Towards Agent-based Coalition Formation for Service Composition

Ingo Müller, Ryszard Kowalczyk
Swinburne University of Technology Melbourne
Faculty of Information and Communication Technologies
Hawthorn, 3122 Victoria, Australia
{imueller, rkowalczyk}@ict.swin.edu.au

Peter Braun
the agent factory GmbH
Intershop Tower
07743 Jena, Germany
braun@the-agent-factory.de

Abstract

The topic of agent-based service composition has been experiencing much attention recently. Researchers are applying agent technology with the aim to improve adaptiveness and flexibility of prevailing static Web service composition solutions. One major characteristic of multi-agent systems in particular is their ability of emergent behavior that allows gaining complex system behavior from small distributed sets of simple rules. This paper describes a multi-agent-based coalition formation approach for service composition that achieves emergent behavior based on a light-weight interaction protocol and decentralized decision making. The paper also presents evaluation results of first experiments to underline the validity of the approach.

1 Introduction

Web service composition describes the process of combining several atomic Web services, each providing a specific piece of functionality, into Web services workflows. The current de-facto standard BPEL4WS provides means to define control flow and data dependencies between atomic services that lead to rigid hard-coded workflow implementations. Current solutions help industry to automate business processes at predictable costs. However, dynamic, more adaptive and more flexible alternatives are feasible that can be realized on top of Semantic Web technologies such as OWL-S and the notion of Quality of Service (QoS).

In the context of Web service composition, added flexibility stands for the ability to dynamically choose atomic services at runtime rather than relying on hard-coded workflows. The decision on particular services can be based on functional requirements and QoS parameters. The other aforementioned aspect adaptability refers to the ability to handle error situations in order to keep a composite Web service alive even though atomic services might be unavailable or fail completely, or QoS parameters are violated.

Intelligent agent and multi-agent technologies provide a variety of approaches that can be applied for realizing improvements of existing Web service composition solutions such as several types of negotiations and auctions [13] and broker-based systems [17]. All approaches generate technically sound solutions, however they are based on centralized data and control flow, complex knowledge of situation and environment, and mostly centralized decision making.

We believe these constraints cannot hold in all possible situations. Therefore, we aim at multi-agent coalition formation because we believe it better matches more uncertain and decentralized environments. Multi-agent systems enable emergent behavior that allows gaining complex system behavior from a population of autonomous agents that embrace simple sets of decision making rules and thus clearly distinguish from algorithmic, centralized, and deterministic solutions.

Our coalition formation approach centers on the concept of cooperative goal-oriented agents for solving complex problems. In case of Web service composition, a complex problem is a request for a composite service. The solution is the formation of coalitions of agents with complementing capabilities (provision of particular atomic services). Each coalition represents an instance of a composite service. Coalitions emerge from interactions between agents based on limited knowledge and local autonomous decision making. The goal of agents is to participate in those coalitions in order to satisfy their preferences or interests.

This paper reports on concept, implementation, and evaluation of our approach. The remainder of the paper is structured as following: Section 2 discusses related work. Section 3 describes the design of our multi-agent system. Section 4 introduces the coalition formation process. Section 5 informs about current decision making and Section 6 presents evaluation of first results. Finally, Section 7 concludes the paper with an outlook to future work.
2 Related Work

The topic of agent-based service composition has been experiencing much attention recently. The enormous effort results into a vast amount of approaches and implementations, e.g. [2, 3, 7, 9, 10, 11, 16, 17]. In general, these approaches make use of software agents to manage the composition and enactment of multiple services. Most papers describe broker or mediator architectures that implement middle agents for centralized decision making based on complete knowledge about available services. Software agents that represent services are usually simple passive software entities acting as proxies. None of these approaches supports emergent behavior, due to static hierarchical interaction protocols. Our approach differs from existing approaches in two aspects. First, it is based on decentralized decision making based on limited knowledge. A light-weight interaction protocol reduces the number of agent types and roles, which potentially increases system robustness. Secondly, agents representing services are proactive. Our approach enables emergent behavior through asynchronous agent interactions. Agents actively engage in the coalition formation process according to private preferences and, therefore, cause the emergence of coalitions.

