Modelling Service Level Agreements for Business Process Outsourcing Services[*]

Adela del–Río–Ortega, Antonio Manuel Gutiérrez, Amador Durán, Manuel Resinas, and Antonio Ruiz–Cortés

Universidad de Sevilla, Spain
{adeladelrio,amgutierrez,amador,resinas,aruiz}@us.es

Abstract. Many proposals to model service level agreements (SLAs) have been elaborated in order to automate different stages of the service lifecycle such as monitoring, implementation or deployment. All of them have been designed for computational services and are not well–suited for other types of services such as business process outsourcing (BPO) services. However, BPO services supported by process–aware information systems could also benefit from modelling SLAs in tasks such as performance monitoring, human resource assignment or process configuration. In this paper, we identify the requirements for modelling such SLAs and detail how they can be faced by combining techniques used to model computational SLAs, business processes, and process performance indicators. Furthermore, our approach has been validated through the modelling of several real BPO SLAs.

1 Introduction

Service level agreements (SLAs) have been used by many proposals in the last decade to automate different stages of the service lifecycle, using a formal definition of the different parts of an SLA such as *service level objectives* (SLOs), penalties, or metrics, to automate their negotiation [1], the provisioning and enforcement of SLA–based services [2], the monitoring and explanation of SLA runtime violations [3], or the prediction of such violations [4]. What all of these proposals have in common is that most of them have been designed for computational services. Therefore, they are aimed at enhancing software that supports the execution of computational services such as network monitors, virtualisation software, or application servers with SLA–aware capabilities.

On the other hand, *business process outsourcing* (BPO) services are non–computational services such as logistics, supply–chain, or IT delivery services, that are based on the provisioning of business processes as services, providing partial or full business process outsourcing. Like computational services, their execution is regulated by SLAs and supported by specific software [5,6]. In this case, since BPO services are process–oriented, the software that supports them is usually a *process–aware information systems* (PAIS) such as ERPs, CRMs, or *business process management systems* (BPMSs). However,

[*] This work has been partially supported by the European Commission (FEDER), the Spanish and the Andalusian R&D&I programmes (grants TIN2012–32273 (TAPAS), TIC–5906 (THEOS) and COPAS (P12–TIC-1867))

unlike computational services, there is little work related to the extension of PAIS with SLA–aware capabilities to support BPO services.

A PAIS with SLA–aware capabilities, i.e. an SLA–aware PAIS, is a PAIS that uses explicit definitions of SLAs to enable or improve the automation of certain tasks related to both the SLAs and their fulfilment such as performance monitoring, human resource assignment or process configuration [7]. For instance, an SLA–aware PAIS could be automatically instrumented according to the metrics defined in the SLA so that when there is a risk of not meeting an SLO, an alert is raised allowing the human actors involved in the process to take measures to mitigate the risk. Another example could be the automated configuration of the process, *e.g.* removing or adding activities, executed by the SLA–aware PAIS depending on the conditions of the SLA agreed with the client.

Apart from the benefits derived from the automation of these tasks, the need for a SLA–aware PAIS becomes more critical in a *business–process–as–a–service* (BPaaS) scenario. A BPaaS is a new category of cloud–delivered service, which, according to Gartner [8], can be defined as "the delivery of BPO services that are sourced from the cloud and constructed for multitenancy. Services are often automated, and where human process actors are required, there is no overtly dedicated labour pool per client. The pricing models are consumption–based or subscription–based commercial terms. As a cloud service, the BPaaS model is accessed via Internet–based technologies." In this setting, the conditions of the SLA agreed with each client may vary. Therefore, it is crucial for the PAIS that supports the BPaaS to behave according to the SLA agreed with the client. An example could be the prioritisation of the execution of tasks for those clients whose SLAs have bigger penalties if they are not met.

In this paper, we focus on the formalization of BPO SLAs as a first step to enable such SLA–aware PAIS. To this end, after analysing the modelling requirements of such SLAs, four main aspects involved in their formalization have been identified, namely: 1) the description of the business process provided by the service; 2) the SLOs guaranteed by the SLA; 3) the penalties and rewards that apply if guarantees are not fulfilled; and 4) the definition of the metrics used in these guarantees. Then, we detail how these aspects can be formalized by means of generic models for the definition of computational SLAs and techniques used to model process performance indicators. Furthermore, we have validated our approach through the modelling of several real BPO SLAs.

The remainder of the paper is structured as follows. In Section 2, a running example is introduced. Section 3 details the four elements that must be formalized in SLAs for BPO services and Section 4 shows how they can be modelled using WS–Agreement. Next, Section 5 reports on how the running example can be formalized using our proposal and discusses some limitations identified during the definition of the SLA metrics. Section 6 reports on work related to the definition of SLAs for BPO services. Finally, conclusions are detailed in Section 7.

