Modeling Service Relationships for Service Networks

Jorge Cardoso

CISUC, Department of Informatics Engineering
University of Coimbra, Polo II, 3030 Coimbra, Portugal
`jcardoso@dei.uc.pt`

Abstract. The last decade has seen an increased interest in the study of networks in many fields of science. Examples are numerous, from sociology to biology, and to physical systems such as power grids. Nonetheless, the field of service networks has received less attention. Previous research has mainly tackled the modeling of single service systems and service compositions, often focusing only on studying temporal relationships between services. The objective of this paper is to propose a computational model to represent the various types of relationships which can be established between services systems to model service networks. This work acquires a particular importance since the study of service networks can bring new scientific discoveries on how service-based economies operate at a global scale.

Keywords: service relationship, service system, business service, open service, service network, semantic Web.

1 Introduction

Many systems around us can be described by network models, which are structures consisting of nodes connected by edges. The examples available are numerous and range from social networks, to the Internet, to supply chains, and to power grids. The global economy is itself a complex network composed of national economies, which are themselves networks of markets, and markets are also networks of providers, brokers, intermediaries, and consumers.

Understanding how services systems[1] evolve as networks and the risks and gains of different topologies is becoming increasingly critical for society [1]. Vargo et al. [2], and others, have also perceived that society is moving into a service-dominant system. Nonetheless, our knowledge on global service networks is limited. Understanding the dynamics and laws governing service networks can provide authoritative insights on *why* and *how* financial service systems fail. For example, it can explain how the 2007–2012 global financial crisis propagated throughout global service networks. It can also provide scientific grounds for the engineering of efficient and robust service network topologies to resist adverse environments.

---

[1] When no ambiguity arises, we will use the term *service* to refer to a *service system.*

J.F. e Cunha, M. Snene, and H. Nóvoa (Eds.): IESS 2013, LNBIP 143, pp. 114–128, 2013.

While service markets are ubiquitous, the study of service networks did not receive the needed attention. Research has been mainly done from a technical perspective by modeling single services as software components (e.g. Web services [3]). Business process and workflow management have also looked into how services can be composed to form process models by establishing temporal dependencies between services (c.f. [4]). The goal of the research conducted was not to model service networks as the representation of economic activities, but to model the technical interfaces that need to be in place to integrate information systems to operate in heterogeneous environments.

In [5,6] we proposed to model service networks by constructing what we call *Open Semantic Service Networks (OSSN)*. These networks are constructed by accessing, retrieving, and combining information from *service systems* and *relationship models* globally distributed. With respect to the modeling of services, we have developed a family of languages named *-USDL (the Unified Service Description Language)[7,8] to provide computer-understandable descriptions for business services. These languages[2,3,4] allow to formalize business services in such a way that they can be used effectively, for example, for dynamic service outsourcing and automatic service contract negotiation.

Since our previous work yielded suitable computational models to represent service systems, the objective of this paper is to propose a model to represent the various types of relationships which can exist in a service network. The model developed, and called *Open Semantic Service Relationship (OSSR)* model, is computer-understandable, is represented with semantic Web languages, and defines the main concepts and properties required to established rich semantic relationships between service models. We believe that the importance and expressiveness of relationships has been overlooked in many fields. Gradde and Snehota [9] also believe that existing studies on relationships in the field of business models usually oversimplify business representations. For example, we consider that the simple relations used by other modeling initiatives such Linked Data [10] to interconnect data – using `rdfs:subClassOf`, `owl:EquivalentClass`, and `owl:sameAs` – the relation `foaf:knows` from FOAF [11] to interconnect people, and the use of `rdfs:seeAlso` by SIOC [12] to interconnect documents are strict and limited relationships not suitable to connect service systems. Therefore, we developed a multi-layer relationship model which links services via multiple types of connecting perspectives (e.g. participating roles, interconnection level, and involvement strength [13,14,15,9]) capturing the richness, complexity, and characteristics of services. This goes well beyond the connection of service systems treated simply as unidimensional nodes.

This paper is organized as follows. In the next section, we present important definitions and illustrate application domains for service networks to serve as motivation scenarios. Section 3 describes the multi-level relationship model developed to connect service systems. Section 4 describes the evaluation and

---

[2] Linked-USDL = `http://linked-usdl.org/`

[3] α-USDL = `http://www.genssiz.org/research/service-modeling/alpha-usdl/`

[4] USDL = `http://www.w3.org/2005/Incubator/usdl/`

implementation of the model. Section 5 presents the related work in this field of research. Section 6 is the conclusion.

