

# Aggregating Service Level Agreements in Services Bundling: A Semiring-Based Approach

Trung-Viet Nguyen<sup>1,2</sup>, Lam-Son Lê<sup>1(✉)</sup>, and Khuong Nguyen-An<sup>1</sup>

<sup>1</sup> Faculty of Computer Science and Engineering,  
HCMC University of Technology, Ho Chi Minh City, Vietnam

lam-son.le@alumni.epfl.ch, nakhuong@hcmut.edu.vn

<sup>2</sup> Faculty of Information Technology, Can Tho University of Technology,  
Can Tho City, Vietnam

ntviet@ctuet.edu.vn

**Abstract.** Business services arguably play a central role in service-based information systems as they fill in the gap between the technicality of Service-Oriented Architecture and the business aspects captured in Enterprise Architecture. Business services have distinctive features that are not typically observed in Web services, e.g. significant portions of the functionality of business services might be executed in a human-mediated fashion. As such, service level agreement (SLA) should be described as a mixture of human-mediated functionality (e.g., service penalty) and computer-interpretable measurement (e.g., reliability, payment). In this paper, we propose a formal framework for reasoning about the SLAs from the perspective of services bundling – the practice of innovatively organizing business services into a bulkier service offering that creates new values. Specifically, we (a) represent multi-level SLA of a business service in terms of service reliability, payment and penalty using the mathematical structure of semiring; (b) provide formality for aggregating SLAs of the constituent services that make up the service bundling; (c) make multi-level SLAs of a bundled service technically comparable. The main contribution of this work is a machinery for handling a large number of SLAs generated through services bundling, allowing to the service consumers to pick up the right service offering according to their preference.

**Keywords:** SLA · Services bundling · Semiring · Formal methods

## 1 Introduction

In the last few years, service-oriented computing has become an emerging research topic in response to the shift from product-oriented economy to service-oriented economy. On the one hand, we now live in a growing services-based economy in which every product today has virtually a service component to it [21]. In this context, services are increasingly provided in different ways in order to meet growing customer demands. Business domains involving large and complex

collection of loosely coupled services provided by autonomous enterprises are becoming increasingly prevalent [1, 24]. On the other hand, Information Technology (IT) has now been thoroughly integrated into our daily life [14] and gradually gives rise to the paradigm of ubiquitous computing. As such, business services are essentially IT-enabled making the border between business services<sup>1</sup> and IT-enabled services blurred. At the high-level operationalization of a business service, we see business activities happening between service stakeholders. We may or may not witness IT operations at this representational level. At lower levels, the operationalization of these services are eventually translated into IT operations as we have seen in the cases of banking services, recruitment services, library services, auctioning services, etc.

Services bundling is a practice of innovatively grouping related business services to come up with new service offerings that create new service values for customers. A typical example of service bundling is car rental, accommodation, travel insurance could be combined to offer a valued travel package to tourists. The customers may experience to-be-bundled business services in two different ways: by consuming them individually and by taking them via service bundling. As such, aggregating service level agreement (SLA) of business services when bundling them poses a number of challenging questions. First, the SLA of business services are to be perceived from at least two different angles, specifically the customer's point of view and through the provider of the service bundling (e.g. travel agency). Second, as the SLA of a service might have multiple levels (hence, the term multi-level SLA), bundling services could result in generating a large number of SLAs that are to be perceived by the customers. We are in need of a machinery that sorts them and helps the customers choose the right SLA according to their preference. While there exists considerable amount of work on reasoning about SLA of Web services, not much effort has been put in coping with the complexity of the SLAs from the business standpoint. In this paper, we propose a formal framework for reasoning about the SLAs for service bundling. Specifically, we (a) represent multi-level SLA of a business service in terms of service reliability, payment and penalty using the mathematical structure of semiring; (b) provide formality for aggregating SLAs of the constituent services that make up the service bundling; (c) make multi-level SLAs of a bundled service technically comparable. This work sheds light on how human-mediated concepts such as business contracts [12, 17] could be reasoned about together with computer-interpretable aspects such as costs and reliability in SLA.

The remaining of this paper is organized as follows. In Sect. 2, we give preliminaries for SLA, service penalty and the semiring. Section 3 expresses our research approach and describes a running example. Section 4 is the core of the paper – we formally define multi-level SLAs and reason about aggregating them

---

<sup>1</sup> By calling them business services, we mean services happening between people or business entities. They are enabled by IT in one way or another. For the sake of simplicity, we shall use the term “business service” or simply “service” to refer to these IT-enabled business services throughout this paper.

in services bundling. We survey related work in Sect. 5. Section 6 ends the paper by drawing some concluding remarks and outlining our future work.

## 2 Preliminaries

### 2.1 SLA Overview

A service level agreement (SLA) [22] is a contract between the service consumer and the service provider, formally defines the level of services. It gives details about the quality and scope of the service provided, which can also be referred to as a “service level contract”.

“A service level agreement, SLA, is a technical contract between two types of businesses, producers and consumers. A SLA captures the agreed upon terms between organizations with respect to quality of service (QoS) and other related concerns. In simple cases, one consumer forms a SLA with a producer.” [4]

In service-oriented computing, a SLA is a collection of service level requirements which have been negotiated and mutually agreed upon by the information providers and the information consumers [5]. Usually, providers define some service levels as a fixed combination of their specific capabilities on a set of quality dimensions, and users must choose one these levels. An SLA could be split into different levels, each solving problems for different groups of customers who have the same services, in the same SLA [16], hence the term multilevel SLA as follows.

- Corporate-level SLA: covers all the generic service level management (often abbreviated as SLM) issues appropriate to every customer throughout the organization. These issues are likely to be less volatile and so updates on this kind of SLA are less frequently required.
- Customer-level SLA: addresses all service level issues relevant to a particular customer group, regardless of the services being used.
- Service-level SLA: is relevant to a specific service, in relation to a specific customer group.

