

# Challenges in Service Value Network Composition

Jaap Gordijn<sup>1</sup>, Ivan Razo-Zapata<sup>1</sup>, Pieter De Leenheer<sup>1, 3</sup>, and Roel Wieringa<sup>2</sup>

<sup>1</sup> Vrije Universiteit, Amsterdam, The Netherlands  
j.gordijn, i.s.razozapata, pieter.de.leenheer@vu.nl,

<sup>2</sup> Universiteit Twente, Enschede, The Netherlands  
R.J.Wieringa@utwente.nl,

<sup>3</sup> Collibra nv/sa, Brussels 12, Belgium

**Abstract.** Commercial services become increasingly important. Complex bundles of these services can be offered by multiple suppliers in a service value network. The *e<sup>3</sup>service* ontology proposes a framework for semi-automatically composing such a network. This paper addresses research challenges in service value network composition. As a demonstration of the state of the art, the *e<sup>3</sup>service* ontology is used. The challenges are explained using an example of an Internet service provider

**Key words:** service value network, bundling, composition, *e<sup>3</sup>service*

## 1 Introduction

Services comprise a significant part of the economy. For instance, in the USA approximately 81.1 % of the employees worked in the service industry in 2011 <sup>1</sup>.

Increasingly, such services are ordered and/or provisioned online. For instance, a cinema ticket can be ordered via the Internet, but the customer still has to travel to the cinema, where the service is delivered. Viewing a film, by contrast, can be ordered *and* provisioned online. Other examples are an email inbox, web-page hosting, or voice over IP (VOIP). The focus of this paper is on services that can be offered and provisioned online, see also Sect. 2, about the virtual ISP example.

Services are ordered and provisioned in a *service value network* (SVN) (see e.g. [13, 17, 1, 23] for SVN and related concepts). At minimum, a SVN consists of two actors, namely a supplier and a customer. However, in many cases, the SVN will consist of multiple suppliers, each offering a service, who together satisfy a complex customer need. The package of services satisfying the complex customer need is called the *service bundle*. By using multi-supplier service bundles, each supplier can concentrate on its own core competence, and can participate in satisfying a complex customer need, which it never could satisfy on its own. Moreover, a SVN may contain the suppliers of the suppliers and so on, until we reach the suppliers for which we safely can assume that their services can be provisioned in a known way.

<sup>1</sup> see <http://www.bls.gov/fls/flscomparelf.htm> table 7, visited June 21st, 2012

The observation that a SVN may consist of many suppliers leads to the conclusion that formation, or *composition* of the SVN is a research question in its own right. Specifically, if the customer need is ordered *and* provisioned online, the composition process should be software supported, and at least be semi-automatic. To this end, we introduce the notion of *computational* services; these are commercial services which are represented in a machine readable way, so that software can (semi) automatically reason about the required service bundle and the corresponding suppliers. We employ ontologies (see Sect.. 3) for representation and reasoning purposes.

The  $e^3$ service ontology[15, 23] and its predecessor *serviguration* [3] is an approach to semi-automatically compose service value networks. The  $e^3$ service approach takes two perspectives on service composition, namely a customer- and a supplier perspective, and tries to generate a multi-supplier service bundle and the corresponding SVN to satisfy a complex customer need. We use the  $e^3$ service ontology as the baseline for service value network composition.

The  $e^3$ service ontology is not to be confused with Web service technologies such as SOAP, WSDL and UDDI [8]. Whereas the focus of  $e^3$ service is on the composition of *commercial* services, SOAP, WSDL and UDDI facilitate interoperability between *software* services executing on various software and hardware platforms. Nevertheless, commercial services can be (partly) implemented by means of web service technology. After sketching the state of the art of  $e^3$ service , the contribution of this paper is to explain research challenges with respect to  $e^3$ service , including potential solution directions. Although the research challenges are described in terms of the  $e^3$ service work, we believe the challenges themselves are present in a broader context.