2 Running Example

Let us take one of the BPO SLAs to which our approach has been applied as running example throughout this paper. The SLA takes place in the context of the definition of *statements of technical requirements* (SoTRs) of a public company of the Andalusian

Autonoumous Government, from now on *Andalusian Public Company*, APC for short. SoTRs are described in natural language and include information about the services required as well as their SLA. Although the running example includes one service only, further information on this or the rest of services, as well as for further application scenarios, is available at `http://www.isa.us.es/ppinot/caise2015`.

The SoTR of this example is defined for the Technical Field Support for the Deployment of the Corporative Telecommunication Network of the Andalusian Autonomous Government. It is presented in a 72–page document written in natural language including the SLAs defined for five of the required services, namely: 1) field interventions; 2) incidents; 3) network maintenance; 4) installations and wiring; and 5) logistics. In particular, we focus on the *field interventions* (FI) service.

From a high–level perspective, the FI service can be defined as follows: the APC requires an FI, which can have different levels of severity, from the contractor staff. Then, the contractor plans the FI and performs it at headquarters. In some cases, it is necessary for the contractor to provide some required documentation and, if such documentation is considered incomplete or inadequate by the APC, it needs to be resubmitted by the contractor until it fulfils the APC's quality requirements.

For this service, the SoTR document presents the following information: 1) the committed times by the contractor (see Table 1); 2) the general objective defined for FIs —the SLO of the SLA— represented as AFIP $> 95\%$, where the AFIP (*accomplished FIs percentage*) metric is defined as:

$$\text{AFIP} = \frac{\text{\# accomplished FIs}}{\text{\# FIs}} \times 100$$

and 3), the penalties applied in case the SLO is not accomplished (see Table 2). These penalties are defined over the monthly billing by the contractor for the FI service. In addition, the SoTR presents the following definitions for the referred times in Table 1:

Response Time Elapsed time between the notification of the FI request to the contractor and its planning, including resources assignment, *i.e.* technicians.

Presence Time Elapsed time between resource (technician) assignment and the beginning of the FI, *i.e.* technician arrival.

Resolution Time Elapsed time between the technician arrival and the end and closure of the FI.

Documentation Time If documentation, *i.e.* reports, is required, it is defined as the elapsed time between the end and closure of the FI and documentation submission.

Table 1. Committed times by the contractor (in hours) for the FI Service SLA

Criticality Level	Response Time	Presence Time	Resolution Time	Document. Time	Timetable	Calendar
Critical	0.5	4	2	4	8:00 – 20:00	Local
High	2	8	4	12	8:00 – 20:00	Local
Mild	5	30	6	24	8:00 – 20:00	Local
Low	5	60	8	48	8:00 – 20:00	Local

Table 2. Penalties definition (in monthly billing percentage) for the FI Service SLA

AFIP	**Penalty**
$94\% \leq \text{AFIP} < 95\%$	-1%
$93\% \leq \text{AFIP} < 94\%$	-2%
$92\% \leq \text{AFIP} < 93\%$	-3%
$91\% \leq \text{AFIP} < 92\%$	-4%
$90\% \leq \text{AFIP} < 91\%$	-5%
$\text{AFIP} < 90\%$	-10%

If the APC considers such documentation as incomplete or inadequate, it will be returned to the contractor and documentation time is again activated and computed.

3 Requirements for Modelling SLAs of BPO Services

After a study of the state of the art in SLAs for both computational and non–computational services, and the analysis of more than 20 different BPO SLAs developed by 4 different organisations, some of the requirements for modelling BPO SLAs in the context of SLA–aware PAIS have been identified. As a result, we conclude that four elements must be formalized in SLAs for BPO services, namely: 1) the business process; 2) the metrics used in the SLA; 3) the SLOs guaranteed by the SLA; and 4) the penalties and rewards that apply if guarantees are not fulfilled. Next we describe each of them.

3.1 Business process

An SLA is always related to one or more specific services. The way such services must be provided is usually defined by describing the underpinning business process, and this is often done in natural language. Consequently, the formalization of SLAs for BPO services requires the formalization of the business process itself. Note that it is not required for the SLA to detail the low level business process that will be enacted by the provider's PAIS since most SLAs do not delve into that level of detail and just focus on main activities and the consumer–provider interaction (cf. Fig 1 for the high–level business process of the running example). However, it should be possible to link this higher level business process to the lower level business process enacted by the PAIS.

