Flexible Coordination in Resource-Constrained Domains

Stephen F. Smith and Katia P. Sycara

The Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213

CMU-RI-TR-93-17

December, 1993

Abstract

In this report, we summarize the research performed under Advanced Research Projects Agency (ARPA) contract F30602-90-C-0119, Flexible Coordination in Resource-Constrained Multi-Agent Domains. The broad goal of this research, which was carried out as part of ARPA/Rome Laboratories Planning Initiative (PI), has been to investigate the use of constraint-based scheduling frameworks and techniques as a basis for more accurate and more flexible decision support at various stages of the military crisis-action planning, deployment and employment process. This work has led to development of a transportation scheduling system called DITOPS, which provides advanced capabilities for construction, analysis and revision of large-scale deployment schedules.

A version of this report also appears as Rome Laboratories Technical Report RL-TR-94-95.

1 Introduction and Overview

In this report, we summarize the research performed under Advanced Research Projects Agency (ARPA) contract F30602-90-C-0119, Flexible Coordination in Resource-Constrained Multi-Agent Domains. The broad goal of this research, which was carried out as part of ARPA/Rome Laboratories Planning Initiative (PI), has been to investigate the use of constraint-based scheduling frameworks and techniques as a basis for more accurate and more flexible decision support at various stages of the military crisis-action planning, deployment and employment process.

The scheduling problems faced in the crisis-action planning domain present significant technical challenges. The problems are large-scale, the planning and execution environment is dynamically changing, and solutions must integrate the actions of multiple decision-makers. Coordinated decision-making about resource apportionment and allocation is required at different levels of detail over different time scales, including (1) the ability to assess deployment transportation feasibility at early stages in the mission planning process and determine overall transportation asset requirements, (2) the ability to generate increasingly detailed deployment schedules that satisfy operational constraints, make efficient use of transportation assets and balance conflicting mission objectives, and (3) the ability to rapidly adapt schedules in light of changing (or evolving) circumstances. Existing transportation planning/scheduling systems and mobility analysis models generally operate with models that simplify important domain constraints (limiting the confidence that can be placed in results), inadequately manage problem complexity (sacrificing optimality), provide only limited support for many planning tasks (forcing extensive manual analysis and manipulation of system inputs and outputs), are inflexible in reactive decision making contexts (limiting responsiveness to change), and provide no support for coordination of different planning tasks (increasing overall planning time).

Toward the development and demonstration of constraint-based scheduling technologies that overcome these limitations, our research objectives have been two-fold:

1. to extend and generalize previously developed theories and techniques for constraint-directed scheduling for application to the military crisis-action logistics domain, and to evaluate their effectiveness in various decision-support contexts, and

2. to augment this constraint-directed scheduling methodology with protocols and coordination strategies to support integrated decision-making by multiple transportation scheduling agents.

We have taken a specific constraint-based scheduling technology, the OPIS manufacturing scheduling system, as our starting point and have focused on generalization and application of its multi-perspective scheduling technology to the deployment scheduling problem. These investigations have led to the development of DITOPS, a constraint-based, transportation scheduling prototype with facilities to support problem decomposition and distributed decision-making (DITOPS stands for Distributed Transportation Scheduling in OPIS). In

parallel, we have also conducted more basic research into new constraint-based scheduling procedures and into mechanisms for coordinating the scheduling actions of multiple agents.

Methodologically, our goal has been to demonstrate the feasibility and potential of constraint-based scheduling technologies under realistic crisis-action planning assumptions. Our augmentation and generalization of the capabilities defined in the underlying OPIS scheduler have been focused by detailed analysis of the military crisis-action planning process, the characteristics and scheduling requirements in this domain, and the capabilities of current transportation scheduling tools. We have experimentally evaluated the resulting DITOPS system on large scale transportation scheduling problems defined within the ARPA PI Common Prototyping Environment (CPE), and additional functional capabilities have been demonstrated through the conduct of Technology Integration Experiments (TIEs) with planning technologies developed at both BBN (FMERG) and SRI (SOCAP). We have also exported scheduling support capabilities provided by DITOPS for use within the TARGET Integrated Feasibility Demonstration system (IFD 3).

1.1 The DITOPS Transportation Scheduler

The constraint-based scheduling concepts and techniques implemented and demonstrated in the DITOPS scheduler offer several sources of leverage in addressing crisis-action deployment problems:

- incorporation of additional deployment constraints - One basic pre-requisite for development of realistic deployment schedules is an ability to accurately model and account for all important domain and problem constraints. DITOPS is able to generate transportation schedules that take into account important classes of constraints that are not currently modeled in current transportation planning practice (e.g., the temporal precedence and synchronization constraints between "TPFDD" movement records). It is also capable of enforcing other classes of constraints that are modeled in practice but are typically ignored by other transportation scheduling tools (e.g., enforcement of "earliest arrival date" (EAD) constraints).

- higher quality schedules - Through reliance on techniques that use information about constraint interactions to guide the scheduling process, DITOPS is able to produce schedules that better optimize transportation objectives than schedulers that operate with conventional, simulation-based procedures. Dramatic improvements to deployment "closure profiles" have been demonstrated, for example, in comparative analysis experiments with other representative TPFDD scheduling approaches at comparable computational cost.

- incrementality and reactive capabilities - The constraint-based scheduling procedures utilized to generate schedules in DITOPS are incremental in nature and thus equally applicable to the problem of incrementally revising a schedule in response to change. DITOPS provides a variety of rescheduling methods, each designed to locally revise

specific decisions in the schedule while emphasizing specific reoptimization objectives. Through use of constraint propagation and analysis methods, DITOPS provides guidance as to what decisions in the schedule must be revised in a given reactive context, what opportunities exist for non-disruptive change, and what reoptimization objectives should be emphasized, all of which can be used to direct the schedule revision process.

- Flexibility to support different planning tasks - DITOPS provides scheduling procedures that can be flexibly adapted to obtain different functional capabilities. One level of flexibility is provided by the ability to selectively specify "relaxable" constraints and overlay preference structures (in the form of a utility function) on their satisfaction. This enables the scheduler to be configured to address qualitatively different deployment questions. For example, arrival dates can be specified as relaxable to perform closure analyses under specific asset apportionment assumptions. (This is the specific task for which most current transportation analysis tools are designed.) It is also possible to instead designate asset capacity constraints as relaxable to estimate the resources required to achieve mission closure dates. (This is a task that is typically not addressable with current tools.)

  Another level of flexibility stems from the extensibility of the underlying modeling framework and scheduling infra-structure. Exploiting object-based representation techniques and an object-oriented programming methodology, the DITOPS scheduler is explicitly designed for extension/customization of modeling capabilities (to effectively incorporate the important idiosynchracies of a given planning domain) and reuse of component constraint management and scheduling techniques (to rapidly adapt scheduling support capabilities to fit the requirements of different decision-support tasks and applications).

1.2 The Larger Vision

From a broader perspective, the constraint-based scheduling capabilities demonstrated in the DITOPS scheduler point the way toward a different paradigm for decision support than is provided in current transportation and mobility analysis tools; a paradigm that more directly matches the requirements and characteristics of the transportation planning and scheduling process. Construction of transportation schedules in practice is an *iterative 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 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 retain 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 (e.g., changes to mission requirements, unexpected unavailability of lift capacity), 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.

In contrast to these decision support requirements, current transportation scheduling tools are typically *batch-oriented* solution generators. In commonly used simulation-based technologies, for example, 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.

Constraint-based scheduling procedures, alternatively, by virtue of their inherently incremental and decomposable nature, enable an *interactive* decision-support paradigm based directly on focused, incremental change to the current solution. Constraint-based scheduling procedures manipulate schedules "from the side" (i.e., placing and rearranging activities on a time line in accordance with resource and process constraints, as opposed to simulating execution forward in time and recording the activity time and resource assignments that result as a by-product), providing a natural framework for selective user exploration and comparison of alternative assumptions, and for direct, controlled convergence to an acceptable solution. It is possible to incrementally commit to subsets of decisions in the current solution (e.g., "locking down" decisions associated with particular forces or transportation resources), to likewise designate sets of activities or regions of the time line that require change/improvement, and to specify constraint changes (e.g., addition of lift capacity, routing changes) to be taken into account as the schedule is revised. The (reactive) scheduling methods implemented in the DITOPS scheduler provide the types of functionality required to support this interactive decision-making model.

Our vision of transportation scheduling tools of the future are decision-support environments similar in spirit to current day spreadsheet programs; sets of scheduling decisions and solution constraints are interactively manipulated by the user at levels consistent with user-task models, with the system applying appropriate (re)scheduling procedures to implement user actions (i.e., manage the details) and provide immediate, localized consequences of each change. Constraint analysis techniques will contribute additional leverage to this incremental scheduling process, providing guidance to users in identifying the principal causes of observed solution deficiencies (e.g., resource bottlenecks) and in analyzing various decision-making options.

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:

   (a) 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.

   (b) 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.

Given the complexity and scale of transportation scheduling problems, a crucial component of such an environment is a framework for interaction that enables the user to visualize, analyze and manipulate solutions at multiple, aggregate levels. The current DITOPS user interface has taken some initial steps in this direction, providing facilities for graphically visualizing and manipulating resource schedules and capacity constraints at different levels of aggregation. But significant challenges remain in effectively bridging the gap between user and system models of transportation schedules; this constitutes a major focus of our current research.

