# Toward the Development of Flexible Mixed-Initiative Scheduling Tools*

Stephen F. Smith and Ora Lassila
The Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213
sfs@cs.cmu.edu / ora@cs.cmu.edu
(412) 268-8811 / (412) 268-2357

March 15, 1994

**Abstract**

In this paper, we discuss work aimed at the development of interactive decision-support tools for complex, large-scale scheduling applications. Our approach is grounded on three basic premises: (1) that system organization and decision-support "services" should directly reflect the inherently reactive nature of decision-making in complex scheduling environments, (2) that diversity in the character and requirements of various user tasks will invariably require different specialized scheduling services in different decision-making contexts, and (3) that problem scale and complexity will necessitate user interaction at aggregate, task-oriented levels. We describe DITOPS, a transportation scheduling tool which integrates a hierarchical modeling infra-structure and reactive scheduling methodology with graphical schedule visualization and manipulation capabilities to provide a flexible interactive environment for construction and management of transportation schedules.

## 1 Introduction

Our specific focus in this paper is the development of interactive tools for military crisis-action transportation scheduling. The requirements for effective decision support in transportation scheduling are not unlike the requirements of any complex, large scale scheduling domain. We can make three broad observations:

- Scheduling can rarely be treated strictly as an optimization problem. Optimizing capabilities are clearly important, but the overarching goal of producing a good executable *behavior* requires attention to the ongoing scheduling process. Schedule construction in practice tends to be a dynamic reactive process. An initial schedule is built, problematic or unsatisfactory aspects of the result are identified, requirements are relaxed or strengthened (typically through negotiation with other planning agents), schedule modifications are made and so

*The research reported in this paper has been supported in part by the Advanced Projects Research Agency under contract F30602-90-C-0119 and the CMU Robotics Institute.

on. Throughout this process, the current schedule provides the planner(s) with an important nominal reference for identifying, specifying and communicating changes, and there is considerable pragmatic value in an ability to maintain continuity (or localize change) in the solutions that are produced across iterations. Such an ability allows the planner to impose structure on an otherwise overwhelmingly complex search process and to converge in a more focused fashion to an acceptable overall solution. Likewise, as unexpected events occur in the execution environment, it is important to preserve continuity in domain activity while making those schedule changes necessary to restore feasibility and insure continued attendance to overall mission performance objectives. Both of these aspects of the scheduling process place a premium on incremental, reactive scheduling capabilities.

- There is generally diversity in the character of the user tasks that must be carried out at different stages of the scheduling process. For example, in transportation scheduling, planners are variably concerned with tasks relating to resource requirements determination and apportionment, transportation feasibility analysis, detailed resource allocation and reactive schedule management at different stages of the process. Though commonality can be found in overall solution and decision-support requirements, each of these tasks often implies different types of dominating constraints and objective criteria. Accordingly, we can expect that the heuristics and solution procedures required for effective decision-support will likewise vary according to the user decision-making context.

- In large-scale domains, it is unreasonable to expect planners to comprehend schedules and interact with a scheduling system at the level of the system solution model. There are too many decisions and details, most of which are unimportant from the standpoint of user tasks and goals, and the complexity of decision-making at this level is overwhelming. Effective decision-support tools must bridge the gap between user and system models of schedules and decision-making, promoting inspection, analysis and manipulation of schedules at aggregate, task-oriented levels and "managing the details" in accordance with user goals and intentions.

The decision support requirements implied by these observations are at odds with most existing scheduling software systems. Current transportation scheduling tools, for example, are typically organized as *batch-oriented* solution generators [7]. Problem input parameters and constraints are specified, the system is run to produce a schedule, and the result is examined for acceptability. In reacting to either unsatisfactory properties of the generated schedule (e.g., unacceptable late closures) or changing circumstances in the world (e.g., the unexpected loss of port capacity), the human planner is forced to hypothesize how changes to system inputs might affect the solution that is produced, and has no control over what aspects of the solution actually will change when the system is rerun with specified input parameter changes. Consequently, there can be considerable "thrashing" in the solutions generated from run to run, and it is quite cumbersome to enforce commitment to specific aspects of any given solution.

