

# The RAPSIDY Repair Solver

Sven Brueckner, Van Parunak  
Altarum Institute

RAPSIDY (Resource Allocation Problem Solver Incorporating Dynamics) is a repair solver based on some of the concepts developed by the Altarum team during the ANT program. By “repair solver” we mean one that begins with an existing schedule and a set of new constraints and attempts to deliver a scheduled that incorporates the new constraints with minimal change to the previous schedule.

RAPSIDY interacts with the ISI SNAP system (Figure 1). It receives the definition of the current scheduling problem as well as user preferences and requirements and it delivers a set of assignments that (partially) solve the problem under the given constraints.

A scheduling problem as presented by SNAP (Figure 2) comprises a set of resources (pilots, planes, ranges, simulators, ...) and a set of tasks (missions). Tasks specify a number of requirements that all have to be met to fulfill the task. Requirements provide the constraints on the eligibility of resources to be assigned to this task. A task is divided into contiguous temporal segments, which jointly define the overall duration of the task. Ant requirement of the task may be associated with one or more of these segments and thus, a resource may be assigned to a task (through the specific requirement) for only parts of the overall task duration.

Many of the constraints on the allocation of resources to the requirements of a task at a given interval within the overall planning horizon are hidden from the operation of a solver (such as RAPSIDY) connected to SNAP. Instead, a solver explores the permissibility and value of potential task assignments through interactions with a domain Oracle inside the SNAP system. This arrangement has the advantage of avoiding the explicit representation of constraints arising in a specific domain within the solver. Instead, all the solver knows about is a general scheduling problem (assign resources to requirements at specific times) with the potential for failure of task configurations that are prohibited by the Oracle even though they meet the requirements of the general problem structure. On the other hand, the lack of the explicit representation of constraints leads to a significant overhead in the operation of the solver, since it needs to explore regions in the search space of the general problem even though they are excluded by the specific domain constraints. This overhead led to a

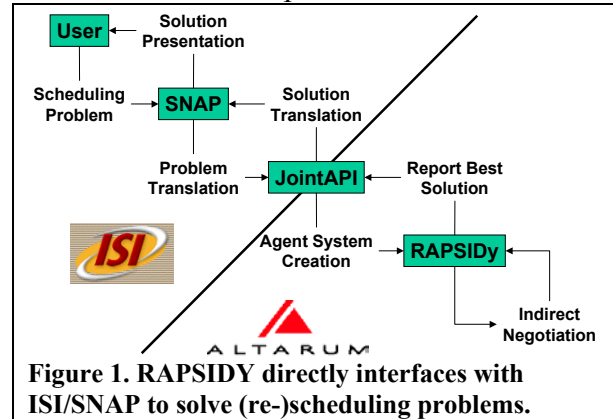


Figure 1. RAPSIDY directly interfaces with ISI/SNAP to solve (re-)scheduling problems.

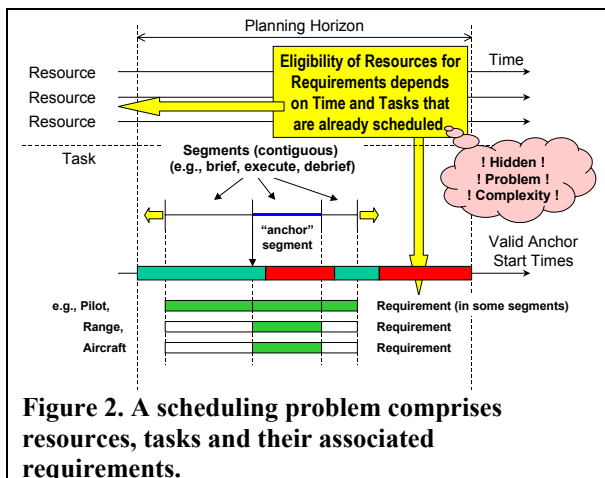


Figure 2. A scheduling problem comprises resources, tasks and their associated requirements.