1.3 Organization of the Report

In Section 2, we summarize the major accomplishments and contributions of the research effort. In Section 3, we summarize the concepts and techniques underlying the current DITOPS transportation scheduler, the experimental results obtained in the context of TPFDD scheduling, the interactive, reactive capabilities for what-if exploration and incremental schedule revision that have been demonstrated, and the extended distributed scheduling prototype implemented. In Section 4, we summarize the additional capabilities and results obtained through participation in various Technology Integration Experiments with other PI sponsored organizations.

2 Summary of Accomplishments

Our research has produced the following major accomplishments:

- *Ontological primitives for modeling transportation scheduling problems* - We have developed a general scheduling ontology (in the form of an exportable class library) that enables specification of transportation domain models that incorporate all important constraints in any given transportation scheduling context. The ontology is explicitly designed to support:
  - *Realistic models of resource allocation constraints, objectives and preferences* - The ontology provides primitives for differentially modeling a wide range of resource types (resuable, consumable, shared, atomic, composite, mobile, stationary, etc.) and allocation constraints (capacity limits, cargo compatibility restrictions and preferences, mobility and availability constraints, allocation preferences, etc.). Likewise, primitives are defined for modeling the component activities of transportation plans (e.g., transporting, loading, unloading, processing, etc.), the temporal relationships that exist among them (e.g., multi-leg plans, synchronized air/sea movements, etc.), absolute timing constraints and preferences on their execution, and their resource requirements.
  - *Multi-level models* - The ontology provides structures and protocols for constructing hierarchical descriptions of transportation processes and required resources, allowing the level of detail at which allocation decisions are considered to be selectively and dynamically varied according to planning context (e.g., high-level course of action analysis, tpfdd-level feasibility analysis, detailed port scheduling) and domain characteristics (e.g., the criticality of various constraints).
  - *Extensibility and reuse* - The ontology provides general protocols for combination and extension/customization of concepts to capture the important idiosynchracies of a given transportation scheduling application.

- *Constraint-based techniques for transportation scheduling* - We have extended and adapted the multi-perspective scheduling techniques of OPIS to incorporate the dominant characteristics of transportation scheduling problems and enable multi-perspective construction and revision of transportation schedules. At the infra-structure level, we have developed constraint management techniques to enforce cargo "batching" constraints, to enable "splitting" of move requirements too large to be accommodated by a given asset, to incorporate capacity requirements that involve multiple resources (e.g., lift asset and port capacity), and to account for resource location constraints. We have developed two general scheduling (or rescheduling) procedures that localize decision-making along two distinct foci: A "resource scheduler", which constructs (or revises) some portion of the schedule of a designate transportation asset (or set of assets), and a "movement scheduler", which constructs (or revises) the schedule of a designated set of temporally connected move requirements (e.g., a multi-leg trip).

More specialized reactive methods for shifting movement schedules and redirecting movements to nearby destinations have also been developed to provide capabilities for qualitatively different types of reactive change.

- *Demonstration and analysis of capabilities in the domain of "TPFDD" scheduling* - In collaboration with BBN (Cambridge), a comparative analysis of TPFDD level deployment scheduling capabilities was carried out with respect to PFE. On the "MEDCOM problem scenario" that was utilized in IFD2, the DITOPS scheduler was shown to (1) to produce deployment schedules for various sea and air assets with 25-50% reduction in movement tardiness over PFE (assuming comparable constraints), and (2) provide an ability to enforce important constraints (e.g., earliest arrival dates) that are currently not handled within PFE. In this latter case, a 6% reduction in movement tardiness over the PFE schedule was still obtained for sea cargo movements (i.e., even when additional, more restrictive constraints were enforced). Another TIE experiment with BBN demonstrated the use of DITOPS scheduling capabilities in support of decisions earlier in the TPFDD generation process, specifically the ability to make transport mode assignments that take better advantage of the capabilities of apportioned transportation assets through generation of aggregate level deployment schedules. Capabilities for incrementally revising deployment schedules to account for changes in problem constraints (e.g., the unexpected unavailability of a POE, a reduction in lift capacity) have also been developed and demonstrated, supporting both reactive management of deployment schedules as well as a basis for pro-actively evaluating the impact of various possible scenarios.

- *Re-engineering and porting of OPIS/DITOPS modeling and scheduling infrastructure into a PI-compatible software/hardware environment* - As part of this project, we have ported the underlying OPIS scheduler from a TI Explorer environment including KnowledgeCraft to CommonLisp/CLOS/CLIM on a Sun Workstation. The new software architecture is heavily based on object-oriented representation and programming techniques, and is organized to promote rapid adaptation and configuration of component scheduling functionality (or tools) into new scheduling services that fit the requirements of specific client applications. The modeling and scheduling infrastructure is defined according to a layered system semantics (which is implemented in the form of class libraries). At the base of the system is an extended object system which adds necessary "frame-like" representation capabilities. Using these basic capabilities, basic kernel scheduling components are then defined (e.g., constraint propagation techniques, general purpose modeling primitives, capacity analysis techniques). More specialized system components (e.g., the transportation scheduling methods and heuristics of DITOPS) are in turn defined by composing relevant kernel scheduling services, and finally, those capabilities specific to a particular application domain are configured (in our work thus far, relating primarily to the joint strategic deployment scheduling domain).

- *Interactive Transportation Scheduling* - We have integrated the ported infra-structure with graphical schedule visualization and manipulation capabilities to provide a flexible

interactive environment for construction and management of transportation schedules. Utilizing the system's hierarchical domain model, the user interface promotes interaction at aggregate levels. The user can view resource schedules, presented graphically as usage profiles over time, at different levels of detail (e.g., for an individual ship, for the cargo ship fleet, for all transportation lift assets). Building in part on capabilities provided by the CPE SciGraph package, activity-centered views (e.g., movement closure profiles) can also be examined for a graphically selected portion of any resource schedule. Changes in availability of various resources (e.g., indicating port closures, addition or loss of transportation assets) can be graphically communicated, utilizing the reactive scheduler to examine effects. Users may also specify changes to various scheduling preferences and objectives utilized by the system (e.g., preferring use of large ships to small ships) to explore the consequences of various tradeoffs.

- *Integration of resource analysis capabilities into higher-level deployment planning processes* - Using the ported infra-structure, we have configured and exported an employment plan constraint checking/scheduling module for integration by BBN/San Diego into the TARGET IFD-3 system. Also, in collaboration with SRI, we developed and provided a resource capacity analysis capability to support plan evaluation within SRI's SOCAP course of action (COA) plan generator.

- *Development and demonstration of distributed, multi-level deployment scheduling* - Through analysis of current transportation planning practice, criteria for problem decomposition (scope, granularity, types of decisions) were identified, leading to the definition of a multi-level model and organizational structure for distributed transportation scheduling and control. We developed and implemented a communication and coordination infra-structure to support this distributed model, and demonstrated its use in integrating the scheduling activities of a global (e.g., "transcom" level) agent and multiple port schedulers.

- *Development and validation of new "constraint-posting" scheduling techniques* - We have developed new procedures for constructing schedules which, in contrast to conventional approaches to scheduling, operate with the more general representational assumptions of contemporary temporal planning frameworks, and thus provide natural opportunities for tighter integration of planning and scheduling processes. Experimental work thus far has concentrated on calibrating performance leverage with respect to classical scheduling approaches on published benchmark problems, and the results obtained thus far are quite impressive. We have demonstrated (1) an ability to produce solutions comparable to micro-opportunistic, "bottleneck-based" approaches on constraint satisfaction problems with orders of magnitude speedup, and (2) an ability to outperform the best known approximation algorithms developed in the Operation Research community in various schedule optimization contexts. (Detailed accounts of this work may be found in [SC93, CS93b, CS93a].)

- *Development and analysis of frameworks for cooperative, multi-agent decision-making* - We have developed an approach for distributed constraint satisfaction based on (1) partitioning the set of constraints into subsets of different types and (2) associating

responsibility for enforcing constraints of each type with different sets of specialized agents. Variable instantiation is the joint responsibility of different teams of these specialized agents, and the final solution emerges through incremental local revisions of an initial, possibly inconsistent, instantiation of all variables. Experimental evaluation of the approach on constraint satisfaction scheduling problems has shown this distributed approach to also perform comparably to micro-opportunistic, bottleneck-based procedures with much greater computational efficiency. (see [LS93b] for details.)

We have also developed a model for collaborative decision-making by teams of specialists, each with unique areas of expertise and limited understanding of the expertise of other agents. The approach is based on a partitioning of agent knowledge into expert and naive models. The naive portion of agents' models provides both a common language and the inferential skeleton needed for the development of shared models. Model refinement occurs when problem solving reaches an impass; structured communications among agents are tied to model manipulations, which dynamically alter agents' evaluations and justifications, and results in more precisely directed overall search. (This work is described further in [LS93a].)

- *Contributions to PI integration and infra-structure activities* - We have been active in supporting numerous joint activities of the PI: serving as co-chair of the working group responsible for developing the PI's overall vision or "Technical Roadmap" for planning and scheduling technology development and identifying critical experiments, and serving as co-chair of the Scheduling Technology Working Group. We have also made contributions to the knowledge representation working group (included in the Knowledge Representation Specification Language (KSRL) document) and to the development of the Common Plan Representation (CPR). We have supported BBN in its development of the Common Prototyping Environment (CPE), including insertion of the DITOPS scheduler into the CPE and development of the interface modules to make the system accessible as a knowledge server through the CRONUS inter-module communication infra-structure.