Inflexibility vis a vis user decision-making requirements and tasks is a second shortcoming of current transportation scheduling tools. Commonly used simulation-based tools, for example, are configured to support transportation feasibility analysis but are of little use in addressing resource requirements analysis tasks [7]. (It is difficult to run a simulation backwards, which is essentially what is required). Though planning leverage and efficiency could be gained from an ability to address these tasks in an integrated fashion, users are forced to consider these decisions sequentially in isolation.

In this paper, we describe DITOPS, a prototype tool for generation, analysis and revision of large-scale transportation schedules. DITOPS integrates a hierarchical modeling infra-structure and a reactive, constraint-based scheduling methodology with graphical schedule visualization and manipulation capabilities to provide a decision-support environment that more directly matches the characteristics and requirements of complex scheduling applications. DITOPS promotes a mixed-initiative scheduling paradigm similar in spirit to current day spreadsheet programs: sets of scheduling decisions and solution constraints are interactively manipulated by the user at (typically aggregate) levels consistent with user-task models. At each step, the system applies appropriate (re)scheduling procedures to impose the changes specified by the user (i.e., manages the details), and provides localized consequences of each change. System look-ahead analysis and scheduling techniques further provide a basis for functionality that transcends the spreadsheet decision-support analogy, enabling user support in identifying principal causes of observed solution deficiencies (e.g., resource bottlenecks), in analyzing various decision-making options (e.g., apportionment of additional resources), and in assessing solution sensitivity to various executional circumstances.

The remainder of the paper is organized as follows. In Section 2, we discuss the appropriateness of constraint-based models for mixed-initiative scheduling and illustrate our vision. In Section 3, we summarize the approach taken in DITOPS and indicate its mixed-initiative scheduling capabilities. Finally, in Section 4, we assess characteristics of the current DITOPS model and identify our research directions.

## 2 Constraint-Based Models for Mixed-Initiative Scheduling

Constraint-based frameworks provide a model well-suited to the reactive decision-making requirements of practical scheduling domains. In broadest generality, this model defines a problem solving organization that distinguishes two components: a *decision-making* component, responsible for making choices among alternative scheduling decisions and retracting those that have since proved undesirable, and a *constraint management* component, whose role is to propagate the consequences of decisions and incrementally maintain a representation of the current set of feasible solutions (detecting inconsistent solution states when they arise). Schedule construction, revision, and improvement proceed iteratively within a basic *decide and commit* cycle. Most project management tools and several interactive scheduling systems[2, 4] are direct implementations of this model, with the user as the decision-making component.

Of more interest and importance in most scheduling environments, are extensions of this basic model that off-load more decision-making responsibility to the system. The "spreadsheet" style of interaction provides a natural framework for "what-if" experimentation and iterative solution development, but, as indicated above, interaction and decision-support is typically required at a much more aggregate decision-making granularity. One extension to the basic constraint-based model that preserves the "direct manipulation" style of interaction is one that refines the *decide* step of the cycle into a two-step process of formulating and executing decision-making actions. Within this model, *action formulation* is concerned with isolating a particular subproblem and *action execution* results in solution of this subproblem. The user, who interacts with system processes by formulating actions, is able to manipulate solutions in terms of higher-level and more comprehensible task-oriented perspectives (e.g., reschedule force module x's movements to arrive by C15), with the system's subproblem solution procedures providing, from the user's viewpoint, an amplification of deductive constraint management functionality; a more sophisticated (and typically heuristic) "propagation of effects".

In the simplest case, there is a direct mapping between user specifiable actions and system solution procedures, in which case the user holds complete responsibility for action formulation. The COMPASS interactive scheduling framework [1], for example, is organized in this fashion. In our view, however, the user should be able to operate in terms of much more ill-structured action specifications (e.g., relax lift capacity constraints and reschedule late movements while minimizing additional lift asset requirements, resolve the conflicts introduced into the schedule by the loss of capacity at port1). This, more flexible viewpoint implies that the system must participate actively in structuring the appropriate subproblem to solve (e.g., in determining the appropriate scope of change, in translating objectives and preferences into appropriate heuristic revision procedures), and that subproblem solution may require coordinated execution of several solution procedures.