2 Definitions and Motivation Scenarios

A *service network* is defined as a graph structure composed of service systems which are nodes connected by one or more specific types of service relationship, the edges. A *service system* is a functional unit with a boundary through which interactions occur with the environment, and, especially, with other service systems. Service networks are similar to social networks in their structure but connect service systems. They are different from process models since they do not place an emphasis on control-flow, temporal dependencies, and cases. We illustrate their possible use with two application domains.

Regulation of Service Markets. The analysis of service networks can detect topological patterns such as oligopolies, monopolies, or 'cartels' in service markets. For example, a power-law distribution pattern can be used to identify oligopolies since it implies that only a few large service providers exist, whereas the occurrence of small providers is extremely common. The identification of such network characteristics or anomalies are of importance for regulatory bodies such as the EU which routinely passes directives for European markets on laws to be followed.

Supply Chain Management. While supply-chain management is crucial for many companies, today, there is no practical and automated solution to analyze global supply-chain networks. The inexistence of global models only enables to study this type of networks from a local, reduced, and naïve view (c.f. [16]). The development of computational models will give firms a better understanding of the dynamic behavior of supply chain networks at a global scale.

3 Multi-layer Relationship Model

When examining previous approaches to model network structures by using semantic Web languages, relationships were often overlooked by placing the emphasis on nodes. For example, the use of simple primitives such as `foaf:knows`, `rdfs:subClassOf`, `owl:EquivalentClass`, `rdf:seeAlso` and `owl:sameAs` to connect people, data, and community generated documents (c.f. [10,11,12]) is limited for service networks. It only enables to create networks with one homogeneous layer thus limiting the types of analysis which can be made. The richness of service systems – which involve people, laws, resources, operations, processes, service levels, etc. – requires a different approach based on the use of multiple layers to construct service networks. For example, two service systems can be related by describing the roles they can take, by representing the strength of the relationship, and by establishing a comparison of the set of functionalities provided. This goes well beyond the connection of entities seen simply as unidimensional nodes.

To construct a comprehensive relationship model, we followed an inductive research approach. We conducted a literature review on work describing and discussing the types of relationships which exist in organizations applicable to the field of services. We electronically searched the titles, abstracts, keywords, and full texts of articles in GoogleScholar (`scholar.google.com`), SpringerLink (`link.springer.com`), Taylor & Francis (`www.tandf.co.uk`), and Google Books (`books.google.com`) for the main word string "relationship". The search included several variations of the original term like "service relationship", "service systems relationship", "business relationship" or "relationship model". Articles were read to determine there relevance for modeling relationships between service systems.

We identified propositions and generalized them in a theoretical multi-layer relationship model composed of six layers: 1) *role*, 2) *level*, 3) *involvement*, 4) *comparison*, 5) *association*, and 6) *causality*. OSSR comprises at total of 15 top level concepts, namely `Relationship`, `Service`, `Source`, `Target`, `Role`, `Level`, `Involvement`, `Comparison`, `Association`, `Causality`, `Cause`, `Link`, `Effect`, `Category`, and `KPI`. The layers are grouped together using the central concept `Relationship`. One of the endpoints of the relationship is the service source (`Source`) and the other one is the service target (`Target`). Both are subclasses of the concept `Service` which represents a service system possibly modeled with a language such as Linked-USDL (see §1 and §4).

The layers and concepts[5] are summarized in Table 1, illustrated in Figure 1, and described in the following sections. While the examples given are mainly from the field of Software-as-a-Service (SaaS), the model was designed to be applied to services that range from human-based services to fully automated software-based services.

Table 1. The multiple layers of the Open Semantic Service Relationship (OSSR) model

Layer	Description
Role [14]	The role of the service sytems involved in a relationship.
Level [15]	The level (e.g. activity, resources, or people) at which a relation is established.
Involvement [9]	The strength of a relationship.
Comparison [17]	The comparison of service systems involved in a relationship.
Association [18]	An expression of the 'a part of' relation between two service systems.
Causality [19]	The influence that key performance indicators of one service system has in another service system.

