

Formation of Service Value Networks for Decentralized Service Provisioning

Sebastian Speiser, Benjamin Blau, Steffen Lamparter, and Stefan Tai

Karlsruhe Service Research Institute (KSRI), Universität Karlsruhe (TH), Germany
firstname.lastname@ksri.uni-karlsruhe.de

Abstract. The provisioning of complex services requires tight collaboration between diverse service providers and their customers harmonizing supply and demand chains to a highly flexible, dynamic and decentralized service value network. Peers in such a network autonomously delegate (sub-)tasks which cannot be done efficiently by themselves to other more suitable peers in their community. In this paper, we propose an architecture for such service communities that features decentralized service provisioning based on current Web technologies. In this context, we present an algorithm for efficient service value network formation and show by means of a simulation that sufficiently sized service networks can fulfill practically all customer requests. When compared to the optimal (central) case, there is a modest price increase for the customers but the overall welfare decreases only insignificantly.

1 Introduction

Complex (or *composite*) services "typically involve the assembly and invocation of many pre-existing services possibly found in diverse enterprises [1]," and thus, a network of service providers and consumers. In services-led economies, these networks increasingly are loosely-coupled configurations of legally independent firms.

The formation of such a network is driven by the value that it generates for its customers. With increasing competition and specialization in the services sector, and the continuous introduction of new services offerings, value-driven network formation and transformation is of predominant importance. Dynamic *service value networks* are often considered as the only strategic alternative to provide complex services [2,3,4].

Service networks describe the possible cooperations between legally independent actors that enable co-generation of value by fulfilling complex customer requests. Due to low lock-in and lock-out costs, service value networks are characterized by a high rate of fluctuation of service providers, making it difficult to maintain a consistent, centralized service repository.

Consequently, this paper analyzes the formation of service value networks in a decentralized service provisioning environment. Based on customers' requests for the completion of complex tasks, peers in such a network autonomously delegate (sub-)tasks to other service providers within their partner networks. This iterative process fosters the evolution of a network-based value generation driven by customers' needs and co-opetition of specialized service providers.

In this paper, we propose an architecture for such service networks that features decentralized service provisioning based on current Web technologies. In this context, we present an algorithm for efficient service value network formation and show by means of a simulation that sufficiently sized service networks can fulfill practically all customer requests. When compared to the optimal (central) case, there is a modest price increase for the customers but the overall welfare decreases only insignificantly.

Several decentralized service discovery mechanisms have been presented in literature. Most of them distribute service descriptions over several peers and use an indexing mechanism to efficiently route the queries through the P2P network (e.g. [5,6]). However, building and maintaining such indexes contradicts our assumption that each peer in the network has only knowledge about his direct neighbors and usually peers are not willing to share this knowledge since it can be an important business asset.

There are a few approaches that explicitly consider a network of providers, e.g. [7]. However, the algorithms assume peers with different capacity (i.e. available resources) but the same homogeneous functionality (such as computing power). Since in our network the services are highly specialized and therefore provide different functionality, these algorithms are not directly applicable.

The paper is structured as follows. In Section 2 we define our model of composed services and service value networks. Based thereon in Section 3 we introduce our proposed algorithms for decentralized service provisioning. Section 4 describes a simulation of the algorithms and compares them to centralized approaches. We conclude and give an outlook on future work in Section 5. An extended version of this paper with more details can be found in [8].

2 Service Value Networks

In this section we introduce a formal model that is the foundation for our understanding of service value networks (SVNs). First we define a *service specification* which determines how *atomic services* are combined to form more complex *composed services*. Afterwards service providers and customers are introduced that are connected in a service network. For the fulfillment of service requests the SVNs are dynamically formed and create a value that is defined in the end of this section.

Service Specification. Services that cannot be decomposed in smaller sub-services are called *atomic*. In contrast a *composed* service is equivalent to its *component* services that can in turn again be atomic or composed. Let S be the set of all services. The set of ordered pairs E_S represents the component relationship meaning that $s_1 \in S$ is a component service of $s_2 \in S$, iff $(s_1, s_2) \in E_S$. The directed graph $G_S = (S, E_S)$ is called *service specification*. We require G_S to be acyclic. Furthermore we define the two disjoint sets of *atomic* services $S_a = \{s \in S \mid \nexists s' \in S : (s', s) \in E_S\}$ and *composed* services $S_c = S \setminus S_a$. We define the complexity of a service s as the number of basic services that are required to create an equivalent service.

