SEMAPLAN: Combining Planning with Semantic Matching to Achieve Web Service Composition

Rama Akkiraju[1], Biplav Srivastava[2], Anca-Andreea Ivan[1], Richard Goodwin[1], Tanveer Syeda-Mahmood[3]
[1]IBM T. J. Watson Research Center, 19 Skyline Drive, Hawthorne, NY, 10532, USA
[2]IBM India Research Laboratory, Block 1, IIT Campus, Hauz Khaus, New Delhi, 11016, India
[3]IBM Almaden Research Center, 650 Harry Road, San Jose, CA 95120, USA
{akkiraju@us, sbiplav@in, ancaivan@us, rgoodwin@us, stf@almaden}.ibm.com

Abstract

In this paper, we present a novel algorithm to compose Web services in the presence of semantic ambiguity by combining semantic matching and AI planning algorithms. We use cues from domain-independent and domain-specific ontologies to compute an overall semantic similarity score between ambiguous terms. This semantic similarity score is used by AI planning algorithms to guide the searching process when composing services. Experimental results indicate that planning with semantic matching produces better results than planning or semantic matching alone. The solution is suitable for semi-automated composition tools or directory browsers.

A Motivating Scenario[1]

We present a scenario from the knowledge management domain to illustrate the need for (semi) automatic composition of Web services. For example, if a user would like to identify names of authors in a given document, text annotators such as a *Tokenizer*, which identifies tokens, a *LexicalAnalyzer*, which identifies parts of speech, and a *NamedEntityRecognizer*, which identifies references to people and things etc. could be composed to meet the request. Figure 1 summarizes this composition flow. Such dynamic composition of functionality, represented as Web services, saves tedious development time since explicating all possible and meaningful combinations of annotators can be prohibitive. AI Planning algorithms are well suited to generate these types of compositions.

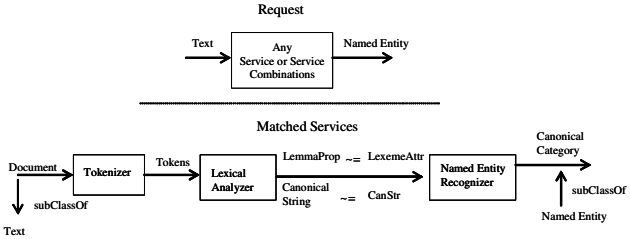

Figure 1. Text Analysis Composition example with semantic matching (~= illustrates semantic match)

However, unlike most planning problems, in business domains often it can not be assumed that Web services are described using terms from a single domain theory. For example, the term *lexemeAttr* may not match with l*emmaProp* unless the word is split into l*exeme* and *Attr* and matched separately using synonyms and word abbreviation expansion. In the absence of such semantic cues, two services that have these terms would go unmatched during planning thereby resulting in fewer results which adversely impacts recall. In the next section, we explain how we enable a planner to use these cues to resolve semantic ambiguities in our system -SEMAPLAN.

Combining Semantic Matching AI with Planning for Web Service Composition

Figure 2 illustrates the components and the control flow in SEMAPLAN system. Details of SEMAPLAN system are described below.

Service Representation. Service representation involves preparing Web Services with semantic annotations and readying the domain dependent and independent ontologies. We use WSDL-S (Akkiraju et al., 2005) to annotate Web Services written in WSDL with semantic concepts from domain ontologies that are represented in OWL (http://www.w3.org/TR/owl-features/).

Term Relationship Indexing. After semantic annotation, the available Web services in the repository are parsed, processed and an efficient index is created consisting of related terms/concepts referred to in the service interface descriptions for easy lookup. This is achieved using the services of a *semantic matcher* which uses both domain-independent and domain-specific cues to discover similarity between application interface concepts.. To achieve domain independent similarity we use techniques similar to the one in (Dong 2004). This is done by tokenizing the words, tagging them via part-of-speech, stop word filtering and abbreviation expansion and then consulting a dictionary to find similar words (eg: synonyms, hyponyms and hypernyms). More details are in (Syeda-Mahmood et al., 2005). For finding related terms using domain-specific ontologies, we use relations such as *subClassOf(A,B), subClassOf(B,A)*, *typeOf (A,B)*, and *equivalenceClass(A,B)* in domain ontologies represented in OWL (OWL 2002). The scores from two sources are then combined to obtain an overall semantic score for a given pair of concepts. As a result an efficient index is created

from this semantic matching which we call a s*emantic similarity map*.