3 Technical Overview of DITOPS

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 at CMU, 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. Just as important, DITOPS also provides flexible capabilities for incrementally revising schedules in response to changed constraints. 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 closures), DITOPS can provide immediate, localized feedback of the effects of these changes on the current schedule. 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.

The DITOPS scheduling framework is founded on three basic principles:

1. decision-making must be rooted in a representational framework sufficient to capture important domain constraints and scheduling preferences - DITOPS provides a general framework for modeling transportation processes, required resources, movement requirements and shipments, which can be instantiated in any specific problem domain to encode all relevant temporal synchronization and resource utilization constraints on solution feasibility. Specific types of constraints (e.g., deadlines) can be selectively modeled as relaxable preferences, and domain models are defined hierarchically to enable different levels of constraint specificity (e.g., to match their relative importance in a given problem context).

2. dynamic look-ahead analysis of the structure of problem constraints is the key to efficient and effective scheduling - At the core of DITOPS is an incremental, reactive framework for generating and revising schedules [Smith 93], which relies on repeated analyses of current problem constraints (e.g. projected resource contention, current scheduling conflicts) to focus attention toward most critical decisions and tradeoffs, and to select appropriate decision-making (or decision revision) procedures.

3. large-scale problem solving invariably involves multiple decision-makers and distributed decision-making - The DITOPS scheduler has been augmented with mechanisms for inter-agent coordination, and initial protocols and interaction strategies consistent with military transportation planning and control requirements have been implemented.

In the subsections below we first summarize the technical approach taken to representation and decision-making within the DITOPS transportation scheduler. We then summarize the performance and interactive/reactive capabilities of the core scheduler. This is followed by an overview of the extended prototype developed for distributed multi-level transportation scheduling. A summary of additional functionality that was configured using components of the scheduler to support various course of action planning processes is presented in Section 4.

3.1 Modeling Transportation Scheduling Constraints

The DITOPS scheduler operates with respect to a hierarchical model of the resources and resource allocation constraints of a given application domain. The use of a hierarchical model serves three basic purposes. First, it enables decision-making at different levels of abstraction to support different stages of the overall planning process (e.g. high level capacity analysis, determination of transport modes, detailed asset assignments and movement schedules). Second, it provides a basis for focusing the scheduler's search process when scheduling (or rescheduling) at a specific level of detail. Finally, it provides a structure for decomposing and distributing a transportation scheduling problem among multiple agents, and a basis for coordinating multi-agent decision-making across different levels.

A DITOPS model of a given application domain is composed from 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* - Resource objects represent the various assets, equipment, and facilities required to carry out deployment requests. A variety of specialized resource classes are defined to support specification of different types of resources. These resource types include unit capacity resources, which must be allocated exclusively to a single request (e.g., a loading/unloading crane), batch capacity resources, which can simultaneously accommodate multiple requests over the same interval (e.g., a sea barge or tanker ship), and a variety of disjunctive and conjunctive aggregate capacity resources, where capacity can be simultaneously allocated to multiple requests without temporal synchronization (e.g., a C-5 plane fleet, a tanker ship fleet, a seaport). Atomic resources can be grouped through the definition of composite resources (e.g. individual tankers into a tanker fleet into an overall sea fleet; unloading equipment, storage capacity, parking places, etc, into a port) to provide consistent descriptions of resources and utilization constraints at multiple levels of abstraction. Such resource models provide the basis for hierarchical specification of transportation processes.

  A central component of each resource class specialization is a set of methods for managing and querying a representation of available capacity over time. These methods define the resource class's allocation semantics from the standpoint of scheduling and

control decision-making. Resources can also be distinguished as mobile (a ship) or stationary (a port); the former case implying the representation (and management) of a second dynamically changing property, location. Other utilization constraints associated with resource descriptions and enforced by allocation methods include constraints on capabilities (e.g., subset of commodity types that can be moved by a given type of transport asset), resource capacity constraints, and batching constraints (e.g., incompatibilities among commodity types that might be carried simultaneously).

- *Operations* - Operation objects are used to represent the constituent actions of transportation processes (or plans). Generally speaking, an operation specifies the set of constraints and effects that define a particular activity (resource requirements, duration constraints, temporal relations relative to other activities, cargo involved). Like resources, a taxonomy of specializations are defined to characterize different activity types. For example "transport operations" specify an origin (POE) and destination (POD), which imposes a setup requirement that the allocated transportation asset be at the origin at the start of the operation and an effect that leaves the allocated asset at the destination location. "Load" and "unload" operations, alternatively, do not change asset location. Through association of temporal relationships and/or synchronization constraints to other operations, operation descriptions can be composed into larger transportation processes. Operations can also be organized hierarchically to provide descriptions of transportation processes at different levels of resource specificity.

- *Move Requirements* - Move requirement objects represent the input requests that the scheduler must attend to. These descriptions specify cargo characteristics (e.g., cargo and commodity types), quantities, origin and destination of the movement (e.g., POE and POD), and relevant absolute time constraints (e.g., ALD, EAD, LAD, etc.), as well as any temporal relations and/or synchronization constraints with other move requirements (e.g., that two movements must arrive within a day of each other). In the context of deployment scheduling, move requirements correspond directly to individual TPFDD records.

- *Shipments* - Shipment objects represent the actual cargo entities (or "packages") that are associated with individual transport operations (e.g., the 25th infantry division, 1000 CBarrels of POL, etc.). Shipments are created in response to the cargo specifications given in move requirements. Generally speaking, accomplishment of a given move requirement may necessitate the transport of several shipments (i.e., require multiple trips), since a move requirement's lift requirements may exceed the capacity of any available transportation asset.

- *Missions* - Mission objects provide a specification of a plan template (or basic plan class) for instantiating the transport plans that must be scheduled. In the strategic deployment domain, for example, the basic plan class corresponding to an individual TPFDD record is specified as an aggregate transport operation (at some level of precision with respect to required asset capacity), which decomposes into a load, travel, unload operation sequence.

Through definition of more specialized object classes, the constraints specified by any of these modeling primitives can be straightforwardly customized. For example, in modeling the IFD2 MEDCOM scenario in terms consistent with PFE (see discussion of experiments below), a specialization of transport operation was defined to incorporate the PFE definition of required capacity as a function of both commodity and asset type. The current DITOPS library of modeling primitives consists of 80 core (i.e., domain independent) classes and 40 additional specializations defined for specific application to military transportation planning domains. Full details of these primitives and their protocols may be found in [LSS+93].

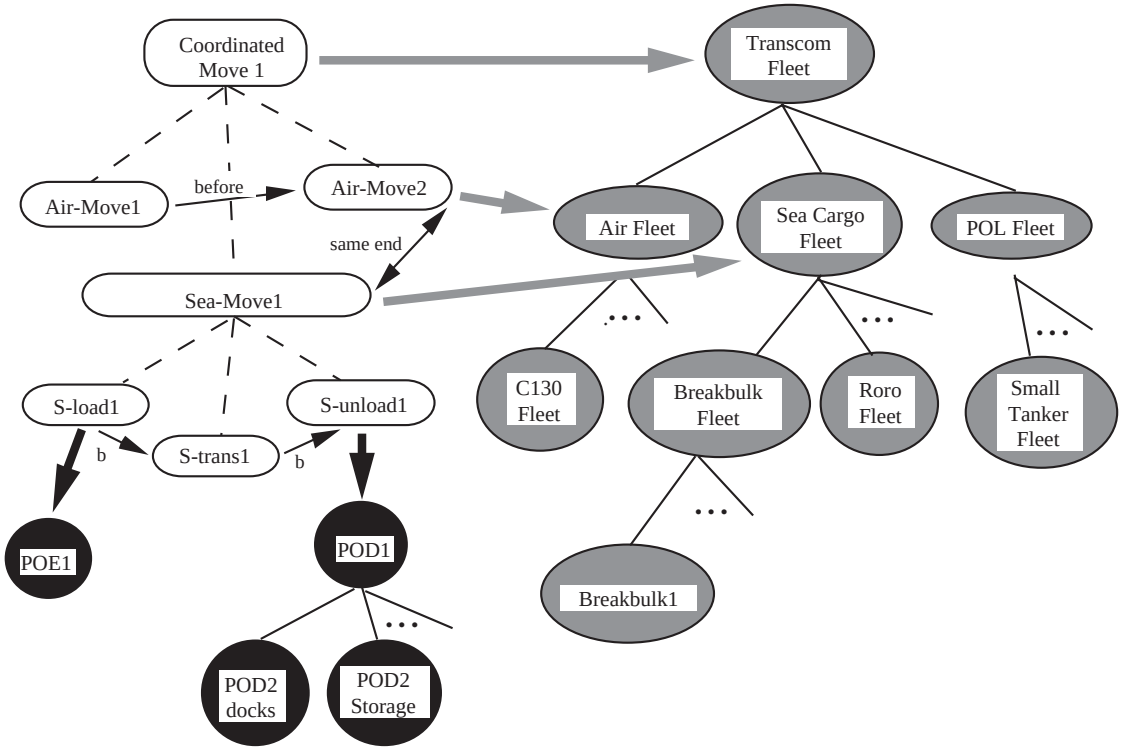

Figure 1: Hierarchical domain models

To give a flavor of the modeling capabilities provided by the DITOPS class library, we consider aspects of the domain model that was constructed for MEDCOM strategic deployment scenario just mentioned. Figure 1 graphically illustrates the defined hierarchical models of required resources and transportation processes.