Quality of service (QoS) is defined as the “collective effect of service performance, which determines the degree of satisfaction of a user of the service” [15]. However, compatibility with the system and the definition can be used in contracts between service providers and customers, here we define QoS as follows: “QoS is the degree to which service providers can offer customers under contracts have been committed”.

One of the most significant QoS concerns of Web services is reliability. For services (cloud) the reliability can be measured by the time the system is ready to serve as intended. For the mission-critical task, the reliability is more important than other aspects [18].

## 2.2 Semiring

Semiring is a mathematical structure that features a domain equipped with two operations satisfying certain properties, as described in Definition 1.

**Definition 1.** A **semiring** is a tuple  $\langle A, \oplus, \otimes, \bar{0}, \bar{1} \rangle$  such that

- $A$  is a set and  $\bar{0}, \bar{1} \in A$
- $\oplus$ , called the additive operation, is a commutative, associative operation having  $\bar{0}$  as its neutral element (i.e.  $a \oplus \bar{0} = a = \bar{0} \oplus a$ )
- $\otimes$ , called the multiplicative operation, is an associative operation such that  $\bar{1}$  is its unit element and  $\bar{0}$  is its absorbing element (i.e.  $a \otimes \bar{0} = \bar{0} = \bar{0} \otimes a$ )
- $\otimes$  distributes over  $\oplus$  (i.e.  $\forall a, b, c \in A \rightarrow a \otimes (b \oplus c) = a \otimes b \oplus a \otimes c$ )

An idempotent semiring is a semiring whose additive operation is idempotent (i.e.  $a \oplus a = a$ ). This idempotence property allows us to endow a semiring with a canonical order defined as  $a \preceq b$  iff  $a \oplus b = b$  [7]. There exists another form of idempotent semiring called c-semiring whereby the  $\oplus$  operator is defined over subsets of a domain and as such it has flattening property [3]. The endowed order of a c-semiring is actually a partial order that would be used for choosing “best” solutions in a constraint satisfaction problem.

**Definition 2** ([2]). A semiring will be called **c-semiring**, where “c” stands for “constraint”, meaning that they are the natural structures to be used when handling constraints. A **c-semiring** is a tuple  $\langle A, \oplus, \otimes, \bar{0}, \bar{1} \rangle$  such that

- $A$  is a set and  $\bar{0}, \bar{1} \in A$
- $\oplus$  is defined over (possibly infinite) sets of elements of  $A$  as follows:
  - For all  $a \in A, \sum(\{a\}) = a$ ;
  - $\sum(\emptyset) = \bar{0}$  and  $\sum(A) = \bar{1}$ ;
  - $\sum(\bigcup A_i, i \in I) = \sum(\{\sum(A_i), i \in I\})$  for all sets of indices  $I$  (flattening property).
- $\otimes$  is a binary, associative and commutative operation such that  $\bar{1}$  is its unit element and  $\bar{0}$  is its absorbing element
- $\otimes$  distributes over  $\oplus$  (i.e. for any  $a \in A$  and  $B \subseteq A, a \otimes \sum(B) = \sum(\{a \otimes b, b \in B\})$ )

To make an idempotent semiring applicable for the representation of QoS, we endow it with a canonical order defined as  $a \preceq b$  iff  $a \oplus b = b$ . A semiring is used to express the domain and the order between values that feature a QoS. To represent QoS factors, we may use the notion of *bounded lattice*. Each bounded lattice has a greatest element (denoted as  $\top$ ) and a least element (denoted as  $\perp$ ) and features two operations: meet (denoted as  $\wedge$ ) and join (denoted as  $\vee$ ) [7].

### 2.3 Business Contract Modeling Based on Deontic Logic

Business contracts specify obligations, permissions and prohibitions as mutual agreements between business parties [19], as well as actions to be taken when a contract is violated. Governatori and Milosevic [12] have proposed such a contract modeling language which includes a non-boolean connective,  $\odot$ , to represent contrary-to-duty obligations (i.e., what should be done if the terms of a contract are violated). Deontic operators capture the contractual modality (i.e. obligations, permissions and prohibitions) [10]. Governatori et al. represent a contractual rule as  $r : A_1, A_2 \dots A_n \vdash C$  where each  $A_i$  is an antecedent of the rule and  $C$  is the consequent. Each  $A_i$  and  $C$  may contain deontic operators but connectives can only appear in  $C$ .

As an example,  $r : \neg p, q \vdash O_{seller}\alpha \odot O_{seller}\beta$  is a contractual rule (identified by  $r$ ) stating that if antecedents  $\neg p$  and  $q$  hold, then a seller is obliged to make sure that  $\alpha$  is brought about. Failure to do so results in a violation, for which a reparation can be made by bringing about  $\beta$  (the connective  $\odot$  can therefore be informally read as “failing which”).

**Definition 3 (Contractual rules of SLA [12]).** *Contractual rules  $r$  and  $r'$  can be merged into rule  $r''$  as follows where  $X$  denotes either an obligation or a permission.*

$$\frac{r : \Gamma \vdash O_s A \odot (\odot_{i=1}^n O_s B_i) \odot O_s C \quad r' : \Delta, \neg B_1, \neg B_2, \dots, \neg B_n \vdash X_s D}{r'' : \Gamma, \Delta \vdash O_s A \odot (\odot_{i=1}^n O_s B_i) \odot X_s D} \quad (1)$$

The  $\otimes$  operator is associative but not commutative. This property matters when reasoning about the subsumption and merging of contractual rules. Definition 3 defines how contract rules might be merged. Governatori et al. also devise a machinery for determining if one contractual rule subsumes another as presented in Definition 4.

