SLA Specification for IoT Operation - The WSN-SLA Framework

Guillaume Gaillard, Dominique Barthel, Fabrice Theoleyre, Fabrice Valois

SLA Specification for IoT Operation - The WSN-SLA Framework Research Report - INRIA Version 1.0, 30/06/2014

Guillaume Gaillard , Dominique Barthel , Fabrice Theoleyre, Fabrice Valois

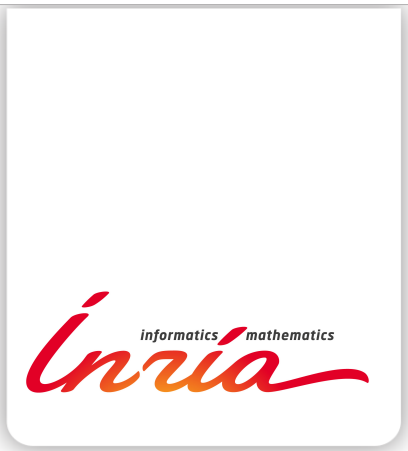
RESEARCH REPORT

N° 8567

Juillet 2014

Project-Team UrbaNet

ISSN 0249-6399 ISRN INRIA/RR--8567--FR+ENG



SLA Specification for IoT Operation - The WSN-SLA Framework Research Report - INRIA Version 1.0, 30/06/2014

Guillaume Gaillard * †, Dominique Barthel *, Fabrice Theoleyre‡, Fabrice Valois †

Project-Team UrbaNet

Research Report n° 8567 — Juillet 2014 — 71 pages

Abstract:
In this specification, we provide a guideline to write Service Level Agreements (SLAs) For the Internet Of Things (IoT) Operation. Current growth in the field of connected devices open the door to Wireless Sensor Network (WSN) Operators that share their operated transmission infrastructure among several clients. Each client company has specific application needs for the transmission of messages between their devices and their Information System. Some require bounded delays, others want minimal loss rate.
The WSN Operator must fulfill these Quality of Service (QoS) requirements, as specified in a contract, the SLA. In the Web Services domain, SLAs have been studied for long, and a specification was given, in XML format, in the WSLA framework [5].
We extend the specification with specific items that integrate the WSN constraints into the SLA framework. We give the possibility to address specific sets of sensors with a given QoS. We facilitate the construction of complex metrics that express the performance of the WSN. Our solution gives useful information for direct implementation in the field.

Key-words: Service Level Agreement, Quality of Service, Wireless Sensor Networks, Internet of Things, XML, Network Management

* Orange Labs R&D, Meylan, France. Email:firstname.lastname@orange.com
† Université de Lyon, INRIA, INSA-Lyon, Laboratoire CITI, F-69621. Email:fabrice.valois@insa-lyon.fr
‡ ICube, Université de Strasbourg / CNRS. Email:theoleyre@unistra.fr

RESEARCH CENTRE GRENOBLE – RHÔNE-ALPES

Inovallée
655 avenue de l'Europe Montbonnot
38334 Saint Ismier Cedex

Spécification de SLA pour l'Internet des Objets - le modèle WSN-SLA
Rapport de Recherche - INRIA
Version 1.0, 30/06/2014

Résumé :

Dans cette spécification, nous proposons un modèle de rédaction de contrats Service Level Agreements (SLA) pour opérer l'Internet des Objets (IoT). L'importance croissante du domaine des objets connectés permet à des opérateurs de réseaux de capteurs (WSN Operators) de mutualiser leur infrastructure radio pour plusieurs clients. Chaque entreprise cliente a des besoins applicatifs spécifiques liés à la transmission de messages entre leurs capteurs et leurs systèmes d'information.

L'opérateur de réseaux de capteurs doit respecter ces exigences de Qualité de Service (QoS), comme cela est spécifié dans un contrat, le SLA. Dans le domaine des Web Services, les contrats SLA ont été étudiés depuis longtemps, et il en a été donné une spécification en XML : c'est le modèle WSLA [5].

Nous étendons cette spécification en y intégrant des éléments qui répondent aux contraintes des réseaux de capteurs. Nous donnons la possibilité de respecter la QoS pour des ensembles particuliers de capteurs (par exemple sur une zone donnée). Nous permettons la construction de métriques complexes qui rendent compte de la performances des réseaux de capteurs.

Notre solution donne des informations utiles pour un déploiement sur le terrain.

Mots-clés : Service Level Agreement, Qualité de Service, Réseaux de Capteurs Sans Fil, Internet des Objets, XML, Mutualisation, Gestion de Réseaux

Contents

Chapter 1

Introduction

The Capillary Networks encompass the transmission of information of a growing number of applications, from the devices to the Information Systems. They need to be intensively managed, in order to work properly. Particularly in the context of smart cities, the radio communication channel must be spatially and temporally shared among the devices that give temperature, car presence, pollution alarms, gas indices, etc.

Instead of the dedicated deployment and maintenance that is done for each application nowadays, we propose that Wireless Sensor Networks operators, WSN Operators, share a relay infrastructure between several clients. They would handle the messages from each client device to each client information system, and vice-versa. The infrastructure sharing is a key point to reducing costs: only one deployment is needed, the WSN Operator has a global view of the network :

- maintenance is easier, and more scalable (we address the problems of several clients at the same time, hence we save our efforts);
- we use more efficiently the communication resources by sharing them among users, and that reduces the costs because the system is well dimensioned (we don't waste resource);
- the WSN Operator makes the devices access the medium intelligently, instead of letting them compete and interfere.

The WSN Operator dedicates himself to manage the radio communications, leaving the clients their expertise on the design of the applications.

The WSN Operator adds value to the Internet of Things (IoT) actors by providing guarantees on a quality of service (QoS) to each client for each of its application flows. The WSN Operator must meet the requirements of the clients in terms of delay, and loss rate. He must be able to prove the contracted conditions are respected.

For example, a gas company wants to collect a daily gas index for each of its meters using the WSN operator radio infrastructure. The client application requires that at least one message of the two that are generated be collected each day, for 95% of the client meters. Moreover, the client wants its messages be collected in less than one hour, in order to answer billing requirements. The WSN Operator must provide the information that its solution works, day after day. He must settle the conditions on which he gives his commitments : position of the devices, number of applicative messages, etc.