In the resource model depicted, individual transportation assets are first composed into disjunctive aggregate resources (or resource pools) representing fleets of specific craft types. These descriptions, in turn, are aggregated into larger disjunctive aggregates representing higher-level pools of allocation alternatives (e.g., cargo/pol sea lift capacity, air/sea lift capacity). In this case, capacity constraints are straightforward mapped by summing the unit capacities of individual resources[1]. Representations of port resources are specified similarly.

[1]Mapping of other constraints (e.g., operating speeds, allowable cargo types) is accomplished by various approximation methods (e.g., weighted averaging, set union).

However, in this case the individual resources associated with a given port (e.g., loading unloading equipment, cargo storage space, etc.) are composed into a conjunctive aggregate resource, which provides a single, higher-level estimate of overall port capacity[2] On one hand, these levels of resource description define multiple levels of possible scheduling precision. For example, in computing transport mode assignments relative to an apportioned set of resources, there is likely little leverage to be gained by computing schedules at the level of individual resources. Alternatively, a level of scheduling precision appropriate for transportation feasibility analysis at the level of USTRANSCOM would include individual craft assignments but only aggregate accounting of port capacity constraints. Detailed models of atomic port resources would, however, become necessary at the stage of detailed execution planning. Having fixed a given level of scheduling precision (say individual craft assignments and aggregate accounting of port capacity), the hierarchical model additionally provides a structure for elaborating the search for a schedule. Summarized allocation constraints and preferences associated with aggregate resources (e.g., current available capacity, usage restrictions, preferred sub-resources) provide a basis for restricting and biasing consideration of resource alternatives.

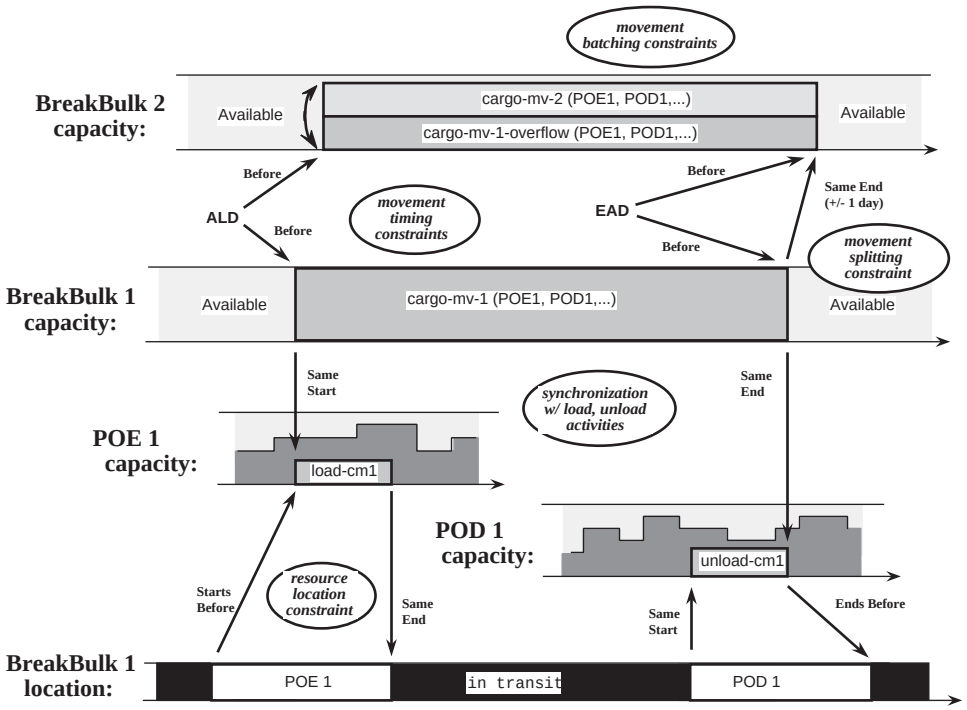

Figure 2: Allocation constraints in "MEDCOM" model

Figure 2 graphically illustrates the range of allocation constraints incorporated within the

[2]Several alternative mappings of capacity are possible here. The mapping might omit all but the most constraining atomic capacity; alternatively, the mapping might compute an overall throughput summary estimate. In some problem contexts, it may not be necessary, desirable or feasible to account for and model individual port capacity constraints, in which cases, aggregate port descriptions simply constitute the most detailed level of the model.

MEDCOM domain model. The figure centers around a particular scheduled transport activity, *cargo-mv-1*. It depicts the available capacity over time of two ships, *breakbulk-1* and *breakbulk-2*, and two seaports, *POE-1* and *POD-1*, as well as the current location of *breakbulk-1* over time. The "box" labeled with *cargo-mv-1* within the available capacity profile of *breakbulk-1* represents the interval over which *cargo-mv-1* is scheduled to occur; since *breakbulk-1* is required and the quantity of *cargo-mv-1* fully consumes the capacity of the ship, it is unavailable for other use over this period (according to the depicted profile, it is currently available both before and after this scheduled trip). The activity *cargo-mv-1* abstracts a more detailed sequence of *load-cm1*, *transport-cm1* and *unload-cm1* operations. Both the *load-cm1* and *unload-cm1* operations additionally require port capacity at *POE1* and *POD1* respectively. According to the defined hierarchical model, *load-cm1* is constrained to commence at the beginning of the overall *cargo-mv-1* operation (i.e. at the same point that *breakbulk-1* is first allocated to *cargo-mv-1*), and, conversely *unload-cm1* is constrained to end coincident with the "release" of *breakbulk-1* by *cargo-mv-1*. During the scheduled interval of both *load-cm1* and *unload-cm1*, the required amount of port capacity (in this case a function of the cargo quantity) is designated as allocated to these operations and otherwise unavailable. Execution of either *load-cm1* or *unload-cm1* also requires the transport resource to be physically present at the port. These constraints are specified in the model as operation "setup" constraints that must be satisfied, and are enforced by ensuring that the resource is at the designated POE at the scheduled start of any load operation. If , during scheduling of a load operation the assigned transport resource is not at the load site, checks are made to ensure that the resource is available sufficiently earlier than the scheduled start to enable it to travel to the load site.

The example depicted in Figure 2 also illustrates three other types of allocation constraints that are specified in the MEDCOM model and taken into account when scheduling transportation activities. In this example, *cargo-mv-1* actually moves only a portion of the cargo designated in its associated move requirement (i.e., the TPFDD record that led to creation of *cargo-mv-1* in the first place). Since the maximum carrying capacity of *breakbulk-1* is not sufficient to accommodate the entire input requirement, the load has been dynamically "split" into two loads. A second *cargo-mv-1-overflow* activity has been created and scheduled on a different resource *breakbulk-2*. In this case, both *cargo-mv-1* and *cargo-mv-1-overflow* share the common ALD (available to load) and EAD (earliest arrival date) constraints on scheduled start and end times that are specified in the associated move requirement. Whenever the cargo associated with a given move requirement must be split across multiple trips, a default constraint on their relative timing is also imposed. In this example, the two transport activities are constrained to finish within one day of each other. Finally, it is often the case that the capacity of a given transportation resource is sufficient to simultaneously support multiple transport activities, transporting their respective cargos as a "batch" on the same trip. This is the case for the *breakbulk-2* trip that is depicted in Figure 2 where *cargo-mv-1-overflow* has been batched with second transport activity *cargo-mv-2*. For transport activities to be batched, several constraints may have to be satisfied. Minimally, the activities must have the same designated POEs and PODs. Although not specified in the MEDCOM model, additional constraints relating to the compatibility of different cargo types might also be defined and enforced.

3.2 Building and Managing Transportation Schedules

Scheduling in DITOPS is formulated as a reactive process, reflecting the fact that a schedule at any level or stage of the deployment planning process is a dynamic evolving entity, and is continuously influenced by changing mission requirements, conflicting decision-making perspectives/goals and changing executional circumstances. This problem solving perspective in large part motivates the above illustrated representations of changing resource state over time (i.e., available capacity, location). These representations are pre-requisite to the specification of procedures for reflecting the consequences of changed constraints and for incrementally managing schedules in response to such changes. These representations also enable use of schedule building and revision procedures other than time-forward simulation, which is inherently myopic and susceptible to sub-optimal decision-making.

Most generally, the DITOPS scheduling model can be seen as a constraint-based scheduling model; instantiated movement plans define sets of start/end time and transport resource decision variables, and decision-making is concerned with establishing (or restoring) an assignment of times and resources to all variables that is consistent with specified temporal synchronization and resource utilization constraints. A constraint-based scheduling model is broadly characterized as an iterative procedure that combines three basic elements:

1. deductive constraint propagation techniques, which are applied to incrementally update the domains of decision variables in an underlying solution constraint graph as changes (or extensions) are made to the schedule and recognize inconsistencies,

2. look-ahead analysis techniques, which estimate the critical tradeoffs (decisions) and opportunities (flexibilities) implied by current solution constraints for purposes of determining which decision (or set of decisions) should be considered next, and

3. a decision procedure, or set of procedures, for carrying out specific solution changes or extensions.