The decision-support capabilities we envision are illustrated by the the following interactive "TPFDD" generation scenario:

1. **Evaluate initial schedule**

   Starting with a set of deployment requirements and initial estimates as to apportioned transportation resources, a USTRANSCOM planner invokes the system to generate an initial schedule that satisfies stated resource capacity and utilization constraints and minimizes late closures. Upon inspection of the results too many late closures are discovered.

2. **Identify principal bottleneck**

   System analysis of the constraints contributing to these results indicates the principal source of lateness to be insufficient throughput capacity at the designated final port of debarkation, POD1.

3. **Propose a solution**

   The planner responds to this information by introducing a second port of debarkation, POD2, into the scenario and indicating that POD1 arrivals be rescheduled to exploit the additional capacity provided by POD2. The number of late closures is substantially reduced by this action.

4. **Identify next bottleneck**

   Analysis of the resulting schedule now indicates that the remaining late closures stem from inadequate sea lift capacity during week 2 of the deployment.

5. **Engage in clarification dialog**

   Several "what-if" actions are carried out to determine additional resource requirements and to clarify alternative options for eliminating late closures:

   - Late movements are rescheduled with the specification that lift capacity constraints may be relaxed (i.e., additional assets may be added), which indicates that two additional transports are needed to meet all specified arrival dates.
   - The sea mode assignment associated with the remaining late arrivals is eliminated to determine whether excess air lift capacity can be utilized to resolve the problem. Results of this action indicate that only 50% of the late cargo can be accommodated by available air capacity (due in part to capacity limitations and in part to the cargo carrying restrictions of available aircraft types).

6. **Locate additional resources**

   At this point, the user decides that acquisition of additional sea assets is the best option and proceeds to obtain use of two commercial transports during the 2nd week of the mission.

7. **Propose a solution**

   The additional lift capacity is added to the model and late movements are rescheduled to complete by their requested arrival dates.

# 3 The DITOPS Transportation Scheduler

Our vision of future transportation scheduling tools is implemented in the DITOPS system. DITOPS is an advanced tool for generation, analysis and revision of crisis-action logistics schedules. The system incorporates concepts of constraint-directed scheduling developed within the OPIS manufacturing scheduling system[8], together with extensions to address the specific characteristics of transportation scheduling problems. Using DITOPS, we have demonstrated an ability to efficiently generate higher quality schedules than conventionally used simulation approaches on large-scale deployment scheduling problems while simultaneously satisfying a wider range of deployment constraints [10]. DITOPS is implemented using object-oriented representation and programming techniques, providing an extensible modeling and scheduling framework that enables straightforward system customization to account for the principal constraints and objectives of different scheduling domains [9].

Of interest in this paper, however, are the complementary capabilities that DITOPS provides for incrementally revising transportation schedules in response to changed constraints, and their application in interactive decision-support contexts. These capabilities allow schedules to be reactively updated to reflect unexpected events that occur during schedule execution (e.g., the closing of a port due to bad weather) while preserving continuity in scheduled activities wherever feasible. They also allow for efficient, controlled convergence to acceptable or improved solutions during advanced planning; as adjustments to various scheduling constraints and preferences are made by human planners in response to observed solution deficiencies (e.g., too many late arrivals), DITOPS can provide immediate, localized feedback of the effects of these changes on the current schedule. In the subsections below, we summarize the system's underlying reactive scheduling framework, discuss implications with respect to coordinating user/system activity, and describe the mixed-initiative scheduling capabilities provided in the current implementation.

## 3.1 Modeling and Scheduling Infra-structure

The DITOPS scheduler operates with respect to hierarchical models of transportation processes and resources. A DITOPS model of a given application domain is composed from an extensible set of pre-defined primitives, which provide object structures (i.e., a class library) for specifying various transportation scheduling constraints and associating an appropriate operational semantics. A transportation scheduling model is specified as a relational configuration of five basic types of "building blocks": *resources*, *activities*, *move requirements*, *shipments*, and *missions*. Resources represent the various assets, equipment, and facilities required to carry out transportation activities (e.g., planes, ships, ports); specializations are defined to model a wide range of resource types (atomic, composite, mobile, stationary) and allocation constraints (capacity limits, cargo compatibility restrictions, availability constraints). Atomic domain resources can be grouped into aggregate resources (e.g, individual C-5s into a C-5-fleet into an air-fleet) to provide descriptions of allocation constraints at multiple levels of specificity. Activities represent the constituent actions of transportation processes and plans (e.g., transporting, loading, unloading, processing). Like resources, activities can be composed into aggregate structures to provide multiple levels of description. Move requirements represent the input requests that constitute the scheduling problem. Shipments represent the actual cargo entities (or "packages") that are associated with individual transport activities (e.g., 1000 CBarrels of POL). Mission descriptions provide a specification of a plan template (or basic plan class) for instantiating the transport plans (i.e., activity networks) that must be scheduled.