3.2 SLA metrics

These are the metrics that need to be computed so that the fulfilment of the SLA can be evaluated. For instance, in the running example, *response time*, *presence time*, or AFIP are examples of such metrics. The mechanism used to define these metrics must have two main features. On the one hand, it must be *expressive*, i.e. it must allow the definition of a wide variety of metrics. On the other hand, it must be traceable with

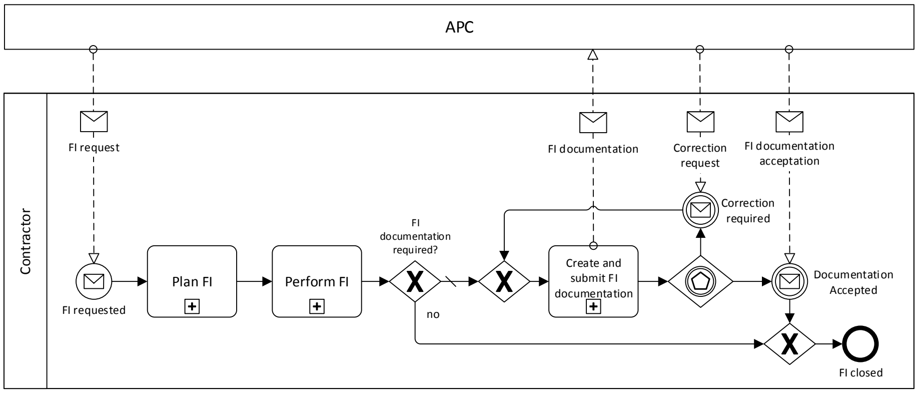


Fig. 1. BPMN model of Field Intervention (FI) service

the business process so that it enables their automated computation. In addition, it is convenient that the metrics are defined in a declarative way because it reduces the gap between the SLA defined in natural language and the formalised SLA and decouples the definition of the metric from its computation.

3.3 Service Level Objectives (SLOs)

These are the assertions over the aforementioned metrics that are guaranteed by the SLA and, hence, must be fulfilled during the execution of the service. For instance, the running example defines AFIP $> 95\%$ as an SLO for AFIP metric of the FI service. In general, SLOs can be defined as mathematical constraints over one or more SLA metrics.

3.4 Penalties and rewards

They are compensations that are applied when the SLO is not fulfilled or is improved, respectively. An example is shown in Table 2, which depicts the penalties that apply for the FI Service SLA in our running example. The specification of penalties and rewards require the definition of a mathematical function, whose domain is one or more SLA metrics and whose range is a real number representing the penalty or reward in terms of a percentage over the price paid for the service in a time period.

From these requirements, we conclude that the structure of SLAs for BPO services is very similar to the structure of SLAs defined for computational services. For instance, Amazon EC2 SLA[1] also includes a definition of the service; some metrics like the *monthly uptime percentage* (MUP); an SLO, which is called *service commitment*, defined as MUP $\geq 99.95\%$; and a penalty based on the MUP and defined in terms of a percentage over the price paid in the last month. Furthermore, the definition of SLOs and penalties and rewards can also be done in the same manner.

[1] `http://aws.amazon.com/ec2/sla/`

In contrast, the description of the service and the definition of the SLA metrics of BPO SLAs and computational SLAs present significant differences. The main reason is that, unlike computational services, BPO services are process–aware and, hence, their description and their SLA metrics are based on that process.

4 Modelling SLAs for BPO Services

Based on the requirements described in the previous section, and on the similarities and differences between BPO SLAs and computational SLAs, we propose modelling the latter SLAs by combining the agreement structure and mechanisms for the definition of SLOs, penalties, and rewards that have been already proposed for computational SLAs, with notations used to model processes and *Process Performance Indicators* (PPIs), such as [9,10,11,12,13]. PPIs are quantifiable metrics that allow the efficiency and effectiveness of business processes to be evaluated; they can be measured directly by data that is generated within the process flow and are aimed at the process controlling and continuous optimization [14].

Specifically, in this paper we propose using WS–Agreement [15] as the agreement structure; BPMN as the language to model business processes; PPINOT [13] as the mechanism to model PPIs; the predicate language defined in iAgree [16] to specify SLOs, and the compensation functions introduced in [17] to model penalties and rewards. These proposals have been chosen because of two reasons. Firstly, they are amongst the most expressive proposals of their kind, which is necessary to model the different scenarios that appear in BPO SLAs. Secondly, they have a formal foundation that enables the development of advanced tooling support that can be reused in a SLA–aware PAIS environments.

In the following, we introduce the basic structure of an SLA in WS–Agreement and then, we detail how it can be used together with other languages and models to define a BPO SLA. Furthermore, we also provide more details about the aforementioned models and the tooling support that has been developed for them.