---

[5] The terms written using the `typewriter` font indicate a concept or property value of the OSSR model.

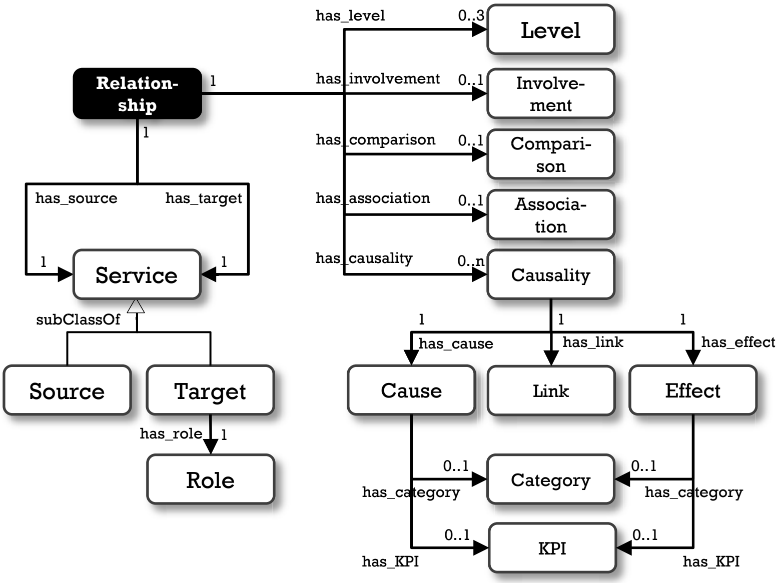

Fig. 1. The structure of the OSSR model

3.1 Participating Roles

Understanding roles is an important aspect to determine the position of a service system in a network. For example, a service can create alliances with complementors to differentiate itself from competition to deliver more value to customers. We rely on the work from Ritter et al. [14] to classify the role of the service systems involved in a relationship in four distinct types captured with the concept `Role`:

1. Customer,
2. Supplier,
3. Competitor, and
4. Complementor.

A service source which establishes a relationship with a service target with the role of `Customer` focuses on a good working mode with customers keeping always in mind the co-creation of value during service provisioning. A relationship with a service target of type `Supplier` focuses often on a durable stream of competitive advantage which maybe hard for others to imitate or break.

Complementors are the mirror image of competitors. In other words, customers value a service more when complementors exist whereas they value a service less when competitors exist [20]. A relationship with a target service of type `Complementor` enables a service source to increase its value by adding external operations to it. A service system S_c is a complementor if customers value service system S_i more when they have service S_c than when they have service S_i alone. One example is joining a flight transportation service, an accommodation service and a car renting service whereby services cooperate in reaching out to customers in the form of value added promotions. Finally, a service system can establish a relationship of type `Competitor` with a service target that belongs to a group of competiting firms. In other words, service S_c is a competitor if customers value service S_i less when they have access to service S_c than when they have service S_i alone.

For example, the SaaS SugarCRM has several competitors including SalesForce.com Sales Cloud, Microsoft On-Demand Dynamics CRM, and Oracle CRM OnDemand. Avis Scandinavia, a company providing rental services, is a customer of SugarCRM. Sage ERP and Sugar ERP Business Suite are complementors of SugarCRM. The SugarCRM service is a customer of Oracle and IBM since it relies on Oracle 11g or IBM DB2 database support services.

3.2 Interconnection Level

It is fundamental that a service system relates its activities, its actors, and its resources to those of other firms' services to streamline integration. Only the consideration of various levels can led to a sound integration of service system layers into networks. Håkansson and Snehota [15] showed that relationships can perform a variety of actions through:

1. Activity links,
2. Actor bonds, and
3. Resource ties.

This classification is captured by the concept `Level` which is associated with the concept `Relationship`. An activity link (modeled with the concept `ActivityLink`) refers to the integration of activities, tasks, or operations executed under the control of two service systems. Many arguments for closely integrating activities between manufacturers, suppliers, and customers originated from the fields of business process management and business process reengineering. In other words, this concept makes it possible to create cross-organizational workflows and business processes underlying a service network.

Actor bonds (modeled with the concept `ActorBond`) refer to the interaction among participants belonging to the human resource structure of distinct services. The objective of this concept is to enable the analysis and reasoning on social enterprise networks.