To facilitate the discussion, we create a hypothetical example about a virtual Internet service provider (ISP) (Sect.. 2). Thereafter, we discuss the state of the art with respect to  $e^3$ service , by using the virtual ISP example (Sect.. 3). Then we briefly state our vision about the composition of SVNs (Sect. 4). Subsequently, we present the research directions (Sect. 5). Finally, we present our conclusions (Sect. 6).

## 2 Example: The virtual Internet service provider

To illustrate the capabilities of, and research issues with respect to  $e^3$ service , we have constructed a hypothetical educational example about a virtual Internet service provider. This example is inspired on the example in [7, 15].

The virtual Internet service provider example assumes that an end user (the customer) wants to compose an Internet service provider out of elementary service offered by potentially different suppliers. For example, an offered service bundle may include only basic Internet access (then the bundle consists of only one service). In contrast, a service bundle may be complex such as basic Internet access, an email inbox, an email sending service (e.g. a SMTP service), web page hosting, voice over IP (telephony), a helpdesk, remote disk storage and back up and news. All these service can potentially be offered by different suppliers, so

that a multi-supplier service bundle emerges. Moreover, some services may be self-services. For example, the helpdesk service may consist of 1st, 2nd and 3rd line support, and the customer performs the 1st line helpdesk by himself.

### 3 $e^3$ service : State of the art

This section summarizes the current state of the art of  $e^3$ service . For a more detailed discussion, the reader is referred to [23, 22, 24] and [14]. Although the identified research challenges exist outside the context of  $e^3$ service , we take the state of the art of  $e^3$ service as our point of departure.

#### 3.1 Impedance mismatch between customer and supplier

A key problem in the composition of service value networks is the *mismatch* between the customer need and the offered service (bundle) by the supplier(s). The service bundle may contain several features (later called *consequences*) which are unwanted by the customer, or the bundle may miss required features as wanted by the customer.

*Example.* The user may want to communicate via text (e.g. email). However, the provider is offering the bundle consisting of email, voice over IP (VoIP), and Internet access. The mismatch is in the VoIP service which is not requested by the customer; the latter one (Internet access) is a required service needed to enable email and VoIP.

To address this mismatch,  $e^3$ service proposes two ontologies: (1) the customer ontology, and (2) the supplier ontology, including automated reasoning capacity.

#### 3.2 Customer ontology

The customer ontology borrows concepts and terminology from marketing (see e.g. [16] and [14]). Key notions in the supplier ontology are *need* [16, 2, 22, 14] and *consequence* [12, 14, 22]. According to [12, 14, 22], a consequence is the result from consuming valuable service outcomes. A need may be specified by various consequences [14, 22]. In the current work on  $e^3$ service (of Razo-Zapata et al., *ibid*) we focus mainly on *functional consequences*. In the previous example, we have already exemplified the notion of need and consequence.

#### 3.3 Supplier ontology

The supplier ontology is fully integrated with the  $e^3$ value ontology [10] and therefore borrows many concepts from the  $e^3$ value ontology [22]. Key concepts in the  $e^3$ value ontology are *actors* who perform *value activities* [10]. Actors can exchange things of economic value (*value objects*) with each other via *value transfers* [10].

*Example.* An actor can be an Internet service provider (ISP) who performs the activities of access provisioning, email inbox provisioning and email SMTP relaying, web / HTTP hosting, and more. To other actors (customers) a range of services (in terms of value objects) is offered, amongst others email inbox, SMTP relay and hosting of web pages.

To be able to connect the supplier ontology with the customer ontology, value objects have *consequences* too [22]. These consequences are from an ontological perspective similar to the consequences identified by the customer ontology. This allows for matching both kinds of consequences. The fact that a value object can have multiple consequences (and vice versa) models the situation that a customer obtains a value object as a *whole* (thus with all the consequences it consists of), whereas the customer might be interested in a subset of consequences. It is not possible to buy consequences separately, as they are packaged into a value object.