4.1 WS–Agreement in a nutshell

WS–Agreement is a specification that describes computational service agreements between different parties. It defines both a protocol and an agreement document metamodel in the form of XML schema [15]. According to this metamodel, an agreement is composed of an optional name, a context and a set of terms. The *context* section provides information about participants in the agreement (*i.e.* service provider and consumer) and agreement's lifetime. The *terms* section describes the agreement itself, including *service* terms and *guarantee* terms.

Figure 2 shows the overall structure of a WS–Agreement document using iAgree syntax [16], which is designed for making WS–Agreement documents more human–readable and compact than with the original XML syntax. All examples included in this paper are defined using iAgree.

Service terms describe the provided service, and are classified in *service description terms*, *service properties* and *service references*. Service description terms (lines 9–10)

```
1  Agreement Example version 1
2  Provider as Responder
3    Metrics
4      ServiceCreditMeasure: Percentage
5      AvailabilityMeasure: Percentage
6      CostMeasure: Integer
7    AgreementTerms
8      Service Example @ http://mycloud.com/service.wsdl
9      DescriptionTerms
10       Cost : CostMeasure = 10
11     MonitorableProperties
12       Availability : AvailabilityMeasure
13     GuaranteeTerms
14       G1: Provider guarantees
15         Availability > 99
16         with monthly penalty of
17         ServiceCredit : ServiceCreditMeasure = 25
18           if Availability ≤ 99
19 EndAgreement
```

Fig. 2. Computational SLA in WS–Agreement using iAgree syntax

describe the features of the service that will be provided under the agreement. They identify the service itself, so there is no reason to monitor them along service lifecycle. Service properties (lines 11–12) are the set of monitorable variables relevant to the agreement, for which a name and a metric are defined. Finally, service references (line 8) point to an electronic service using endpoints references.

Guarantee terms (lines 13–18) define SLOs that the obligated party must fulfil together with the corresponding penalties and rewards. An SLO in WS–Agreement is an assertion over monitorable properties that must be fulfilled during the execution of the service. SLOs can be guarded by a *qualifying condition* (QC), which indicates a precondition to apply the constraint in the SLO. Both SLOs and QCs are expressed using any suitable user–defined assertion language. penalties and rewards.

4.2 Materialising BPO SLAs with WS–Agreement

WS–Agreement leaves consciously undefined the languages for the specification of service description terms, SLOs, or QCs. This flexibility makes WS–Agreement a good choice for modelling BPO SLAs since it allows embedding any kind of model in its terms. In this paper, we propose the following *WS–Agreement Configuration* [16] for defining BPO SLAs:

Service Description Terms In BPO services, this description can be provided in terms of the underpinning business process. In this paper we use the BPMN (*Business Process Model and Notation*) standard since it is a well–known standard widely used in both industry and academy.

Service Properties In BPO services, these metrics can be specified using a PPI–oriented approach. In this paper, we have chosen PPINOT [13] because of its expressiveness and its traceability with BPMN models. Furthermore, PPINOT has been used at the core of a software tool called the *PPINOT Tool Suite* [18], which includes the definition of PPIs using either a graphical or a template–based textual notation [19], their automated analysis at design–time, and their automated computation based on the instrumentation of open source BPMSs.

Specifically, metrics are defined using PPINOT measure definitions. As described in [13], they can be classified into three main categories depending on the number of process instances involved and the nature of the measure: base measures, aggregated measures, and derived measures.

Base measures They are obtained directly from a single process instance and do not require any other measure to be computed. Aspects that can be measured include: 1) the duration between two time instants (*time measures*); 2) the number of times something happens (*count measures*); 3) the fulfilment of certain condition in both running or finished process instances (*condition measures*); and 4) the value of a certain part of a data object (*data measures*).

Aggregated measures Sometimes, it is interesting not only knowing the value of a measure for a single process instance (*base measures*) but an aggregation of the values corresponding to the multiple instances of a process. For these cases, *aggregated measures* are used, together with an aggregation function such as *average*, *maximum*, etc.

Derived measures They are defined as functions of other measures. Depending on whether the derivation function is defined over single or multi–instance measures, derived measures are classified accordingly as *derived single–instance measures* or *derived multi–instance measures* (see [13] for details).

Guarantee Terms To define SLOs, we use the predicate language defined in iAgree [16], which includes relational, logical and common arithmetic operators. Apart from a concrete syntax, iAgree also provides semantics to define SLOs expressions as logic constraints, which enable the automation of analysis operations on SLAs such as detecting conflicts within an agreement document [16] or explaining SLA violations at run–time [3]. Concerning penalties and rewards, they are defined using iAgree syntax as well together with the notion of *compensation functions* defined in [17].