In fine granularity scheduling models (e.g., [Sad94]), the look-ahead analysis and decision procedures map directly to the variable and value ordering heuristics of traditional constraint satisfaction problem solving procedures. DITOPS, alternatively, implements a "coarser granularity" model [Smi94]. Look-ahead analysis is instead used as a basis for heuristic problem structuring and subproblem formulation, which involves selection of a particular set of decision variables to focus on (i.e., assign or revise) and selection of a particular decision (or local search) procedure to apply to this set of decisions. In either type of model, preference or utility structures (e.g., reflecting objective criteria and preferences) can be associated with decision variable values to bias the overall search process. In the case of DITOPS, alternative decision-making procedures are specifically designed to provide differential optimization and conflict resolution capabilities. In the absence of explicit user guidance, control heuristics which map analyses of the current solution state to important optimization (or reoptimization) needs and opportunities are used to opportunistically select the most appropriate decision procedure on each control cycle. For example, suppose a capacity conflict has

been introduced into the schedule of a particular cargo ship due to it having been temporarily disabled. Activities scheduled over the expected period of unavailability must now be reassigned to other ships and loss of transport capacity implies the need to (re)optimize existing ship capacity to maximize utilization; a decision procedure with this optimizing property is the preferred procedure to apply. At the same time, reconsideration of the schedules of other ships capable of carrying the now stranded cargo should take into account current flexibilities in the solution. If in examining the available capacity of the fleet to which the failed ship belongs it is estimated that sufficient extra capacity to resolve the problem, there is no reason to consider any other viable resource alternatives; the scope of the change is restricted to this smaller set of resources. Within DITOPS, this subproblem formulation activity is carried out by a designated procedure referred to as the top-level manager. The underlying system control architecture is graphically shown in Figure 3.

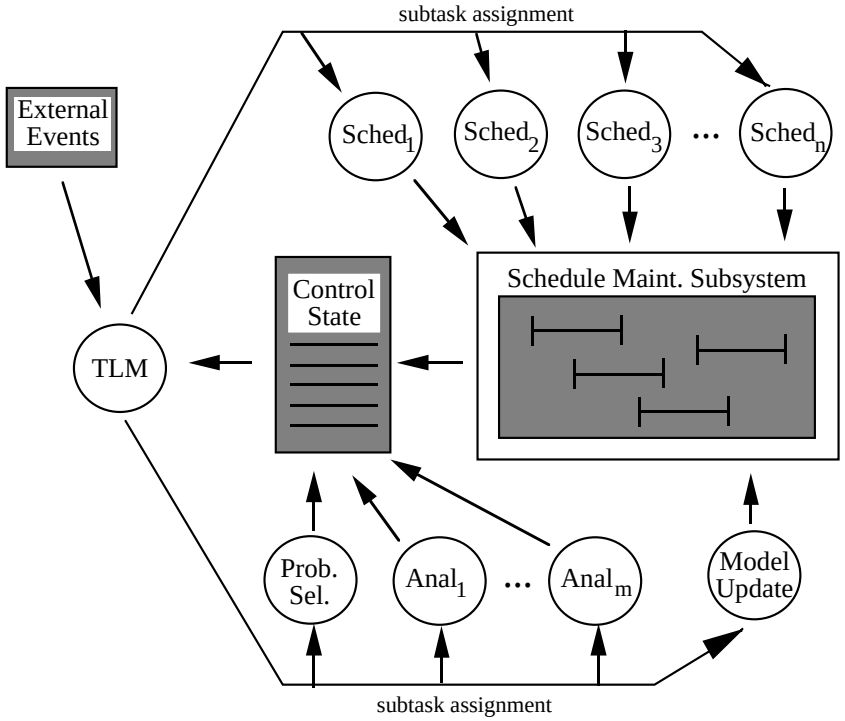

Figure 3: The DITOPS Scheduling Architecture

Two constraint analysis procedures are available within the DITOPS scheduler to support the control decisions of the top-level manager. In situations where scheduling decisions remain to be made, a capacity analysis procedure provides estimations of likely resource bottlenecks. In situations of detected constraint conflicts, a conflict analysis procedure computes a set of metrics, some of which estimate the severity of the problem and some of which characterize the looseness or tightness of time and capacity constraints in the local "neighborhood" of the schedule that contains the conflict.

A number of decision-making procedures are available for application in different scheduling or rescheduling contexts. 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 a common search infra-structure that

- incorporates machinery for incrementally propagating consequences of scheduling decisions and detecting constraint conflicts (referred to as the "schedule maintenance subsystem" in Figure 3),
- provides primitives for generating feasible decision alternatives (based on the use of aggregate resource and activity descriptions defined in the underlying domain model), and
- allows incorporation of additional allocation preferences, which are expressed in the domain model as utility functions over the possible values of decision variables (e.g., possible resource assignments, possible activity start times) and integrated as terms of the search procedures' evaluation function.

A number of more specialized revision procedures have also been defined, providing additional capabilities to shift the scheduled interval of scheduled "trips", to swap scheduled batches of particular transportation assets, and to balance cargo load to exploit increases in port capacity. The search infra-structure and decision procedures are defined and implemented compositionally using object-oriented techniques, providing a functional "tool box" for constructing additional decision-making procedures.[SL93]

3.3 Experimentation and Performance Analysis

In demonstrating and evaluating the capabilities of the DITOPS transportation scheduler, we have focused principally on the strategic deployment planning task addressed by the US Joint Transportation Command (USTRANSCOM). At this level of the logistics planning process, planning is concerned with the development, analysis and management of a Time-Phased Force Deployment Database (or TPFDD), which specifies the complete set of personnel and cargo movements required to support a given employment plan and all associated deployment constraints (e.g., earliest/latest departure and arrival dates, transport modes, origins and destinations, etc.). We have utilized a representative TPFDD provided within the ARPA PI CPE (referred to as the MEDCOM scenario) to demonstrate a range of decision support and decision making capabilities provided by the DITOPS transportation scheduler. We summarize this work in the subsections below.

Item	Description
POE	Port of embarkation (origin)
POD	Port of debarkation (destination)
ALD	Available to load date
EAD	Earliest arrival date
LAD	Latest arrival date
cargo-type	categorization of type of cargo used to formulate usage (or carrying) constraints for different resource types. One of: 'bulk outsize, oversize, pax, nat, pol, container, roro, breakbulk'
commodity-type	finer categorization of cargo type used to determine capacity requirements on feasible asset types. One of: 'airborne, air-cavalry, air-mobile, armored, infantry, mechanized, combat-support, combat-service-support, navy, air-force, marines, resupply, ammunition, pol, pax'
MTONS	sea cargo quantity in metric tons (weight)
STONS	air cargo quantity in short tons (volume)
PAX	quantity for PAX cargo - number of passengers
CBARRELS	quantity of POL cargo - number of 100 barrels

Figure 4: TPFDD Record Information

3.3.1 Generating TPFDD level schedules

The principal task supported by current scheduling tools at USTRANSCOM is *transportation feasibility analysis*: given a fully specified TPFDD and a profile of apportioned sea and air lift assets, generate a deployment schedule that assigns personnel and cargo to be moved to specific lift assets over time in accordance with specified constraints. To assess the capabilities of DITOPS in this capacity, we conducted a comparative experiment with a BBN developed feasibility estimator called PFE. PFE is a simulation-based technology based directly on the now operational DART simulation tool, and is quite representative of the tools currently in use at USTRANSCOM.[SMS+91]

The experimental comparison was carried out using the MEDCOM TPFDD that was generated during the course of the 2nd PI Integrated Feasibility Demonstration (IFD2). This TPFDD contains a total of 3001 movement requirements, of which 1187 are pre-designated as air movements and 1814 are pre-designated as sea movements. The information provided with each movement requirement is listed in Figure 4. Sea movements can be further decomposed into 1323 sea cargo movements (requiring capacity on some subset of five different types of cargo carrying vessels) and POL movements (requiring capacity on oil tankers). Given the pre-assignment of transport mode and the absence of temporal constraints on the relative timings of various sea cargo, air, and pol movements in this scenario, the problem is decomposable into 3 mutually exclusive subproblems. Air and sea assets apportioned to support the deployment consisted of 369 aircraft, 36 cargo ships, and 4 tankers, with initial locations and staged availability as indicated in Figure 5.

Craft Type		Total Number	Initial Location	Availability							
				C0	C1	C2	C3	C4	C5	C7	C10
Air											
	C5	56	TMKH	56							
	C130	128	TMKH	128							
	C141B	185	TMKH	185							
Sea Cargo											
	FBB	20	BBNV					6			
			DKSD	8	3			1			2
	FLASH	6	BBNV				2				
			DKSD		3		1				
	FRORO	6	BBNV		1		1				
			DKSD	1	1		2				
	FSSC	3	DKSD		2			1			
	FSEAB	2	DKSD		1		1				
Sea POL											
	SMTNK	2	DKSD		1		1				
	MDTNK	1	DKSD				1				
	LGTNK	1	DKSD							1	

Figure 5: MEDCOM scenario lift assets and availability

In collaboration with BBN, a model of scenario resources and resource utilization/allocation constraints equivalent to that employed in PFE was configured and instantiated. In particular, asset usage constraints were defined by associating a specific subset of allowable cargo types with each type of craft (e.g., C141Bs can only carry 'bulk' cargo). Asset capacity constraints were specified for each asset type as a vector of (commodity type quantity) pairs, over which capacity requirements for a movement of a specific commodity type on a specific type of resource were formulated as a function of the percentage of the resource required. Availability and locations of specific transportation assets were initialized according to the constraints in Figure 5, and travel times were based on identical models of resource operating speeds and inter-port distances. Equivalent port throughput capacity constraints were specified, including a reduced throughput capacity of 50,000 Mtons/day at one POD (Tunis) which was called for in the scenario to introduce greater congestion. There was one point of difference between the PFE and DITOPS models with respect to modeling both port and air-lift capacity. In DITOPS, capacity constraints were defined with respect to continuous time (i.e., how much capacity is available at any point in time), while PFE relies on less precise "capacity per day" models. In this regard, load and unload durations were assumed to be one day each for sea movements (consistent with PFE) and one hour each for air movements (below the granularity of the PFE simulation). Complete details of all port and asset capacity constraints can be found in the CPE description of the MEDCOM scenario.