In order to make this possible, we proposed a framework in [4]. Several mechanisms have been studied, that provide differentiated QoS for WSN [11]. However, the novelty of our contribution resides in the link we propose between the clients applicative requirements and the network

management system. The architecture presented in [4] gives the opportunity to the clients to negotiate their requirements in contracts named Service Level Agreements (SLAs), and allows the WSN Operator to manage the performance of its network infrastructure, and to inform the clients about the fulfillment of the clauses of the SLA.

This current document provides a method for building SLA for the IoT. It is based on a specification in XML format, of possible contents of the SLAs that can be negotiated between a client and a WSN Operator. The proposed schema fits the constraints and parameters that are required for operating WSNs.

We explain more precisely in Chapter 2 what is addressed by our proposal. Then, we provide in Chapter 3 a brief summary of the XML Language and functionalities. We complete the related work in Chapter 4 by explaining why the previous work inspires us, what hard points are already addressed, and what is specific to WSNs and needs to be addressed. We describe our specification, comparing it to the existing efforts, in Chapter 5. Finally, in Chapter 6, we provide an example of SLA on a known use case : the collection of gas meter indices. This is illustrated by samples of the whole SLA document given in Appendix A.

Chapter 2

Problem Statement

Service Level Agreements (SLAs) are contracts of QoS between a service provider, a service consumer, and possible supporting parties. They specify the way the service is proposed, and the obligations that each actor must meet. They have been used for a very long time, in several IT services, particularly on IP [9]. They were used in telephony, for example, to specify the quality of the voice transmission or the quantity of possible simultaneous calls. Operators (i.e telco service providers) and clients have their relationships ruled by these SLAs.

The notion of SLA has been revived ten years ago with the appearance of the Web Services and the Service Oriented Architecture. In 2001-2003, Keller and Ludwig provided a framework written in XML for having SLAs for Web Services [5]. This publication has become a key reference, and is still frequently cited by new publications. However, SLAs are often too specific to a domain [2], or too generic and less applicable to concrete use cases [6].

Nowadays, researchers tend to only take into account the WSN data collection through an abstraction of network resources and trough the Big Data paradigm. Whereas we aim at providing a framework for WSN management, that answers at the MAC and network layer to the different applicative requirements. Our solutions must be applicable in the scope of standards and technologies such as RPL [10] or 6TiSCH [8].

The services specified here concern the transmission of applicative flows between client devices and client Information Systems. Our goal is to define guarantees on the network performance, regarding end-to-end parameters of the networking layers, such as the message delivery ratio or the delay of transmission. The considered applications (telemetering, pollution monitoring, etc. See [1]) depend on these parameters in order to work properly. Besides, the clients have to bound the traffic they generate according to specified constraints.

In this context, we have to make appear information specific to WSNs in the SLAs. That corresponds to the background of the negotiation of the SLA between the WSN Operator and a client.

We need to write:

- the quantity and the 3D geo-location of the devices: the Operator must check the capacity of its infrastructure, and the reachability of the new devices;
- parameters of a specific set of devices (e.g. the sensors of a specific building, or the temperature sensors of a zone, or 2 sensors by 50 square meters zone in a city, etc.);
- specific calculations and mathematical operators (for example the way to calculate a message delivery ratio during a given period);

- non ambiguous basic metrics, provided by the WSN Operator;
- the evaluation of Service Level Objectives (SLOs) on sliding periods or on fixed date to date periods.

All these pieces of information are specified in Part 5.

Chapter 3

XML Background

In order to describe the WSLA framework in Chapter 4, and then illustrate our contributions in Chapters 5, 6, and in Appendix A, we need to give a big picture of how the XML works.

Section 3.1 explains the syntax of a XML file, while Section 3.2 gives the components of an XML Schema.

Readers that have experience in the use of XML can directly begin Chapter 4.

3.1 The XML Language

More than a language, Extensible Markup Language (XML) is an expression format at the interface between human understanding and computer automatized processing [3]. It has few main syntactical elements:

- Text contents:
  Text contents are the basic elements of an XML specification: they set the values of whatever is expressed.
- Tags:
  Tags constitute the identifier of each element of an XML file: they may be interpreted specifically by a software. They are delimited by < and >. They may contain no value, text contents, or other sub-elements delimited by tags.
- Attributes:
  Tags can have attributes, further specifying them (e.g. a name). Attributes take the form of a parameter and a value.
- Comments:
  The author of an XML file may insert comments for human better understanding of the structure of the document.

Here are some examples and illustrations:

```
1 <!-- Comment -->
2 <!-- the Element FirstElement contains a tag and a text content -->
3 <FirstElement> Text content </FirstElement>
4
5 <!-- the Element SecondElement contains text and a third element, named example,
      without any text content -->
6 <SecondElement attribute="attributeValue">
7   Other text content of second element
```

```
8     <ThirdElement name="example"/>
9     <AnotherElement> example2</AnotherElement>
10
11  </SecondElement>
```

Listing 3.1: XML syntax illustration.

Listing 3.1 shows comments (e.g. line 1, 2, 5), a tag (e.g. line 9), attributes (e.g. line 6, 8), and text contents (e.g. line 3).

3.2 The XML Schema Description

The XML files and syntax leave the author completely free to build structured documents of chosen elements and values. But one may need to restrict these liberties by limiting the list and contents of possible elements. In order to specify such a class of particular documents, it is possible to use XML Schema files and syntax. The XML Schema rules the content of an XML file, specifying:

- the elements that can appear in the document;
- the type of the text contents (e.g. string, float, integer, date, etc.);
- the tags expected for each element and their number of occurrences;
- the attributes expected for each tag and their number of occurrences.