### 3.4 Reasoning support

In [23] different reasoning processes are employed than in [14]. We restrict ourselves to [23]. In [23], reasoning is explained as a Propose-Critique-Modify (PCM) [6] problem solving method, consisting of the following reasoning steps:

- Propose
  - Laddering: A technique to refine needs in terms of functional consequences [12, 23]. E.g. a complex need ( $N_1$ ) such as Assuring Business Continuity can be expressed in terms of: Data available in case of emergency ( $FC_1$ ), Application available 24/7 ( $FC_2$ ) and Regulatory compliance ( $FC_3$ ).
  - Offering: Determination what functional consequences can be offered by suppliers [23, 22, 21]. E.g. a backup service ( $S_1$ ) can offer FCs such as: Data available in case of emergency  $FC_A$ , Redundancy  $FC_B$ , Regulatory compliance ( $FC_C$ ), among others.
  - Matching: Match customer-desired consequences with supplier-offered consequences [23, 22, 21]. E.g. the customer-desired  $FC_1$  can be matched with the supplier-offered  $FC_A$  and  $FC_3$  with  $FC_C$ .
  - Bundling: Finding multi-supplier service bundles that satisfy the customer need [23, 22]. Bundles may partly satisfy the need, may overlap, or may precisely satisfy the need. E.g. since  $S_1$  cannot provide all the customer-desired  $FCs$ , an extra service such as remote desktop ( $S_2$ ) offering  $FC_2$  can be combined with  $S_1$  to generate a solution bundle.
  - Linking: Finding additional services needed by the suppliers that provide the service bundle [23, 21, 11]. E.g. a bundle composed of  $S_1$  and  $S_2$  might need to solve dependencies for  $S_2$  such as a versioning service that provides the updated O.S. to  $S_2$ .
- Verify
  - Analysis of provided, missing and non-required functional consequences and rating the importance of these consequences using a fuzzy inference system

[23]. E.g. an *SVN* providing  $FC_1$ ,  $FC_2$  and  $FC_3$  will fit better the customer-desired consequences (and will have a higher score) than any *SVN* providing only  $FC_1$  and  $FC_2$  or only  $FC_2$  and  $FC_3$ .

- Critique
  - In case the configuration task is unsuccessful: Identification of the source of failure [6, 23]. E.g. after composing an *SVN* offering  $FC_1$ ,  $FC_2$  and  $FC_3$ , the customer might realize that  $FC_3$  is not relevant for him. In this case the customer can indicate that he would like to get alternative *SVNs* only offering  $FC_1$  and  $FC_2$ .
- Modify
  - Modify the service network of the service bundle based on the results of the critique step cf. [6, 23]. E.g. based on the critique's output, new *SVNs* can be composed to better fit the customer-desired consequences.

## 4 Vision on composition of service value networks

Our long term vision can be characterized as follows:

- A multi-perspective view on composition and operation of service value networks. For instance, a business *value* perspective, a business *process* perspective, and an *IT* perspective may be relevant.
- Integration of the forementioned perspectives (e.g. cf. [20]). These perspectives *together* provide a blueprint of the *SVN* at hand.
- Various ways of composing *SVNs*, for instance hierarchical composition [15, 23] with one party that executes the composition in contrast to self-organization composition, in which participants themselves configure a *SVN*.
- Operationalization of the *SVN* in terms of processes and supporting IT. In some cases, IT can be dominant, as is for instance the case for the virtual ISP example.
- Reconfiguration of the *SVN*. In some cases it is necessary to reconfigure the *SVN* based on quality monitoring, disappearing actors, etc.

Although the issues described above might seem only applicable to our vision, areas such as Service-oriented Enterprise Architecture also deal with them by aiming at transparently merging business services (commercial services), software services (web services), platform services and infrastructure services (IT architecture) (see e.g. [25]).

## 5 Research challenges in service value networks

### 5.1 Terminologies for customer and supplier ontologies may differ

*Theme.* Ontology.

*Description of the challenge.* The current  $e^3$  service has two important assumptions. First, it is assumed that the customer and supplier ontology are linked to each other via a single *consequence* construct. Perhaps, multiple (e.g. more detailed) customer consequences may map onto one supplier consequence, or vice versa. Second, it is assumed that *both* the customer and suppliers *all* use the *same* terminology for stating the consequences. This challenge supposes first that links via the customer and supplier can involve more complex constructs than is the case right now (so, just one concept: the consequence). Second the challenge includes the idea that - given certain constructs to express what is needed and offered - the customer and suppliers can do so by using different terminology.