In conducting the experiment, we focused on three dimensions of system performance:

1. the ability to enforce important deployment constraints - this dimension concerns the

System/Config.	Air Movements (1187 total)	Sea Cargo Mvmts. (1323 total)	POL Movements (491 total)
PFE w/o EAD enforc:	% tardy: 0	% tardy: 84	% tardy: 59
DITOPS w/o EAD enforc:	% tardy: 0	% tardy: 58	% tardy: 1
DITOPS w/ EAD enforc:	% tardy: 0	% tardy: 78	% tardy: 90

Figure 6: Comparative performance of DITOPS and PFE

reliability of the schedule as an indicator of deployment feasibility. In the case of the MEDCOM scenario, earliest arrival date constraints (EADs) were intended to be enforced as hard constraints (to preserve the element of surprise), but this was not possible withing PFE. We conducted runs with DITOPS with and without the assumption that EADs should be enforced as hard constraints, to demonstrate the potential variance in results.[3]

2. ability to optimize with respect to important deployment objectives - this dimension measures the system ability to produce better quality schedules, and hence better guidance to more detailed (e.g., component command) planning processes. Here our principal measure of performance was level of tardiness observed in the 'closure' of various movements under both generated schedules. We also tracked resource utilization over time, but due to the nature of the PFE simulation (as run by BBN), comparison was not possible along this dimension.

3. the computational cost of the scheduling process - the issue here is system efficiency and scalability.

Comparative results with respect to tardiness on the IFD2 problem are given in Table 6. As can be seen, there is sufficient air lift capacity to meet movement delivery dates (LADs) and the deployment schedules of both PFE and DITOPS show no tardiness. The situation is different with respect to the sea transport portion of the problem. With respect to sea cargo, for example, DITOPS produced a deployment schedule with a 6% reduction in late closures over the PFE schedule. Average resource utilization by resource type ranged from 51% to 100%, with an overall average utilization of 85%. It was not possible to compute and compare average tardiness figures, since the PFE simulation apparently terminates after 70 days (in this case, failing to schedule 32% of the movements).

This reduction in tardiness is significant for a couple of reasons. First, it was achieved while enforcing EAD constraints that were ignored by PFE and adversely affect the scheduler's ability to minimize late closures. A run of DITOPS where these constraints were also ignored (also included in Table 1) yielded a 26% reduction in tardiness. The results are even more dramatic in the case of POL movements, where, without EAD constraint enforcement,

[3]In actuality, it is not the case in this scenario that all EADs are in fact hard constraints; however, since information needed to differentiate was not available we assumed the worst case.

DITOPS produces a schedule with only 1% of the movements tardy as compared to PFE's schedule with 59% of the movements tardy. Second, these initial results were obtained with fairly generic scheduling methods. We expect even better results as heuristics that further exploit the structure of the problem are incorporated.

Using the ported CommonLisp/CLOS system, the schedules reported above are generated in just over 10 minutes on a SUN Sparcstation 10, indicating the ability of the DITOPS scheduler to scale to realistically sized problems. Experiments have also been performed under assumptions that port capacity constraints are not limiting and can therefore be ignored (which, for example, matches the modeling assumptions of the Kestrel scheduler[Smi92]). If the port capacity constraints specified for the MEDCOM problem are ignored, schedule generation time is reduced to about 7 minutes.

3.3.2 TPFDD Mode Assignment

We have also conducted experiments to demonstrate capabilities in support of decisions made earlier in the deployment planning process that in current practice are made without consideration of resource capacity constraints. We have performed preliminary experiments using (a variant of the IFD2 scenario) that demonstrate the potential impact of basing "transport mode" decisions on resource capacity information. Specifically, the mode decisions designated in the input IFD2 TPFDD were stripped off, and, using the constraints relating to various asset capabilities and cargo commodity types, aggregate level schedules were generated which assigned either air or sea lift capacity to specific move requirements. The results obtained varied considerably from the original mode assignments, exploiting the excess air lift capacity implied by the detailed TPFDD scheduling experiments described above. Although it is not clear whether represented resource usage and cargo commodity constraints are sufficient alone to determine feasible air and sea assignments, in this case, the redistribution of mode assignments resulted in better closure profiles.

3.4 Interactive/Reactive Schedule Revision

The reactive scheduling framework of DITOPS provides equally important capabilities for incrementally revising schedules, either in response to changes in external circumstances (e.g., the unexpected fog-in of a port or the receipt of additional deployment requirements) or for purposes of improving a schedule with observed deficiencies (e.g., by apportioning additional transport resources). From a mixed-initiative scheduling perspective, this reactive 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 system's subproblem formulation

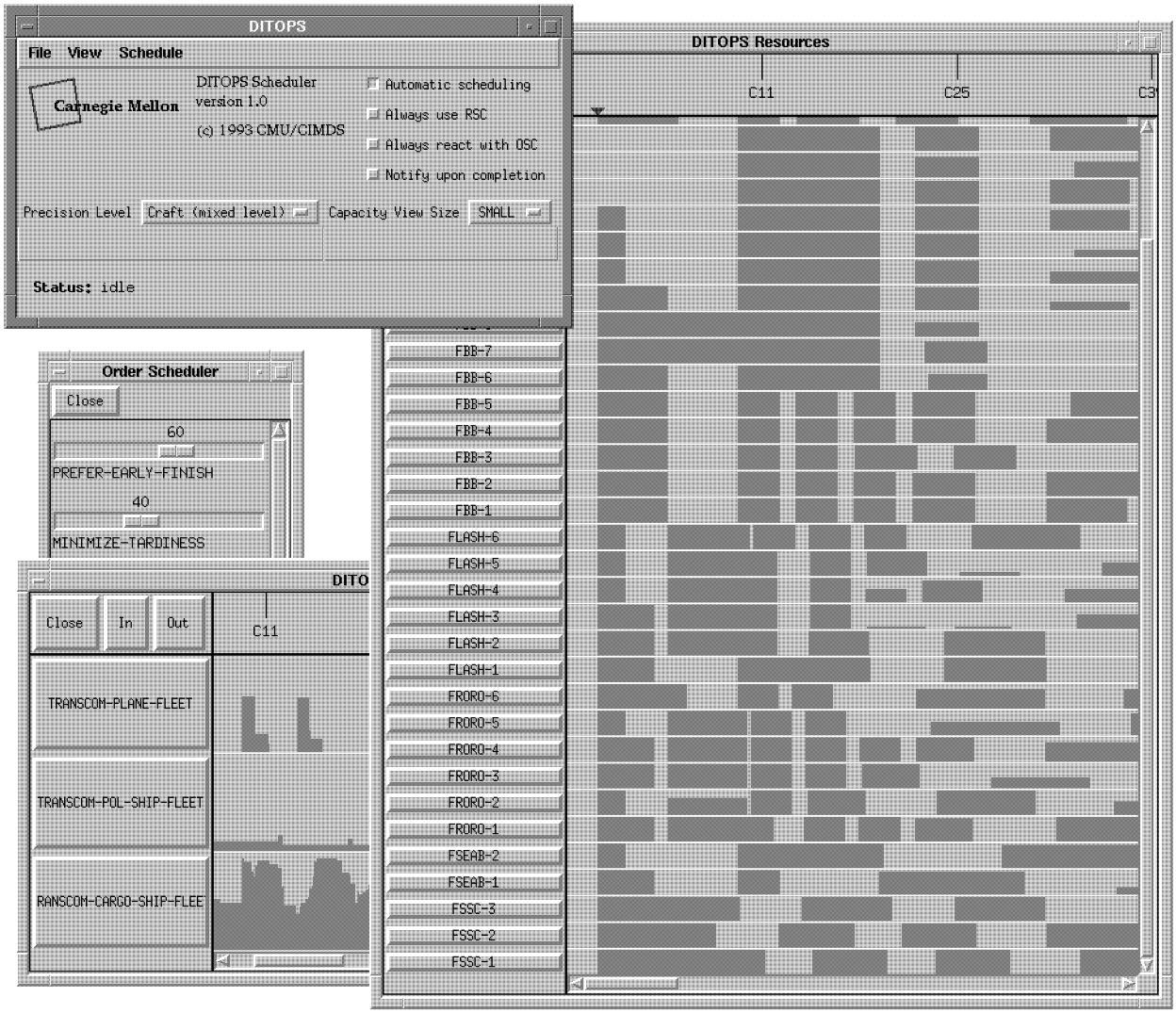

Figure 7: The DITOPS User Interface

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 7). 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 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 7), 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.).

3.5 Multi-Level, Distributed Scheduling

Transportation scheduling is inherently a distributed problem. Given its overall size and complexity, as well as the component structure of the military command, responsibility for different parts of the problem at different stages of the process are distributed among many

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

planning agents. In current military transportation practice, schedules are produced by different agents along different lines of decomposition. For example, (a) different agents (e.g., CENTCOM, PACOM) produce schedules at the same level of aggregation for different military operations (e.g., multiple simultaneous crises), (b) different agents (e.g., USTRANSCOM, MAC, MITMIC) produce schedules for the same operational scenario at different levels of aggregation, or (c) different agents produce schedules for different resources (e.g., tankers, crews, cargo-handling equipment at a port). In all cases, resolution of conflicts is an integral issue. Although decomposition is an effective means of reducing problem complexity, effective and efficient decision-making requires mechanisms for coordinated interaction.