The syntax used in a XML Schema is the same as in a XML document. There are specific XML elements in an XML Schema file (See Figure 3.1). The principal ones are:

- Simple and complex elements: simple elements refer to XML elements only constituted by a text content.
- Simple and complex types: to define elements with attributes, or elements including other elements, we use complex elements in the XML Schema. A list of simple types, such as digits, dates, or strings, are defined in XML. But one can create complex types based on them.
- Extensions and restrictions: an element can be extended from another one by a new attribute. A type can be restricted (e.g. an element may be restricted to values upper than 10).
- Sequences and choices: an XML element may have a sequence of ordered sub-elements. Elements can be of mixed nature (and contain text contents, elements, etc.) It can also have a sub-element chosen in a list. Elements can have several declaration of the same sub-element, or no occurrence of a the same element. In the XML Schema, one can bound the total count of occurrences of a sub-element.

Listing 3.2 is one example of a XML Schema corresponding to the XML instance showed in Listing 3.1.

```
1  <!-- the Element FirstElement is simple and contains a string. It has no attribute -->
2  <xsd:element name="FirstElement" type="xsd:string"/>
3
4  <!-- the Element SecondElement contains text and a list of two elements, with one
       attribute -->
```

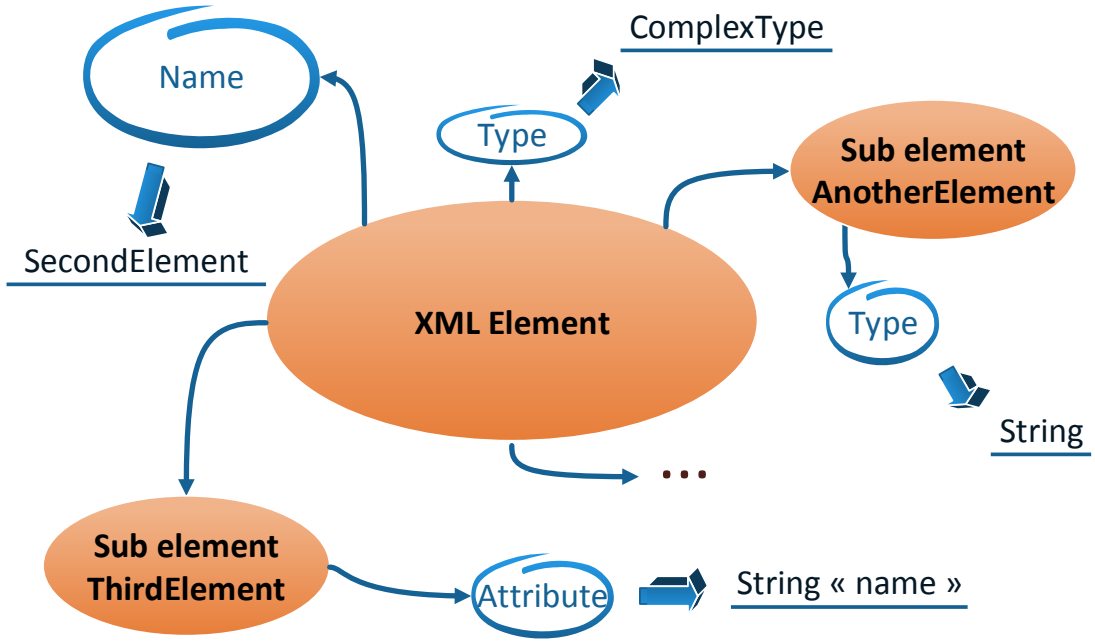


Figure 3.1: The XML Schema.

```
5  <xsd:element name="SecondElement">
6    <xsd:complexType mixed="true">
7      <xsd:sequence>
8            <xsd:element name="ThirdElement">
9          <xsd:complexType>
10           <xsd:attribute name="name" type="xsd:string"/>
11         </xsd:complexType>
12           <xsd:element name="AnotherElement" type="xsd:string"/>
13     </xsd:sequence>
14   </xsd:complexType>
15 </xsd:element>
```

Listing 3.2: XML Schema illustration.

It shows the declaration of the simple element *FirstElement* at line 2. It shows the declaration of the complex element *SecondElement* at line 5. *SecondElement* is mixed, composed of a sequence of two elements, *ThirdElement* and *AnotherElement* (line 12). *ThirdElement* is complex, it has an attribute, *name*, that takes string values (line 10).

Figure 3.1 illustrates the XML Schema of the second element of Listing 3.2. At the center, we see that the XML element comprises a type (complex), a name (SecondElement), and another sub-element that is an attribute, the string name.

Chapter 4

Back to the WSLA Framework [5]

In this chapter, we go into details of the WSLA proposition[1] [7]. In Section 4.1, we explain the background of the proposal, and why it is the same for our problem. In Section 4.2, we describe the components of the WSLA documents. Finally, in Section 4.3, we highlight the implementation parts that are specific to web services, and what is missing to address WSNs.

4.1 A Common Scenario

Keller and Ludwig proposed a SLA specification for Web Services [5]. They defined a framework for the contractual relationships between the actors of the web.

Similarly, we define here a set of parameters and conditions that compose the contracts between the clients (e.g. utilities such as gas companies) and the WSN Operator, for a specific service (e.g. the telemetering on a zone of a city).

4.1.1 Why having SLAs as a framework for writing contracts for IT services ?

The diversity of IT actors make them prone to compete on the same market. SLAs are a way to explicitly define what services are offered by each actor. Hence, a client is able to compare the services through the SLAs, and choose what best suits its needs.

For the comparison to be possible, the contracts must be written in non-ambiguous terms. The SLA framework, specifying all the set of possible parameters, is a solution to address this interoperability.

By following a template of SLA, the parties ensure that the contract they are writing is complete and coherent, because they suppose the same template has been used previously without any trouble. Moreover, you are able to re-use the clauses of the existing SLAs and adapt them to your needs (e.g. the actions guaranteed by the Service Provider in case of service degradation).

As a summary, SLAs offer clarity, interoperability, expression exhaustiveness and flexibility, and examples and templates that everybody may use.

[1] Note that all the listings of this chapter are extracted from [7], Copyright 2001, 2002, 2003 IBM Corp.

4.1.2 Why choosing XML for the data format?

The XML format is of common usage, partly because of the growing importance of the web, and its use in HTML.

It is:

1. expressive: it allows to write documents with in a structured and easily human readable form.
2. interoperable: defining a identical XML Schema for all the SLA templates makes the entities understand each other.
3. unambiguous: the XML Schema provides several constraints on the sequences of elements, their attributes, and values. And this document is written in the purpose of avoiding bad interpretations of the XML files. Hence, clients, operators, and developers should not have any doubt when interpreting the XML code.