**Definition 4 ([12]).** *Let's consider two rules  $r_1 : \Gamma \vdash A \odot B \odot C$  and  $r_2 : \Delta \vdash D$  where  $A = \odot_{i=1}^m A_i$ ,  $B = \odot_{i=1}^n B_i$  and  $C = \odot_{i=1}^p C_i$ . Then  $r_1$  subsumes  $r_2$  (i.e.  $r_2$  can safely be discarded if we have  $r_1$ ) iff*

1.  $\Gamma = \Delta$  and  $D = A$ ; or
2.  $\Gamma \cup \{\neg A_1, \dots, \neg A_m\} = \Delta$  and  $D = B$ ; or
3.  $\Gamma \cup \{\neg B_1, \dots, \neg B_n\} = \Delta$  and  $D = A \odot \odot_{i=0}^{k \leq p} C_i$

## 3 Research Motivation

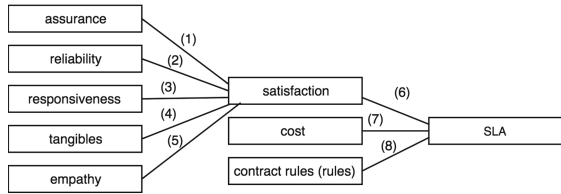
In this section, we first describe a running example<sup>2</sup> (Subsect. 3.1) and then come up with our research statements (Subsect. 3.2).

<sup>2</sup> The running example will be used for explaining our research questions and exemplifying our formality.

### 3.1 Running Example

Let’s consider the business of a tour company that is performed on the theme of Future Internet [20] – a recently-emerging trend that aims to offer integrated access to people, media, services, etc. using an underlying platform. It seeks to enable new styles of social and economic interactions on an unprecedented scale, offering both flexibility and quality. Besides being the constituting building block of the so-called Internet of Services, the Future Internet, through the metaphor of the Internet of Things, will provide location-independent, interoperable, scalable, secure and efficient access to a coordinated set of services. However, to turn the promise of this principle into realized benefits, services must be accompanied by exact definitions as to the conditions of their usage. These conditions can be specified by Service Level Agreement (SLA).

Businesses using SERVQUAL to measure and manage service quality deploy a questionnaire that measures both the customer expectations of service quality in terms of these five dimensions (i.e., reliability, assurance, tangibles, empathy and responsiveness) [9], and their



**Fig. 1.** Breakdown of the SERVQUAL-based SLA

perceptions of the service they receive. When customer expectations are greater than their perceptions of received delivery, service quality is deemed low. According to SERVQUAL, the quality of a business service includes assurance, reliability, responsiveness, tangibles and empathy. Table 1 describes them in detail. Each dimension descriptor has an informal definition in the tourist industry. In our framework, these items together make up the first element to arrange and evaluate the SLA, i.e., satisfaction have a reliability of service and weight of evaluating the importance of a service in the SLA. The other two SLA elements are cost and penalty rules. Figure 1 illustrate our standpoint. When customers

**Table 1.** Components of SLA in SERVQUAL [9]

Num	Dimensions	Definition
(1)	Assurance	Knowledge and courtesy of employees and their ability to convey trust and confidence
(2)	Reliability	Ability to perform the promised service dependably and accurately
(3)	Responsiveness	Willingness to help customers and provide prompt service
(4)	Tangibles	Physical facilities, equipment, and appearance of personal
(5)	Empathy	Caring, individualized attention the firm provides its customers
(6)	Satisfactions	Satisfaction have a reliability of service and weight of evaluating the importance of a service in the SLA
(7)	Cost	Cost is a payment or price of service
(8)	Rules	The penalty clauses in contract are also represented as contractual rules

book a trip through a travel tour company’s website, they get travel information such as destinations, restaurants, hotels. For example:

“This trip departs within 60 days and requires full payment to confirm a place. Upon receipt of payment we will confirm your booking within 2–4 days. We recommend waiting for our confirmation before purchasing air tickets or other non-refundable travel arrangements.”

In Fig. 2, services provided by car rental companies, airlines and hotels will be bundled and offered to customers via a tour company. Customers will interact directly with the system via a booking agency website. This system will be connected to many different types each of which could be provided by at different SLAs by multiple companies. The SLAs will be combined and arranged to send to customers. Customers will choose the SLA have been combined to send to the travel agency. This agency will manage the customer contracts with companies providing services.