*Example.* With respect to the first assumption, the virtual ISP example may suppose a global customer consequence ‘being online’ that maps onto a supplier consequence ‘email’ + ‘internet access’. Considering the second assumption, in the virtual ISP example, the desired customer consequence can be ‘communicate via text’, whereas the stated supplier consequence can be ‘electronic mail’.

*Foreseen solution direction.* Concerning the *consequence* as matching construct, the goal is to allow for composition of consequences into more complex consequence constructs. E.g. various kind of relationships between consequences can be identified. For instance, in [15], a consequence can depend on other consequences, and can be in a core/enhancing bundling and an optional bundling relationship with other consequences. This can be extended with composition relationships.

With respect to the use of different terminologies, in [19, 4] a solution is proposed to match various *functionalities*, expressed in different terminologies, in the context of web services. Perhaps this kind of solution is also of use for commercial services, which are expressed by different terminologies.

## 5.2 The notion of consequence is a too high-level construct

*Theme.* Ontology.

*Description of the challenge.* Currently, the  $e^3$  service ontology matches customer needs with supplier service offerings via the notion of *consequence*. In [15], a distinction is made between *functional* consequences, and *quality* consequences. However, a more detailed structuring of the notion of consequences can be useful. It is for instance possible to distinguish various quality consequences, such as timely provisioning of the service, stability of the service supplier, etc.

*Example.* In case of the virtual ISP example it is possible that a supplier offers Internet access and another supplier offers VoIP (via the offered Internet access). In such a case, it is important that Internet access has sufficient quality for the VoIP service. In this context, quality can be stated by the bandwidth and latency of the network connection, which should be sufficient to carry the VoIP connection.

*Foreseen solution direction.* An ontology of both functional consequences and quality consequences should be made. Functional consequences are highly domain dependent but for quality consequences, theories on software quality can be of use and SERQUAL [18], a theory on quality properties of commercial services. Finally, the Unified Service Description Language (USDL)<sup>2</sup> [5] may be a source.

### 5.3 Matching of customer needs with supplier service offerings is broker-based

*Theme.* Reasoning.

*Description of the challenge.* Different approaches can be followed to match customer needs with supplier offerings. In this paper, we distinguish *hierarchical* matching and *self organizing* matching. In case of hierarchical matching, there is a party (e.g. a broker) that controls and executes the matching process. Suppliers are simply told by the broker to provide their services in a bundle. In the current work ([23]) matching is done via a broker. The position of the party who performs the matching is powerful from a business perspective, since such a party determines which actors are providing which services. Other matching models can be distinguished, for instance self organizing models, in which actors collaborate and negotiate about the service bundle to be provided to the customer and there is no central coordinator.

*Example.* In the virtual ISP example, there would be in the current *e<sup>3</sup>service* implementation a specific party (the broker) who performs the matching process. This process includes eliciting customer needs, finding the appropriate service bundles and assigning specific services in the bundle to individual suppliers.

*Foreseen solution direction.* Hierarchical matching is currently partly supported by *e<sup>3</sup>service* as an intermediate top-level party performing the matching process. The current process can be extended by supporting multiple matching parties who are organized in a matching hierarchy. Additionally, self organizing matching should be supported (as this is an entirely different business model), e.g. via gossiping protocols, which avoid central components such as a broker (see e.g. [9] for gossiping in computer networks).

### 5.4 Restricted knowledge used for need and consequence elicitation

*Theme.* Reasoning.

<sup>2</sup> <http://www.internet-of-services.com/index.php?id=570&L=0>, visited June 21st, 2012

*Description of the challenge.* The current implementation of  $e^3$ service supposes business-to-consumer (B2C) and business-to-business (B2B) relationships. B2C interaction plays a role while executing customer need and consequence elicitation, based on the customer need and service catalogues. B2B relationships play a role during the linking process: if a supplier offers a service to the customer, it is possible that the supplier itself requires services from other suppliers. This is referred to as linking. It is possible that customer to customer (C2C) interaction may play a role during the customer need and consequence elicitation process. For instance, a service value network with the belonging consequences may be built that closely resembles the service value network (and consequences) generated for another customer.

