Towards Automated Service Matchmaking and Planning for Multi-Agent Systems with OWL-S – Approach and Challenges

Johannes Fähndrich, Nils Masuch, Hilmi Yildirim, and Sahin Albayrak

DAI-Labor, TU Berlin
Ernst-Reuter-Platz 7, 10587 Berlin, Germany
forename.surname@dai-labor.de

Abstract. In the past, the demand for modular, distributed and dynamic computer systems has increased rapidly. In the field of multi-agent systems (MAS) many of the current approaches try to account for these requirements. In this paper we discuss the shortcomings of the semantic service selection component SeMa^2, propose improvements and describe an integration concept into a multi-agent framework. Further, we illustrate how this system can be extended by an automated service composition component using methods from the AI planning community.

Keywords: OWL-S, Automated Service Selection, Automated Service Composition, Planning, Multi-Agent Systems.

1 Introduction

Distributed systems based on the Service Oriented Architecture (SOA) paradigm have become more and more popular in recent years. One of its inherent strengths is the definition of a clear autonomy of each service, which means that it is represented as a separate module. Further, services are designed for enhancing the reusability as well as the interoperability which is one of the key issues for distributed systems. Especially when talking about huge computer systems with different providers and parties involved these attributes are essential.

In order to cope with dynamic aspects in huge systems, such as the immediate (dis-)appearance of services, solutions to adapt the process via an automated service selection and composition are desirable. As a first step, service matching techniques have been developed that enable the automated selection of services. However, this is not enough when the system has to deal with complex goals, where the involvement of different services is necessary. In this case, the system needs some form of automated service composition solution, which can also be interpreted as planning. In the area of multi-agent systems and AI in general there has been done research leading to approaches, such as hierarchical task networks (HTN) or STRIPS.

In this paper we propose to combine semantic service technologies of the SOA community with the planning techniques of the AI community. We do so by using our semantic service matchmaking component *SeMa*^{2} [12] as a fundament and discuss the

A.R. Lomuscio et al. (Eds.): ICSOC 2013 Workshops, LNCS 8377, pp. 240–247, 2014.

adaptions necessary to set up an HTN planning component, which is capable of being integrated into a service-oriented multi-agent framework.

The remainder of the paper is structured as follows. In section 2 we will shortly present the current status of our service matching component *SeMa*^{2} and provide new concepts for its improvement in detail. In section 3 we describe our concept of extending *SeMa*^{2} by a planning component integrated into a multi-agent system. Section 4 presents the related work in automated service composition. Finally, we close with a conclusion.

2 Automated Service Matchmaking - The SeMa^2 Approach

The service matcher *SeMa*^{2} follows a hybrid approach combining logic-based and non-logic-based matching techniques using OWL-S and SWRL. Figure 1 shows all relevant components, for example the *OWLS-ServiceAnalyzer* as the document parser/writer and the *MatcherController* which triggers all different matching techniques and aggregates them to a single result. As for the non-logic-based evaluation *SeMa*^{2} processes syntactical comparison on service names (*ServiceName Matcher*) and service descriptions (*TextSimilarity Matcher*) based upon well-known lexicographic techniques, such as Jaccard index or Hamming distance. Further, three different approaches are used for logic-based matching, namely the *Taxonomy Matcher*, the *RuleStructure Matcher* and the *Rule Evaluator*.

All these results are combined via linear weighted aggregation, with no adaptability so far. At the S3 Contest 2012 *SeMa*^{2} performed well regarding the precision coming with the best matching accuracy in graded relevance ranking. However, at the contest Rule Evaluation (due to missing ABox information) and RuleStructure Matching (due to syntactic incompatibility of the SWRL services) were not integrated. Internal tests with modified service descriptions have shown, that the integration of rule structure matching has even a minimal negative influence on the results lowering the average precision based on the nDCG-measure from 92,7% to 92,1%. Since there is no obvious reason for that and we consider rule matching as an important part for the matcher to be used in a planning component we decided to formalize our approach at first and then focus on improving the aggregation concept of the different matching techniques.