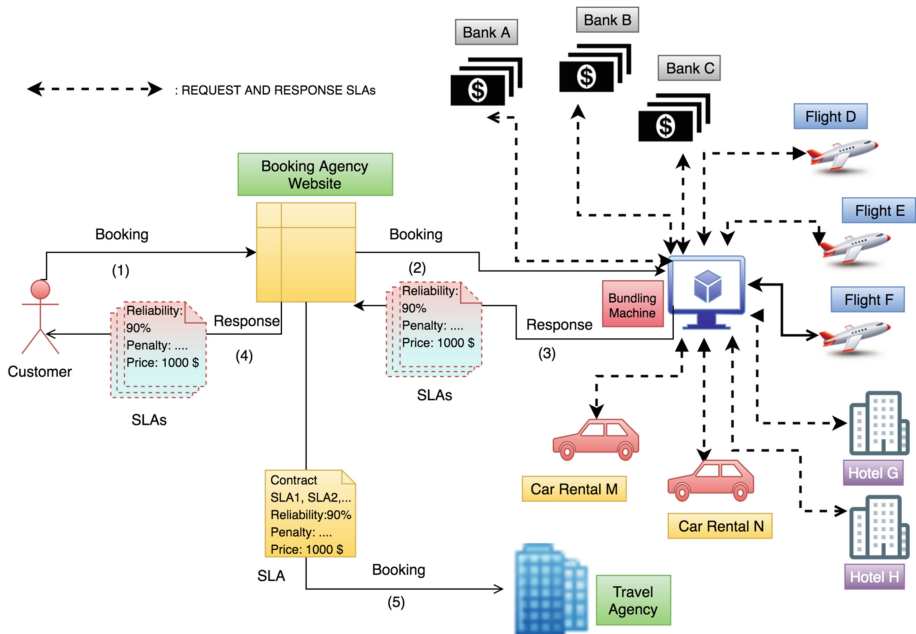


Fig. 2. Bundling car rental agents, airlines and hotels

The tour was made up of a combination SLA of companies providing services as: Bank A, Bank B, Flight D, Flight E, Flight F, Hotel G, Hotel H, Car Rental M, Car Rental N,... Tables 2 and 3 are used to illustrate the service level of the Hotel G, Car rental M in Fig. 2. In fact, each company will release many different SLA and need a screening inspection and combined. Table 2, the Hotel G provided more level services (level SLA) a customer might want as: Luxury ( $SLA_{H1}$ ), Bussiness ( $SLA_{H2}$ ), Personal ( $SLA_{H3}$ ), helping customers to select

the one that's best for them. Each level SLA consist of several sub-agreements, the Luxury ( $SLA_{H1}$ ) level Hotel G include 3 sub-agreement:  $sa_{h1}$ ,  $sa_{h2}$ ,  $sa_{h3}$ . The sub-agreements is a service, has the weight of evaluating the importance of a services in the SLA. The total weight of a level SLA must equal 1 (100%), the weight of a sub-agreement must be one minus the weight of another. The total weight of a level SLA =  $sa_{h1} + sa_{h2} + sa_{h3} = 1$ . Satisfaction of an agreement for a sub-agreements is an operation multiplication between the reliability and weight. In  $SLA_{H1}$  level, the satisfaction (sat) of  $sa_{h1}$  = weight of  $sa_{h1}$  \* reliability (rel) of  $sa_{h1}$  =  $0.2 * 0.99 = 0.198$ . The total satisfaction of several sub-agreements in the level SLA is satisfaction of the level SLA. The satisfaction of  $SLA_{H1} = sa_{h1} + sa_{h2} + sa_{h3} = 0.198 + 0.49 + 0.285 = 0.973$ . The Car rental M provided more level SLA for customer as: Full size( $SLA_{R1}$ ), SUV( $SLA_{R2}$ ), Economy( $SLA_{R3}$ ), the Full size( $SLA_{R1}$ ) include 3 sub-agreement:  $sa_{r1}$ ,  $sa_{r2}$ ,  $sa_{r3}$ . A SLA of a tour is a combination of multiple SLA of 2 Tables 2 and 3 as  $\{SLA_{H1}, SLA_{R1}\}, \{SLA_{H1}, SLA_{R1}, SLA_{H1}\}, \{SLA_{H1}, SLA_{R1}, SLA_{R2}\}, \{SLA_{H1}, SLA_{R2}\}$ :

**Table 2.** Multi-level SLA for hotel or restaurant

With: Weight(w), Commitment(Com), Reliability (Rel), Satisfactions (Sat)										
Sub-Agreement	w	Luxury( $SLA_{H1}$ )			Bussiness( $SLA_{H2}$ )			Personal( $SLA_{H3}$ )		
		Com	Rel	Sat	Com	Rel	Sat	Com	Rel	Sat
$sa_{h1}$ -Phone	0.2	24/7	0.99	0.198	24/7	0.99	0.198	24/7	0.99	0.198
$sa_{h2}$ -Free Breakfast	0.5	Type1	0.98	0.490	Type2	0.90	0.450	No	0	0
$sa_{h3}$ -Pick up: At airport	0.3	Yes	0.95	0.285	Yes	0.93	0.279	No	0	0
Price (Cost)		\$1500		0.973	\$1000		0.927	\$500		0.198

**Table 3.** Multi-level SLA for rental car

With: Weight(w), Commitment(Com), Reliability (Rel), Satisfactions (Sat)										
Sub-Agreement	w	Full size( $SLA_{R1}$ )			SUV( $SLA_{R2}$ )			Economy( $SLA_{R3}$ )		
		Com	Rel	Sat	Com	Rel	Sat	Com	Rel	Sat
$sa_{r1}$ -Support of a rental car	0.4	Yes	0.99	0.396	Yes	0.98	0.392	Yes	0.87	0.348
$sa_{r2}$ -Time of delivery car	0.3	Yes	0.98	0.294	Yes	0.93	0.279	Yes	0.92	0.278
$sa_{r3}$ -Pick up: At airport	0.3	Yes	0.95	0.285	Yes	0.95	0.285	No	0	0
Price (Cost)		\$350		0.975	\$300		0.956	\$250		0.624

### 3.2 Research Statements

Let's formulate our research statements based on the running example described above. First, business services usually come with multi-level SLAs. We need a formal representation that captures SLA items described in Table 1. We aim to shorten them to have a more succinct representation for the sake of formal



reasoning. In our running example, the representational items in Table 1 will be combined in a way to describe the SLA using no more than 3 elements.

Second, bundling business services involves multiple service providers. As such, we need to represent the multi-level SLAs from at least two perspectives, namely the customers's point of view and the viewpoint of the providers who provide the services to be bundled. In our running example, we take into account the travellers' point of view and the viewpoint of car rental companies, airlines and hotels when it comes to services bundling and SLAs.

Third, bundling business services will lead to the aggregation of their multi-level SLAs. As same service provider may offer multiple business services under different SLAs, bundling services generate a large number of SLAs many of which are multi-level. We need to come up with an approach to systematically handling the generated SLAs and helping customers choose the one that best suits their needs. This is observable in our running example where multiple car rental companies, hotels, etc. make offerings to travellers to pick up a travel package.

## 4 A Framework for Reasoning About SLA in Services Bundling

### 4.1 Formal Representation

Our fundamental idea is the definitions to create a SLA document from services of providers. It can be used to create a new SLA from the provider's SLA when they provide services. This means that the relationship of providers and new SLA. Therefore, we can define an new SLA document as follows.