4.1.3 The Scope and Conditions of WSLA

Keller and Ludwig have provided an XML Schema that specifies all the components of an SLA for Web Services [7]. They don't make any assumptions about the way to build these documents. This can be the result of a negotiation between the actors, or the proposal of one actor to the others. Else, the services can be previously defined by the WSN operator, whereas the values and thresholds can be let to the negotiation.

Besides, the deployment of the configuration of the system according to a new SLA, is proprietary and out of scope of the WSLA. The IS of the clients are supposed reachable by the Service Provider.

4.2 The Structure of a Service Level Agreement

A WSLA document consists in three parts (Listing 4.1 shows the elements of type *wslaType*):

- the Parties: See the *wsla:Parties* element at line 8, and Section 4.2.1;
- the Service Definitions, where the SLA parameters and metrics are defined. See the *ServiceDefinition* element at line 9 and Section 4.2.2;
- the Obligations, where the Service Level Objectives and corresponding Actions are defined. See the *Obligations* element at line 10 and Section 4.2.3.

```
1  <!-- Copyright 2001, 2002, 2003 IBM Corp -->
2  <!-- -->
3  <!-- Global WSLA structure -->
4  <!-- -->
5
6  <xsd:complexType name="wslaType">
7    <xsd:sequence>
8      <xsd:element ref="wsla:Parties"/>
9      <xsd:element name="ServiceDefinition" type="wsla:ServiceDefinitionType" maxOccurs="
       unbounded"/>
10     <xsd:element name="Obligations" type="wsla:ObligationsType"/>
11   </xsd:sequence>
12   <xsd:attribute name="name" type="xsd:string"/>
13 </xsd:complexType>
14
15 <xsd:element name="SLA" type="wsla:WSLAType">
16
```

```
17 </xsd:element>
```

Listing 4.1: XML Schema for the SLA structure, from [7].

we describe them in the following Subsections.

4.2.1 The Parties

```
1 <!-- Copyright 2001, 2002, 2003 IBM Corp -->
2 <xsd:complexType name="PartiesType">
3   <xsd:sequence>
4     <xsd:element name="ServiceProvider" type="wsla:SignatoryPartyType"/>
5     <xsd:element name="ServiceConsumer" type="wsla:SignatoryPartyType"/>
6     <xsd:element name="SupportingParty" type="wsla:SupportingPartyType" minOccurs="0"
      maxOccurs="unbounded"/>
7   </xsd:sequence>
8 </xsd:complexType>
```

Listing 4.2: XML Schema for the SLA Parties, from [7].

The parties are the actors (juridical entities, mainly companies) that are cited in the SLA document.

In WSLA, three sorts of parties are possible, as shown in Listing 4.2:

- the Service Provider, at line 4: the actor that offers a service to the clients. See the element description in Figure 4.1, and Listings 4.3 and 4.5;
- the Service Consumer, at line 5: the considered client. It has the same type *wsla:SignatoryPartyType* as the Service Provider element, and hence the same structure;
- the Supporting Parties, at line 6: other actors responsible for specific tasks defined in the document. For example, they may provide values of some performance parameters, or have specific actions on the services. There may be several Supporting Parties, or no one, in an SLA document.

Figure 4.1 illustrates the sequence of elements defined in the XML Schema, that permits to build the Service Provider. It shows the two XML extensions (*WSDLSOAPAction* and *SignatoryParty*, designed by arrows) and the components of the parties.

```
1  <!-- Copyright 2001, 2002, 2003 IBM Corp -->
2  <xsd:element name="ServiceProvider" type="wsla:SignatoryPartyType"/>
3
4  <xsd:complexType name="SignatoryPartyType">
5    <xsd:complexContent>
6      <xsd:extension base="wsla:PartyType">
7      </xsd:extension>
8    </xsd:complexContent>
9  </xsd:complexType>
10
11 <xsd:complexType name="PartyType" abstract="true">
12   <xsd:sequence>
13     <xsd:element name="Contact" type="wsla:ContactInformationType"/>
14     <xsd:element ref="wsla:Action" minOccurs="0" maxOccurs="unbounded"/>
15   </xsd:sequence>
16   <xsd:attribute name="name" type="xsd:string"/>
17 </xsd:complexType>
18
19 <xsd:complexType name="ContactInformationType">
20   <xsd:sequence>
21     <xsd:element name="Street" type="xsd:string"/>
22     <xsd:element name="City" type="xsd:string"/>
23   </xsd:sequence>
24 </xsd:complexType>
```

Listing 4.3: XML Schema for the Service Provider, from [7].