2.1 Scoring and Aggregation of Different Matching Techniques

Due to the best-first search we are aiming to use in our planning component, the results of the Precondition and Effect (PE) matching are crucial. The rule structure component has multiple matching layers and thus has many decision points on how to rate full, partial or other matches. A further challenge is the assignment of weights to the different matching results to achieve a single score for a service. Each scoring for the equivalence of two concepts can be seen as an expert opinion assigning a probability to the match. For example there could be an expert on *semantic distance based scoring* and one on *logic based scoring*. The resulting probability of equivalence is published to be used by an aggregation method. Doing so, the scorings of an expert will be formalized as $p_i(R,S) \in [0,1]$ with R,S being a request and a service and $i \in \{1,\dots,n\}$ representing

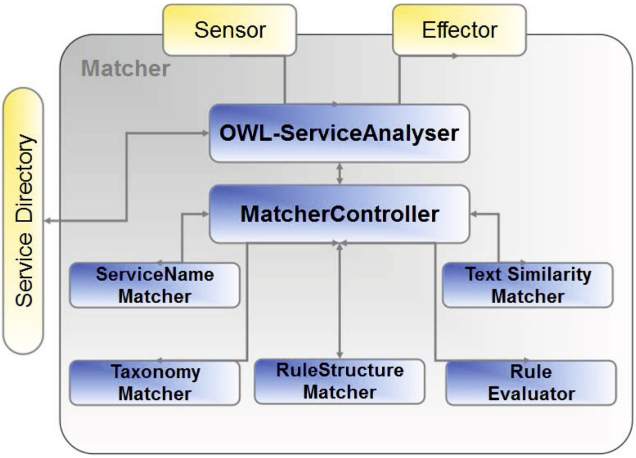

Fig. 1. The component architecture of SeMa^2

one of the n different expert opinions. Right now, each of the expert opinions are evaluated in a static way mapping a concrete result (e.g. full match, sub match, super match) to a fixed value.

In the following we propose different scoring methods to extend the PE matching and present one example in a probabilistic framework similar to [1,7]. Afterwards some aggregation methods taken from information fusion will be presented to create a probabilistic matching score. In SeMa^2 the comparison of concepts is reduced to the equivalence of the URI of ontology and concept. Here a collection of possible extensions is presented. *Semantic distance based scoring* [19] analyzes the embedded ontology of two concepts to find the shortest path from one concept to another. *WordNet based scoring* [1] can be used to find lexical similarity in used words. If two concepts out of different ontologies need to be matched a *bipartite matching score* is able to rate the similarity by i.e. the maximum cardinality match counting the edges between the different concepts. More sophisticated methods use ontology matching to find a semantic relation between concepts. Logic based scoring like proposed in *Approximated Logical Matching* [7] are further scoring methods using reasoning on formal features of the rules describing the preconditions and effects.

Probabilistic Model of Opinion. To formalize such different scoring methods we apply the results of *Morris* [13] and have modeled expert opinions as probabilities $p_i(R,S)$. The following section will detail this model. As an expert observes two concepts and elaborates their semantic distance we can abstract his opinion as $p_i(\Theta|d)$ where Θ is the subject of interest and d are the observations. An expert can then collect evidence for his opinion by conducting multiple observations d_i. Each observation might then be interpreted as evidence to strengthen his opinion. Following *Beyerer* [2] a Bayesian interpretation of the probability the conditional probability $p_i(\Theta|d)$ could be interpreted as a degree of confidence or better a degree of belief. With such an interpretation we can use this formalism to model the expert opinions as described in equation 1.

$$\underbrace{p(\Theta|d)}_{A-Posteriori} = \frac{p(d|\Theta)p(\Theta)}{p(d)} \propto \overbrace{p(d|\Theta)}^{Likelihood-Function} \underbrace{p(\Theta)}_{A-Priori} \tag{1}$$

Here the subject of interest is Θ e.g. equivalence of a Horn-clause. The observations or information used by the expert to assess its opinion is formalized in d. An example of this d could be the attached ontologies to the concept in order to calculate the semantic distance. The expert can update its opinion after observing another d using Bayesian fusion by calculating the product described in equation 1. If one concept for example is a hypernym of the other, $p(d|\Theta)$ could be proportional to the minimal distance between those two concepts [19]. Further, $p(\Theta)$ allows the expert to formalize a-priory knowledge about probability of Θ.

