HDL Systems in the Robotics Domain

In the previous chapter, the terminological concept of HTN planning was defined. In order to generate a valid HTN planning problem for SHOP2, a terminological concept that models the environment needs to be defined. In addition, the HTN planning operators, actors and objects must be instantiated and inserted into the corresponding ABox. In this chapter, a generic method for modelling the environment in HDL and filling the HDL system is introduced. Two implementations from the robotics domain are also presented, namely the robot navigation domain and the pick-and-place domain.

3.1 Modelling the HTN Planning Problem in the HDL System

There is no fixed method for describing terminological concepts in the knowledge base. However, a rule of thumb is that such a method should be refined through several iterations. In the previous chapter, the terminologies for representing HTN planning in the DL representation were defined. However, the instances of the planning itself are not yet modelled.

Describing a planning problem for the planning system is not trivial. It requires a process involving use cases and testing similar to that followed in the software engineering discipline. Therefore, a method for modelling HTN planning problems in the HDL system has been defined and successfully tested over several use cases.

This method has 7 steps and is described below:

1. Define the actions and objectives
   This first step reduces the size of the problem by clearly defining the objectives. They are modelled by simple actions which will achieve them. This step thus requires us to define the test domain where the actions should take place.
2. Define the task networks
   As previously described in Section 2.2.3, HTN planning is a heuristic approach. The heuristic comes from human knowledge. The result is a plan that is very intuitive in that it solves the problem in a manner that a human being would. In the previous step, some actions and the objectives were defined. In this step the task networks for achieving the objectives is built.

3. Program the planning domain description
   This step instantiates the network, defined in the previous step, into a particular planning-understandable syntax, in this case SHOP2. The relations between methods and operators are connected through pre- and post-conditions. The post-conditions might change some particular states by removing or adding some facts.

4. Test the planning domain
   It is necessary to test the planning domain description to validate its results. If it does not successfully extract a plan for a simple problem or if some error occurs, it is necessary to repeat the previous step(s). Only if a working planning domain description, that runs under the test cases, is found, it is possible to proceed to the next step.

5. Define the HTN ABox in the DL system
   Once a valid planning domain is fixed, its description in DL representation can be inserted into the ABox. This can be done by describing parts of methods or operators. However, it is necessary to capture and model the relationship of each of them such that it can be derived later automatically using Algorithms 2.1 and 2.2. It is also possible to define a complete planning-domain that involves the particular methods and operators.

6. Modelling and instantiating the states in the HDL system
   The states of the planning problem need to be conceptualised in the TBox of the DL system. The real problem can then be instantiated in the HDL system.

7. Testing the HDL system
   The final step tests the HDL system on its ability to generate a correct planning domain and planning problem.

Steps 1 to 4 above quickly describe the process used in designing and writing planning problems for existing planning systems. Three additional steps are needed to translate the problem into DL representation such that it can be used by the HDL system. These steps provide a rough guide to successfully modelling a planning problem in the HDL system. One might be tempted to model the planning problem directly in DL and then try it. If a problem appears though, it might be more difficult to find the root of the error.