Resource ties (modeled with the concept `ResourceTie`) refer to the exchange of resources types. According to the resource-based view [21], service systems can differentiate themselves and increase their competitiveness by using heterogeneous, immobile, valuable, rare, inimitable, and non-substitutable resources. The concept of resource ties is aligned with the concept of value exchanged within companies in e^{3}value [22]. For example, the Progress Apama SaaS for complex event processing for capital markets establishes resource ties in the form of events with the stock exchange market service. This latter service executes trading operations and continuously sends events to Progress Apama.

These three levels of integration can benefit from extending the OSSR model by referencing Linked Data concepts to provide more information on the activities, actors, and resources exchanged among two service systems. For example, references using `URI` can be made to the type of resources being exchanged: `'US dollars'` or 'BMW PART NR. 11127790052'.

3.3 Involvement Strength

The concept `Involvement` represents stakeholders willingness to establish a partnership. Gradde and Snehota [9] proposed to make a qualitative evaluation of relationships strength by using intensity properties. Higher levels of involvement usually mean that both parties are more interested to establish a long-term partnership while lower levels of involvement suggest that both parties choose for a more simplified relationship. The concept can take two forms:

1. Low-involvement and
2. High-involvement.

While `High-involvement` relationships are associated with investment logic, `Low-involvement` relationships can be handled with limited coordination, adaptation and interaction costs. For example, the SaaS SugarCRM establishes partnerships with technology partners such as Epicom Corporation[6], a company providing customization services, among others services. A relationship between these two service systems can be classified as low- or high-involvement depending on the number of customized business applications made by Epicom for SugarCRM, and the number of customers and users the customizations have.

3.4 Functional Comparison

Comparison consists in the identification of similarities and differences between service systems. A service system can be described by the functionalities and characteristics it provides (i.e. activities, operations, functions, options, etc.). Let us consider that the set of functionalities and characteristics provided by a service is represented by $fc(S_i)$. When comparing two service systems, we can identify five possible comparison cases (the cases were derived from set theory [17] and object-oriented programming [18]) expressing the degree of equivalence between two services:

1. $fc(S_i)$ is `equivalent` to $fc(S_j)$, $(fc(S_i) \equiv fc(S_j))$,
2. $fc(S_i)$ is a `generalization` of $fc(S_j)$, $(fc(S_i) \subsetneq fc(S_j))$,

---

[6] `http://www.epicom.com/`

3. $fc(S_i)$ is a `specialization` of $fc(S_j)$, $(fc(S_i) \supsetneq fc(S_j))$,
4. $fc(S_i)$ is `similar` to $fc(S_j)$, $(FC = fc(S_i) \cap fc(S_j), FC \neq \varnothing \wedge fc(S_i) \wedge FC \neq fc(S_i) \neq fc(S_j))$, and
5. $fc(S_i)$ is `different` from $fc(S_j)$, $(fc(S_i) \cap fc(S_j) \in \varnothing)$.

These cases are captured by the concept `Comparison`. Comparing two service systems cannot be viewed as a precise science and has often a high degree of subjectivity, especially when services involve sociotechnical subsystems. Subjectivity is an intrinsic aspect of the physical world. For example, the characteristics of an object in the physical world depends on the direction from which it is viewed. Therefore, all observations of physical characteristics are relative to the frame of reference of the observer, and the results reflect the state of observer.

Two service systems are `Equivalent` (full equivalence) when they are identical in their functionalities and characteristics. The specialization and generalization relationships are both reciprocal and hierarchical. The value `Generalization` (partial equivalence) expresses that a service has a narrower set of functionalities than another. The value `Specialization` (partial equivalence) expresses that a service has a broader set of functionalities than the other one. A `specialization` has the same semantics of the generalization relation but works in the opposite direction.

For example, the SaaS SugarCRM provides four packages: professional, corporate, enterprise, and ultimate. The base service is the same but the packages offer a different set of functionalities and characteristics. In other words, there is a implicit containment hierarchy $fc(S_{professional}) \subset fc(S_{corporate}) \subset fc(S_{enterprise}) \subset fc(S_{ultimate})$. The professional service has all the features that the corporate service has but does not include the option Sugar Mobile Plus; and the ultimate service is the only service providing 250GB Sugar On-Demand Storage. Therefore, the SugarCRM professional service is a `generalization` of the corporate service and the ultimate service is a `specialization` of all the others.