**Definition 5.** *Each multi-level SLA consist of several sub-agreements. The satisfaction of the SLA is intended to reflect the customer satisfaction, which is the sum of the sub-Agreements's satisfaction [26]. Let  $S = \sum_{i=1}^n s_i$  where  $S$  is the **Satisfaction** of the SLA and  $s$  is satisfaction of an agreement of SLA, clearly  $S \in [0,1]$ .*

The SLA can include many sub-Agreements with different policies defined in each of them. A sub-Agreements (subcontract) is understood as a part of SLA. Each sub-Agreements is a service which has the reliability and weight of evaluating the importance of a services in the SLA. Satisfaction of an agreement for a service is an operation multiplication between the reliability and weight  $w$  by the formula:  $s = r * w$  Where  $s$  is satisfaction of an agreement,  $r$  is reliability. The reliability is defined in Difinition 6.

**Definition 6. Reliability** *is defined as follow:  $r = 1 - fr$  where  $r$  is reliability of a service,  $fr$  is failure rate.*

Reliability of a service, can be represented by the failure rate as shown below and the failure rate of a service can be measured by the ratio of the number of times the service damaged to the total service requests.

*Example 1.* A customer rented a car which was provided by the company Rental Car. Rental Car contacted the owner determine the time delivery for customers. But all the time the owner had a problem and couldn't deliver for customers. It was recorded as a case of cancelled service.

**Definition 7.** Let  $C_0 = \{c_1, c_2, \dots, c_n\}$  be the initial set of values of Cost. Cost is a payment or price of a service which is in the closure of cost values under addition. By closure of  $C_0$ , we mean the smallest set containing all summation of elements in  $C_0$ :  $C_0^+ = \{\sum_{k=1}^{\infty} (C_{i_1} + \dots + C_{i_k}) | C_{i_k} \in C_0\}$ . Based on those conditions, Cost is defined as  $C = C_0^+ \cap [0, Cost_{max}]$ , where  $Cost_{max}$  is the highest cost that the customer can pay.

**Definition 8.** The complete SLA is a combination of multilevel SLA from service providers that offer several types of services. The complete SLA is defined **SLA (Combining SLAs)** which has included the several types of services. Let  $K_0 = \{k_1, k_2, \dots, k_n\}$  be the initial set of types of services in CSLAs. With type of service  $k_i$ , We have more service providers to choose SLAs. The service providers will deliver services to make up the initial set  $J_0^{k_i} = \{j_1^{k_i}, j_2^{k_i}, \dots, j_{m_{k_i}}^{k_i}\}$ ,  $m_{k_i}$  is defined  $l$ . A set types of services and services of provider is defined as follows:  $\{SLA(k_i, j_l^{k_i})\}$  where  $k_i \in K_0$  and  $j_l^{k_i} \in J_0^{k_i}$ . Each element of  $SLA(k_i, j_l^{k_i})$  is the triple  $\langle S, C, R \rangle$ , where  $S$ : Satisfaction;  $C$ : Cost;  $R$ : Rule. A **CSLA** is combined from  $SLA(k_i, j_l^{k_i})$  as follows:

$$\begin{aligned}
 CSLA &= \bigodot_{i=1}^n SLA(k_i, j_l^{k_i}) \\
 &= \langle \min S_{SLA(k_i, j_l^{k_i})}, \sum_{k=1}^n (C_{SLA(k_i, j_l^{k_i})}), \text{merge}R_{SLA(k_i, j_l^{k_i})} \rangle.
 \end{aligned}
 \tag{2}$$

*Example 2.* The running example in Sect. 3.1, a tour is combined services' SLA in Tables 2 and 3. Let  $K_0 = \{k_1, k_2\}$  where  $k_1$  is the type of service for hotel and  $k_2$  is type of services for the car rental. We call  $j_l^{k_1}$  is SLA in the type of service for hotel ( $k_1$ ). The  $SLA_{H1}$  will be converted to  $SLA(k_1, j_1^{k_1})$ . The conversion of SLAs is presented in Table 4.

**Table 4.** Combining multi-level SLAs into one CSLA

Services	Provider	Multi-level SLAs are provided for customers			
		Level	SLA	Satisfaction	Cost
Hotel	Hotel G	Luxury( $SLA_{H1}$ )	$SLA(k_1, j_1^{k_1})$	0.973	\$1500
		Bussiness( $SLA_{H2}$ )	$SLA(k_1, j_2^{k_1})$	0.927	\$1000
		Personal( $SLA_{H3}$ )	$SLA(k_1, j_3^{k_1})$	0.198	\$500
Car rental	Car Rental M	Full size( $SLA_{R1}$ )	$SLA(k_2, j_1^{k_2})$	0.975	\$350
		SUV( $SLA_{R2}$ )	$SLA(k_2, j_2^{k_2})$	0.956	\$300
		Economy( $SLA_{R3}$ )	$SLA(k_2, j_3^{k_2})$	0.624	\$250

In Definition 8, the satisfactions of CSLA is the smallest satisfaction in the SLA set. The combination of multilevel SLA from service providers created a collection of CSLA objects. The table 5 is a range of CSLAs that shows how combining two or more SLA, which displays the satisfaction index of CSLA.

