Challenges and Solutions for Hierarchical Task Network Planning in E-Learning

Carsten Ullrich[a], Okhtay Ilghami[b]
[a]*DFKI GmbH, Germany,* [b]*University of Maryland, USA*

1. Motivation

This paper describes a collaboration between two PhD students. The first author is developing a course generator (CG) for a Web-based e-learning environments (WBLE) [3]. A CG generates sequences of learning objects according to the learner's goals and individual properties. The assembling process is knowledge intensive: assembling courses that implement modern pedagogical theories requires a framework that allows to represent complex pedagogical strategies. The strategies are implemented using the Hierarchical Task Network (HTN) planner JSHOP2 [1], developed by the second author. In HTN planning, the goal of the planner is to achieve an ordered list of top tasks, where each task is a symbolic representation of an activity to be performed. The planner formulates a plan by applying methods that decompose these top tasks into smaller and smaller subtasks until *primitive* tasks are reached that can be carried out directly.

When we applied the HTN framework to WBLE, we encountered several challenges, which we will describe in this paper, together with our solutions. We believe that this work is of general interest, as today's WBLE are excellent examples of distributed information systems and illustrate the challenges that can arise in complex scenarios.

2. Challenges and Solutions

Vast Amounts of Resources Typically, a WBLE uses vast amounts of learning objects (LOs), potentially distributed over distinct repositories, which each may contain thousands of LOs. This creates a problem because a CG needs to reason about the LOs but traditional AI planning requires evaluating a method's precondition against the planner's world state. In a naive approach, this would require mirroring in the world state all the information available about the resources in all the repositories. In real world applications, this is simply infeasible. Additionally, only a subset of all the stored resources may be relevant for the planning, but which subset is unknown beforehand.

A possible solution is to access the information on-demand by using external functions. An external function serves to calculate information not directly available in the world state using procedures not native to the planning algorithm. Thus, instead of the operator's (or method's) preconditions to match against the logical atoms that make up the world state, they invoke function calls that return possible substitutions.

Distributed and Heterogeneous Resources To make things even more difficult, despite standardization efforts, each repository often uses a (at least partly) different knowledge representation format. In Information Integration, these difficulties are tackled using a mediator architecture: a mediator acts as a link between the application and resource layer, thus providing a uniform query interface to a multitude of autonomous data sources. It translates queries for a specified set of connected repositories and passes the translated queries to the repositories.

The advantage of the mediating component is that the querying component, i.e., the planner, does not need to know the specific specification of the data sources and their query languages, because the mediator translates the queries. The planner simply accesses the mediator using an external function. However, one needs an abstract representation of the resources (a mediated representation) used during planning, and mappings to the representations used in the repositories. For CG, we developed such a representation [2].

Third-Party Services In WBLE, a vast range of services that support the learning process in various ways have been developed. A course should integrate these services in a pedagogically sensible way: during the learning process, at specific times the usage of a tool will be more beneficial than at some other time.

The problem is that in a Web-based environment, availability of services may vary. However, one wants to avoid using different domain descriptions for each potential configuration. Therefore, the methods need to encode in their preconditions whether a service is available, which is easily realizable by using external functions. In case the service is not available and the plan generation should not fail, fallback methods should be specified that specify alternative actions. The advantage of adding fallback methods is that the domain description remains reusable, regardless of the actual configuration. In the case of course generation, several methods encode the knowledge at what time in the course the learner preferably should use a learning supporting tool, and insert corresponding calls to the tool or simply text at the appropriate place in the course.

3. Conclusion

The HTN framework was used to implement pedagogical strategies formulated by didactical experts. The experts were very comfortable with expressing their knowledge in an hierarchical manner, which eased the formalization into planning operators and methods. The current implementation consists of about 250 methods, 20 operators, and 40 axioms. A technical analysis yielded satisfactory results: generating a course takes between several seconds and minute, which is acceptable. The CG is already used in several schools and universities and evaluations assessing its pedagogical effectiveness are underway.

To conclude, a principal characteristic of our solutions to the described challenges is that they do not require extending the planning algorithm, as they are built "on top" of the algorithm. The extensive use of external functions allows us to access information stored in large and heterogeneous resources whenever it is necessary, and to flexibly integrate learning supporting tools.

References

[1] O. Ilghami. Documentation for JSHOP2. Technical Report CS-TR-4694, Department of Computer Science, University of Maryland, February 2005.

[2] C. Ullrich. The learning-resource-type is dead, long live the learning- resource-type! *Learning Objects and Learning Designs*, 1(1):7–15, 2005.

[3] C. Ullrich. Course generation based on HTN planning. *Proc. of 13th Workshop of the SIG Adaptivity and User Modeling in Interactive Systems*, pages 74–79, 2005.