Service Provider. Each service provider p is able to execute the services denoted by his capability set $\phi_p \subseteq S$. For the execution of a service $s \in \phi_p$ the provider is charged with internal costs, determined by his cost function $c_p : S \rightarrow \mathbb{R}$. As a provider aims

at making a profit he charges a price that adds a margin to his costs, given by the margin function $m_p : \mathbb{R} \rightarrow \mathbb{R}$. The charged price for $s \in S$ is $f_p(s) = m_p(c_p(s))$. The provider is able to subcontract services to a set of cooperating service providers, denoted as V_p . The costs for p when subcontracting s are given by $c_p(s) = \min_{p' \in V_p} f_{p'}(s)$.

A provider p maintains a function $d_p : S \rightarrow \{false, true\}$ for all services that he can either offer himself or in cooperation. The function returns true for services that should be decomposed and false for services that are executed or delegated as a whole.

Customer. Besides service providers we consider the set of service customers W that consume services but do not provide any services nor forward requests. A customer $w \in W$ has a valuation for the services he requests, given by his utility function $u_w : S \rightarrow \mathbb{R}$. Service requests are sent to w 's partner network $V_w \subseteq P$.

Service Value Network. The relationships between customers, providers and among providers are represented by the directed graph $G = (P \cup W, E)$ where an edge $(x, y) \in E$ denotes that x sends requests to y . The edges are given by the partner networks as $E = \{(x, y) \in (P \cup W) \times P \mid y \in V_x\}$. The graph G is called *service network*. The customer that requests a service and the providers that are involved in the fulfillment of the request form a *service value network*. As this process is invoked for every service request it is a *dynamic formation*.

Let $P' \subset P \cup W$ be the set of participants in a service value network fulfilling the request for service s by customer w . Service delegations are represented as tuples $(p_1, p_2, s) \in P' \times P' \times S$, meaning that p_1 delegates service s to p_2 . Internal executions are also treated as service delegations with $p_1 = p_2$. Let E' be the set of all service delegations then the directed graph $G' = (P', E')$ denotes the service value network. For the reader's convenience we define the following three sets for a service value network: (I) The customer's request $R_{G'} = \{(w, p, s) \in E' \mid w \in W\}$, (II) the internal executions $I_{G'} = \{(p_1, p_2, s) \in E' \mid p_1 = p_2\}$, and (III) the set of service cooperations $D_{G'} = E' \setminus (R_{G'} \cup I_{G'})$.

Welfare in Service Value Networks. The welfare co-generated in a service value network is given by the sum of received values for all participants. For a service value network $G' = (P', E')$ which serves customer w with service s , we define the welfare $wf_{G'} = u_w(s) - \sum_{(p, p, s') \in I_{G'}} c_p(s')$. This shows that the welfare can be maximized if the internal costs that occur during the execution of a service are minimized. For each service delegation $(p_1, p_2, s') \in D_{G'}$ the payment of $f_{p_2}(s')$ increases p_2 's value by the same amount as it decreases p_1 's value. Therefore payments do not influence the welfare, except in the case where the total price is above the customer's valuation meaning that no transaction will take place and thus no value at all is generated.

3 Network Formation and Service Delivery Algorithms

The algorithms can be divided into two groups. The first initializes and maintains the data structures a provider keeps to determine the best executions strategies for each service. Based on this data the second group generates concrete offers upon service requests.

Every new provider p that joins the service value network assigns all services $s \in S$ to himself with costs of ∞ . Then he updates the services in his capability set to have a price given by his internal costs plus his margin ($\forall s \in \phi_p : f_p(s) := m_p(c_p(s))$). Afterwards the providers notifies his partner network about his capabilities.

A provider p that receives a notification about the capability of p' to deliver s at price $f_{p'}$, first checks if the new price is better than his current costs. If this is the case he updates the preferred provider for the service and sets the costs to the received price. He also updates the price $f_p(s)$ he charges for the service to be the new costs plus his margin. Then he updates his cost structure.

This function is done by first notifying the partner network about the new capability respectively the new price. Then it is checked if the updated service is part of composed services. In that case for each composed service the sum of the prices for the components is compared to the current total price. If a decomposition is cheaper this is saved in the provider's data structure and the cost structure is updated recursively for the composed services.

When a provider p is requested a concrete offer for a service s then p simply adds his margin on top of his costs associated with s . The associated costs are calculated with the following three cases. If s is marked for decomposed execution, it costs the sum of the recursively calculated costs of its components. If p is the preferred provider for s , the internal costs are taken. Else an offer from the provider that p has assigned for s is requested.