**Table 5.** Listing CSLA in details of SLAs

CSLA	Combination SLA	$\langle S, C, R \rangle$
$CSLA_1$	$SLA(k_1, j_1^{k_1}), SLA(k_2, j_1^{k_2})$	$\langle 0.975; c; r \rangle$
$CSLA_2$	$SLA(k_1, j_1^{k_1}), SLA(k_2, j_2^{k_2})$	$\langle 0.956; c; r \rangle$
$CSLA_3$	$SLA(k_1, j_1^{k_1}), SLA(k_2, j_3^{k_2})$	$\langle 0.624; c; r \rangle$
$CSLA_4$	$SLA(k_1, j_2^{k_1}), SLA(k_2, j_1^{k_2})$	$\langle 0.931; c; r \rangle$
$CSLA_5$	$SLA(k_1, j_2^{k_1}), SLA(k_2, j_2^{k_2})$	$\langle 0.931; c; r \rangle$
$CSLA_6$	$SLA(k_1, j_1^{k_1}), SLA(k_2, j_1^{k_2}), SLA(k_2, j_2^{k_2})$	$\langle 0.956; c; r \rangle$
$CSLA_7$	$SLA(k_1, j_1^{k_1}), SLA(k_2, j_1^{k_2}), SLA(k_2, j_3^{k_2})$	$\langle 0.624; c; r \rangle$
$CSLA_8$	$SLA(k_1, j_1^{k_1}), SLA(k_2, j_2^{k_2}), SLA(k_2, j_3^{k_2})$	$\langle 0.624; c; r \rangle$

## 4.2 Aggregating SLAs

Mathematically combining Satisfaction, Cost and Rule results in the set of triple  $A = \{\langle S, C, R \rangle\}$ , where  $S$  is Satisfaction,  $C$  is Cost,  $R$  is Rule.

- The Satisfaction has become an aspect of quality of service. It has been proven by a number of quality research services related to customer satisfaction:  $Satisfaction \in [0, 1]$ ;
- Cost is a payment or price of service which is presented in Definition 8
- The penalty clauses in contract are also represented as contractual rules. Rules are formulas in first order logic.

We consider the following ordering on set  $A$ . Let  $a = \langle s_1, c_1, r_1 \rangle$  and  $b = \langle s_2, c_2, r_2 \rangle$ . We can say that  $a \leq b$  iff  $(s_1 \leq s_2)$  or  $(s_1 = s_2) \wedge (c_1 \leq c_2)$  or  $(s_1 = s_2) \wedge (c_1 = c_2) \wedge (r_2 \vdash r_1)$ . In the case that  $(s_1 = s_2) \wedge (c_1 = c_2) \wedge (r_2 \not\vdash r_1) \wedge (r_1 \not\vdash r_2)$ , we define  $a = b$ .

Clearly, the relation “ $\leq$ ” defines a total ordering over the set  $A$ . We define the  $\oplus$  operation as the max operation with respect to this order.

The  $\otimes$  operator is the multiplication acting on each component an element in the set  $A$  differently. The  $\otimes$  operator’s action on  $S$  is the min operation. The  $\otimes$  operator’s action on  $C$  is ordinary addition. The  $\otimes$  operator’s action on  $R$  is the merging of two different rules into one rule. More precisely, let  $a = \langle s_1, c_1, r_1 \rangle$  and  $b = \langle s_2, c_2, r_2 \rangle$ , then  $a \otimes b$  is defined as follow

$$a \otimes b := \langle \min \{s_1, s_2\}, c_1 + c_2, \text{merge}(r_1, r_2) \rangle.$$

**Proposition 1.** Let  $\bar{0} = \langle \top, 0, 0 \rangle$ ,  $\bar{1} = \langle \perp, 0, 1 \rangle$ , with  $\top$  and  $\perp$  are tautology and the empty set, respectively. Then the tuple  $\langle A, \oplus, \otimes, \bar{0}, \bar{1} \rangle$  is a semiring.

*Proof.* Clearly,  $\bar{0}$  is the neutral element of  $\oplus$  and  $\bar{1}$  is the unit element of  $\otimes$ . It suffices to show that the  $\otimes$  operator is distributive over  $\oplus$ . Let  $a = \langle s_1, c_1, r_1 \rangle$ ,  $b = \langle s_2, c_2, r_2 \rangle$ , and  $c = \langle s_3, c_3, r_3 \rangle \in A$ . We show that  $a \otimes (b \oplus c) = (a \otimes b) \oplus (a \otimes c)$ .

Indeed, without loss of generality, we may assume that  $b \oplus c = c$  (i.e.,  $b \leq c$ ) and  $b \neq c$ . Then we have  $r_3 \vdash r_2$ . Therefore,  $\text{merge}(r_1, r_3) \vdash \text{merge}(r_1, r_2)$ . Hence

$$\begin{aligned} \text{R.H.S.} &= \max(\langle \text{merge}(r_1, r_2), c_1 + c_2, \min\{s_1, s_2\} \rangle, \\ &\quad \langle \text{merge}(r_1, r_3), c_1 + c_3, \min\{s_1, s_3\} \rangle) \\ &= \langle \text{merge}(r_1, r_3), c_1 + c_3, \min\{s_1, s_3\} \rangle \\ &= a \otimes c \\ &= \text{L.H.S.} \end{aligned}$$

### 4.3 Sorting

In Example 2, more conditions can be added: the ability to pay of customer, types of services needed for the trip. The ability to pay of customer is  $Cost_{max}$  and it can hit \$2000. A customer can pick up a tour that comes with an accommodation service and one or two car rental services.

We define a total order for sorting the SLAs in services bundling. This order can intuitively be explained as follows. First, the system looks at CSLAs' satisfaction as a criteria to establish the order. If the system encounters two CSLAs having same satisfaction, the system will look into their cost. The one having a lower cost is preferred to the other. In case the two CSLAs being compared have exactly the same satisfaction and cost, the system will check their penalty rules. Note that penalty rules are sorted by applying the entailment. Formally, we can define this total order as:  $CSLA_i \geq CSLA_j$  iff  $(S_{CSLA_i} > S_{CSLA_j})$  or  $(S_{CSLA_i} = S_{CSLA_j}) \wedge (C_{CSLA_i} \leq C_{CSLA_j})$  or  $(S_{CSLA_i} = S_{CSLA_j}) \wedge (C_{CSLA_i} = C_{CSLA_j}) \wedge (R_{CSLA_i} \vdash R_{CSLA_j})$  where  $S$ : satisfaction,  $C$ : cost,  $R$ : rule.

