

# HTN Planning for the Web Service Composition Using SHOP2

*Summary by:* Ben Rockstroh

July 15, 2009

Web Service Composition

OWL-S

Composite Example

Process Properties

SHOP2 HTN planning system

Translate from OWL-S to SHOP2

Problems Encountered

Use with the Semantic Web

Future Work

# Web Services Composition

Sometimes thought of as the process of generating a (potentially) complex process description which is to be executed.

**Goal:** Find a collection of atomic processes which form an execution path for a composite process.

# OWL-S Process Ontology

## Processes

- ▶ atomic - self contained steps, Web Services.
  - ▶ A web service that verifies log on credentials.
- ▶ simple - abstract view of an atomic or simple process
- ▶ composite - made up of atomic and simple processes.
  - ▶ Amazon.com - collection of web services provided for purchasing a book.

# Amazon.com Composite Process

- ▶ If LogOn is successful
- ▶ then show logon screen
- ▶ else report failed logon

# Process Properties

- ▶ preconditions - Physical conditions that must be true before execution.
- ▶ inputs (*Optional*) - knowledge of preconditions
- ▶ effects (*Conditional*) - physical side-effects of execution.
- ▶ outputs (*Conditional*) - knowledge of effects.

# Definitions

**HTN** - Hierarchical Task Network

**SHOP2** - domain-independent HTN planning system.

SHOP2 differs from other planning systems in that it plans tasks in the same order they will be executed. This allows for the planner to know all environment variables at time planning.

# Translation from OWL-S to SHOP2

## Assumptions

- ▶ Process can either have effects or outputs, but not both.
- ▶ SHOP2 does not handle concurrency  $\Rightarrow$  OWL-S's **Split** and **Split+Join** controls can not be present in the plan.
- ▶ Preconditions for all atomic processes have been satisfied at time of execution.



# Problems Encountered

**Nested If-Then-Else** treat each control construct within a composite process as a separate composite process.

**Sequence** - Process  $K_1$  output is process  $K_2$  input. SHOP2 has no concept of output, this is solved by treating all outputs as knowledge effects. **OWL-S 1.0** - Does not have concrete specifications for preconditions and effects. This is solved with OWL-S 1.1

## Using with Semantic Web KB

SHOP2 can be used for *simple* HTN planning with Web Services described by OWL-S. **Problem:** SHOP2's interfering capabilities are not currently capable of working with KBs as large as the Semantic Web.

**Solution:** Replace SHOP2 theorem prover with one capable of working with OWL DL.

## Future Work

Currently the world is assumed to be static. Preconditions are checked at most once before each step. i.e. All effects are considered physical KB does not change.  
Concurrency is not supported by SHOP2