Shopper: A System for Executing and Simulating Expressive Plans

John Maraist and **Robert P. Goldman**
SIFT, LLC
211 North First Street, Suite 300
Minneapolis, MN 55401-1480
{jmaraist,rpgoldman}@sift.info

Abstract

In this demo we present Shopper, a plan execution engine that facilitates experimental evaluation of plans. Shopper interprets the LTML plan language, which extends PDDL in two major ways: with more expressive control structures, and with support for semantic web services marked up in OWL-S. LTML's command structures include not only conventional ones such as branching, iteration, and procedure calls, but also features needed to handle HTN plans, such as precondition-filtered method choice. Unlike conventional programming languages, LTML supports interaction with the agent's belief store, so that its execution semantics line up with those assumed by planners. LTML actions extend PDDL actions in having outputs as well as effects, which means that they can support actions that sense the world; an important special case of this is semantic web services, which reveal information about a state hidden from the agent. To support experimentation as well as action in the real world, Shopper accommodates multiple, swappable implementations of its primitive action API. For example, one may interact with real web services through SOAP and WSDL, or with simulated web services through local procedure calls. In our demonstration we will show what LTML plans look like, relating them to their PDDL ancestors. We will show Shopper interpreting a plan whose individual steps are web service invocations. Then we will show how one may program simulated web services (or other actions) for use in experimentation, and show how Shopper operates with this simulation.