Neptune: A Mixed-Initiative Environment for Planning and Scheduling

Pauline M. Berry[1], Blazej Bulka[2], Bart Peintner[1], Mark Roberts[3], and Neil Yorke-Smith[1]

[1]Artificial Intelligence Center
SRI International
Menlo Park, CA 94025
{berry,peintner,nysmith}@ai.sri.com

[2]Dept. of Comp Science and Elec. Eng.
University of Maryland Baltimore County
Baltimore, MD 21250
bulka1@umbc.edu

[3]Computer Science Dept.
Colorado State University
Fort Collins, CO 80523
mroberts@cs.colostate.edu

Abstract

We describe the design of a user-centric, integrated planning and scheduling system that assists the user in exploring the rich space of plans and associated resource assignment options in complex, real-world domains. Each component of the system (user, planner, and scheduler) reacts to the actions of the other, resolving conflicts, and iteratively refining the solution until acceptable.

Introduction

Real-world problems frequently show the need for integration of planning and scheduling activities. Prominent examples include military operations, in which achieving strategic goals requires complex plans with significant scheduling constraints, and space operations, in which interdependent scientific and maintenance tasks vie for limited resources. Traditionally, these problems are approached as decomposable; highly efficient technologies have resulted for both planning and scheduling in isolation. However, in many domains, including manufacturing, space, and service industries, these processes are in fact highly interdependent. In addition, existing technologies exhibit considerable degrees of automation that obscure the processes from users, and removes users' ability to control or influence solutions. This is especially true in the case of complex, knowledge-rich, multi-faceted domains where significant user control is required to address imperfect domain knowledge. Exploiting advances in the integration of planning and scheduling [Myers et al. '01, Rodriguez-Moreno et al. '06] and mixed-initiative planning or scheduling systems [Myers et al. '02, Smith et al. '05] we present a new approach to such complex domains.

Neptune retains the strength of human experts in understanding their domain, its nuances, and the context of their goals, while leveraging the power and flexibility of tightly integrated planning and scheduling algorithms. The system is user-centric, allowing the user to view, understand, and impact the underlying processes. Having given the system a set of initial goals, the user absorbs the developing solution, directing it with general and specific feedback until the result is acceptable.

In this short paper, we describe how Neptune's planner, scheduler, and user work autonomously, each reacting to the results developed by the others. We details the conflict-directed mechanism by which the planner and scheduler work together to refine the solution as the user directs.

The Neptune Components and Interactions

Four main components compose Neptune: a hierarchical task network (HTN) planner, a constraint-based scheduling engine [Berry et al. '07], the user interface (UI), and the user. In addition, a process management component facilitates the ongoing work. The UI should be understood as being the conduit through which three autonomous, cooperative agents (the planner, the scheduler, and the user) work on a common problem continuously and in parallel, each reacting to the operations of the others.

As the planner expands HTN templates to create a sequence of partial plans from the initial goals, the scheduler uses resource summaries propagated from primitive plan nodes to find schedules and conflicts for each partial plan. The planner tailors its search through the plan space to the actions of the user and the scheduler: it focuses on plans near the plan currently being viewed and manipulated by the user, and it actively works to develop plans that resolve resource and temporal conflicts uncovered by the scheduler.

The scheduler accepts both specific and high-level guidance from the user while updating the schedule and conflicts for each new plan branch created by the planner. For example, the user can request lower utilization of a particular resource or can state that balancing resource usage is more important than other high-level criteria.

While the planner and scheduler work in the background to explore the plan space, the user can inspect single plans from the space uncovered so far. Each node in a plan is annotated with results from the scheduler, including whether the current instantiation or any development of the node leads to a feasible plan. Easily determining which nodes participate in conflicts, the user can focus on selecting among the suggested resolutions, leaving the details of planning and scheduling to the rest of the system.

Figure 1 shows a small portion of the UI: the top half shows part of the HTN plan display, and the bottom half shows one of several views of the schedule associated with the displayed plan. The user can quickly focus on the conflicted elements of the solution (colored red and pink), and begin addressing any conflicts. The user may right-click on any element to give specific instructions to the planner or scheduler. In Figure 1, for example, the context menu shown allows the user to “Lock activity to this time” or “Do not schedule this activity”. The latter will tell the planner to avoid any expansion that includes this activity.

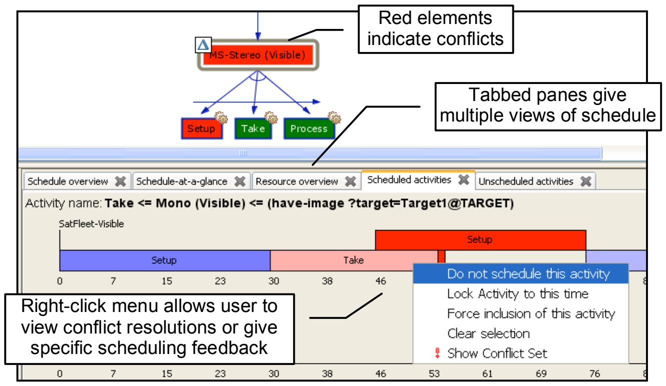
Detecting Conflicts and Providing Resolutions

The philosophy of Neptune is to leverage the strengths of automated planning and scheduling technologies for the minutiae of the solution crafting process, while providing transparency and deference to the user for strategic and informed decisions. Thus, with user included, the whole system overcomes the common problem of domain models that do not fully capture the nuances of the real world situation they model, and trust is built for the solutions.