Correctness and Scalability. Initially a provider assigns all services to himself and memorizes as costs either his internal costs or infinity for services that he is not capable of doing. In the succeeding phase the provider broadcasts his capabilities. We can assume that each provider adds a margin that leads to increasing prices ($\forall x \in \mathbb{R}, p \in P : m_p(x) > x$). Therefore at some point providers will dismiss further notifications based on their price. The notification phase is very similar to the *Routing Information Protocol* (RIP) [9]. With RIP routers on the internet exchange information about which networks they can reach. The costs are measured as the number of intermediary routers that are used by a router to reach the network. The different network speeds are equivalent to the margins charged by providers. RIP has proven to be correct by operating reliably the internet and other networks. However it has some scalability issues and is therefore replaced more and more by link-state based routing protocols. These protocols are not applicable to our problem domain, as they require that providers can gather complete topological information about the service network, including the margins of other providers. This is not a realistic option. The scalability issues will not be a problem in the near future as the service networks are supposed to be much smaller than the internet.

Routers operating with RIP resend their routing information every 30 seconds and delete routes that are not confirmed by such resendings. In this way the protocol deals with the removal of links or routers. This can also be applied to our algorithms in order to react to price increases or false advertisements. False advertisements are notifications of providers that they can deliver a service for a given price but always return higher prices in the offer phase. Other providers can detect such a behavior and remove the provider from their partner network.

In the offer making phase a provider knows exactly if he should decompose a service and to which other providers services are delegated. This efficiency for the frequent service offers is bought with increased costs for the propagation of changes in the service value network.

4 Network Simulation

We ran simulations of service value networks in order to compare them to an approach with a central registry. The following questions are analyzed with the simulation results:

1. What is the performance of the notification algorithm?
2. How many service requests can be fulfilled?
3. How large is the price increase for customers?
4. What is the impact on the welfare?

For the experiment we first create a service specification consisting of n_s services. Each of the n_p providers is capable of doing a randomly selected service s with internal costs randomly selected between $0.2 \cdot \text{complexity}(s)$ and $0.8 \cdot \text{complexity}(s)$. It is insured that every service has at least one provider. A provider p is assigned the margin function $m_p(x) = (1 + M_p) \cdot x$, where M_p is randomly chosen for every provider between 10% and 20%. The partner network is created by establishing $0.05 \cdot n_p^2$ random partnerships. The customer's valuation of a service s is given by $\text{complexity}(s)$.

For the evaluation of our approach we compare it to the case where a central service registry exists. This can be modelled as a provider that has partnerships with all other providers and charges no margin. The registry is able to serve a customer all services with optimal price. In some comparisons we assume that the registry also knows about the internal costs of individual providers and can therefore provide service executions with optimal welfare.

We ran the experiment with $n_s = 50$ services and varied the number of providers from 50 to 400 in steps of 25. We repeated this 100 times for each n_p and based on the average values, we came to the following results.

Performance of Algorithm (Question 1). The number of notifications that are sent between providers until all service pricing information is exchanged grows with speed of $O(n^3)$. As the number of providers also grows, this means an average number of $O(n^2)$ notifications per provider. Therefore the algorithm can be considered efficient.

Decentral Execution Ratio (Question 2). The decentral execution ratio is the probability that a request to a random provider for a random service results in a price that is lower than the valuation of the service. This ratio converges fast to the optimal value of 100% and is even for small numbers of providers ($n_p \leq 100$) in an acceptable range between 80% and 90%.

Price Increase for Customer (Question 3). We see in Figure 1 that the price increase for a customer requesting a random service from a random provider gets smaller with increasing size of the service network. In our simulation there is an inherent reason why

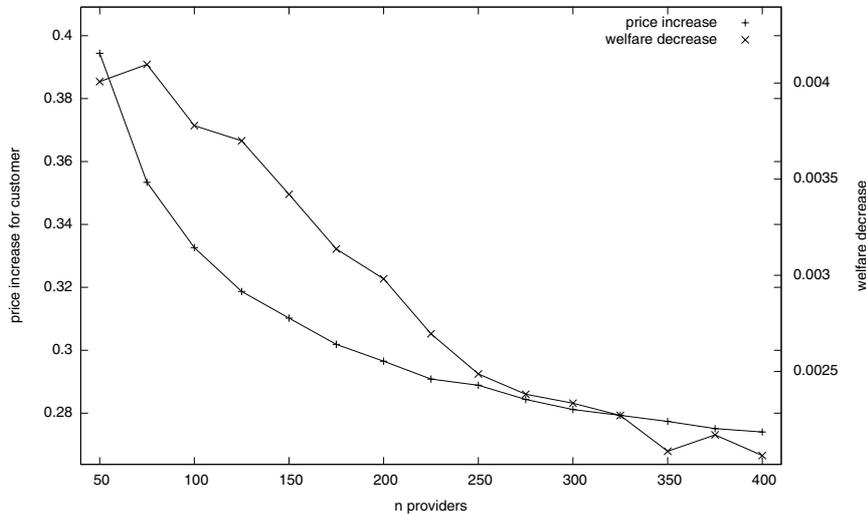


Fig. 1. Price increase for customer and welfare decrease for varying n_p from 50 to 400