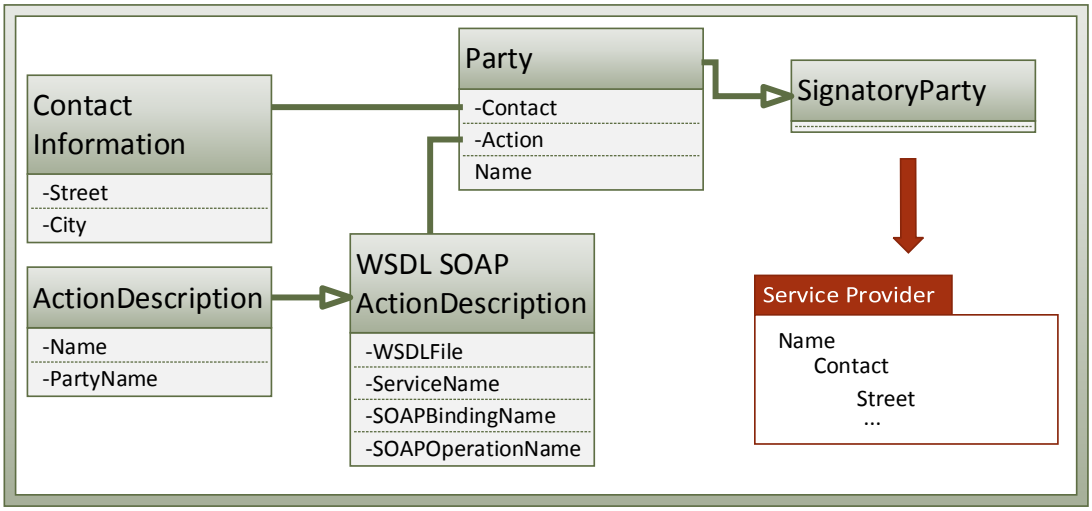


Figure 4.1: The elements of the Service Provider Schema.

```
<!-- Copyright 2001, 2002, 2003 IBM Corp -->
<xsd:element name="Action" type="wsla:ActionDescriptionType"/>
<xsd:complexType name="ActionDescriptionType" abstract="true">
  <xsd:attribute name="name" type="xsd:string"/>
  <xsd:attribute name="partyName" type="xsd:string"/>
</xsd:complexType>

<xsd:complexType name="WSDLSOAPActionDescriptionType">
  <xsd:complexContent>
    <xsd:extension base="wsla:ActionDescriptionType">
      <xsd:sequence>
        <xsd:element name="WSDLFile" type="xsd:anyURI"/>
        <xsd:element name="ServiceName" type="xsd:string" minOccurs="0"/>
        <xsd:element name="SOAPBindingName" type="xsd:ID"/>
        <xsd:element name="SOAPOperationName" type="xsd:ID"/>
      </xsd:sequence>
    </xsd:extension>
  </xsd:complexContent>
</xsd:complexType>
```

Listing 4.4: XML Schema for the Action Description Type, from [7].

```
<!-- Copyright 2001, 2002, 2003 IBM Corp -->
<ServiceProvider name="ACMEProvider">
  <Contact>
    <Street>PO BOX 218</Street>
    <City>Yorktown, NY 10598, USA</City>
  </Contact>
  <Action name="notification" partyName="ACMEProvider" xsi:type="
    WSDLSOAPActionDescriptionType">
    <WSDLFile>notification.wsdl</WSDLFile>
    <SOAPBindingName>soapnotification</SOAPBindingName>
    <SOAPOperationName>notification</SOAPOperationName>
  </Action>
</ServiceProvider>
```

Listing 4.5: XML Instance of the ACMEP Service Provider, from [7].

```
<!-- Copyright 2001, 2002, 2003 IBM Corp -->
```

```
2  <xsd:simpleType name="RoleType">
3    <xsd:restriction base="xsd:string">
4      <xsd:enumeration value="MeasurementService"/>
5      <xsd:enumeration value="ManagementService"/>
6      <xsd:enumeration value="ConditionEvaluationService"/>
7    </xsd:restriction>
8  </xsd:simpleType>
9
10 <xsd:complexType name="SupportingPartyType">
11   <xsd:complexContent>
12     <xsd:extension base="wsla:PartyType">
13       <xsd:sequence>
14         <xsd:element name="Sponsor" type="xsd:string" maxOccurs="2"/>
15         <xsd:element name="Role" type="wsla:RoleType" maxOccurs="unbounded"/>
16       </xsd:sequence>
17     </xsd:extension>
18   </xsd:complexContent>
19 </xsd:complexType>
```

Listing 4.6: XML Schema for the Supporting Parties, from [7].

Listing 4.3 shows the XML Schema of the element *ServiceProvider* at line 2. It is of type *wsla:SignatoryPartyType*, as described at line 4. *SignatoryPartyType* is an extension of type *PartyType* (line 11). This extension is just a name specification, it does not add any attribute or sub-element.

PartyType is composed of a name attribute, an element *Contact* (described at line 19) and an element *wsla:Action*. Line 14 specifies that this last element *Action* appears from zero to several times in the Party. Indeed, each party may have the responsibility of many actions.

Actions are of type *ActionDescriptionType*, as described in Listing 4.4 at line 2. The *ActionDescriptionType* is extended as *WSDLSOAPActionDescriptionType* at line 8. The extension adds four elements that are specific to this type of action (parameters WSDL and SOAP).

Listing 4.5 specifies the instance of the Service Provider named ACMEProvider with its contact and action.

The Supporting Parties are of type *wsla:SupportingPartyType*, as described in Listing 4.6. Line 12 shows that it extends the *partyType*.

Listing 4.6 shows that the Supporting Parties are specified with:

- one or two sponsors, at line 14: they define whose Signatory Party they are in agreement with;
- a role, of type *RoleType*, at line 2: what the Supporting Party is responsible for, e.g. a Measurement Service, at line 4;

4.2.2 The Service Definitions

The main part of the SLA is the description of the service. In this section, the Service Provider and the Customers agree on the parameters and the values they want to guaranty. They define how the parameters are evaluated, what are the corresponding metrics, and who is responsible for providing them.

Hence, in WSLA, there is several possible elements in the Service Definition part of the SLA document. Listing 4.7 shows all the elements one can combine in order to define a service.

Each SLA Parameter corresponds to a variable that is evaluated and used in the further described Obligations part. The SLA Parameters are built by a sequence of functions applied on Metrics, during a time frame specified in Schedules.

Once built, one can organize the SLA document by associating the elements in Operations and Operation Groups. This structure allows an easier understanding of the document.

We describe these elements in the following Listings:

- the Service Objects, at line 28: the elements of a task:

  the Schedule element, see Listing 4.8;

  the Metric element, see Listing 4.11;

  the SLA Parameter element, see Listing 4.10;