The Neptune user may manually expand plan nodes, or allow the system to make decisions. The other components pre-explore possible expansions and use the discoveries to make available consequences of decisions to the user, and to suggest recommended decisions.

Nonetheless, the user is not prevented from pursuing a direction marked as a dead end. If the user continues down such a path, the planner and scheduler work together to generate all conflicts in the solution related to this decision, as well as possible resolutions for them. For example, Figure 1 shows a conflict between a Setup task in one part of the plan (the HTN sub-tree shown in plan view) and the Take and Process tasks in another part (not shown). The conflict concerns over-use of the SatFleet-Visible resource. Other conflicts include temporal constraint conflicts, and logical plan conflicts (not supported yet).

There are two possible resolutions for the conflict in Figure 1. The HTN tree shown can be unexpanded, and a sibling of MS-Stereo (Visible) can be used instead. Alternatively, the HTN sub-tree not shown can be expanded in a way that uses another resource. To generate these resolutions, the planner examines each scheduling conflict, and iterates through each task it contains. For each task, the planner performs a local search near the expansion that created it. Each partial plan searched is evaluated by the scheduler, which determines whether the


Red elements indicate conflicts
MS-Stereo (Visible)
Setup
Take
Process
Tabbed panes give multiple views of schedule
Schedule overview
Schedule-at-a-glance
Resource overview
Scheduled activities
Unscheduled activities
SatFleet-Visible
Setup
Setup
Take
0 7 15 23 30 38 46
Do not schedule this activity
Lock Activity to this time
Force inclusion of this activity
Clear selection
Show Conflict Set
Right-click menu allows user to view conflict resolutions or give specific scheduling feedback
0 7 15 23 30 38 46 53 61 69 76

Activity name: **Take <= Mono (Visible) <= (have-image ?target=Target1@TARGET)**

Figure 1: The user may view, understand, and address conflicts in both the planning and scheduling parts of the UI.

conflict is resolved (and whether new conflicts have emerged). The search on each iteration ends when a conflict-free plan (now called a *resolution*) is found. The resolution is attached to the plan and the process continues.

At any time, the conflicts and resolutions found so far are available to the user. Resolutions are ordered heuristically for the user to view. The user can preview the consequences of each before picking one, or permit the conflict to remain. Thus, the executive decision of how to resolve a conflict is left to the domain expert, while the task of working through the details is undertaken by the other components even before the user asks for them.

Future Directions

The degree of interaction offered to the user combined with the granularity of interaction enabled between planner and scheduler sets Neptune apart from systems such as COMIREM, IPSS, and MAPGEN. For example, Neptune can schedule both hierarchical plans and partially expanded hierarchical plans; COMIREM requires fully expanded plans and MAPGEN does not handle hierarchical plans at all. However, much work is left to explore the interaction between the user, planner, and scheduler. Experience with several real-world domains is required before we can claim our methodology is effective in practice. We must incorporate uncertainty into our domain models, in task durations, resource usage, and task success. Finally, our evolving implementation does not yet scale to large domains; engineering is needed to incorporate incremental versions of each algorithm.

References

Berry, P. M.; Moffitt, M. D.; Peintner, B.; and Yorke-Smith, N. 2007. The Design of a User-Centric Scheduling System for Multi-Faceted Real-World Problems. In: *Proc. of ICAPS’07 Workshop on Moving Planning and Scheduling Systems into the Real World*.

Myers, K. L.; Smith, S.; Hildum, D.; Jarvis, P. A.; and de Lacaze, R. 2001. Integrating Planning and Scheduling through Intensity Adaptation. In: *Proc. of IJCAI-01 Workshop on Planning with Resources*.

Myers, K. L.; Tyson, W. M.; Wolverton, M. J.; Jarvis, P. A.; Lee, T. J.; and desJardins, M. 2002. PASSAT: A User-centric Planning Framework. In: *Proc. of Third Intl. NASA Workshop on Planning and Scheduling for Space*.

Rodriguez-Moreno, M. D.; Borrajo, D.; and Cesta, A. 2006. IPSS: A Hybrid Approach to Planning and Scheduling Integration. *IEEE Trans. on Knowledge and Data Engineering* 18(12).

Smith, S.; Hildum, D.; and Crimm, D. R. 2005. Comirem: an intelligent form for resource management. *IEEE Intelligent Systems* 20(2).