At the core of DITOPS is an incremental, reactive approach to generating and revising schedules, based on two inter-related concepts [11]:

- *constraint-based focus of attention* - analysis of the structure of current problem constraints provides a useful basis for identifying critical decision tradeoffs and flexibilities, and structuring the scheduling process, and
- *multi-perspective scheduling* - this guidance can be effectively exploited through selective use of a set of complementary revision procedures.

In DITOPS, alternative decision-making procedures are specifically designed to provide differential optimization and conflict resolution capabilities. Local search methods are defined for both "resource" and "movement" centered scheduling, providing capabilities, respectively, for manipulating (i.e., revising or extending) the schedules associated with particular sets of resources (e.g., the cargo ship fleet) or particular sets of temporally related movements (e.g., the movements associated with a particular force module). By virtue of search orientation, each of these methods emphasizes specific optimization biases; resource scheduling promotes efficient use of available transport capacity while attempting to minimize the tardiness of scheduled movements. Movement scheduling, alternatively, promotes enforcement of arrival constraints and efficient synchronization of dependent movements, while attempting to minimize asset capacity requirements. Both of these methods share common search infra-structure that allows incorporation of additional allocation preferences. A number of more specialized revision procedures are also defined, providing additional capabilities to shift the scheduled interval of scheduled "trips", swap batches associated with particular transportation assets, and balance load to exploit increases in port capacity.

Scheduling is cast as an iterative process of subproblem formulation and subproblem solution. In brief, this iterative process proceeds according the following control cycle:

1. *Constraint propagation* - At the start of each decision-making cycle, the consequences of newly introduced constraints (originating either from external sources or from decisions made on the previous cycle) are derived. Resulting constraint conflicts, as well as improvement opportunities (i.e. complementary situations where constraints have loosened) are collected as potential modification focal points, and a particular focal point is selected for consideration on this cycle.

2. *Look-ahead analysis* - Relative to the selected focal point, constraint analysis methods are applied to estimate the critical tradeoffs and opportunities for solution revision (or extension) implied by the current solution state.

3. *Subproblem formulation* - Analysis results are then used to formulate a specific scheduling task, which isolates a particular set of decision variables to focus on (i.e., assign or revise) and designates a particular decision-making procedure to apply to this set of decisions. In the absence of further external revision guidance or constraints, this is accomplished through application of a set of control heuristics, which map the important optimization (or reoptimization) needs and opportunities implied by analysis results to the respective capabilities of alternative scheduling procedures.

4. *Subproblem solution* - Once the appropriate revision task has been formulated, it is carried out, changes are introduced into the current solution constraint network, and the cycle repeats.

The DITOPS scheduling framework thus provides a control infra-structure and heuristic model for reactively responding to changes in problem constraints. In situations where changed constraints undermine current decisions (i.e., result in inconsistencies or conflicts), response is aimed at restoring solution feasibility; in situations where constraints are loosened, response seeks to exploit this added flexibility.

From a mixed-initiative scheduling perspective, this framework promotes a default style of interaction grounded in user manipulation of problem constraints (e.g., resource capacity and availability, activity deadlines, etc.) and system determination of consequences (using internal strategies for reconciling conflict resolution and solution improvement possibilities with the desire to minimize schedule disruption). Though this division of responsibility may match user decision-making goals in some cases, it will more frequently be the case that realization of system activity consistent with user expectations will necessitate greater user involvement in the subproblem formulation process. Within DITOPS, this subproblem formulation activity is opened up to the user through a graphical interface that emphasizes visualization and manipulation of the schedule from aggregate perspectives, using the hierarchical descriptions of transportation processes and required resources defined in the underlying domain model. We first consider the types of revision constraints and biases relevant to coordinating user/system activity, and then discuss the system's current user interface and interactive scheduling capabilities.