- the Operations, at line 18: the tasks constituting a service, e.g. *Database Provisioning*;
- the Operation Groups, at line 12: the considered service. Operation Groups define logical aggregations of simple operations, e.g. *Database Management*;

```
1  <!-- Copyright 2001, 2002, 2003 IBM Corp -->
2  <!-- -->
3  <!-- Service Definitions -->
4  <!-- -->
5
6
7  <xsd:complexType name="ServiceDefinitionType">
8    <xsd:complexContent>
9      <xsd:extension base="wsla:ServiceObjectType">
10       <xsd:sequence>
11         <xsd:element ref="wsla:Operation" minOccurs="0" maxOccurs="unbounded"/>
12         <xsd:element ref="wsla:OperationGroup" minOccurs="0" maxOccurs="unbounded"/>
13       </xsd:sequence>
14     </xsd:extension>
15   </xsd:complexContent>
16 </xsd:complexType>
17
18 <xsd:element name="Operation" type="wsla:OperationDescriptionType"/>
19 <xsd:complexType name="OperationDescriptionType">
20   <xsd:complexContent>
21     <xsd:extension base="wsla:ServiceObjectType">
22     </xsd:extension>
23   </xsd:complexContent>
24 </xsd:complexType>
25
26 <!-- Service Objects -->
27
28 <xsd:complexType name="ServiceObjectType" abstract="true">
29   <xsd:sequence>
30     <xsd:element ref="wsla:Schedule" minOccurs="0" maxOccurs="unbounded"/>
31     <xsd:element ref="wsla:SLAParameter" minOccurs="0" maxOccurs="unbounded"/>
32     <xsd:element ref="wsla:Metric" minOccurs="0" maxOccurs="unbounded"/>
33     <!-- V2003 -->
34     <xsd:element ref="wsla:Trigger" minOccurs="0" maxOccurs="unbounded"/>
35     <xsd:element ref="wsla:Constant" minOccurs="0" maxOccurs="unbounded"/>
36     <xsd:element ref="wsla:MetricMacroDefinition" minOccurs="0" maxOccurs="unbounded"/>
37     <xsd:element ref="wsla:MetricMacroExpansion" minOccurs="0" maxOccurs="unbounded"/>
38   </xsd:sequence>
39   <xsd:attribute name="name" type="xsd:string"/>
40 </xsd:complexType>
```

Listing 4.7: XML Schema for the SLA Service Definitions, from [7].

The first elements that constitutes the Service Definition are the Schedules.

```
1  <!-- Copyright 2001, 2002, 2003 IBM Corp -->
2  <xsd:element name="Schedule" type="wsla:ScheduleType"/>
3  <xsd:complexType name="ScheduleType">
4    <xsd:sequence>
5      <xsd:element name="Period" type="wsla:PeriodType"/>
6      <xsd:element name="Interval" type="wsla:IntervalType"/>
7    </xsd:sequence>
8    <xsd:attribute name="name" type="xsd:string"/>
9  </xsd:complexType>
```

Listing 4.8: XML Schema for the Service Schedule, from [7].

Listing 4.8 shows that *Schedules* are series of slots that divide a given period of time, (*Period*, line 5). These slots are named *Intervals*, see line 6.

```
1  <!-- Copyright 2001, 2002, 2003 IBM Corp -->
2  <Schedule name="Daily">
3    <Period>
4      <Start>2014-07-31T14:00:00.000-05:00</Start>
5      <End>2020-12-31T14:00:00.000-05:00</End>
6    </Period>
7    <Interval>
8      <Days>1</Days>
9    </Interval>
10 </Schedule>
```

Listing 4.9: XML example of a Service Schedule.

Listing 4.9 shows an example of a daily schedule, valid between 2014 an 2020. A metric that corresponds to this schedule would be evaluated each day.

The next elements that constitutes the Service Definition are the SLA Parameters.

```
1  <!-- Copyright 2001, 2002, 2003 IBM Corp -->
2  <xsd:element name="SLAParameter" type="wsla:SLAParameterType"/>
3
4  <xsd:complexType name="SLAParameterType">
5    <xsd:sequence>
6      <xsd:element name="Metric" type="xsd:string"/>
7      <xsd:element name="Communication" type="wsla:SLAParameterCommunicationType"
       minOccurs="0"/>
8    </xsd:sequence>
9    <xsd:attribute name="name" type="xsd:string"/>
10   <xsd:attribute name="type" type="wsla:Type"/>
11   <xsd:attribute name="unit" type="xsd:string"/>
12 </xsd:complexType>
```

Listing 4.10: XML Schema for the SLA Parameter, from [7].

Listing 4.10 shows that an SLA Parameter is build on:

- a metric: the metrics are the variables that allow to evaluate the systems performances. They are build by the system executing the SLA Management [4]. See Listing 4.11 for more details;
- an optional communication element: it defines the way the SLA parameter is transmitted to the parties;
- three attributes *name*, *type*, and *unit*. They characterize the SLA Parameter.

Finally, the last important elements of a Service Definition are the Metrics:

```
1  <!-- Copyright 2001, 2002, 2003 IBM Corp -->
2  <xsd:complexType name="MetricType">
3    <xsd:sequence>
4      <xsd:element name="Source" type="xsd:string"/>
5      <xsd:element name="MetricURI" type="xsd:anyURI" minOccurs="0"/>
6      <xsd:choice>
7        <xsd:element name="MeasurementDirective" type="wsla:MeasurementDirectiveType"/>
8        <xsd:element name="Function" type="wsla:FunctionType"/>
9        <!-- Version 2003 -->
10       <xsd:element name="MeasurementDirectiveVariable" type="wsla:MDVariableType"/>
11     </xsd:choice>
12   </xsd:sequence>
13   <xsd:attribute name="name" type="xsd:string"/>
14   <xsd:attribute name="type" type="wsla:Type"/>
15   <xsd:attribute name="unit" type="xsd:string"/>
16   <xsd:attribute name="counter" type="xsd:boolean" use="optional"/>
17 </xsd:complexType>
18
```

```
19 <xsd:element name="Metric" type="wsla:MetricType"/>
20
21 <!-- Measurement directives -->
22
23 <xsd:complexType name="MeasurementDirectiveType" abstract="true">
24   <xsd:sequence>
25     <xsd:element name="ReadingSchedule" type="xsd:string" minOccurs="0"/>
26   </xsd:sequence>
27   <xsd:attribute name="resultType" type="wsla:Type"/>
28 </xsd:complexType>
```

Listing 4.11: XML Schema for the SLA Metrics, from [7].

Listing 4.11 shows that Metrics comprises:

- a Source, at line 4, indicating the party responsible for providing the Metric;
- a choice between two constitutive elements, at line 6:

  either a Measurement Directive, defined at line 23;

  or a function of Metrics (e.g. a sum, a mean, etc. See the following Listings);
- four attributes *name*, *type*, *unit*, and *counter*. They characterize the properties of the Metric.