Secondly, we consider existing work on agent-based coalition formation. In general, coalition formation related literature distinguishes between two major approaches. A game theory based approach employs sets of self-interested agents that compete for resources, whereas a complementary approach bases on agents that can achieve their goals only through collaboration. The degree of collaboration depends on the level of social commitment of agents. Self-interested agents [15] aim at maximizing their individual utilities, whereas group-oriented agents maximize group utilities [14]. These approaches are based on restrictive assumptions in order to achieve stable or optimal agent coalitions, such as a priori knowledge about the value of a coalition, the set of possible coalitions, and/or agents and their capabilities that are known to a central coalition leader [1]. In addition, the quest for stable or optimal solutions is paid off with infeasible computational complexity, e.g. exponential complexity [4] or polynomial complexity [15]. We do not limit our approach to the mentioned assumptions; especially do we consider limited knowledge. Our approach does not guarantee optimal utility for coalition initiators nor fair benefit sharing amongst coalition members. It is also possible, that no coalitions emerge in a given period of time. However, due to characteristics of emergent agent behavior, we expect reasonable outcomes of our approach for all involved parties.

Finally we would like to refer to two approaches that are in a sense similar to our approach, because they refer to multi-agent systems of autonomous decision makers [6, 18]. Both works describe scenarios in which an agent has its own local utility and is free in choice of joining or leaving coalitions. Both approaches aim at maximizing local utility and reduction of coordination costs for long term coalitions (e.g. in the context of e-commerce [18]). Our approach focuses in contrast on repetitive coalition formations. In addition, coalitions are not a mean to reduce coordination costs or to improve the utility of groups of agents. Our coalitions satisfy requests for composite services and allow agents to find partners according to local preferences. But this process starts all over again and partners may vary from iteration to iteration depending on the requested composite service. However, agents can keep history for establishing and optimizing long term relationships with other agents.

3 Multi-Agent System Design

Our multi-agent system design is based on a blackboard architecture [8] as depicted in Fig. 1. A blackboard is an information space providing agents the opportunity to exchange and to share information. Besides a blackboard the multi-agent system also comprises two different agent types. User agents act on behalf of users and service agents act on behalf of service providers.

![Figure 1. Multi-Agent System Design.](image)

A user agent provides the opportunity to publish requests for composite services. A request for a composite service contains the description of a generic service workflow that does not refer to particular service instances, but more general to semantic service type definitions. At this stage, we leave open the question for a concrete service type specification standard. In addition, the request also defines a deadline when the user expects coalitions to be available. In the following, this deadline is denoted as coalition registration deadline. A user agent places a new composite service request on the blackboard and waits afterwards until
the coalition registration deadline is due, before it becomes active again.

A service agent provides exactly one atomic service on behalf of a service provider. It obtains composite service requests from the blackboard. A service agent performs matchmaking between service type definitions of a request and its own associated service in order to determine whether it can contribute to a coalition. Thus, one major goal of our approach is to break down complexity. We replace one central complex matchmaker with complex knowledge, with multiple distributed simple matchmakers with limited knowledge. A service agent also uses the blackboard for exchanging service advertisements with other service agents regarding a single composite service request, before actively engaging in a coalition formation process. A service advertisement contains the service type definition of the atomic service associated with a service agent. Service agents determine potential coalition partners based on simple matchings of service type definitions of a request and available advertisements.

During the coalition formation process, service agents team up in coalitions (a service agent can only join one coalition at a time); each of them able to provide the requested composite service. Hence, service agents do not need to understand the workflow layout of a composite service. A coalition of service agents merely contains a complete set of atomic services rather than a concrete workflow representation. This set of service agents is both, necessary and sufficient to instantiate a composite service. Therefore, simple matching of service type definitions is a sufficient base for the coalition formation process. A second major goal of our approach is to decentralize decision making among multiple actors with the effect that service providers now have direct influence on the composite services they are serving in and the business partners they are collaborating with.

The multi-agent system design results in a number of constraints:

- Service agents have very limited knowledge about their environment. A service agent knows the service type definition of its associated service. Furthermore, it can gain knowledge about composite service requests and advertisements.
- Besides advertisements, service agents do not reveal any private data.
- Information about advertisements is aging and may become obsolete.
- Service agents autonomously perform decentralized decision-making for picking partners.
- The multi-agent system design defines a framework for various agent-based coalition formation approaches.

Each agent possesses reactive and proactive behavioral elements in order to successfully perform tasks.

- Service agents do not act maliciously and they have the inherent incentive to form and join coalitions.
- We do not consider any security and trust issues.

Our current implementation only supports single composite service requests, which could be easily extended. As described in [12], the current design could also be used for extended functionality such as QoS negotiations and contracting issues, giving user agents a more autonomous and proactive character. In addition, the design could be changed to a truly decentralized approach by distributing the blackboard or by replacing the blackboard with a peer-to-peer like communication structure.