The value `Similar` (inexact equivalence) expresses that services are similar. Some functionalities intersect while others are disjoint.

A relationship of type `Different` indicates that two services do not have any functionality in common.

3.5 Service Association

The association of service systems enables to combine simpler services into more complex service systems. Associations are a critical building block of many fields of science (e.g. biology, physics, and programming). The concept `Association` can take the form of an:

1. Aggregation or a
2. Composition.

An association of type aggregation expresses 'a part of' or 'has a' relationship between two service systems. One of the services has the role of assembly and the other one has the role of component. The value `AggregationBy` indicates that the service source has the role of assembly and the service target has the role of component. The value `AggregationOf` is the inverse relation. It indicates that the service source has the role of component and the target service has the role of assembly. For example, an airline service is an aggregation of security, check-in, catering, handling, and cleaning services. Another example from the SaaS arena is the Internet self-service named IT Incident Management Service (ITIMS) adapted from ITIL best practices and described in [23]. The service is an aggregation which relies on three SaaS components to operate: the platform provider `Heroku.com`, the database provider `MongoDB.com`, and the email gateway provider `McAfee.com`. In other words, ITIMS establishes an `AggregationBy` with three other SaaS which take the role of role of components.

A composition is a specialized form of strong aggregation where component services cease to exist, or are not needed, if the assembly service ceases to exist. The value `CompositionBy` is the inverse of value `CompositionOf`. It indicates that the service source has the role of component and the target service has the role of assembly.

3.6 Causality between Services

A relationship has also associated a `Causality` concept. Causality or cause-effect describe how a *cause* event occurring in a service system has an *effect* in another service system. Causality is expressed using key performance indicators (KPI) of service systems which are connected. KPIs are often associated with service level (see *-USDL) and quality of service (QoS), and include parameters such as availability, cost, downtime, errors, response rime, etc. For example, the Invoice Accuracy KPI of a service provider to control the quality of service S_p can be connected to the Time Delivery KPI of a service customer S_c. An increase of the first KPI will originate an increase in the second KPI since it will take more time to resolve errors. This cause-effect relation between KPIs enables to conduct the quantitative analysis of the propagation of changes or domino effect in a service network.

The system dynamics or systems thinking approach [19] is used to capture and enable the posteriori analysis of service networks. Instead of looking at causes (captured with the concept `Cause`) and their effects (captured with the concept `Effect`) in isolation, systems thinking enables to look at service networks as a system made up of interacting parts. The concept `Link` connects a cause to and sets the sign that a directed link can take: `Positive` or `Negative`. A positive link indicates that a change (increase or decrease) in a service KPI results in the same type of change (increase or decrease) in another service KPI. A negative link indicates that a change (increase or decrease) in a service KPI results in the opposite change (decrease or increase) in another service KPI.

By using the concept of causality it becomes possible to express and quantify the impact that one service system has in other service systems. This capability brings an important contribution to service networks. It enables to think about a service network as a complex dynamic system to study how a service behavior affects the provisioning of other services. Its application to global networks will make it possible to discover new scientific insights on global digital service economies.

Since KPIs are often domain dependent and their semantics may not always be clear to analysts, individual measures of performance in a cause-effects relation are classified with a schema composed of five elements (c.f. [24]) captured by the concept `Category`:

1. Quality,
2. Time,
3. Cost,
4. Flexibility, and
5. Other.

The use of this schema provides a level of abstraction which enables, for example, a time-based analysis of service network. Since the meaning of the elements that compose the schema is intuitive they will not be further explained.

4 Evaluation and Implementation

To evaluate the OSSR model from a user, expert, and ontology engineer point-of-view, we have followed the frame of reference proposed in [25]. It consisted in verifying several aspects of the model.