5 Applicability of our approach

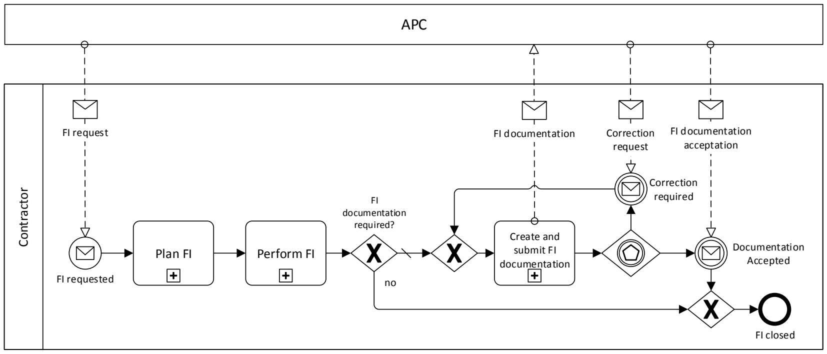
In order to validate the applicability of our approach, we have used it to model the SLAs of 9 different services designed by 3 different organisations. In the following, we show how WS–Agreement and PPINOT can be used to model the running example and then, discuss the limitations we have found and how they can be solved. The remaining SLAs that have been modelled are available at `http://www.isa.us.es/ppinot/caise2015`.

5.1 SLA for the running example

Figure 3 shows an excerpt of the SLA for the running example, in which the three elements of the BPO SLA are specified as follows.

```
Agreement FI_Service_SLA version 1
Provider Corporate as Responder;
  Metrics for FI_Service:
    ResponseTime: LinearTimeMeasure
       from event FI requested is triggered
       to activity Plan FI becomes active
       considering only working hours and local calendar
    PresenceTime: LinearTimeMeasure ...
    ResolutionTime: LinearTimeMeasure ...
    DocumentationTime: CyclicTimeMeasure aggregation Sum
       from activity Create and submit doc becomes active
       to activity Create and submit FI documentation becomes completed
       considering only working hours and local calendar
    CLevel: DataMeasure criticalityLevel of Intervention
    AFI_Measure: AggregatedMeasure with function sum
       aggregates DerivedMeasure with function A & B & C & D where
          A: DerivedMeasure with function
             CLevel = critical => ResponseTime   < 0.5 & PresenceTime      < 4 &
                                  ResolutionTime < 2   & DocumentationTime < 4
          B: DerivedMeasure with function
             CLevel = high     => ResponseTime   < 2 & PresenceTime      < 8 &
                                  ResolutionTime < 4 & DocumentationTime < 12
          C: DerivedMeasure with function
             CLevel = mild     => ResponseTime   < 5 & PresenceTime      < 30 &
                                  ResolutionTime < 6 & DocumentationTime < 24
          D: DerivedMeasure with function
             CLevel = low      => ResponseTime   < 5 & PresenceTime      < 60 &
                                  ResolutionTime < 8 & DocumentationTime < 48
    FI_Measure: AggregatedMeasure with function sum
        aggregates CountMeasure when event FI closed is triggered
    AFIP_Measure: DerivedMeasure with function ( AFI_Measure / FI_Measure ) * 100

AgreementTerms
  Service FI_Service
    process:
```



```

  MonitorableProperties
    AFIP: AFIP_Measure
  Guarantee Terms
    G1: Provider guarantees AFIP > 95%
      with monthly penalty
      of Penalty = 95 - AFIP if 90% ≤ AFIP < 95%
      of Penalty = 10        if AFIP < 90%
    ...
```

Fig. 3. Excerpt of the FI service SLA in *iAgree* syntax

Service Description Terms Service description terms (lines 34–36) specify the high level BPMN model associated to the FI service derived from the corresponding SoTR, as described in Section 2.

Service Properties Once the high level business process has been modelled, service properties relevant to the SLA are defined, namely AFIP (lines 37–38). This service property is computed according to the AFIP_Measure metric (lines 15–28), that measures the percentage of *accomplished* FIs (AFI_Measure) with respect to the total number of FIs (FI_Measure), as described informally in Section 2. The definition of these metrics is done by means of the measure definitions that PPINOT provides to detail how PPIs are measured (see [13] for details).

Guarantee Terms Finally, the guarantee terms of the SLA including its SLOs and penalties are specified. In this case, according to Tables 1 and 2, the percentage of accomplished interventions must be greater than 95%. This can be defined in terms of the previously defined service properties as AFIP > 95% (line 40). Additionally, penalties are defined as a percentage discount of the monthly billing if the SLO is not achieved. This is 1% of discount per each 1% of accomplished percentage under the objective, or 10% if the percentage is under 90%.