4 Coalition Formation Process

The concept underlying the coalition formation process are goal-oriented service agents that interact with each other based on autonomous decisions. They cooperate and also compete in order to form coalitions that can provide requested composite services. Therefore, a well-defined interaction protocol needs to be defined in order to structure and guide agent interactions. Various alternatives exist for designing valid interaction protocols. We detail about our choice and the coalition formation process according to it in the remainder of this section without motivating design decisions.

The interaction protocol is organized in three different conversations depending on three different roles that service agents can fulfill:

- A service agent is in role Candidate, when it has no coalition partners.
- A service agent acts in role Leader, when it has at least one partner in its coalition that accepted this service agent as leader.
- A service agent is in role Member, when it has joined the coalition of another service agent. By joining, the service agent accepts the current coalition leader.

A service agent can change its role at any point in time during a coalition formation process. However, it always can only perform one role at a time. A role change depends on messages received from other service agents and own decisions.

The coalition formation process commences when a user agent publishes a composite service request on the blackboard. All idle service agents that successfully match their service type description with the request, adopt the Candidate role and leave their service advertisements with the
blackboard. The next thing to do for every Candidate service agent is either to proactively determine potential partners or to reactively respond to requests. Both activities refer to the Coalition Request conversation as depicted in Fig 2.

Let us have a look at a simple example. Assume service agent A decided to contact service agent B in order to form a coalition. Thus, A sends a Coalition Request message to B. Service agent B then decides whether to accept the request and replies accordingly, either with a Coalition Accept or Coalition Reject message. The consequence of replying with Coalition Accept is that service agent A changes to role Leader, meanwhile service agent B transfers to role Member. Both agents remain in their original roles in case of Coalition Reject.

In our example, service agent B would send a Member Leave Request message to its coalition leader - service agent A. The leader can then decide whether to accept or reject the request based on the current status of its coalition. In case the leader is already engaged in another interaction that could lead to the completion of the coalition, it rejects the leave request. In all other cases it sends a Member Leave Accept message back to the requesting Member agent, followed by sending a Member Leave Inform messages to all other coalition members to inform about the loss.

Due to multi-agent system design, characteristics of autonomous decision making, and most importantly asynchronous message passing, it is non-deterministic when and with whom a service agent starts interacting. In particular, it is non-decidable for all other agents, if, when, and how a service agent processes incoming messages. Thus no strict order of actions can be determined and coalitions solely emerge through interactions. Another consequence is that the message handling is more complicated than illustrated in this section. Because most messages of the interaction
5.1 Pick Next Candidate Decision

The task is to choose the next potential coalition partner based on all advertisements stored locally with a service agent. Let \(A\) be the set of all advertisements.

\[
A = \{a_1, a_2, ..., a_n\}
\]

Each advertisement is represented by a pair of attributes.

\[
a_i = (c_i, s_i) : c_i \in N \land s_i \in \{0, 1, 2\}
\]

Attribute \(c_i\) represents a counter of the number of previously exchanged messages with the associated agent during the current coalition formation process. Attribute \(s_i\) denotes the interaction status with the associated agent, which is an element in the set \{unknown = 0, rejected = 1, not understood = 2\}. To obtain the next potential partner, a service agent determines the subset \(S\) of all advertisements with minimum counter and, in addition, minimum status.

\[
S = \{a_i : min_s(min_c(c_i, s_i))\}
\]

Set \(S\) is never empty as long as advertisements exist. If the size of \(S\) is greater than 1, one element is randomly selected. The attributes of an advertisement change with every further interaction with the associated agent. If set \(S\) is empty before a coalition is completed, an agent cannot perform pro-active decisions until new advertisements have been received.

5.2 Coalition Leave Decision

The task is to decide whether to leave the current coalition. If \(m\) denotes the number of coalition partners of the agent’s current coalition and \(n\) denotes the number of prospective partners in the other coalition, a service agent decides to leave its current coalition when the following equation is true: \(m < n\).

6 Evaluation

This section reports on the evaluation of an experiment aiming at demonstrating functioning and output of the coalition formation process. The experiment was conducted with the Tracy agent toolkit [5]. Four test scenarios were created, as depicted in Tab. 1, to verify our approach with different types of composite service requests. Property service request configuration defines the number of different service types of a composite service request. Property service agent configuration denotes the number of service agents per service type representing a particular atomic service. All four
scenarios vary only in two parameters: service request configuration and the total number of service agents. As a consequence, any coalition formation process based on the scenarios terminates after 3000 milliseconds with the number of registered coalitions varying between 0 and the possible maximum of 3 registered coalitions. 1000 coalition formation test runs were performed per scenario. The data obtained with the experiment forms the base of the evaluation of the remainder of this section.