## 3.2 Coordinating User/System Activity

Generalizing from the subproblem formulation framework defined in the original OPIS scheduling architecture, we can identify several types of information that must be communicated to manage user actions:

- *action focus* - The action focus indicates the specific set of activities (or more precisely decision variables) for which change is mandated. Depending on the specific solution constraints manipulated by the user, the focus may or may not be directly inferrable. If, for example, a tightening of resource capacity constraints results in the introduction of conflicts (e.g., a ship is designated as unavailable over some interval during which it is scheduled to be transporting cargo), then the set of activities now in conflict constitute the action focus. However, if capacity constraints are loosened (e.g., an additional ship is made available), then the action focus remains ambiguous. In this case, the user has a manipulation goal (e.g., reduce late arrivals) that is not directly communicated by changed solution constraints, and appropriate system action requires identification of a target activity set (e.g., the set of transport activities currently scheduled to arrive late).

- *action scope* - The action focus only places lower bounds on the extent of change to the current solution. Managing the overall scope of change is a second central issue in aligning system activity with user manipulation goals. Most generally, the challenge here is one of maintaining sufficient stability in the schedule to meet user expectations and maintain decision-making continuity while at the same time exploiting current (re)optimization opportunities. Left to its own devices, the heuristic strategy currently implemented in DITOPS (descended from the original OPIS strategy[11]) will extend the scope of solution change to improve the quality of the schedule in circumstances where this tradeoff presents itself[1]. However, we assume that several types of additional information may be communicated by the user to further elaborate manipulation goals and intentions, and alter these default revision assumptions:
  - *revision dimensions* - The description of an activity identifies several types of decision variables. In the current DITOPS transportation model, for example, a scheduled transport activity includes assignments of execution start (end) times, a supporting transportation asset, an origin (POE) and a destination (POD). Each of these assignments reflect different dimensions along which change might be desired. If the user has loosened the capacity constraints associated with lift assets (as in our example above), then asset reassignment is obviously a dimension along which change is to be considered. Suppose alternatively, that the user has indicated the unavailability of a port for some period (e.g., due to weather conditions). One objective might be to examine the direct impact on movement arrivals (i.e., the delays that will be incurred given the assets "frozen" in the disabled port, the inability to move cargo in/out of the port for some period, etc.). In this case, system revision of the schedule should avoid any reassignment of lift assets. Consider finally a user change which increases destination port capacity through introduction of an additional POD. Though it may be reasonable to generally assume that activity origins and destinations are fixed variables, in this case, user change would be meaningless if reassignment of PODs were not included within the scope of allowable change.
  - *revision boundaries* - It may also be desirable to localize change to specific "regions" of the schedule, or alternatively to exclude some sets of decisions as fixed.[2] In some cases, direct manipulation of solution constraints is sufficient to infer appropriate change restrictions (e.g., recognition that "in-process" activities cannot have their resources reassigned). However, in many other cases, localization constraints are instead a function of past user actions, current user goals or both, and must be explicitly communicated. For example, the current schedule of a particular set of transportation assets may reflect previously negotiated agreements with other planning agents which enable their interleaved use within another ongoing (but independently managed) crisis operation. The user may prefer to "lock in" these decisions to avoid further complication. Likewise, in examining the impact of augmenting transport fleet capacity to reduce lateness, it may be desirable to constrain reassignment of assets to a particular subset of the overall fleet.
  - *relaxable constraints* - Generally speaking, there is no guarantee that a feasible schedule can be obtained if all solution constraints are interpreted as non-relaxable, and de-

[1] Other reactive scheduling approaches have adopted an opposite stance (e.g., [6, 12]); either rescheduling bias can be in conflict with user manipulation goals and lead to problematic results in specific decision-making contexts.