a zero price increase is not possible: every provider can execute himself only one service and therefore has to delegate all other services, which means for almost all requests at least two providers are involved that both charge a margin. In the central case we assumed that the registry does not charge money and can therefore offer the best available price. The operation of such a central registry however is associated with costs that have to be reimbursed either by charging a margin or receiving payments from the service providers which will increase their costs. Therefore we conclude that with increasing service network size the prices get more competitive compared to a central scenario.

Welfare Decrease (Question 4). The participants in a service network are self-interested. The proposed algorithms aim at keeping prices low in order to stay competitive while ensuring that a given margin is earned. Our main concern is how the welfare of a decentral formed service value network is compared to a centrally planed cooperation. We observed in the simulation that the average welfare decrease over all services and providers behaves similar to the price increase (see Figure 1). However the decrease is always very small (below 0.6%) even for small networks.

We showed that the algorithm is efficient and delivers results that are competitive to a central approach with a registry that operates for free, which is optimal but unrealistic. Although customers have to pay slightly higher prices, practically all requests can be fulfilled without a significant welfare decrease. We also observed that a larger service network is better both for the customer and the overall welfare.

5 Conclusion

In this paper we considered the problem of decentralized service value network formation. We provided an algorithm that distributes a service request over a network of

self-interested, non-cooperative service providers and thereby creates an efficient service value network. The algorithm is novel compared to existing approaches as peers in the network do not have to provide any information about their business network to their customers. Thereby, new business models for service intermediaries are enabled, whose only business asset is a strong partner network. We showed by means of a simulation that the algorithm is tractable for reasonable sized scenarios as the number of required notifications in a network with n providers is $O(n^3)$. In addition, the results show that the algorithm performs quite well in terms of welfare decrease and price increase compared to a central scenario. In fact, the total loss in welfare is below 0.6%.

There are several directions in which we plan to extend this work. First, we plan to replace the current uniform distribution used to create the service network with a power-law distribution which seems to be a more realistic assumption for social networks as well as Web environments [10]. We plan an analytical and experimental evaluation how this change impacts the performance of our algorithms. Second, we plan to extend the algorithms for quality of service aspects. Modeling the trade-off between quality and price requires the introduction of multi-attribute price and value functions. Third, we plan to assess whether introducing a market mechanism such as a path auction might further increase the efficiency of the service allocation.

Acknowledgement. This work was partially funded by the German Research Foundation (DFG) in scope of the Graduate School Information Management and Market Engineering.

References

1. Papazoglou, M.: Web Services: Principles and Technologies. Prentice Hall, Englewood Cliffs (2007)
2. Tapscott, D., Lowy, A., Ticoll, D.: Digital Capital: Harnessing the Power of Business Webs. Harvard Business School Press (2000)
3. Hagel III, J.: Spider versus Spider. The McKinsey Quarterly (1), 4–5 (1996)
4. Steiner, F.: Formation and Early Growth of Business Webs: Modular Product Systems in Network Markets. Physica-Verlag, Heidelberg (2004)
5. Schmidt, C., Parashar, M.: A peer-to-peer approach to web service discovery. World Wide Web Journal 7(2) (2004)
6. Vu, L.H., Hauswirth, M., Aberer, K.: Towards P2P-based Semantic Web Service Discovery with QoS Support. In: Workshop on Business Processes and Services (BPS), pp. 18–31 (2006)
7. de Weerd, M., Zhang, Y., Klos, T.: Distributed task allocation in social networks. In: AAMAS 2007, pp. 1–8 (2007)
8. Speiser, S., Blau, B., Lamparter, S., Tai, S.: Formation of service value networks for decentralized service provisioning. Technical report, KSRI, Universität Karlsruhe, TH (2008)
9. Hedrick, C.: Routing Information Protocol. RFC 1058 (Historic) (June 1988) Updated by RFCs 1388, 1723
10. Faloutsos, M., Faloutsos, P., Faloutsos, C.: On power-law relationships of the internet topology. In: SIGCOMM 1999, pp. 251–262 (1999)