Opinion Aggregation. The opinions $p_i(\Theta|d)$ are collected and need to be fused to one score. Since the experts are not always equally important the possibility to prioritize the weightings of the different expert opinions is a requirement for the fusion method. With the probabilistic formalization of the expert opinions method like the *Dempster-Schafer theory of evidence* [16], fuzzy logic or artificial neuronal networks can be used for information fusion [3]. This work introduces a method of opinion aggregation called pooling method formalized in a function $K(p_1,\dots,p_n)(\Theta)$. It is acquired by adapting a weighted mean to the aggregation of opinions. We choose a weighted arithmetic mean called linear opinion pool [18]. This arithmetic mean has been generalized by *Genest* [5] to be able to use weights in the interval $[-1,1]$ in a more general class of linear opinion pools (GenLinOP). This opinion pool has the form of equation 2.

$$K(p_1,\dots,p_n)(\Theta) = \sum_{i=1}^{n} w_i p_i(\Theta) + \left[1 - \sum_{i=1}^{n} w_i\right] R(\Theta) \tag{2}$$

$w_1,\dots,w_n \in [-1,1]$ are weights and R is an arbitrary probability function, with the restriction: $\forall J \subseteq \{1,\dots,n\} : \left|\sum_{j\in J} w_j\right| \leq 1$.

The method shown in equation 2 has been chosen because of its theoretical sound standing. Other pooling methods have been and are continued to be evaluated which is subject to research. The GenLinOP has the possibility to include – besides the opinion of the group – an a-priori established probability which can be modeled as $R(\Theta)$.

Taking this theoretical framework as a basis we implement the different measures used in the service matching as experts returning a probability $p_i(\Theta)$ and aggregate them with a pooling method $K(p_1,\dots,p_n)(\Theta)$. For an example we have adapted the comparison of the arguments of a predicate. The probability here is as follows:

$$p(\Theta) = \begin{cases} \frac{1}{dist(a_r,a_s)} & \text{, if } 1 \geq dist(a_r,a_s) > 0 \\ 1.0 & \text{, if } a_r.getURI() \equiv a_s.getURI() \\ 0 & \text{, else} \end{cases} \tag{3}$$

$dist(a_r, a_s)$ defines the distance between the two concepts as proposed above. In a similar manner all other fixed values will be turned into probabilistic expert opinions. With this change, we are able to distinguish partial argument matches. We want to emphasize the importance of such a partial match for planning tasks. Here multiple services can be used to fulfill the arguments of a predicate in a precondition. Thus on a higher level: we are able to use multiple services to fulfill the preconditions of a successor task or state. For planning, we do not only need the probability after defuzzification, but also the already matched elements with their matches. The extension from a binary to a probabilistic representation is one step towards this goal.

Selecting one service to satisfy a query assumes that this given query has been foreseen and a corresponding service has been implemented. Without loss of generality we assume that this is not always the case, making it necessary to compose multiple services to fulfill a task. Thus using the service matcher as part of a planning component rises the next challenge.

3 Automated Service Composition

Similar to *Klusch* [6] the approach of this paper aims at connecting the research area of the semantic web with the flexibility and adaptiveness of agent planning. Here we see services as actions and a plan as equivalent to a service composition. A goal state in agent planning is modeled with the fulfillment of a query in the semantic web community. With this mapping of terms, we aim at building a *Hierarchical Task Network* (HTN) planner, which uses web services to achieve a defined goal state. As basis for our approach, we use the multi-agent system *JIAC V* [10] in which the agents have the capability to publish their actions as web services including semantic service descriptions [11]. The published service can then be used, like all other services contained in the service directory. The agent provides the planning component with a goal description and its knowledge base. This is necessary so that the current state can be assessed by the planner.

3.1 Challenges

In the following we will have a look at some challenges which arise by service composition. First of all, the preconditions of a service need to be split up as fine granular as possible, enabling more services to fulfill a subset of them. This means to invert the *Lloyd-Topor Transformation* [9]. Another challenge is to narrow the search space of possible actions for each state. In the planning domain heuristics are used to choose via *best-first search*, in the semantic web research area semantic descriptions are used to decide if a service is useful for a given task. These semantic descriptions might allow sound heuristics and thus narrow the search space in the same way as in traditional planning. Services can be separated into two classes: *information gathering service* and *world altering service*. We postulate that the execution of information gathering services at plan time might be helpful for the planning process. The challenge here is how the adaption of an information gathering service (executed during plan time) is reflected in the knowledge base, how they are reverted if the planner comes to a backtracking point and which adaption is communicated back to the agent instead of adding the information gathering service to the plan and re-executing it at plan execution time. To be able to generalize a created plan as a non-primitive task in the HTN, the plan including the information gathering services would be needed at the agent side. A more technical design decision is the placement of the planner. To avoid a centralized solution which would rise privacy issues, every agent could have a planning component which would get the agent closer to the BDI paradigm. But in order to avoid the overhead every node should instead provide a planner for its agents. In an multi-agent system, a heterogeneous landscape of ontologies can be used among the agents, making ontology matching a challenge worth while facing.

