Automated Composition of Semantic Web Services into Executable Processes

P. Traverso and M. Pistore
ITC-IRST - University of Trento
traverso@itc.it - pistore@dit.unitn.it

Summary of Article:

One of the largest obstacles for the automation of web services, as a composition of services that allows for the decrease in efforts, time, and cost in their manual development, integration, and maintenance, is finding a way to handle this planning problem, which is far from trivial. At the time of this article (2004), there were two approaches to tackle this problem. First, most of the giants of industry proposed a low level process modeling and execution language. The benefit to this approach was that it allowed programmers to create complex web services as distributed processes and to connect them in a simple way: like simple conditionals and loops. The drawback was that new processes/definition that interact/link to already defined components had to be manually programmed in, this is not only challenging but time consuming and error prone. Second, research from within the Semantic Web community proposed a top-down unambiguous description of web services capabilities. The benefit of this approach is that it enables the use of reasoning in web services, and to automate web service tasks such as discovery and composition. They major challenge is the ability to automatically compose services that can directly execute, if this was achieved, it would greatly reduce effort, time, and errors dues to manual composition.

This paper tackles such a task through automating composition of web services described in OWL-S, which allows for the automated generation of executable process. Goals of services are automatically generated and described by the EaGle language, which provides clear semantic that express complex requirements. They handle the planning problem using the "Planning as Model Checking", which takes a formal model of a system and compares it against a logical specification of its requirements to discover inconsistencies. This framework offers partial observability at each plan execution step, the planner than has to consider all the sets of domain states, which are all plausible states based from the initial knowledge. Planning in this framework requires searching through all possible evolutions of initial state, in order to retrieve a conditional course of action that leads to the goal state. This planning algorithm generates plans that are automata (self-operating machine). They further implemented these techniques and preformed experimental evaluation. The results indicate that automated composition at the high-level is orders of magnitude faster than execution at the lower level.