Measurement Directives are basic instructions given to the system to collect raw performance metrics. They are the formal elements allowing to define all the other metrics. The *MeasurementDirectiveType* comprises:

1. an element *ReadingSchedule*, line 25, that defines the schedule element ruling the evaluation of the raw metric.
2. necessary extensions: it must include other elements in the directives (e.g. a URI, a trigger, etc.).

Functions are used to combine metrics. They are named in the XML Schema, but they must be defined and specified in another document. Indeed, the meaning of each function must be provided unambiguously in a specification, in order to avoid misleading implementations from the developers. However, some functions are really clear.

```
1  <!-- Copyright 2001, 2002, 2003 IBM Corp -->
2  <xsd:complexType name="Plus">
3    <xsd:complexContent>
4      <xsd:extension base="wsla:FunctionType">
5        <xsd:sequence>
6          <xsd:element name="Operand" type="wsla:OperandType" minOccurs="2" maxOccurs="2"
       />
7        </xsd:sequence>
8      </xsd:extension>
9    </xsd:complexContent>
10 </xsd:complexType>
```

Listing 4.12: XML Schema for the Plus function, from [7].

Listing 4.12 shows the declaration of the Plus function. It is an extension of the *FunctionType*, that is abstract and only provides the name Function. See line 4. The extension adds a sequence of exactly two operand elements, at line 6. So you can sum a delay with another, for example.

Note that the authors do not provide the interpretation by the Information System of each function. In the previous example, Plus is a function named Plus, and it has 2 operands, but it is not related anywhere with a $+$. We provide a formal description of ambiguous notions, in our scope, in Section 5.5.

Keller and Ludwig define a more complex function: the Time Series Constructor. It builds Time Series, that are lists of values on a set of time intervals.

```
1  <!-- Copyright 2001, 2002, 2003 IBM Corp -->
2  <xsd:complexType name="TSConstructor">
3    <xsd:complexContent>
4      <xsd:extension base="wsla:FunctionType">
5        <xsd:sequence>
6          <xsd:element name="Schedule" type="xsd:string"/>
7          <xsd:choice>
8            <xsd:element name="Metric" type="xsd:string"/>
9            <xsd:element name="Function" type="wsla:FunctionType"/>
10         </xsd:choice>
11         <xsd:element name="Window" type="xsd:long" minOccurs="0"/>
12       </xsd:sequence>
13     </xsd:extension>
14   </xsd:complexContent>
15 </xsd:complexType>
```

Listing 4.13: XML Schema for the Time Series Constructor, from [7].

Listing 4.13 shows that this extension of the *FunctionType* adds:

- a Schedule, line 6, that gives the time frame allowing to build the time series. It indicates the start time, the end, and the nature of the time interval [7] (for example 2 min);
- a reference either to a function or a metric, that gives the values;
- a Window, line 11, that gives the number of intervals kept in consideration. For example, 8 means the last eight values. This may be used to define conditions on sliding windows or else on date to date windows.

A last example of proposed function is the *PercentageLessThanThreshold* function.

```
1  <!-- Copyright 2001, 2002, 2003 IBM Corp -->
2  <xsd:complexType name="PercentageLessThanThreshold">
3    <xsd:complexContent>
4      <xsd:extension base="wsla:FunctionType">
5        <xsd:sequence>
6          <xsd:choice>
7            <xsd:element name="Metric" type="xsd:string"/>
8            <xsd:element name="Function" type="wsla:FunctionType"/>
9          </xsd:choice>
10         <xsd:element name="Value" type="wsla:OperandType"/>
11       </xsd:sequence>
12     </xsd:extension>
13   </xsd:complexContent>
14 </xsd:complexType>
```

Listing 4.14: XML Schema for the *PercentageLessThanThreshold* function, from [7].

Listing 4.14 shows the contents of the *PercentageLessThanThreshold* function. It takes as first parameter a metric or a function whose result type is a Time Series. It takes as second parameter a value (the threshold), at line 10. The result of the function is the percentage of time intervals during which the considered value has been lower than the threshold value.

Figure 4.2 illustrates the sequence of elements defined in the XML Schema, that permits to build the Service Definition. It shows the relationships between the elements *Metrics*, *SLA Parameters* and *Schedules* and their construction. The SLA parameters comprises a *Communication* element that indicates who is in charge of the publication of the value of the SLA Parameter. It illustrates the extensions *Operations* and *Operation Groups*, that give the structure of the document.

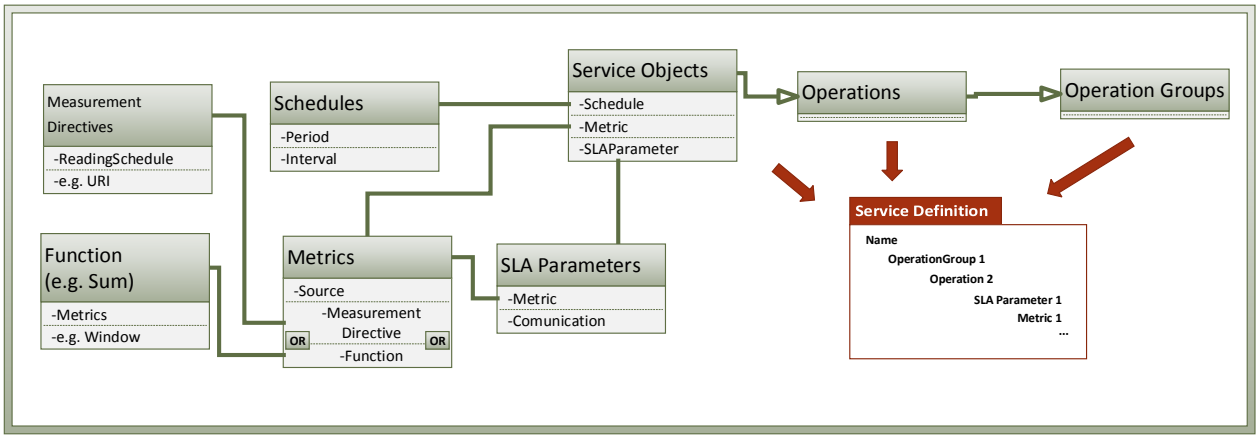


Figure 4.2: The elements of the Service Definition Schema.

