaction. He received his BS in computer science from the University of Pittsburgh. Contact him at the Robotics Inst., Carnegie Mellon Univ., 5000 Forbes Ave., Pittsburgh, PA 15213; dcrimm@cs.cmu.edu; www.ri.cmu.edu/people/crimm_david.html.

Unlike black-box, batch-oriented solvers designed to automate planning and scheduling, Comirem is an intelligent planning and scheduling form: an interactive device oriented instead toward amplifying user-driven decision making. Our current research aims at extending Comirem in several directions. One area of interest is in enhancing Comirem's planning capabilities. On one hand, we're interested in providing direct interactive support for creating and manipulating activities and activity networks. On the other, we're investigating extensions that allow incorporation of explicit goal-directed reasoning about the resource support plans required to enable various resource assignments.⁵ A second general area of interest is the design of user-directed strategies for flexible-times scheduling. ■

Acknowledgments

We thank Doug Dyer for his insight and feedback throughout the Active Templates program, which contributed greatly to the design and development of several key features of Comirem. We also thank Gabriella Cortellessa and Christian Ohler for their contributions to the conflict analysis mechanism. Our research was partly sponsored by DARPA under contract F30602-00-2-0503 and by the CMU Robotics Institute.

References

1. S.F. Smith and M.A. Becker, "An Ontology for Constructing Scheduling Systems," *Ontological Engineering: Papers from the AAAI Spring Symp. Ontological Eng.*, tech. report SS-97-06, AAAI Press, 1997, pp. 120–129.
2. T.E. Morton and D.W. Pentico, *Heuristic Scheduling Systems*, John Wiley & Sons, 1993.
3. R. Dechter, I. Meiri, and J. Pearl, "Temporal Constraint Networks," *Artificial Intelligence*, vol. 49, nos. 1–3, 1991, pp. 61–95.
4. S.F. Smith et al., "Using a Scheduling Domain Ontology to Compute User-Oriented Explanations," *Proc. 16th European Conf. Artificial Intelligence (ECAI'04) Workshop Planning and Scheduling: Bridging Theory to Practice*, 2004; <http://galahad.plg.inf.uc3m.es/~dborrajoe/ecai04-workshop/papers/SmithEtAl.pdf>.
5. S.F. Smith and T. Zimmerman, "Planning Tactics within Scheduling Problems," *Proc. 14th Int'l Conf. Automated Planning and Scheduling (ICAPS-04) Workshop Integrating Planning into Scheduling*, 2004, pp. 83–90.

tivity precedence relations, and so on. 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 practical planning is a problem-solving process that promotes incremental decision making. In complex, ill-structured planning domains, all relevant constraints generally aren't known at the outset, and those that are can frequently change as planning and execution progresses. 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 with 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 shouldn't arbitrarily result in wholesale changes to the plan. The system should instead act to preserve prior decisions wherever possible and meaningful. We designed the Comirem planning framework with an incremental, change-based problem-solving process in mind. The system lets the user explicitly manage the scope of allowable change in specific decision contexts.

**QUESTIONS?
COMMENTS?**
**IEEE Intelligent Systems wants to
hear from you!**
EMAIL: isystems@computer.org