Table 6 illustrates how the first step in this sorting procedure changes the order of the CSLAs listed in Table 5. Note that  $CSLA_6$  and  $CSLA_3$  have been moved and marked with \* to their new location.

**Table 6.** A CSLA List are ordered by satisfaction

CSLA	Combination SLA	$\langle S, C, R \rangle$
$CSLA_1$	$SLA(k_1, j_1^{k_1}), SLA(k_2, j_1^{k_2})$	$\langle 0.975; c; r \rangle$
$CSLA_2$	$SLA(k_1, j_1^{k_1}), SLA(k_2, j_2^{k_2})$	$\langle 0.956; c; r \rangle$
* $CSLA_6$	$SLA(k_1, j_1^{k_1}), SLA(k_2, j_1^{k_2}), SLA(k_2, j_2^{k_2})$	$\langle 0.956; c; r \rangle$
$CSLA_4$	$SLA(k_1, j_2^{k_1}), SLA(k_2, j_1^{k_2})$	$\langle 0.931; c; r \rangle$
$CSLA_5$	$SLA(k_1, j_2^{k_1}), SLA(k_2, j_2^{k_2})$	$\langle 0.931; c; r \rangle$
* $CSLA_3$	$SLA(k_1, j_1^{k_1}), SLA(k_2, j_3^{k_2})$	$\langle 0.624; c; r \rangle$
$CSLA_7$	$SLA(k_1, j_1^{k_1}), SLA(k_2, j_1^{k_2}), SLA(k_2, j_3^{k_2})$	$\langle 0.624; c; r \rangle$
$CSLA_8$	$SLA(k_1, j_1^{k_1}), SLA(k_2, j_2^{k_2}), SLA(k_2, j_3^{k_2})$	$\langle 0.624; c; r \rangle$

After finding the CSLA with equal satisfaction, the system continue to add the cost of CSLA to Table 6. The CSLA's cost are calculated by the formula  $C = C_0^+ \cap [0, Cost_{max}]$  in the Definition 7. By comparing the maximum value of the two closures, we will choose the better CSLA as follows:  $C_{CSLA_1} > C_{CSLA_2}$  iff  $\max\{C_{CSLA_1}^+ \cap [0, Cost_{max}]\} < \max\{C_{CSLA_2}^+ \cap [0, Cost_{max}]\}$ . In Table 6, we have:

$$\begin{aligned} CSLA_5 > CSLA_4 &\Leftrightarrow \max\{C_{CSLA_5}^+ \cap [0, Cost_{max}]\} < \max\{C_{CSLA_4}^+ \cap [0, Cost_{max}]\} \\ &\Leftrightarrow \max\{C_{CSLA_5}^+ \cap [0, 2000]\} < \max\{C_{CSLA_4}^+ \cap [0, 2000]\} \\ &\Leftrightarrow 1900 < 2000. \end{aligned}$$

Table 7 illustrates how the second step in our sorting procedure makes a new order between the CSLAs. If CSLAs changed positions they will be marked \*\*.

**Table 7.** A CSLA List are ordered by cost

CSLA	Combination SLA	$\langle S, C, R \rangle$
$CSLA_1$	$SLA(k_1, j_1^{k_1}), SLA(k_2, j_2^{k_2})$	$\langle 0.975; 1500, 350; r \rangle$
$CSLA_2$	$SLA(k_1, j_1^{k_1}), SLA(k_2, j_2^{k_2})$	$\langle 0.956; 1500, 300; r \rangle$
* $CSLA_6$	$SLA(k_1, j_1^{k_1}), SLA(k_2, j_1^{k_2}), SLA(k_2, j_2^{k_2})$	$\langle 0.956; 1500, 350, 300; r \rangle$
** $CSLA_5$	$SLA(k_1, j_2^{k_1}), SLA(k_2, j_2^{k_2})$	$\langle 0.931; 1000, 300; r \rangle$
$CSLA_4$	$SLA(k_1, j_2^{k_1}), SLA(k_2, j_1^{k_2})$	$\langle 0.931; 1000, 350; r \rangle$
** $CSLA_8$	$SLA(k_1, j_1^{k_1}), SLA(k_2, j_2^{k_2}), SLA(k_2, j_3^{k_2})$	$\langle 0.624; 1500, 300, 250; r \rangle$
* $CSLA_3$	$SLA(k_1, j_1^{k_1}), SLA(k_2, j_3^{k_2})$	$\langle 0.624; 1000, 250; r \rangle$
$CSLA_7$	$SLA(k_1, j_1^{k_1}), SLA(k_2, j_1^{k_2}), SLA(k_2, j_3^{k_2})$	$\langle 0.624; 1000, 350, 250; r \rangle$

## 5 Related Work

In general, SLA management has been studied in the past few years and it has been mainly concentrated on the definition of languages and the specification of standards for SLA [6, 23]. However, these standards are still evolving as they present some limitations. The existing frameworks for SLA management only define the format and types of messages that can be exchanged during the negotiation between service providers and service consumers. Some projects particularize the management of SLAs to specific domains, such as military, database management, or information systems [8, 13, 25]. Garvin proposed a quality management grid featuring a total of eight dimensions including performance and reliability [11].

In our paper, we provide insights of how the SLAs of business services could be aggregated in services bundling. Our approach to combining SLAs is focused on cost, reliability and penalty. Our goal is to provide a mechanism for sorting the SLAs generated.

## 6 Conclusions

The representation of business services requires that we view human activity and human-mediated functionality through the lens of computing and systems engineering. Services bundling is a practice that creates new service values by purposely combining business services. The resulting combination, called a new service offering, poses a few challenging questions of how to aggregate the SLAs of the bundled services. We give insights into the modeling of SLA for high-level business services taking into account their human-mediating nature. Concretely, the SLA should be multi-level and incorporate technical attributes such as reliability and contract-like statements such as payment and penalty rules. We use Deontic logic for formally reasoning about penalty rules. Altogether, we leverage the mathematical structure of semiring to represent the SLA as a whole, which helps explain the intuitive meaning of aggregating the SLAs of business services when bundling them.