5.2 Limitations of our approach

The application of the proposed approach for defining SLAs of BPO services to real scenarios showed up some limitations concerning the definition of SLA metrics, whereas WS–Agreement and the models used to define business processes, SLOs, penalties, and rewards proved to be capable to model all possible situations.

Concerning SLA metrics, although most of them could be successfully modelled using PPINOT, there were a few types that could not be represented properly. As far as we know, this limitation is not specific to PPINOT, since there is not any other PPI modelling approach that can model all of the metrics that appear in the analysed SLAs. We believe that the main reason why we have found this limitation is that, although related, the purpose of PPIs and SLA metrics are slightly different. PPIs are used internally by the organisation that performs the process as a mechanism to improve its performance. In contrast, SLA metrics are aimed at providing service–level guarantees to the service consumer or defining penalties when guarantees are not met. As a consequence, SLA metrics are much more focused on the customer and its expectations than the former.

Specifically, we found four types of metrics that cannot be modelled neither with PPINOT nor with most of the other PPI modelling approaches:

Metrics that involve exclusion of idle time, suspend time, calendars or timetables
In the running example, when defining times like *resolution* time, *documentation* time, *etc*, the SoTR document usually specified that idle time should be ignored for those measures, and that the local calendar and working hours were considered to compute time for them. This ability to exclude time according to some criteria is not usually present in PPI modelling approaches.

Metrics that involve delays with respect to a date given in a data object These metrics require comparing the time instant when an activity had started or finished, or when an event was triggered, with respect to a due date contained in a document like a project plan, a replacement requirement or any other in order to compute possible delays. This is a rather frequent metric in SLAs since it is directly related with customer expectations. However, it is much less frequent as a PPI metric and, hence, it is not supported by PPI modelling approaches.

Metrics that involve human resources These metrics are used in SLAs in which the task performer profile must be taken into account when applying penalties, so that the penalty had a different coefficient to be applied according to the different profiles. This metric is again closely related with the customer. In this case, with the fact that the customer expects a fair compensation depending on the task performer profile that failed to fulfilled the guarantees. However, current PPI modelling approaches do not support any metric that involve information related with the human resources that performed the task.

Metrics that involve different processes Some SLA metrics have to be defined over two or more process instances. This happens when a metric require execution information from two different processes to be computed. Again, this metric cannot be modelled using current PPI modelling approaches, since a PPI focus on just one process by definition.

Some of these limitations could be easily addressed in PPINOT just by doing minor changes in its metamodel. However, others are left as future work since they require more significant changes. In particular, the first two type of metrics can be supported just by defining filters over *time measures*, so that idle time, suspend time, calendars or timetables can be taken into account when computing the time for the measure; and by adding a new type of measure, *time instant measure*, that measures the date and time in which an event takes place instead of the duration between two events. The metrics that involve human resources can be partially addressed using an extension to PPINOT to define resource–aware PPIs [20]. Finally, the metrics that involve different processes can be defined as a *derived measure* that relates measures in each process instance, but it is necessary to include information on how to correlate process instances when defining them, which is something that will be addressed in future work.

6 Related Work

A number of research efforts have focused on proposing models for SLA definition in computational and non–computational domains. In [21], *WSLA Framework* is introduced. This framework provides an *agreement document model* (WSLA), which is the origin of the WS–Agreement specification, and provides foundations to monitor SLA fulfillment. Sauvé *et al.* [22] propose a methodology to calculate SLO thresholds to sign IT services SLAs according to service function cost from a business perspective. In all these cases, guarantees are proposed upon computational metrics (e.g. response time or availability). Therefore, it is useful only for SLAs that apply to the software infrastructure that support business processes and not for the business processes offered as a service. Kieninger *et al.* [23] describe a categorization of IT services and outline a mechanism to obtain efficient SLOs for them. However, they do that in a conceptual level and do not detail how they can be formalised to enable their automated management. Daly et al. [24] propose an SLA model based on the different elements in the service provision, *i.e.* application, servers, network, etc, related to service provision system. Cardoso et al. [25] propose a description language for services that include business characteristics together with technical or operational parameters. Unlike our proposal of managing a business process as a service, this work is focused on managing services including business perspective. Finally, Wieder et al. [26] define a *Service Oriented Architecture* with their own SLA model. The model has to be refined on each specific domain and there is a independent proposal to define measurements. The problem with all these approaches is that the SLA model proposed offers no mechanism to model a business process nor to define metrics in terms of this business process. This seriously limits their applicability for building SLA–aware PAIS, in which processes play a key role.