[2] We define a region of the schedule generally to be any set of scheduled activities, but assume that user intentions are generally structurable in terms of aggregate model structures (e.g., the activities associated with specific resource groups in the schedule) and temporal partitions of the scheduling horizon.

termination of feasibility is a computationally intractable problem. Moreover, even if feasibility were to be heuristically assessed through the use of approximate solution procedures, a style of interaction where a user specifies constraints, receives a "not feasible" response, alters constraints, etc. is quite information poor in the negative case. The reactive scheduling procedures in DITOPS alternatively assume that solution constraints are in fact relaxable along at least one solution dimension (e.g., deadlines), and attempt to minimize the extent of constraint relaxation through incorporation of appropriate objective criteria (e.g., minimize tardiness). At the same time, different user manipulation goals may imply relaxation along different dimensions. Analysis of transportation feasibility implies relaxation of required arrival dates if necessary. Determination of lift resource requirements, alternatively, implies that arrival dates be respected and capacity constraints be relaxed instead if necessary.

- *scheduling biases* - Information relating to action scope imposes constraints on the overall set of decisions variables that may be reconsidered in responding to user changes, and on the sets of allowable values for each variable. Within the space of alternatives defined by these constraints, the user will typically have preferences. The just mentioned association of an objective criterion relating to relaxable constraints is one basic example of injected search bias. There may clearly be others, following in many cases from external user knowledge of domain characteristics. For example, a bias to minimize splitting of movement activities across multiple transport resources may be desirable, given that this introduces the additional complication of reconstituting the original cargo at the destination. Likewise, the use of faster ships might be preferable if hostility is expected at arrival destinations. Since alternative scheduling biases will often be conflicting in nature, the introduction of multiple biases must include prioritization information.[3]

### 3.3 Interactive Transportation Scheduling

Interaction between a user and the current DITOPS scheduler occurs through a graphical direct-manipulation interface which emphasizes visualization and manipulation of schedules in terms of resource capacity utilization over time (see Figure 1). Based on the underlying hierarchical resource model, the user can create resource capacity views at various levels of aggregation. This allows the user to examine either individual craft assets, fleets or ports. The resource capacity views support zooming and scrolling for localizing attention on particular resources and/or regions of the overall schedule horizon. The user can select temporal intervals by "boxing" the area of interest with the mouse. Any querying and manipulation of schedules and solution constraints is based on these uniform time selections; once a selection has been made a variety of actions is possible through a menu associated with the resource in question.

Given a selected interval of time, the user may choose to examine properties of the delineated portion of the resource schedule. If the resource is an individual craft asset, for example, the

[3] In DITOPS, preferences are expressed as utility functions over the domains of specific decision variables, which are combined using associated "importance" weights into an overall objective function. This provides a flexible basis for altering search bias but is not immune to the standard problems associated with search objective functions (e.g., getting the weights right). In some cases, the use of scoping constraints can provide a more direct basis for achieving the desired tradeoff. For example, the above preference for faster ships might instead be incorporated by first restricting scheduling attention to this set of resources, identifying the resulting solution deficiencies (e.g., unacceptably late arrivals) and then framing a more selective reassignment of activities to slower moving assets.

transport activities supported by scheduled trips are accessible. At aggregate resource levels, graphical displays of various properties of the solution can be retrieved (e.g., movement closure profiles, accumulated cargo tonnage over time[4]). This provides a basis for identification of solution deficiencies.

User manipulation of problem constraints and schedules also centers around a selected resource profile interval. A transport or port resource can be made unavailable over a selected interval. In this case, any inconsistencies in the schedule that result are highlighted. Conversely, resource capacity of a given fleet can be increased for a specified interval by moving to the appropriate aggregate resource display (this translates to adding craft to the fleet). As indicated earlier, such a "relaxation" of capacity constraints should generally be accompanied by an indication of the action focus and scope (reflecting the specific rescheduling goal that motivates the change). Within the current implementation, only fixed choices relating to activities that are currently late and resource usage restrictions are available for narrowing system focus and scope. The "current time" indicator at the top of the resource displays can be moved along the schedule horizon to simulate states during the execution of the schedule. Default rescheduling biases are adjustable through a "slider" display which represents the relative importance to be attributed to each system known preference. In imposing any given change to the current schedule, there is no obligation to the user to provide additional revision constraints and guidance; generally speaking, user decisions along these lines are considered to be defaults until they are changed.

Overall system activity is managed through a "control panel" (upper left corner of Figure 1), which provides capabilities for creating various displays, loading scenario descriptions and deployment problems (sets of move requirements), saving and reloading generated schedules, and adjusting global system parameters and preferences (e.g., level of scheduling precision, automatic or selectable system response to changes, etc.).