To support investigation into and experimentation with protocols and strategies for coordinating multiple scheduling agents, the DITOPS infra-structure also incorporates primitives for asynchronous communication. These primitives allow easy implementation of agents, their control architectures and inter-agent messages. In addition, some of the basic services utilized within the DITOPS scheduler (e.g., time services) are designed to allow experimentation in a distributed, asynchronous environment. The design of the DITOPS communication substrate, like the rest of the DITOPS system, relies heavily on object-oriented programming concepts, and is influenced by earlier work reported in [LT91].

This system base has been used to define and implement an initial prototype system for distributed, multi-agent scheduling. We summarize the basic properties of the model underlying the prototype and the demonstration experiment that was performed below.

3.5.1 Decomposition and Interaction Assumptions

The hierarchical descriptions of resources and resource constraints advocated by the DITOPS modeling framework provide a natural basis for decomposing and structuring solutions to the overall transportation planning/scheduling problem. As previously observed, they provide a basis for specifying schedules at different levels to support decision-making at different stages of the planning/scheduling process. They likewise provide a structure for decomposing and distributing problem solving responsibility, where different agents are responsible for allocation/apportionment of specific sets of resources at a given level of detail (e.g. overall transport capacity, sea/air transport assets, port resources). Building from this basic problem decomposition perspective, we have developed a specific model for distributed, multi-level generation and management of transportation schedules. The model assumes a hierarchical organization of scheduling sgents, with each agent having access to specific levels of underlying hierarchical domain model (in effect, the "full" hierarchical model is distributed among scheduling agents). Thus, there is heterogeneity in the portion and level of description of the overall problem accessible to each agent.

Given the scale of the overall problem and the use of abstractions of resource allocation constraints as a basis for specifying problems and solutions at different levels, two further decomposition assumptions follow directly:

- **Decision-making scope and granularity** - The portion of overall problem that is visible and of concern to the decision-maker and correspondingly the level of detail of supporting models can be seen in relation to particular stages of the overall process. For example, transportation feasibility analysis during course of action development requires a global (and necessarily coarse) view of the whole problem. Management of day to day activities at a port, alternatively, requires much more detailed models of temporal process constraints and resource constraints, but only with respect to activities surrounding the use of the port.

- **Horizon of decision-making** - Corresponding to decreasing scope and increasing model detail is a decrease in the temporal horizon of decision-making. This assumption is supported by two considerations: problem scale and presence of environmental uncertainty. The problem solver's computational burden can remain almost invariant at each level by balancing decreasing scope and increasing model detail. The extent of uncertainty in the operating environment makes the executability of more detailed models more suspect further into the future. Thus a given decision-maker's horizon must balance the computational burden of maintaining the solution over time (or equivalently the extent to which it really provides a useful projection of future events)

These collective assumptions lead to a distributed model that resembles the organization and roles of current transportation planning command and control structures. This is illustrated in Figure 8.

Within this model there are two basic types of agent interactions:

- **Vertical**: The results of a given agent's scheduling (or rescheduling) actions are communicated downward as scheduling constraints/objectives; an agent's ability to satisfy imposed constraints/objectives, or responses to lower-level results are communicated upward. At each level of abstraction, an agent produces the best solution it can, given currently imposed global/constraints and objectives and the currently known results communicated from lower level agent results (or the execution environment)

- **Lateral**: Agents at the same level communicate to resolve local conflicts and produce solutions within bounds of constraints that have been imposed through downward constraint communication.

Coordination of the overall organization of agents is achieved by the following "interaction policies":

- Each agent is responsible for generating scheduling constraints for the agents directly under it (in the subtree of which it is the root). At the same time, since a lower level agent has a more detailed model of its own activities than a higher level agent has of it, the lower level agent can react more effectively to schedule deviations that are encountered at its level. Hence reaction starts at the level where the schedule deviation

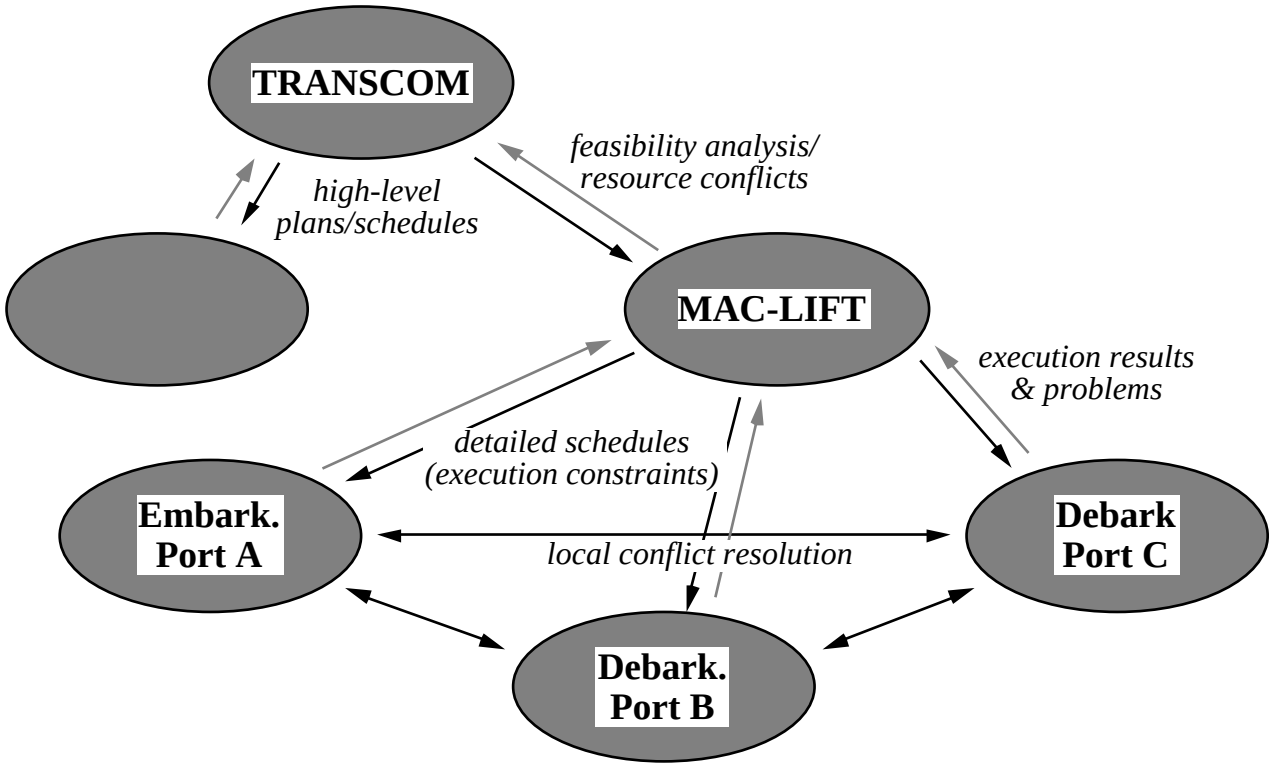

Figure 8: Distributed, multi-level transportation scheduling

occurs and its effects are propagated both downwards (in terms of new constraints) and upwards (in terms of violations of imposed constraints that may result in potential rescheduling decisions at a higher level).

- Deviations will always be responded to locally to the extent possible (engaging other local agents as necessary) - i.e. there must always be the ability to drive execution without communication to the "superior" agent,
- (re)scheduling results produced in response to deviations are always communicated upward if previously imposed guidelines (constraints) have been broken (and of course always propagated downward). If it can be recognized without local schedule revision that the deviation will break imposed constraints (e.g. a port fogin), then the deviation can be communicated upward immediately. In this case, a local agent must still try to make do (resolve problems) until its superior agent responds,
- If it becomes time to act (either in response to execution demands or to respond to lower level agents), then whatever current local solution exists is followed.
- If a superior agent, in response to either deviation or revised solution received from below, revises its more global solution, then revised constraints/guidelines are communicated downward to its inferior agents. Assuming a cooperative framework, these new constraints are given priority in resolving conflicts (i.e. if the revised constraints are inconsistent with current inferior agent schedules, then the inferior agent is obliged

to revise). One issue that arises in this protocol is the detail/granularity of the superior agent's model (and the possible mismatch with the inferior agent's more detailed model). However, if this is ultimately the reason to prefer an inferior agent's solution, then that will subsequently be discovered by its inability to meet the newly imposed constraints - in which case the best solution that the agent can produce is communicated upward.

3.5.2 Distributed Experiments

To demonstrate the above framework for multi-level transportation scheduling, a simple system configuration consisting of a single high-level (i.e., TRANSCOM-level) agent and two more detailed port scheduling agents was implemented and tested. The high-level agent was responsible of generating a deployment schedule for movements from port A to port B (under the aggregate port capacity models utilized in the DITOPS TPFDD scheduling experiments). Both port A and port B were assumed to have their own scheduling agents, whose responsibility was to develop more detailed schedules, involving allocation of constituent port resources (docking berths, loading equipment, etc). The high level agent operated with an overall horizon matching the total duration of the deployment at a temporal granularity of days, while the low-level port agents scheduled over a shorter horizon, defined relative to the travel time required for transport assets to move from port A to port B, and and produced hourly schedules.