*Example.* Suppose that a particular customer uses a bundle of Internet access + email (inbox and SMTP) + VoIP, and is satisfied with the bundle. Via a recommender system, this customer may publish his/her experiences with the used service bundle at hand. The service value web configuration components may use information about this published bundle as an example for other service bundles.

*Foreseen solution direction.* Customer to customer recommendation systems may be used as an input to create a recommender system that registers customer's scores on particular consequences. These scores can then be used in the customer need and consequences elicitation process.

## 5.5 Implementation of $e^3$ service by web services

*Theme.* Software tool support.

*Description of the challenge.* The software implementation of  $e^3$ service is currently Java- and RDF-based. It is possible to think of the software as a set of web services and associated processes that perform the composition of the SVN. Moreover, these web services may be offered (and requested) by multiple suppliers and the customer, so that the composition becomes a distributed task.

*Example.* In the virtual ISP example, each enterprise that potentially wants to participate in a SVN can offer a set web services. These web services allow the enterprise to participate in the composition process.

*Foreseen solution direction.* We foresee the use of web-service standards, such SOAP and WSDL to build a configurator that can run as decentralized (meaning: at the customer and suppliers sites) as much as possible. Moreover, a self-organizing implementation obviously should support a *fully* decentralized architecture.

## 6 Conclusion

In this paper, we have introduced a number of research challenges with respect to commercial service value networks in general and the  $e^3$ service ontology in particular. By no means, the list of challenges is complete. The first challenge is to allow a more complex conceptualisation of service characteristics as well as the use of different terminology by the customer and suppliers of services. Another research challenge is to develop a more detailed ontology for functional and quality consequences. Currently,  $e^3$ service uses a brokerage approach for matching. A different approach to be investigated is the self-organizing approach. Furthermore, a research challenge is how to use customer to customer interactions in the process of elicitation of customer needs and consequences. Finally, a research challenge is how to enable the current  $e^3$ service framework in terms of web services.

**Acknowledgments.** The research leading to these results has received funding from the NWO/Jacquard project VALUE-IT no 630.001.205.

## References

1. Verna Allee. A value network approach for modeling and measuring intangibles. In *Transparent Enterprise Conference*, 2002.
2. J. Arnd. How broad should the marketing concept be? *Journal of Marketing*, 42(1):101–103, january 1978.
3. Ziv Baida. *Software-aided Service Bundling - Intelligent Methods & Tools for Graphical Service Modeling*. PhD thesis, Vrije Universiteit, Amsterdam, NL, 2006.
4. Christian Bizer, Tom Heath, and Tim Berners-Lee. Linked Data - The Story So Far. *International Journal on Semantic Web and Information Systems (IJSWIS)*, 5(3):1–22, MarMar 2009.
5. Jorge Cardoso, Alistair Barros, Norman May, and Uwe Kylaue. Towards a unified service description language for the internet of services: Requirements and first developments. In *IEEE International Conference on Services Computing. Florida, USA*. IEEE Computer Society Press, 2010.
6. Balakrishnan Chandrasekaran. Design problem solving: A task analysis. *AI Magazine*, (11):59–71, 1990.
7. Ania Chmielowiec, Guillaume Pierre, Jaap Gordijn, and Maarten Van Steen. Technical challenges in market-driven automated service provisioning. In Karl M. Goschka, Schahram Dustdar, Frank Leymann, and Vladimir Tasic, editors, *Proceedings of the 3rd Workshop on Middleware for Service Oriented Computing, MW4SOC*, pages 25–30. ACM, 2008.
8. Francisco Curbera, Matthew Duftler, Rania Khalaf, William Nagy, Nirmal Mukhi, and Sanjiva Weerawarana. Unraveling the web services web: An introduction to soap, wsdl, and uddi. *IEEE Internet Computing*, 6:86–93, 2002.
9. Anwitaman Datta, Silvia Quarteroni, and Karl Aberer. Autonomous gossiping: A self-organizing epidemic algorithm for selective information dissemination in wireless mobile ad-hoc networks. In Mokrane Bouzeghoub, Carole Goble, Vipul Kashyap, and Stefano Spaccapietra, editors, *Semantics of a Networked World*,