The first issue we would like to discuss concerns the completeness of our coalition formation approach. Or in other words, how often do coalition formation processes terminate with the maximal number of coalitions? Fig. 5 shows that 100 per cent of scenario A test runs returned with the maximum number of coalitions. However, with increasing number of service types and service agents (as in scenarios B, C, and D), the proportion of test runs with less than maximal number of registered coalitions also increases.

Test runs based on scenario D for example return only in 50 per cent with 3 coalitions, in 37.5 per cent with 2 coalitions, in 11.6 per cent with 1 coalition, and almost 1 per cent of the test runs terminate with no registered coalitions. The relationship between increased number of not maximal outcomes and increased number of service types and service agents can be explained with more interactions necessary in order to complete coalitions, as depicted in Fig. 6.

In order to confirm this suspicion, we need to have a look at how coalitions are formed. Fig. 7 illustrates service agent behavior during the coalition formation process. The figure depicts the average frequency of service agents leaving or abandoning coalitions. Service agents do not possess any other mean to reduce or dissolve a coalition besides these two actions. Fig. 7 shows that with increasing number of service types and service agents the frequency of coalition leave and abandon actions increases as well. No service agents leave or abandon coalitions in case of only two service types of scenario A, because a coalition is already completed after only one successful coalition request conversation (cmp. Sec. 4). In test runs of scenario D with five service types, on average 2.48 service agents leave a coalition and 1.85 service agents abandon a coalition.

The reason for an increased number of situations with a need for coalition leave or abandon actions is illustrated with Fig. 8. The higher the numbers of service types
and service agents, the higher the probability that incomplete coalitions exist that occupy sets of service agents and thus prevent other coalitions from completion. Accordingly more interactions among service agents are necessary to complete coalitions. More communication ‘slows down’ the coalition formation process, because firstly, more message timeouts occur (that delay agent execution) and secondly, more active service agents generate a higher load. A way to improve performance could be the incorporation of adaptable message timeouts that can be adjusted according to previous response times and network delays. Currently, message timeouts are only variable (a random value between 500 and 1000 msec.). An increased registration deadline would be another solution to improve the total number of registered coalitions.

Fig. 7 also underpins that the coalition formation process is stable because a significant but not dramatic increase of actions can be observed. This claim is supported by Fig. 10, which depicts the average frequency of coalition leave and abandon actions of the different result classes of scenario D. Test runs terminating with 0 coalitions experience on average 1 coalition leave action, whereas test runs resulting in 3 coalitions show on average 2.94 coalition leave and 1.94 coalition abandon actions. Fig. 10 demonstrates the coalition formation process does not produce volatile agent behavior that possibly could decrease the number of registered coalitions. In fact, it shows that a higher number of coalition leave and abandon actions supports the completion of one or multiple coalitions, as the similarity of the numbers for test runs terminating with 1, 2 or 3 registered coalitions in Fig. 10 underlines.

Finally, Fig. 9 illustrates the fairness of the coalition formation process. The figure is based on data of test scenario D and shows for each service agent (ids 1-15) the total number of coalitions the agent lead in the entire 1000 test runs. The id of an agent denotes its position in the starting order. In general the approach is fair, since each agent is able to lead coalitions. Currently we cannot explain the peaks for example for service agents 4, 14, and 15. We suspect dependencies between starting order of agents and random message timeout generation on the one hand and the operating system thread scheduling on the other hand to be responsible for the effect. However we need to verify this suspicion with more experiments.

7 Conclusion and Future Work

The motivation for the work presented in this paper is the static inflexible character of prevailing service composition solutions. Researchers trying to increase flexibility and adaptability based on Semantic Web technologies. However, current approaches are limited by constraints such as centralized data and control flow, complex knowledge, and
mostly centralized decision making. Constraints, we believe cannot hold in open uncertain environments. This paper reports on concept, implementation, and evaluation of a novel approach for agent-based service composition, which differs from existing solutions in decentralized decision making, limited knowledge, and emergent behavior of interacting goal-oriented agents. Our current findings prove the concept works; however we need to learn more about several issues such as message timeouts, registration deadlines, or the influence of network delays and thread scheduling in order to improve fairness and performance. In addition, we plan to extend the current approach, e.g. with an adaptive message timeout mechanism, more complex decision making, and learning techniques with the aim to investigate the performance of the coalition formation process in general and to investigate system behavior with agents with different reasoning capabilities or preferences in particular.

References