**Future Investigations.** Our future work includes: (i) devising algorithms for generating the SLAs when bundling services; (ii) designing a recommendation mechanism that would suggest the service consumers pick up an aggregated SLA according to their service preference.

## References

1. Bishop, K., Bolan, G., Bowen, D., Cromack, C., Evans, S., Fisk, R.P., Ganz, W., Gregory, M., Johnston, R., Lemmink, J., et al.: Succeeding through service innovation: a service perspective for education, research, business and government (2008)
2. Bistarelli, S.: Semirings for soft constraint solving and programming. In: Bistarelli, S. (ed.) *Semirings for Soft Constraint Solving and Programming*. LNCS, vol. 2962, pp. 1–20. Springer, Heidelberg (2004)
3. Bistarelli, S., Montanari, U., Rossi, F.: Semiring-based constraint satisfaction and optimization. *J. ACM (JACM)* **44**(2), 201–236 (1997)
4. Blake, M.B., Cummings, D.J., Bansal, A., Bansal, S.K.: Workflow composition of service level agreements for web services. *Decis. Support Syst.* **53**(1), 234–244 (2012)
5. Cappiello, C., Comuzzi, M.: Efficient allocation of quality improvement efforts to support the definition of data service offerings. In: *Proceedings of the 12th International Conference on Information Quality (ICIQ 2007)*, pp. 209–220 (2007)
6. Czajkowski, K., Foster, I., Kesselman, C., Sander, V., Tuecke, S.: SNAP: a protocol for negotiating service level agreements and coordinating resource management in distributed systems. In: Feitelson, D.G., Rudolph, L., Schwiegelshohn, U. (eds.) *JSSPP 2002*. LNCS, vol. 2537, pp. 153–183. Springer, Heidelberg (2002)
7. Davey, B.A., Priestley, H.A.: *Introduction to Lattices and Order*. Cambridge University Press, Cambridge (2002)
8. Falkowski, T., Voß, S.: Application service providing as part of intelligent decision support for supply chain management. In: *Proceedings of the 36th Annual Hawaii International Conference on System Sciences*. IEEE Computer Society (2003)
9. Fick, G.R., Ritchie, J.B.: Measuring service quality in the travel and tourism industry. *J. Travel Res.* **30**(2), 2–9 (1991)

10. Gabbay, D.M., Woods, J.: Logic and the Modalities in the Twentieth Century. Handbook of the History of Logic, vol. 7 (2006)
11. Garvin, D.A.: Managing Quality: The Strategic and Competitive Edge. Simon & Schuster, New York (1988)
12. Governatori, G., Milosevic, Z.: A formal analysis of a business contract language. *Int. J. Coop. Inf. Syst.* **15**(04), 659–685 (2006)
13. Greenwood, D., Vitaglione, G., Keller, L., Calisti, M.: Service level agreement management with adaptive coordination. In: Proceedings of the International Conference on Networking and Services (ICNS 2006), pp. 45–45. IEEE Computer Society (2006)
14. Hansmann, U., Merk, L., Nicklous, M.S., Stober, T.: Pervasive Computing: The Mobile World. Springer, Heidelberg (2003)
15. ITU: ITU-T E.800 E.800: Definitions of Terms Related to Quality of Service (2008)
16. Jineja, R., Sharma, D.: Multi-level SLA management architecture for cloud computing. *Int. J. Adv. Res. Comput. Sci. Softw. Eng.* **4**(5), 401–406 (2014)
17. Lê, L.-S., Ghose, A.: Contracts + goals = roles? In: Atzeni, P., Cheung, D., Ram, S. (eds.) ER 2012 Main Conference 2012. LNCS, vol. 7532, pp. 252–266. Springer, Heidelberg (2012)
18. Li, J., Tang, W., Wang, X.: Adding QoS to web service transaction management. In: Proceedings of the IEEE Conference on Cybernetics and Intelligent Systems, vol. 2, pp. 1308–1313. IEEE Computer Society (2004)
19. Lington, P.F., Milosevic, Z., Cole, J., Gibson, S., Kulkarni, S., Neal, S.: A unified behavioural model and a contract language for extended enterprise. *Data Knowl. Eng.* **51**(1), 5–29 (2004)
20. Pan, J., Paul, S., Jain, R.: A survey of the research on future internet architectures. *IEEE Commun. Mag.* **49**(7), 26–36 (2011)
21. Paulson, L.D.: Services science: a new field for today's economy. *Computer* **39**(8), 18–21 (2006)
22. Philip, P.B., Lewis, G., Merson, P.: Service level agreements in service-oriented architecture environments. Technical Note of Software Engineering Institute (2008)
23. Rahwan, I., Kowalczyk, R., Pham, H.H.: Intelligent agents for automated one-to-many e-commerce negotiation. In: Proceedings of the 25th Australasian Conference on Computer Science, pp. 197–204. Australian Computer Society, Inc. (2002)
24. Singh, M.P., Huhns, M.N.: Service-Oriented Computing: Semantics, Processes, Agents. Wiley, Hoboken (2006)
25. Yan, J., Kowalczyk, R., Lin, J., Chhetri, M.B., Goh, S.K., Zhang, J.: Autonomous service level agreement negotiation for service composition provision. *Future Gener. Comput. Syst.* **23**(6), 748–759 (2007)
26. Yang, S.J., Zhang, J., Lan, B.C.: Service level agreement-based QoS analysis for web services discovery and composition. *Int. J. Internet Enterp. Manag.* **5**(1), 39–58 (2006)