- *Consistency.* In this phase, we tried to identify possible design errors. We did not find circular definitions; the model was syntactically correct; it was validated using Protégé and Jena; several instances of the model were created; no contradictory knowledge was detected; and all concepts were consistent with the theoretical definitions of relationships.
- *Completeness, expandability, and sensitiveness.* In a second phase, we tried to locate concepts whose modeling was incomplete by reexamining the literature on relationships. We looked at the OSSR model from a holistic perspective and we have identified that the causality concept required an additional element to enable the dynamic analysis of service networks: the direction of the cause-effect link. We believe that the model is not complete and additional relations types will be added in the future as the model is experimented in industrial settings. The model is expandable since it is constructed based on the notion of layers: new relations can be added without altering the set of well-defined relations that are already guaranteed. The use of layers also make the model relatively insensitive to small changes.
- *Conciseness.* We proved the conciseness of the model by asserting that it did not contained redundant or unnecessary definitions. Redundancies could not be inferred using other knowledge.

The OSSR model was considered to be valid from a conceptual and formal point-of-view.

Our idea behind the implementation of service relationships is pragmatic and it is based on the objective to create a linked global service network using machine-readable descriptions [26]. Therefore, the model was implemented using the Resource Description Framework (RDF) which allows semantic information to be expressed as a graph. To improve the integration with other semantic Web initiatives, the model establishes links with various existing ontologies to reuse concepts from vertical and horizontal domains such as SKOS (taxonomies), Dublin Core (documents), Linked-USDL (service descriptions) and so on. The implementation is available at `http://rdfs.genssiz.org/ossr.rdf`.

The OSSR model was designed to integrate with Linked-USDL. In other words, a relationship connects two service systems which can be represented with Linked-USDL. Since the model references services using a URI, other service descriptions can be used (e.g. WSDL, OWL-S, and *-USDL [7,8]). Compared to the various choices available, the use of USDL has many benefits since it bridges a business, an operational, and a technical perspective to describe services. Once services and relationships are described in RDF with Linked-USDL and OSSR, respectively, it becomes possible to make queries over distributed service networks using the SPARQL RDF query language [27].

Listing 1 illustrates the use of the OSSR model. The relationship created, identified by `http://rdfs.genssiz.org/ossr/2012/10/`, relates two services: SurgarCRM and MySQL. The first service was modeled with Linked-USDL and its modeling is available at `http://rdfs.genssiz.org/SurgarCRM.ttl`. The model was based on the SaaS customer relationship management software available at `http://www.sugarcrm.com/`. The second service modeled was MySQL (`http://www.mysql.com/`). The relationship was modeled from the SugarCRM point-of-view since the OSSR model specifies that it is the source element.

```
<ossr:Relationship rdf:about=”http://rdfs.genssiz.org/ossr/2012/10/”>
    <ossr:has_source>
      <ossr:Source>
      </ossr:Source>
        <ossr:has_service>http://rdfs.genssiz.org/SurgarCRM#
            offering_SugarCRM</ossr:has_service>
    </ossr:has_source>
    <ossr:has_target>
      <ossr:Target>
        <ossr:has_service>http://rdfs.genssiz.org/MySQL#
            offering_MySQL</ossr:has_service>
         <ossr:has_role>Provider</ossr:has_role>
      </ossr:Target>
    </ossr:has_target>
    <ossr:has_involvement>HighInvolvement</ossr:has_involvement>
    <ossr:has_level>ActivityLink</ossr:has_level>
    <ossr:has_comparison>Different</ossr:has_comparison>
    <ossr:has_association>AggregationOf</ossr:has_association>
    <ossr:has_causality>
```

```
        <ossr:Causality>
            <ossr:has_cause>
                <ossr:Cause>
                  <ossr:has_KPI>http://rdfs.genssiz.org/MySQL#
                      var_MySQL_Reliability </ossr:has_KPI>
                    <ossr:has_category>Quality</ossr:has_category>
                </ossr:Cause>
            </ossr:has_cause>
            <ossr:has_link>
                <ossr:Link>
                    <ossr:has_direction>Positive</ossr:has_direction>
                </ossr:Link>
            </ossr:has_link>
            <ossr:has_effect>
                <ossr:Effect>
              <ossr:has_KPI>
            http://rdfs.genssiz.org/SurgarCRM#
                var_SugarCRM_AvailabilityGuarantee_Value
              </ossr:has_KPI>
                    <ossr:has_category> Quality </ossr:has_category>
                </ossr:Effect>
            </ossr:has_effect>
        </ossr:Causality>
    </ossr:has_causality>
</ossr:Relationship>
```

Listing 1. Example of an OSSR relationship relating two service systems

The relationship indicates that the MySQL service system is a provider of the SugarCRM service; the two services have a high degree of involvement; they establish a relation at the activity level; the service systems are different; and MySQL is a component of the SugarCRM service offering. Finally, a cause-effect relation on quality is established between the KPI `Availability` of the SugarCRM and the KPI `Reliability` of MySQL. This causality is positive since when the `Reliability` of MySQL increases/decreases the `Availability` of the SugarCRM service also increases/decreases. In this exercise, only one relationship was created, but several relationships can be created between two service systems, for example, to express more complex cause-effect relations.