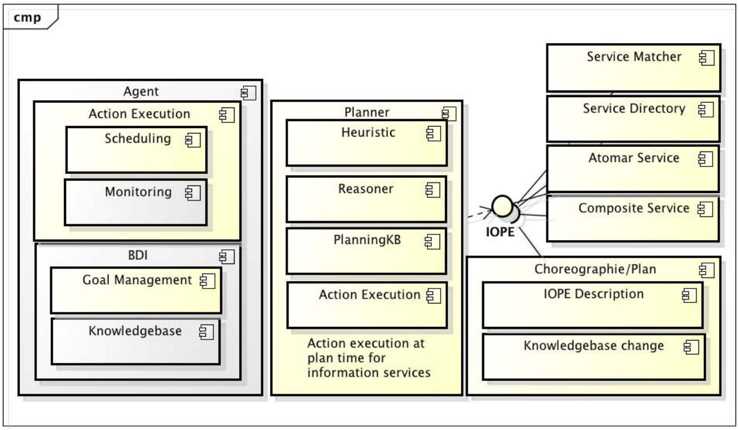

Fig. 2. Component diagram of the extended AI Planner

In more detail, we extend a Hierarchical Task Network (HTN) Planner as shown in Figure 2. The gray components are off-topic for this paper. During a planning process the retrieval of applicable services will be implemented by the SeMa^2 service matcher, searching for services in a *service directory* consisting of *atomic services* and *composite services*. The composite services are published plans or non-primitive tasks of the HTN planner. The *monitoring* component measures the quality of the plan/orchestration during execution and enables the heuristic to adapt to changes. Further plan execution failures can be detected to initiate re-planning. The *heuristic* component evaluates the matching results and creates a heuristic for the *search* component to guide its search. The search is implemented as a *best-first search*. The planner has a copy of the knowledge base of the agent which will be held in the *planning knowledge base* component, which is responsible to be able to revert changes and to keep the knowledge base consistent. The result of a planning process is a *plan* which consists of two parts. The first part are the descriptions of the used services containing their grounding information. The second part is the knowledge base needed by the agent to schedule and execute the plan.

4 Related Work

The literature provides a huge set of different service composition approaches and concepts. In this section we introduce some of these frameworks. One service composition approach is WSPlan, developed by *Peer* [15]. WSPlan uses a knowledge base and services described in WSDL extended by semantic annotations in a PDDL syntax. The knowledge base and the annotations of web services are transformed into PDDL documents. It uses an online planning method for service composition which means that the planning and execution is interleaved. Another service composition solution is OWLS-XPlan, developed by *Klusch* [6]. It transforms OWL-S descriptions of services into PDDLXML, an XML dialect for PDDL. For service composition it uses a combination of a Fast-Forward-planner and an HTN planner. There are other solutions which also transform service descriptions into another description language for service composition. The solution developed by *Okutan et al.* [14] transforms OWL-S descriptions of services into the Event Calculus framework in which actions and their effects are expressed, the solution by *Kuzu and Cicekli* [8] transforms OWL-S descriptions into PDDL and the solution by *Sirin et al.* [17] transforms OWL-S descriptions into the SHOP2 domain to use SHOP2 as an HTN planner. There are also service composition solutions which use multi-agent systems for load balancing. One approach which uses a multi-agent system is the approach by *El Falou et al.* [4]. There is one central agent which receives a request from a client which includes the initial and goal state. It forwards the request to service agents, each managing a group of web services. All service agents compute a local partial plan and send it back to the central agent. The central agent merges the partial plans together to obtain a global partial plan. Then it applies it on the initial state to obtain a new state and sends a new request based on the new state to the service agents. They in turn compute a new plan iterating until the goal state is reached. A simular approach is DPAWSC (Distributed Planning Algorithm for Web Service Composition) which also uses a multi-agent system for service composition.

The overview reveals that there exist different, mostly domain specific approaches for solving the task of service composition with AI planning. The planner proposed in this work will not transform service descriptions into a PDDL like description language. Instead, the planning is done directly on the results of the semantic service matcher and semantic service descriptions. This requires the creation of sound heuristics and backtracking from dead-ends in the planning process.