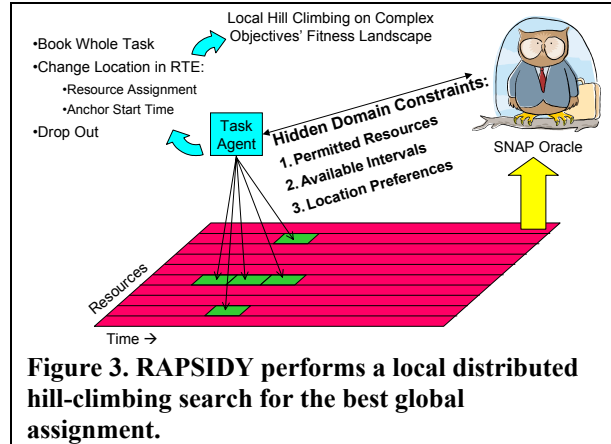
requirements at specific times) with the potential for failure of task configurations that are prohibited by the Oracle even though they meet the requirements of the general problem structure. On the other hand, the lack of the explicit representation of constraints leads to a significant overhead in the operation of the solver, since it needs to explore regions in the search space of the general problem even though they are excluded by the specific domain constraints. This overhead led to a

## The RAPSIDY Repair Solver

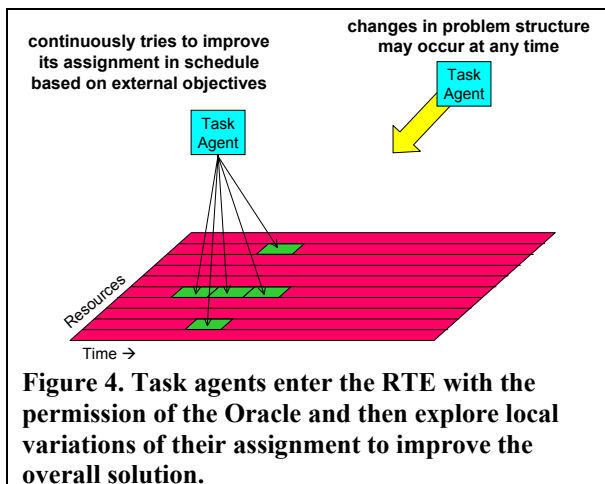
significant slow-down of the solver operation, especially in the region of problem space, where the abstract capacity of the system (availability pattern of resources) is near the abstract demand by the tasks (requirement pattern). In this case, the resulting pattern of permissible and forbidden configurations in search space is very complex and the Oracle needs longest in deciding whether a configuration is acceptable or not (peak in effort curve).

RAPSIDY implements a local distributed hill-climbing approach (Figure 3). Each task in the problem definition is assigned an agent. The agent's goal is to find an acceptable assignment for all its requirements within the space of resources and the planning horizon interval. This Resource-Timeline Environment (RTE) provides the shared environment of the task agents, in which they may experience the domain-independent scheduling constraints (e.g., tasks may not overlap) without consulting the Oracle. For all domain-specific constraints (e.g., eligibility of a resources for a requirement at a specific time), task agents interact with the Oracle.

Once the task agent has found an acceptable assignment in the RTE, it then explores local variations of this assignment to improve its evaluation by the Oracle with respect to the user preferences (Figure 4). Local variations of the assignment include changes in the temporal allocation of the task (slide up or down the timeline) or in the individual assignment of resources to specific requirements. At this point, RAPSIDY only considers user preferences with respect to a previously constructed schedule. Tasks seek to change their assignment towards configurations that match this "old" schedule. Thus, the RAPSIDY solver attempts to repair an existing schedule that needs to be modified either because new tasks have been added to the problem, or the resource availability has changed in some way. It is the goal of RAPSIDY to create a new schedule that resembles the old schedule as much as possible and thus minimizes the changeover costs involved with distributing the new schedule information.



**Figure 3. RAPSIDY performs a local distributed hill-climbing search for the best global assignment.**



**Figure 4. Task agents enter the RTE with the permission of the Oracle and then explore local variations of their assignment to improve the overall solution.**

We were able to compare the RAPSIDY solver with two other scheduling approaches constructed by other ANTS teams. Compared to the SerialCrawler, a greedy assignment approach with systematic backtracking, RAPSIDY performs slower but delivers better solutions especially for more complex problems. Alternatively, when compared to the PseudoBoolean solver, RAPSIDY's lack of detailed temporal reasoning reduces the quality of solutions to problems of extreme complexity. In

## The RAPSIDY Repair Solver

such problems, the space of acceptable solutions is reduced to single points rather than regions that could be explored by our solver. But RAPSIDY finds slightly less valuable solutions in much shorter time.

RAPSIDY was integrated with the ISI SNAP system and demonstrated at the 3 June 2003 scheduling workshop.