5 Related Work

The work on relationships has mainly been carried out in the fields of business management, supply chain management, and operation management. The main contributions (e.g. [13,14,15,9]) have generally discussed the objectives, motivation, and benefits of relationships for businesses. While business relationships look at relationships from a macro perspective, service relationships look at relationships from a micro perspective. According to Jensen and Petersen [28], in service-based economies there is a fundamental need to move from a macro strategic business orientation to a fine-grained activity-based service analysis. Furthermore, previous work does not proposes conceptual models nor formalisms to build computer-understandable descriptions of relationships as described in this paper.

e^{3}service [22] provides an ontology to model e-business models and services. The model targets to represent very simple relations between services from an internal perspective, e.g. core-enhancing, core-supporting, and substitute. From an external perspective, the value chains proposed do not capture explicitly service networks across agents and do not try to analyze quantitatively the effect of relationships.

In [29], the authors look at service networks from a Business Process Management (BPM) and Service Oriented Architecture (SOA) perspectives and present the Service Network Notation (SNN). SNN provides UML artifacts to model value chain relationships of economic value. These relationships take the form of what we can call 'weak' relationships since they only capture offerings and rewards which occur between services. The notation is to be used to describe how a new service can be composed from a network of existing services. The focus is on compositions, processes, and on establishing how new services can be created using BPM to describe the interactions of existing SOA-based services.

Allee [30] uses a graph-based notation to model value flows inside a network of agents such as the exchange of goods, services, revenue, knowledge, and intangible values. In the same lines, Weill and Vitale [31] have developed a formalism, called the e-business model schematic, to analyze businesses. The schematic is a graphical representation aiming at identifying a business model's important elements. This includes the firm relationships with its suppliers and allies, benefits each participant receives, and the major flows of product, information, and money. Both approaches only take into account value flows and do not consider other types of relationships that can be established between agents.

In all these works, relationships can benefit from a deeper study to increase their expressiveness rather than simply connecting flows, cross-organizational processes, or calculating the global added value of distributed activities. Roles, categorization, KPI dependencies, and cause-effect relations also need to be considered. Furthermore, existing modeling approaches fail to adhere to service-dominant logic [2] and focus too much inward the company instead of the service network they belong to.

6 Conclusions

To provide theories and methods to analyze service networks there is the essential prerequisite to model service systems and service relationships. In this paper we addressed the latter: the modeling of service relationships. Our approach considers that service systems are represented with existing description languages, such as Linked-USDL, and derives a rich, multi-level relationship model – named Open Semantic Service Relationship (OSSR) model – from an extensive literature review process. Service relationships are very different from the temporal and control-flow relations found in business process models. They need to relate service systems accounting for various perspectives such as roles, associations, dependencies, and comparisons. After designing the OSSR conceptual model, it was evaluated and implemented. The encoding was based on Linked Data principles to retain simplicity for computation, reuse existing vocabularies to maximize compatibility, and provide a simple - yet effective - means for publishing and interlinking distributed service descriptions for automated computer analysis.