In the scenario demonstrated, the high-level agent would generate an initial deployment schedule and communicate these results "downward" to port agents, requesting them each to generate a port schedule for the designated movements, given arrival and departure dates based on the high-level agent's schedule. The port agents would then generate a schedule for their own resources. Finding this impossible, the port agents would communicate with each other, possibly requesting arrival and departure dates to be shifted to arrive in a feasible solution. Once feasible detailed schedules were obtained, the port agents would then communicate their results "upwards" to the high-level agent.

Initially this scenario was simulated in a simple single-process environment. After developing an understanding of the necessary message types required for this type of distributed scheduling, the scenario was converted to function in a multi-process, multi-agent environment. This work has led to the design and implementation of a class library for asynchronous agents, providing primitives for the construction of the internal structure of agents (messages, message queues), agent control architectures (event processing mechanisms, tasks) and low-level services (network communication services, time and synchronization services). Further details of these mechanisms can be found in [LSS+93].

4 Additional Technology Integration Experiments and Support Services

In addition to the tpfdd-level scheduling capabilities summarized above, component scheduling "services" were also configured to provide constraint analysis support for higher-level course of action (COA) development:

- In collaboration with SRI, the resource capacity analysis capability used within the DITOPS scheduler was adapted for integration with the SOCAP planning system to provide feedback on transportation feasibility during generation of the deployment actions required to support the COA.
- Adapting component constraint propagation and conflict analysis techniques utilized within the DITOPS scheduler, a COA feasibility checker was developed and exported for incorporation into the TARGET IFD3 planning system. designed to verify consistency of the temporal constraints and force assignments in a given employment plan, and identify the set of conflicting constraints in inconsistent situations.

These auxiliary subsystems are summarized in the following two subsections. Complete details may be found in the DITOPS Design Reference Manual [LSS+93].

4.1 Integrating Capacity Analysis into SOCAP

The DITOPS capacity analysis service was developed to show (in collaboration with SRI) the utility of integrating resource contention analysis into higher level COA planning and was demonstrated within the SOCAP planner. It was constructed as a direct extension of the capacity analysis procedure utilized within the DITOPS scheduler itself. In brief, this base procedure operates by first computing an infinite capacity schedule (i.e., a schedule in which all temporal constraints are satisfied) at some specified level of time and resource granularity, and then relating the resource capacity required by the schedule to the amount of resource capacity that is actually available over time. Subintervals of the scheduling horizon in which the demand for capacity on some resource (or set of resources) exceeds the available supply are then identified and returned as likely scheduling bottlenecks. For use within SOCAP, a protocol for mapping COA plan nets into the DITOPS schedule representation (and likewise for communicating results back) was developed and integrated with the base capacity analysis procedure.

The capacity analyzer was incorporated into SOCAP as an additional "plan critic" for use after completion of the deployment planning phase of its overall COA generation process. Its use was demonstrated in the context of the original IFD2 (MEDCOM) problem scenario; upon generation of a deployment plan which assumed only a single in-theater POD, application of capacity analyzer was found to (correctly) identify the insufficiency of this one port

assumption from the standpoint of required port throughput capacity. Information relating to this detected port capacity bottleneck subsequently resulted in the triggering of a plan revision process in SOCAP, wherein a second in-theater POD was added to the plan. The overall point illustrated in this technology integration experiment was that consideration of capacity constraints early on in the planning process can lead to early detection of problems that, in current planning practices, might only be discovered after the initial deployment plan had been "exploded" to the detailed TPFDD level.

The DITOPS capacity analysis service is currently installed as a CPE knowledge service. In brief, it accepts a (typically high level) deployment plan along with a specification of available resources and resource capacity (i.e., ship, plane, port) identified as required in the plan. It returns as output, a set of any "bottleneck" resource intervals, and a resource usage profile for each resource over the plan horizon. The plan is communicated as a list of activities and temporal constraints. Each input activity description (corresponding to a node in the SOCAP plan net) contains the transport resource or resource type it requires, its POE, its POD, its cargo quantities (in terms of stons, mtons and/or pax), and any current bounds on its start time, end time, and duration. Each temporal constraint description identifies a binary temporal relation (e.g., before, same-start), the two activities that are constrained by the relation, and any quantitative bounds on the relation. Available transport resources are also communicated, with each resource description designating both an asset type (e.g., tairlift) and a set of instances (e.g., (1st-C130 2nd-C130 ...)). An additional set of inputs corresponds to periods of reduced available capacity for a given resource (transport asset or port), with each description indicating a specific resource, the amount of capacity lost, and the start and end time of the reduced capacity interval. Finally, parameters which establish the desired temporal granularity of the analysis and the threshold to be used in detecting bottlenecks are passed. The usage (or capacity) profile that is computed for each resource is returned as an ordered list of capacity intervals of the specified granularity (e.g., 24 hours), each of which specifies a start time, an end time, the projected demand for capacity over the interval, the available supply of capacity over the interval, and the demand/supply ratio. Any subsequence of intervals with a demand/supply ratio greater than the specified threshold is returned as an identified bottleneck.

4.2 COA Feasibility Checker

The COA Feasibility Checker subsystem was developed for use within the IFD3 TARGET system, and is currently a functioning component of this system. Similar in spirit to the capacity analyzer integrated into the SOCAP system, it performs feasibility checks on COA plans that are developed interactively within TARGET. However, there are important functional differences. First, the problem context is employment planning as opposed to deployment planning. In employment planning, forces are interpreted as the resources required by plan activities and whose availability must checked. Second, the qualitative temporal constraints on plan activities provided to the feasibility checker were not assumed to be consistent (as was the case with communicated SOCAP constraints); one objective of the feasibility checker is to provide guidance to the human planner in generating these temporal

constraints. Third, the objective is not to estimate resource contention per se, but to instead identify and isolate sets of conflicting constraints.

The COA Feasibility Checker integrates time bound propagation techniques defined within the kernel DITOPS infra-structure with newly developed extensions to recognize and diagnose specific types of constraint conflicts. When provided with an input plan from TARGET, a topological sorting procedure is used in conjunction with time bound propagation to first check the plan for the presence of *cycles*. Detection of a cycle in this case implies the existence of some set of inconsistent temporal relations (e.g., A before B, B before C, C before A). If detected, a characterization of this constraint conflict, including the set of temporal relations that are involved, are returned for use in highlighting to the user which constraints must be changed to achieve a feasible plan. In cases of a cycle-free plan, checks are also performed to detect *time bound violations*, which indicate that the metric constraints imposed on the plan (e.g., mission start and end dates, activity durations) are not feasible, and *resource availability violations*, which indicate situations where the resources required by an activity (in this case, forces) are not available during the activity's inferred time window. Upon detection of either type of conflict, a description identifying the activities and resources involved in the conflict is returned, again to provide guidance in directing the plan change process. If an input employment plan is found to be conflict free, then the inferred time bounds for each constituent activity are returned as output.

References

[CS93a] Cheng C. and S.F. Smith. Integrating ai and or algorithms to solve job shop scheduling problems. Working paper, The Robotics Institute, Carnegie Mellon University, October 1993.

[CS93b] Cheng C. and S.F. Smith. Large job shop scheduling to minimize makespan. Working paper, The Robotics Institute, Carnegie Mellon University, September 1993.

[LS93a] M. Lewis and K.P. Sycara. Reaching informed agreement in multi-specialist cooperation. *Group Decision and Negotiation*, 2:279–299, 1993.

[LS93b] J.S. Lui and K.P. Sycara. Distributed constraint satisfaction through constraint partition and coordinated reaction. In K.P Sycara, editor, *Proceedings 1993 Workshop on Distributed AI*, Hidden Valley, PA, May 1993.

[LSS+93] O. Lassila, S.F. Smith, K. Sycara, G. Amiri, M. Becker, and C. Young. The ditops design reference manual. Technical report, The Robotics Institute, Carnegie Mellon University, September 1993.

[LT91] O. Lassila and S. Torma. Using a distributed frame system to implement distributed problem solvers. Technical Report HTKK-TKO-B68, Department of Computer Science, Helsinki University of Technology, Finland, 1991.

[Sad94] N. Sadeh. The micro-boss factory scheduling system. In *Intelligent Scheduling*, chapter 4. Morgan Kaufmann Publishers, Palo Alto, CA, in press, 1994.

[SC93] Smith S.F. and C. Cheng. Slack-based heuristics for constraint satisfaction scheduling. In *Proceedings 11th National Conference on AI*, Washington DC, July 1993. Morgan Kaufmann.

[SL93] S.F. Smith and O. Lassila. Configurable systems for reactive production management. In *Proceedings IFIP TC 5 / WG 5.7 International Workshop on Knowledge-Based Reactive Scheduling*, Athens, Greece, October 1993.

[Smi92] D.R. Smith. Transformational approach to scheduling. Technical Report KES.U.92.2, Kestrel Institute, December 1992.

[Smi94] S.F. Smith. Opis: A methodology and architecture for reactive scheduling. In *Intelligent Scheduling*, chapter 2. Morgan Kaufmann Publishers, Palo Alto, CA, in press, 1994.

[SMS+91] J. Schank, M. Mattock, G. Sumner, I. Greenberg, J. Rothenberg, and J.P Stucker. A review of strategic mobility models and analysis. Technical Report R-3926-JS, Rand Corporation, 1991.