- volume 3226 of *Lecture Notes in Computer Science*, pages 126–143. Springer Berlin / Heidelberg, 2004.
10. Jaap Gordijn and Hans Akkermans. Value based requirements engineering: Exploring innovative e-commerce idea. *Requirements Engineering Journal*, 8(2):114–134, 2003.
  11. Jaap Gordijn, Pieter De Leenheer, and Ivan Razo-Zapata. Generating service valuwewebs by hierarchical configuration: An ipr case. In *Proceedings of HICSS 44*, 2011.
  12. J. Gutman and T.J. Reynolds. Laddering theory-analysis and interpretation. *Journal of Advertising Research*, 28(1):11–31, february/march 1988.
  13. John Hamilton. Service value networks: Value, performance and strategy for the services industry. *Journal of Systems Science and Systems Engineering*, 13(4):469–489, 2004.
  14. Sybren Kinderen, de. *Needs-driven service bundling in a multi-supplier setting ? The computational e3service approach*. PhD thesis, Vrije Universiteit, 2010.
  15. Sybren Kinderen, de, Jaap Gordijn, and Hans Akkermans. Reasoning about customer needs in multi-supplier ict service bundles using decision models. In *Proceedings of the 11th International Conference on Enterprise Information Systems*, pages 131–136, 2009.
  16. Philip Kotler and Kevin Keller. *Marketing Management*. Prentice Hall, Englewood Cliffs, NJ, 2006.
  17. Christopher H. Lovelock and Jochen Wirtz. *Services Marketing: People, Technology, Strategy, 7th edition*. Pearson Higher Education, 2010.
  18. A. Parasuraman, V. A. Zeithaml, and L. L. Berry. A conceptual model of service quality and its implication. *The Journal of Marketing*, 49(4):4150, 1985.
  19. Carlos Pedrinaci and John Domingue. Toward the next wave of services: Linked services for the web of data. *J. UCS*, 16(13):1694–1719, 2010.
  20. Vincent Pijpers, Pieter De Leenheer, Jaap Gordijn, and Hans Akkermans. Using conceptual models to explore business-ict alignment in networked value constellations case studies from the dutch aviation industry, spanish electricity industry and dutch telecom industry. *Requirements Engineering Journal*, 2011, online available.
  21. Ivan Razo-Zapata, Pieter De Leenheer, and Hans Akkermans. Service value networks for competency-driven educational services: A case study. In *Proceedings of the 6th international BUSITAL workshop, 2011*, 2011.
  22. Ivan S. Razo-Zapata, Jaap Gordijn, Pieter De Leenheer, and Hans Akkermans. Dynamic cluster-based service bundling: A value-oriented framework. In *Commerce and Enterprise Computing (CEC), 2011 IEEE 13th Conference on*, 2011.
  23. Ivan S. Razo-Zapata, Pieter De Leenheer, Jaap Gordijn, and Hans Akkermans. Fuzzy verification of service value networks. In Jolita Ralyte and Xavier Franch, editors, *24th International Conference on Advanced Information Systems Engineering (CAiSE'12)*. Springer, 2012.
  24. Ivan S. Razo-Zapata, Pieter De Leenheer, Jaap Gordijn, and Hans Akkermans. *Handbook of Service Description: USDL and its Methods*, chapter Service Network Approaches, pages 45 – 74. Springer, 2012.
  25. Alain Wegmann, Gil Regev, Irina Rychkova, Lam-Son Lê, José Diego De La Cruz, and Philippe Julia. Business-IT Alignment with SEAM for Enterprise Architecture. In *The 11th IEEE International EDOC Conference (EDOC 2007)*. IEEE, 2007.