Trading Services in Ontology-driven Markets

Steffen Lamparter
Institute AIFB
University of Karlsruhe (TH), Germany
lamparter@aifb.uni-karlsruhe.de

Björn Schnizler
Information Management and Systems
University of Karlsruhe (TH), Germany
schnizler@iw.uni-karlsruhe.de

Abstract

This paper presents an architecture of an ontology-driven market for trading Semantic Web Services. An auction schema is enriched by a set of components enabling semantics based matching as well as price-based allocations. The use of background knowledge in auction mechanisms also reduces the overall complexity of the system which is shown by means of a simulation.

1 Introduction

In recent years, Web Services have become the key technology for building flexible and interoperable computing infrastructure. However, to realize the vision of a full-fletched service oriented architecture, efficient service discovery and allocation is required to coordinate the interplay between service providers and requesters. The Semantic Web community suggested the use of Semantic Web Services (e.g. using OWL-S, WSMO, WSDL-S) which enrich services by a formal and explicit description of their capabilities. Instead of syntactically based matchmaking algorithms, services can be semantically matched using concepts and relations formalized by means of ontologies.

However, the direct application of matchmaking mechanisms for allocating Web Services has several drawbacks: Firstly, these algorithms do not guarantee that those requesters will receive the supplied services who value them most. Secondly, they ignore the fact that users will only offer their services if they are adequately compensated. Compensation requires determining how the offered services are allocated among potential requesters and how the prices for the services are set. Thirdly, semantics based matchmaking algorithms are typically computational demanding and thus not adequate for large-scaled negotiations. However, these aspects are crucial for implementing economic efficient Web Service infrastructures.

This paper attempts to tailor an ontology-driven market mechanism for allocating Semantic Web Services. This is realized by merging the advantages of classical auction algorithms and semantically based matchmakers into one framework. Furthermore, it is shown that the complexity of a classical auction mechanism is reduced by means of formalized background knowledge.

2 Design Requirements

The design of a market platform for trading services mainly affects two components: (i) *A communication language* which defines how bids (i.e. offers and requests) and agreements can be formalized and submitted to the market mechanism. (ii) *An outcome determination* by means of an allocation (i.e. who gets which service) and a pricing component. Both components as well as the interplay between them have to be designed carefully to reflect the users' requirements [Neumann, 2004]. The following requirements for a Web Service market platform can be identified [Schnizler *et al.*, 2004]:

The platform should allow multiple buyers and sellers to *trade simultaneously* (R1) and ensure *immediate execution* in case a suitable counterpart is found (R2). The mechanism should support the *trading of heterogenous services* like, for instance, computational services and storage services (R3). A *meaningful matchmaking of orders* should be realized by the market infrastructure to allow a matching of services based on the semantics of an order instead of their syntactical representation (R4). For instance, a request for a storage service should be matched with an offer for a hard disk service.

Additionally, the market should support the *trading of dependent services* (R5), as service are complementarities. Participants have super-additive valuations for the resources, as the sum of the valuations for the single resources is less than the valuation for the whole bundle $(v(A) + v(B) \leq v(AB))$. Furthermore, participants may want to submit more than one bid on a bundle but many that are excluding each other. In this case, the resources of the bundles are substitutes due to sub-additive valuations for the resources $(v(A) + v(B) \geq v(AB))$. Furthermore, services can differ in their quality characteristics and their policies, e.g. a storage service by its capacity or access time; a billing service by its age restriction. As such, the mechanism should support *multiattribute services* (R6).

A *computational and communication efficient determination of the outcome* (R7) is required by the mechanism in order to converge on a desirable global outcome by minimizing the computational effort.

3 Marketplace Architecture

In this section the overall architecture of an ontology-driven marketplace is introduced. The architecture, as outlined in

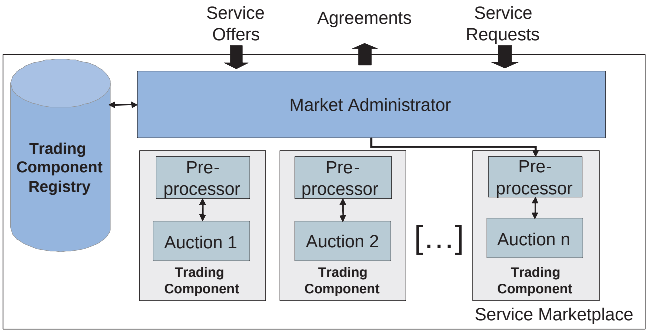

Figure 1: Overall architecture of the service marketplace

figure 1, consists of the following components: a Marketplace Administrator, a Trading Component Registry, and several instances of Trading Components which encapsulate Preprocessors and Auction instances.