Perhaps, the proposal closer to ours is done by Chau *et al.* [27]. It relates SLAs and business process artifacts where guarantees over the process are defined through process events. However, although similar to our work, this approach has a couple of limitations. First, the language to define metrics is imperative. Instead, PPINOT expressions are declarative, which eases the adaptation to different PAIS and makes it possible to define them in an user-friendly way by means of linguistic patterns as detailed in [19]. Second, the authors use their own model for SLA definitions, which limits the interoperability of their proposal and limits the reusability of existing proposals to analyse SLAs such as [16,3].

7 Conclusions and Future Work

In this paper, we have shown how BPO SLAs can be modelled by combining mechanisms for modelling computational SLAs with mechanisms to model business processes and PPIs. Specifically, we first analysed the requirements for modelling BPO SLAs after a study of the state of the art in SLAs for both computational and non–computational services and the analysis of more than 20 different BPO SLAs developed by 4 different organisations. The conclusion of this analysis was that the structure of SLAs for BPO services and the definition of SLOs, penalties, and rewards are very similar to those of SLAs defined for computational services. However, the service description and the definition of the SLA metrics of BPO SLAs and computational SLAs present significant differences. The reason is that, unlike computational services, BPO services are process–aware and this has an strong influence on how they are described.

On the light of these requirements, our proposal to model BPO SLAs combines well founded approaches and standards for modelling computational SLAs and PPIs. Specifically, we rely on WS–Agreement [15], which provides the general SLA structure, BPMN [28], which is used to model the business process related to the service, PPINOT [13], which allows the definition of metrics, and iAgree [16], which provides a language to define SLOs and penalties.

The application of the proposed approach to a number of real scenarios allowed us to conclude that our approach is able to model all possible situations in these scenarios except for some limitations concerning the definition of SLA metrics as detailed in Section 5.2. Some of them could be solved by applying minor changes to the PPINOT metamodel. However, other limitations require more significant changes that shall be carried out in future work.

Apart from addressing these limitations, there are two lines of future work. On the one hand, we want to build a SLA–aware PAIS that uses these models to improve the automation of certain tasks related to both the SLAs and their fulfilment. To this end, we plan to take advantage of the existing tool support for iAgree and PPINOT to automate the definition, monitoring and analysis of the aforementioned SLAs for BPO services. On the other hand, we want to include additional information in SLAs to cover not only performance guarantees, but other aspects that are relevant for the customer such as compliance or audit–related issues [29].

References

1. Resinas, M., Fernández, P., Corchuelo, R.: A bargaining-specific architecture for supporting automated service agreement negotiation systems. Science of Computer Programming **77**(1) (2012) 4–28
2. Emeakaroha, V.C., Brandic, I., Maurer, M., Dustdar, S.: Cloud resource provisioning and SLA enforcement via LoM2HiS framework. Concurrency and Computation: Practice and Experience **25**(10) (2013) 1462–1481
3. Müller, C., Oriol, M., Franch, X., Marco, J., Resinas, M., Ruiz-Cortés, A., Rodriguez, M.: Comprehensive Explanation of SLA Violations at Runtime. IEEE Transactions on Services Computing **7**(2) (2014) 163–183
4. Leitner, P., Ferner, J., Hummer, W., Dustdar, S.: Data-driven and automated prediction of service level agreement violations in service compositions. Distributed and Parallel Databases **31**(3) (2013) 447–470
5. Gutiérrez, A., Cassales Marquezan, C., Resinas, M., Metzger, A., Ruiz-Cortés, A., Pohl, K.: Extending ws-agreement to support automated conformity check on transport and logistics service agreements. In Basu, S., Pautasso, C., Zhang, L., Fu, X., eds.: Service-Oriented Computing. Volume 8274 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2013) 567–574
6. Marquezan, C., Metzger, A., Franklin, R., Pohl, K.: Runtime management of multi-level slas for transport and logistics services. In Franch, X., Ghose, A., Lewis, G., Bhiri, S., eds.: Service-Oriented Computing. Volume 8831 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2014) 560–574
7. van der Aalst, W.M.P.: Business process configuration in the cloud: How to support and analyze multi-tenant processes? In: 9th IEEE European Conference on Web Services, ECOWS 2011, Lugano, Switzerland, September 14-16, 2011. (2011) 3–10
8. Gartner, Inc.: Business Process as a Service (BPaaS). Gartner IT Glossary (2013) Available from `http://www.gartner.com/it-glossary/business-process-as-a-service-bpaas`.
9. Popova, V., Sharpanskykh, A.: Modeling organizational performance indicators. Information Systems **35**(4) (2010) 505–527
10. Wetzstein, B., Ma, Z., Leymann, F.: Towards measuring key performance indicators of semantic business processes. Business Information Systems **7** (2008) 227–238

