UMCP: A Sound and Complete Procedure for Hierarchical Task-Network Planning

Available at

http://www.cs.umd.edu/~nau/papers/erol1994umcp.pdf

Hierarchical Task-Network Planning (HTN) is different than the STRIPS that we learn at
the course as HTN represents the desireble change that we want in the modeled world,
instead of a search to a unique goal, as in STRIPS.

Despite the previous works describing HTN planners before this paper any of them
presented a structured a clear and concise algorithm to solve this problems.
This clear and concise algorithm is present in this paper as follows:

HTN replaces STRIPS goals to tasks and task networks, with is more powerful,
but increases the complexity. There are three types of tasks:

1. *Goal Tasks*, like the Goal in STRIPS, are properties in the model that
   we want to turn into true, e.g. At(Cargo1, JFK) as

shown in the AIND-Planning project

2. *Primitive tasks*, are tasks that we can achieve by executing actions,
   like Fly() on the AIND-Planning project
3. *Compound tasks*, are tasks that cannot be described as a single task,
   but a network of *Goal Tasks* or *Primitive Tasks", e.g.
   “Deliver Cargo 1 at JFK” is the result of a execution of several Load/Unload/Fly/Goal tasks.

Tasks are connected through the Task Network, and a problem is solved following
these steps:

1. Input the Planning Problem P (the task network)
2. If P contains only primitive tasks, then resolve the conflicts in P and
   return the result. If the conflicts cannot be resolved, return failure.
3. Choose a non-primitive task t in P
4. Choose an expansion for t
5. Replace t with the expansion
6. Use critics to find the interactions among the tasks in P, and suggest ways
   to handle them.
7. Apply one of the ways suggested in step 6.
8. Go to step 2.

The paper proposes the semantic and syntax for expressing HTN problems and
demonstrates how implement basic expansions and reductions, as future work
the researchers plan to study the application of HTN solvers in various domains.