Although the market is seen as a monolithic unit, internally it may consist of several independent trading components each with its own auction mechanism and preprocessors. In each of these individual trading components, only a subset of all available services is traded. Although a central auction mechanism comprising all available services would fulfill the economic properties, a distinction between several independent trading components reduces the complexity of the auction (R7) since the allocation time increases exponentially with respect to the number of bids in the order book.

Trading Component Registry The Component Registry is a repository that contains an ontology-based description of each trading component. These descriptions basically specify the capability of the services that are traded in a specific trading component.

Marketplace Administrator The Marketplace Administrator manages the internal mechanisms, i.e. the administrator creates, removes, splits or merges trading components. Furthermore, the administrator receives orders from requesters and suppliers. In order to allow meaningful matchmaking (R5) these orders are formally expressed via ontologies. The Marketplace Administrator compares the capabilities specified in the order with those stored in the registry on a semantic level (R4) and forwards the order to the corresponding internal trading component.

Auction In order to compute the optimal outcome, a multi-attribute combinatorial double auction is applied [Schnizler *et al.*, 2004]. This mechanism meets the requirements mentioned in the last section by allowing simultaneous trading (R1), immediate executions (R2), and trading heterogeneous (R3), dependent (R4), as well as multiattributive (R6) services. However, existing implementations of such mechanisms rely only on pure syntactic matching of orders and, thus, requirement (R4) cannot be met. Therefore, an additional preprocessing of the order book is necessary.

Preprocessor The Preprocessor administrates an order book containing ontology based orders and is responsible for preparing this order book in a way that it can be handled by a traditional auction system. Thereby, two aspects can be distinguished: (i) *Semantic preprocessing:* By analyzing information implicitly contained in the ontology new XOR-orders are introduced in a way that that the subsequent syntactic auction mechanism results in the same set of matches as a semantic approach, while inappropriate allocations are avoided (R4). For example, a storage service is requested and a hard disc service offered, syntactic matching will fail due to the lack of taxonomic information. Therefore, semantic preprocessing adds an additional bid, in this case a hard disc service request, which is XOR connected with the request for a storage service. Hence, the hard disc request can be matched with the corresponding offer while preserving the consistency of the order book. (ii) *Syntactic preprocessing:* Once this is done, the ontology based communication primitives are translated into the syntactic bidding language that is understood by the auction mechanism. Thereby, the ontology instances are serialized to strings.

4 Performance Simulation

As mentioned above, the proposed architecture is capable of splitting markets, if two or more disjoint sets of orders exist. This may lead to high performance gains especially if the auction mechanism is computational demanding. In the following, this performance gain is measured by means of a stochastic based simulation. Therefore, bundles are generated using the Decay distribution, which is shown to lead to hard instances of general combinatorial allocation problems [Sandholm *et al.*, 2002].

In the first treatment, the order stream is computed using a single instance of the mechanism. In the second treatment, the order stream is split into the disjoint subsets and computed using two independent instances of the mechanism. It is shown that an intelligent order book splitting leads to performance gains. For instance, the winner determination for 900 bundle bids takes more than 140 seconds in scenario 1. Splitting the subsets and computing the order books separately (each with 450 bundle bids) takes only 40 seconds.

5 Conclusion

This paper outlined the design of an ontology-driven market for trading Web Services. Based upon a requirement analysis for a Web Service market, a marketplace was designed which is up to these marks. The marketplace transforms ontology-based requests and offers into syntactically represented orders so that an existing auction mechanism can be used while still allocating on a semantic level. Furthermore, by means of ontological information the overall market is split into several independent sub-markets in order to reduce the complexity of the allocation.

References

[Neumann, 2004] Dirk Neumann. *Market Engineering - A Structured Design Process for Electronic Markets*. PhD thesis, University of Karlsruhe (TH), 2004.

[Sandholm *et al.*, 2002] Tuomas Sandholm, Subhash Suri, Andrew Gilpin, and David Levine. Winner determination in combinatorial auction generalizations. In *Proc. of 1st Int. Joint Conf. on Autonomous Agents and Multiagent Systems*. ACM Press, 2002.

[Schnizler *et al.*, 2004] Björn Schnizler, Dirk Neumann, and Christof Weinhardt. Resource allocation in computational grids - a market engineering approach. In *Proc. of the WeB 2004, Washington*, 2004.