11. Pedrinaci, C., Lambert, D., Wetzstein, B., van Lessen, T., Cekov, L., Dimitrov, M.: Sentinel: a semantic business process monitoring tool. In: international Workshop on Ontology-Supported Business Intelligence (OBI). (2008) 26–30
12. Delgado, A., Weber, B., Ruiz, F., de Guzmán, I.G.R., Piattini, M.: An integrated approach based on execution measures for the continuous improvement of business processes realized by services. Information & Software Technology **56**(2) (2014) 134–162
13. del-Río-Ortega, A., Resinas, M., Cabanillas, C., Ruiz-Cortés, A.: On the Definition and Design-time Analysis of Process Performance Indicators. Information Systems **38**(4) (2012) 470–490
14. Chase, G., Rosenberg, A., Omar, R., Taylor, J., Rosing, M.: Applying Real-World BPM in an SAP Environment. SAP Press. Galileo Press (2011)
15. Andrieux, A., Czajkowski, K., Dan, A., Keahey, K., Ludwig, H., Nakata, T., Pruyne, J., Rofrano, J., Tuecke, S., Xu, M.: Web services agreement specification (ws-agreement). Specification from the Open Grid Forum (OGF). (2007)
16. Müller, C., Resinas, M., Ruiz-Cortés, A.: Automated Analysis of Conflicts in WS-Agreement. IEEE Transactions on Services Computing (2013) 1–1
17. Müller, C., Gutiérrez, A.M., Martín-Díaz, O., Resinas, M., Fernandez, P., Cortés, A.R.: Towards a Formal Specification of SLAs with Compensations. In Meersman, R., Panetto, H., Dillon, T.S., Missikoff, M., Liu, L., Pastor, O., Cuzzocrea, A., Sellis, T., eds.: On the Move to Meaningful Internet Systems: {OTM} 2014 Conferences - Confederated International Conferences: CoopIS, and {ODBASE} 2014, Amantea, Italy, October 27-31, 2014, Proceedings. Volume 8841 of Lecture Notes in Computer Science., Springer (2014) 295–312
18. del-Río-Ortega, A., Cabanillas, C., Resinas, M., Ruiz-Cortés, A.: PPINOT tool suite: A performance management solution for process-oriented organisations. In: Proc. of the 11th International Conference on Service-Oriented Computing (ICSOC). (2013) 675–678
19. del-Río-Ortega, A., Resinas, M., Durán, A., Ruiz-Cortés, A.: Using templates and linguistic patterns to define process performance indicators. Enterprise Information Systems **In Press** (2014)
20. del-Río-Ortega, A., Resinas, M., Cabanillas, C., Ruiz-Cortés, A.: Defining and analysing resource-aware process performance indicators. In: Proc. of the CAiSE'13 Forum at the 25th International Conference on Advanced Information Systems Engineering (CAiSE). (2013) 57–64
21. Keller, A., Ludwig, H.: The wsla framework: Specifying and monitoring service level agreements for web services. Journal of Network and Systems Management **11**(1) (2003) 57–81
22. Sauvé, J., Marques, F., Moura, A., Sampaio, M.: SLA design from a business perspective. Ambient Networks (2005) 72–83
23. Kieninger, A., Baltadzhiev, D., Schmitz, B., Satzger, G.: Towards Service Level Engineering for IT Services: Defining IT Services from a Line of Business Perspective. 2011 Annual SRII Global Conference (2011) 759–766
24. Daly, D., Kar, G., Sanders, W.: Modeling of service-level agreements for composed services. In Feridun, M., Kropf, P., Babin, G., eds.: Management Technologies for E-Commerce and E-Business Applications. Volume 2506 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2002) 4–15
25. Cardoso, J., Barros, A., May, N., Kylau, U.: Towards a unified service description language for the internet of services: Requirements and first developments. In: Services Computing (SCC), 2010 IEEE International Conference on. (2010) 602–609
26. Wieder, P., Butler, J., Theilmann, W., Yahyapour, R., eds.: Service Level Agreements for Cloud Computing. Volume 2506. Springer (2011)
27. Chau, T., Muthusamy, V., Jacobsen, H.a., Litani, E., Chan, A., Coulthard, P.: Automating SLA Modeling. In: Proceedings of the 2008 Conference of the Center for Advanced Studies on Collaborative Research: Meeting of Minds. (2008) 10:126–10:143

28. Object Management Group (OMG): Business process model and notation (BPMN) version 2.0 (2011) Available from: http://www.omg.org/spec/BPMN/2.0/PDF.
29. Accorsi, R., Lowis, L., Sato, Y.: Automated Certification for Compliant Cloud-based Business Processes. Business & Information Systems Engineering **3**(3) (2011) 145–154