References

1. Spohrer, J., Maglio, P.P.: Service Science: Toward a Smarter Planet, pp. 1–30. John Wiley & Sons, Inc. (2010)
2. Vargo, S.L., Lusch, R.F.: Evolving to a new marketing dominant logic for marketing. Journal of Marketing 68(1), 1–17 (2004)
3. Erl, T.: Service-Oriented Architecture: Concepts, Technology, and Design. Prentice Hall PTR, Upper Saddle River (2005)
4. Cardoso, J., Sheth, A.: Semantic e-workflow composition. J. Intell. Inf. Syst. 21, 191–225 (2003)
5. Cardoso, J., Pedrinaci, C., Leidig, T., Rupino, P., Leenheer, P.D.: Open semantic service networks. In: The International Symposium on Services Science (ISSS 2012), Leipzig, Germany, pp. 1–15 (2012)
6. Cardoso, J., Pedrinaci, C., De Leenheer, P.: Open Semantic Service Networks: Modeling and Analysis. In: Novoa, H., Snene, M. (eds.) IESS 2013. LNBIP, vol. 143, pp. 141–154. Springer, Heidelberg (2013)
7. Cardoso, J., Barros, A., May, N., Kylau, U.: Towards a unified service description language for the internet of services: Requirements and first developments. In: IEEE International Conference on Services Computing. IEEE Computer Society Press, Florida (2010)
8. Barros, A., Oberle, D.: Handbook of Service Description: USDL and Its Methods. Springer (2012)
9. Gadde, L.-E., Snehota, I.: Making the most of supplier relationships. Industrial Marketing Management 29(4), 305–316 (2000)
10. Bizer, C., Heath, T., Berners-Lee, T.: Linked data - the story so far. Int. J. Semantic Web Inf. Syst. 5(3), 1–22 (2009)
11. Brickley, D., Miller, L.: FOAF Vocabulary Specification (2005)
12. Bojars, U., Breslin, J., Peristeras, V., Tummarello, G., Decker, S.: Interlinking the social web with semantics. IEEE Intelligent Systems 23(3), 29–40 (2008)
13. Anderson, J.C., Narus, J.A.: Partnering as a focused market strategy. California Management Review 33(3), 95–113 (1991)
14. Ritter, T., Wilkinson, I.F., Johnston, W.J.: Managing in complex business networks. Industrial Marketing Management 33(3), 175–183 (2004)
15. Håkansson, H., Snehota, I.: Developing relationships in business networks. Routledge (1995)

16. Chen, C.-L., Lee, W.-C.: Multi-objective optimization of multi-echelon supply chain networks with uncertain product demands and prices. Computers & Cal Engineering 28(6-7), 1131–1144 (2004)
17. Hausdorff, F.: Set Theory. AMS Chelsea Publishing Series. American Mathematical Society (1957)
18. Booch, G.: Object-Oriented Analysis and Design with Applications, 3rd edn. Addison Wesley Longman Publishing Co., Inc., Redwood City (2004)
19. Forrester, J.: Industrial dynamics. MIT Press, Cambridge (1961)
20. Brandenburger, A., Nalebuff, B.: Co-opetition. A terrific book. Doubleday (1996)
21. Wernerfelt, B.: A Resource-Based View of the Firm. Strategic Management Journal 5(2), 171–180 (1984)
22. Gordijn, J., Yu, E., van der Raadt, B.: e-service design using i* and e3value modeling. IEEE Software 23, 26–33 (2006)
23. Cardoso, J., Miller, J.A.: Internet-based self-services: from analysis and design to deployment. In: The 2012 IEEE International Conference on Services Economics (SE 2012). IEEE Computer Society, Hawaii (2012)
24. Cai, J., Liu, X., Xiao, Z., Liu, J.: Improving supply chain performance management: A systematic approach to analyzing iterative KPI accomplishment. Decision Support Systems 46(2), 512–521 (2009)
25. Gómez-Pérez, A.: Evaluation of ontologies. International Journal of Intelligent Systems 16(3), 391–409 (2001)
26. Davies, J., Fensel, D., van Harmelen, F. (eds.): Towards the Semantic Web: Ontology-driven Knowledge Management. John Wiley & Sons, Inc., New York (2003)
27. Prud'hommeaux, E., Seaborne, A.: SPARQL query language for RDF (2005), `http://www.w3.org/TR/2005/WD-rdf-sparql-query-20050217/`
28. Ørberg Jensen, P.D., Petersen, B.: Global sourcing of services versus manufacturing activities: is it any different? The Service Industries Journal 32(4), 591–604 (2012)
29. Danylevych, O., Karastoyanova, D., Leymann, F.: Service networks modelling: An SOA & BPM standpoint. Journal of Universal Computer Science 16(13), 1668–1693 (2010)
30. Allee, V.: Reconfiguring the value network. Journal of Business Strategy 21(4), 1–6 (2000)
31. Weill, P., Vitale, M.R.: Place to space: migrating to ebusiness models. Harvard Business School Press (2001)