5 Conclusion

Within this paper we discussed how our automated service selection component approach *SeMa*^{2} can be extended to fulfill needed requirements for service composition. We presented *SeMa*^{2} shortly, specified the current shortcomings and proposed a formalism how to aggregate the partial results in an adaptive way. Further, we presented our initial concept of extending *SeMa*^{2} by an automated service composition component using HTN planning on SWRL. Finally, we have shown how this work can fit into a comprehensive multi-agent framework.

References

1. Bener, A.B., Ozadali, V., Ilhan, E.S.: Semantic matchmaker with precondition and effect matching using SWRL. Expert Systems with Applications 36(5), 9371–9377 (2009)
2. Beyerer, J.: Verfahren zur quantitativen statistischen Bewertung von Zusatzwissen in der Messtechnik. VDI Fortschritt-Bericht, vol. 8. VDI/Verl. (1999)
3. Fähndrich, J.: Analyse von Verfahren zur Kombination von Expertenwissen in Form von Wahrscheinlichkeitsverteilungen im Hinblick auf die verteilte lokale Bayes'sche Fusion. Diploma thesis. Karlsruhe Institut of Technology (May 2010)
4. Falou, M.E., Bouzid, M., Mouaddib, A.-I., Vidal, T.: Automated Web Service Composition: A Decentralised Multi-agent Approach. In: IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technologies, vol. 1, pp. 387–394 (2009)
5. Genest, C.: Pooling operators with the marginalization property. The Canadian Journal of Statistics/La Revue Canadienne de Statistique 12(2), 153–163 (1984)
6. Klusch, M., Gerber, A., Schmidt, M.: Semantic Web Service Composition Planning with OWLS-Xplan. In: Proceedings of the 1st Int. AAAI Fall Symposium on Agents and the Semantic Web, pp. 55–62 (2005)
7. Klusch, M., Kapahnke, P.: The iSeM matchmaker: A flexible approach for adaptive hybrid semantic service selection. Web Semantics: Science, Services and Agents on the World Wide Web 15, 1–14 (2012)
8. Kuzu, M., Cicekli, N.K.: Dynamic planning approach to automated web service composition. Applied Intelligence 36(1), 1–28 (2012)
9. Leuschel, M., Craig, S.-J.: A Reconstruction of the Lloyd-Topor Transformation using Partial Evaluation. In: Pre-Proceedings of LOPSTR, 2005 (2005)
10. Lützenberger, M., Küster, T., Konnerth, T., Thiele, A., Masuch, N., Heßler, A., Burkhardt, M., Tonn, J., Kaiser, S., Keiser, J., Albayrak, S.: JIAC V — A MAS Framework for Industrial Applications (Extended Abstract). In: Proceedings of the AAMAS 2013, Saint Paul, MN, United States of America (2013)
11. Masuch, N., Brock, P.: Integration of semantic service description techniques into a multi-agent framework. In: Trends in Practical Applications of Agents and Multiagent Systems, pp. 155–162. Springer International Publishing (2013)
12. Masuch, N., Hirsch, B., Burkhardt, M., Heßler, A., Albayrak, S.: SeMa^2: A Hybrid Semantic Service Matching Approach. In: Semantic Web Services, pp. 35–47. Springer, Heidelberg (2012)
13. Morris, P.A.: Combining expert judgments: A Bayesian approach. Management Science 23(7), 679–693 (1977)
14. Okutan, C., Cicekli, N.K.: A monolithic approach to automated composition of semantic web services with the Event Calculus. Knowledge-Based Systems 23(5), 440–454 (2010)
15. Peer, J.: A PDDL Based Tool for Automatic Web Service Composition. In: Ohlbach, H.J., Schaffert, S. (eds.) PPSWR 2004. LNCS, vol. 3208, pp. 149–163. Springer, Heidelberg (2004)
16. Shafer, G.: A mathematical theory of evidence, vol. 1. Princeton University Press (1976)
17. Sirin, E., Parsia, B., Wu, D., Hendler, J., Nau, D.: HTN planning for Web Service composition using SHOP2. Web Semantics: Science, Services and Agents on the World Wide Web 1(4), 377 (2004)
18. Stone, M.: The opinion pool. The Annals of Mathematical Statistics 32(4), 1339–1342 (1961)
19. Wu, J., Wu, Z.: Similarity-based web service matchmaking. Services Computing (2005)