## 4 Current Directions

There are several aspects of the DITOPS interactive scheduling framework that remain underdeveloped in the current implementation. One important issue concerns flexibility in structuring user interaction. Schedule visualization and manipulation is currently structured more or less exclusively by means of the hierarchical resource models defined in the underlying domain model. Though this interaction basis is clearly an essential one, there are orthogonal dimensions of aggregation that more directly isolate specific properties of the current solution and enable more direct specification of actions aimed at manipulating these properties. For example, aggregate activity-centered structures (e.g., the transport activities associated with a specific force module) provide an obvious basis for manipulating temporal constraints. Such capabilities are not yet incorporated in the current interface, and this is one focus of our current efforts. One short-term goal is to apply the scheduler in a specific operational environment, to obtain direct user input and feedback as to the usefulness of different aggregate structures and modeling assumptions, and to better understand user manipulation requirements.

More generally, however, we believe that reliance on pre-defined hierarchical structures will ultimately prove confining as a sole basis for structuring presentation of solutions and user formulation of actions. User analysis requirements depend heavily on the task at hand and the decision-making

[4] In part, these capabilities draw on the SciGraph package.

state, and can never be fully anticipated in advance. Similarly, user manipulation goals and expectations will evolve as decision-making proceeds, and appropriate rescheduling focal points and scoping constraints are likely to become more and more idiosyncratic. To accommodate such anticipated user needs, we are currently exploring integration of our interactive scheduling framework with a data exploration and visualization tool that enables dynamic formation of aggregate structures [3].

**Acknowledgements.** The current DITOPS scheduler is the collective result of the efforts of several individuals, including Gilad Amiri, Marcel Becker, Dina Berkowitz, Casper Cheng, Jyi-Shane Liu, Ali Safavi, Katia Sycara, and Chris Young.

## References

[1] Fox, B., "Compass 2.0 User's Guide", Planning and Scheduling Group, McDonnell Douglas Space Systems Co., December 1992.

[2] Elleby, P., H.E. Fargher, and T.R. Addis, "Reactive Constraint-Based Job-Shop Scheduling", in *Expert Systems and Intelligent Manufacturing*, M.D. Oliff (ed.), North-Holland, NY, 1988.

[3] Goldstein, J. and S.F. Roth, "Using Aggregation and Dynamic Queries for Exploring Large Data Sets", to appear in *Proceedings CHI'94: Human Factors in Computing Systems*, Boston, MA, April, 1994.

[4] Meng, C.C. and M.Sullivan, "LOGOS: a Constraint-Directed Reasoning Shell for Operations Management", *IEEE Expert*, 6(1), 1991.

[5] Ow, P.S., S.F. Smith and A. Thiriez, "Reactive Plan Revision", *Proceedings AAAI-88*, St. Paul, MN, August, 1988.

[6] Safavi, A. and S.F. Smith, "A Heuristic Search Approach to Planning and Scheduling of Software Projects", in *Advances in Artificial Intelligence: Natural Language, and Knowledge-Based Systems*, (ed.) M. Golumbic, Springer-Verlag Publishers, 1990.

[7] Schank, J., M. Mattock, G. Sumner, I. Greenberg, J. Rothenberg and J.P Stucker, "A Review of Strategic Mobility Models and Analysis", Rand Corp. Report Number R-3926-JS, 1991.

[8] Smith, S.F., P.S. Ow, N. Muscettola, J.Y. Potvin, and D. Matthys, "An Integrated Framework for Generating and Revising Factory Schedules", *Journal of the Operational Research Society*, 41(6), June, 1990.

[9] Smith, S.F. and O. Lassila, "Configurable Systems for Reactive Production Management", *Proc. IFIP TC 5 / WG 5.7 Int. Workshop on Knowledge-Based Reactive Scheduling*, Athens, Greece, Oct. 1993.

[10] Smith, S.F. and K.P. Sycara, "A Constraint-Based Framework for Multi-Level Transportation Scheduling", CMU Robotics Institute Technical Report CMU-RI-TR-93-17, December, 1993.

[11] Smith, S.F.,"OPIS: A Methodology and Architecture for Reactive Scheduling", in *Intelligent Scheduling*, (eds. M. Fox and M. Zweben), Morgan Kaufmann Publishers, to appear, 1994.