Prefiltering. Once indexing of related concepts is accomplished, we perform prefiltering using heuristics to obtain a list of candidate matching services for a given request. We implemented a simple backward searching algorithm to select candidate services in the prefiltering stage.

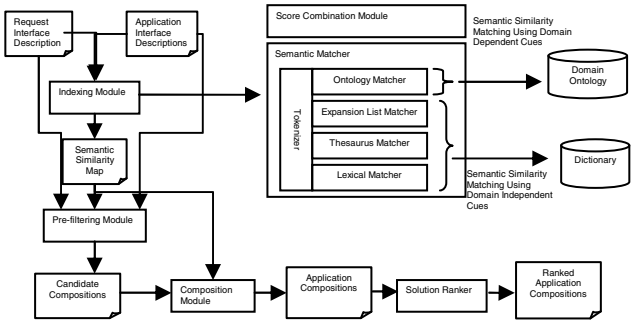

Figure 2. SEMAPLAN system and its components

Generating Compositions using Metric Planner. The candidate application interfaces from pre-filtering stage are passed to a *metric planner* along with the request interface description, and the *semantic similarity map*. A metric planning problem is a planning problem where actions can incur different costs. A metric planner finds plans that not only satisfies the goal but also does in lesser cost. We have a number of choices for calculating the plan cost. In selecting the semantic cost of the action, schemes such as min, max, aggregate of the constituents can be used. In computing the semantic cost of the plan addition, multiplication schemes are possible. These mechanisms are configurable in SEMAPLAN. We have implemented such a metric planner in the Java-based Planner4J framework (Srivastava 2003).

Solution Ranking. Finally, the alternative compositions generated by the planner are ranked by the *ranker.*

Experimental Results

We ran several experiments on a collection of over 100 Web services. Keeping space considerations in mind, we report the result of the first experiment. Details of other experiments are available in (Akkiraju et al., 2006).

Metric Planner alone Vs. SEMAPLAN. In this experiment, our hypothesis was that a planner with semantic inferencing would produce more relevant compositions than a planner alone. The intuition is that the semantic matcher allows concepts such as *lexemeAttr* and *lemmaProp* to be considered matches because it considers relationships such as word tokenization, synonyms, and other closely related concepts (such as *subClassOf, typeOf, instanceOf, equivalentClass*) defined by the domain ontologies; such relationships are not usually considered by the planner. As Figure 3 shows, SEMAPLAN finds more relevant results than a classic metric planner, thus confirming our hypothesis.

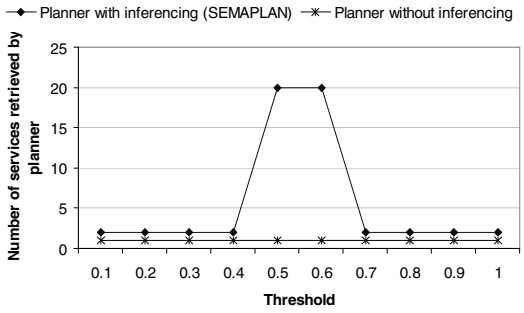

Figure 3: Comparison of Metric Planner Vs. SEMAPLAN

The increased number of solutions is more prevalent with certain semantic thresholds as planner search space is optimized at these thresholds.

Conclusions

In this paper, we presented SEMAPLAN, a novel approach and system to compose Web services in the presence of semantic ambiguity using a combination of semantic matching and AI planning algorithms. The experimental results confirmed our intuition that planning with semantic matching produces more as well as better (when optimal semantic threshold is set) results than using a planning alone. The notion of planning in the presence of semantic ambiguity is conceptually similar to planning under uncertainty. In the future we intend to investigate the application of probabilistic planning techniques to consider semantic differences and compare the results.

References

Akkiraju R, Farrell J, Miller, et al. 2005 Web Services Semantics - WSDL-S. A W3C submission.

Akkiraju R, Srivastava, B, et al. 2006. SEMAPLAN: Combining Planning with Semantic Matching to Achieve Web Service Composition, IBM Research Report

Dong X. et al. 2004. Similarity search for Web services. *In Proc. VLDB, pp.372-283*, Toronto, CA.

Srivastava B. 2004. A Framework for Building Planners. In the Proc. Knowledge Based Computer Systems.

Syeda-Mahmood T., Shah G., et al. 2005. Searching Service Repositories by Combining Semantic and Ontological Matching. ICWS 2005. Florida.