[12] Zweben, Monte, Eugene Davis and Michael Deale, "Iterative Repair for Scheduling and Rescheduling", NASA Ames Technical Report, 1991.

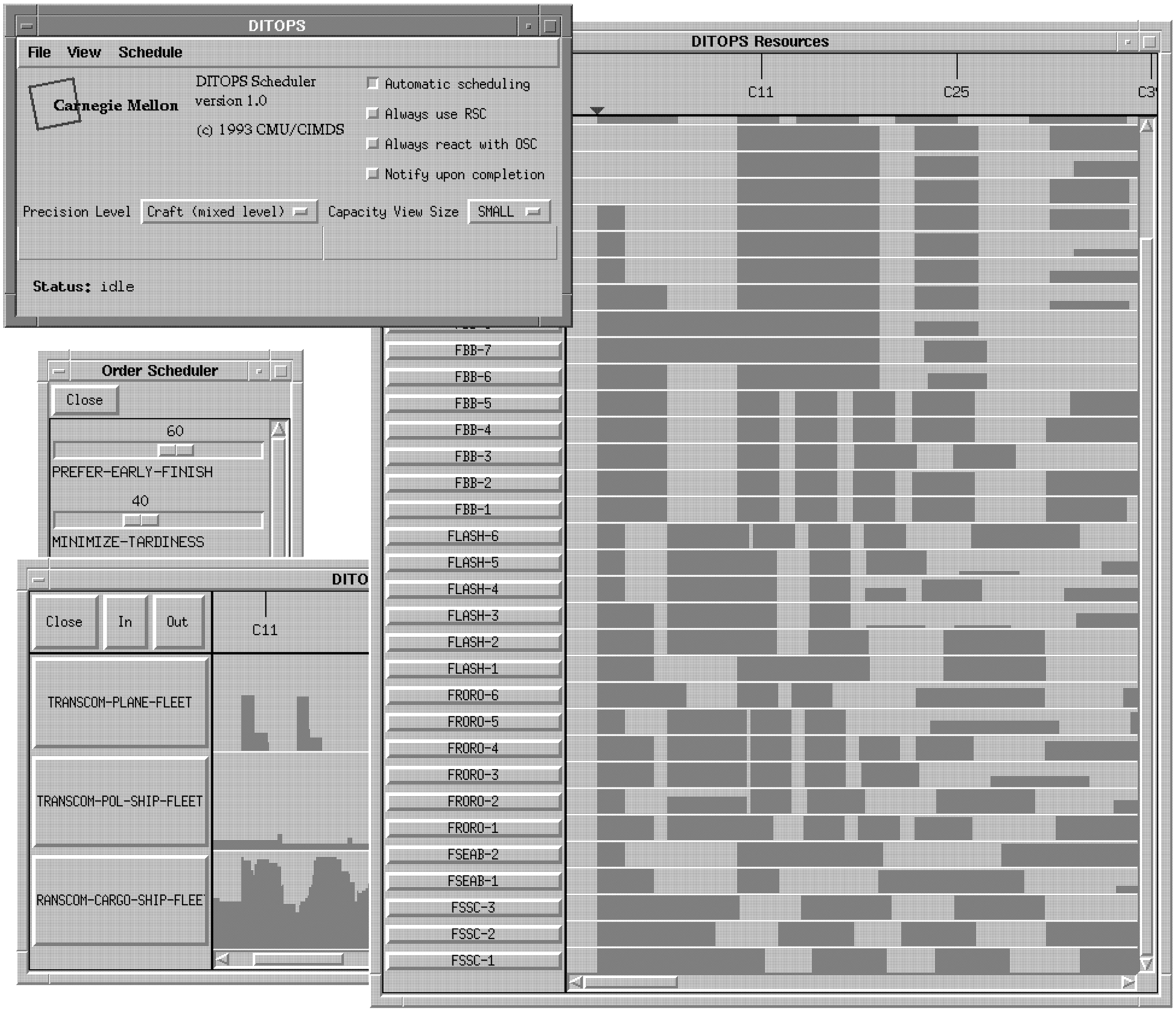


Figure 1: The DITOPS User Interface