4.2.3 The Obligations

Based on the Service definitions, the WSLA allows to build obligations for both parties. They can take two forms:

- a predicate, i.e. a boolean expression that must be True, e.g. *the latency of the application must be less than 10 seconds*. These predicates are the basis of the construction of Service Level Objectives (SLOs). See Section 4.2.4;
- conditional actions that depend on an SLO fulfillment: e.g. *if the latency is greater than 10 seconds, the Service Provider must Notify the Customer*. See Section 4.2.5 for the XML Schema of these actions.

```
1 <!-- Copyright 2001, 2002, 2003 IBM Corp -->
2 <xsd:complexType name="ObligationObjectType">
3   <xsd:sequence>
4     <xsd:element ref="wsla:Constant" minOccurs="0" maxOccurs="unbounded"/>
5     <xsd:element name="ServiceLevelObjective" type="wsla:ServiceLevelObjectiveType"
       minOccurs="0" maxOccurs="unbounded"/>
6     <xsd:element name="ActionGuarantee" type="wsla:ActionGuaranteeType" minOccurs="0"
       maxOccurs="unbounded"/>
7   </xsd:sequence>
8   <xsd:attribute name="name" type="xsd:string"/>
9 </xsd:complexType>
```

Listing 4.15: XML Schema for the Obligations, from [7].

Listing 4.15 shows the construction of the type *ObligationObjectType*. An Obligation Object can be composed of constants, SLOs, and conditional actions (*ActionGuarantee*).

Note that similarly as for the Service Definition, the WSLA framework allows to build Obligation Groups. This gives the possibility to follow the same structure in the Obligations part as in the Service Definitions part of the document.

4.2.4 The Service Level Objectives (SLOs)

The Service Level Objectives (SLOs) express conditions using the SLA Parameters defined in the Service Definitions. The expression of these conditions comprises predicates or combination

of predicates. Predicates are boolean evaluations of a comparison between an SLA Parameter and a defined value. See the construction of the predicate *Greater* on Listing 4.17. SLOs give a boolean result (whether they are met or not, e.g. *SLA Greater than value*).

```
1  <!-- Copyright 2001, 2002, 2003 IBM Corp -->
2  <xsd:complexType name="ServiceLevelObjectiveType">
3    <xsd:sequence>
4      <xsd:element name="Obliged" type="xsd:string" />
5      <xsd:element name="Validity" type="wsla:PeriodType" maxOccurs="unbounded"/>
6      <xsd:element name="Expression" type="wsla:LogicExpressionType"/>
7      <xsd:choice>
8        <xsd:element name="EvaluationEvent" type="wsla:EvaluationEventType"/>
9        <xsd:element name="Schedule" type="xsd:string"/>
10     </xsd:choice>
11   </xsd:sequence>
12   <xsd:attribute name="name" type="xsd:string"/>
13 </xsd:complexType>
```

Listing 4.16: XML Schema for the Service Level Objectives, from [7].

Listing 4.16 shows the contents of an SLO, and how to combine predicates to form the conditions in the SLOs. In the WSLA framework, a SLO consist in:

- the obliged party, at line 4, that is the party that has to meet the SLO;
- a validity period, during which the SLO is applicable;
- an expression, at line 6:
  The expression can be a predicate, or a logic combination of predicates, using Logic Operators. See Section 5.5.2;
- a time frame of evaluation, that defines when the SLO is evaluated. It can be periodic or when a new value is available.

Here is an example of construction of a predicate.

```
1  <!-- Copyright 2001, 2002, 2003 IBM Corp -->
2  <xsd:complexType name="Greater">
3    <xsd:complexContent>
4      <xsd:extension base="wsla:PredicateType">
5        <xsd:sequence>
6          <xsd:element name="SLAParameter" type="xsd:string"/>
7          <xsd:element name="Value" type="xsd:double"/>
8        </xsd:sequence>
9      </xsd:extension>
10   </xsd:complexContent>
11 </xsd:complexType>
```

Listing 4.17: XML Schema for the *Greater* predicate, from [7].

Listing 4.17 shows that *Greater* extends the type *PredicateType* with a sequence of two elements: first, a string reference to an SLA Parameter, at line 6; and second, a value of type double. An example of a predicate is *Delay Greater than 10 seconds*.

SLOs are then used for defining Action Guarantees.

4.2.5 The Actions Guarantees

The elements *ActionGuarantee* define what the parties must do when SLOs are violated.

They are similar in their contents to the SLOs.

```
1  <!-- Copyright 2001, 2002, 2003 IBM Corp -->
2  <xsd:complexType name="ActionGuaranteeType">
3    <xsd:sequence>
4      <xsd:element name="Obliged" type="xsd:string" maxOccurs="unbounded"/>
5      <xsd:element name="Expression" type="wsla:LogicExpressionType"/>
6      <xsd:choice>
7        <xsd:element name="EvaluationEvent" type="wsla:EvaluationEventType"/>
8        <xsd:element name="Schedule" type="xsd:string"/>
9      </xsd:choice>
10     <xsd:element ref="wsla:QualifiedAction" maxOccurs="unbounded"/>
11     <xsd:element name="ExecutionModality" type="wsla:ExecutionModalityType"/>
12   </xsd:sequence>
13   <xsd:attribute name="name" type="xsd:string"/>
14 </xsd:complexType>
```

Listing 4.18: XML for the Actions Guarantees, from [7].

Listing 4.18 shows the components of an Action Guarantee:

- the obliged party, that has to execute the Action (line 2);
- the expression that is a logic combination of predicates, using Logic Operators. It reflects the condition that triggers the Action (line 5);
- a time frame of evaluation, during which the condition of Action is evaluated. It can be periodic or when a new value is available.
- some possible elements *wsla:QualifiedAction*, at line 10, that further describe the concrete actions to be overcome;
- the execution modality of these actions (e.g. *always*).

In WSLA, the only actions provided in the XML Schema are Notification of Violation and Notification of Information.

```
1  <!-- Copyright 2001, 2002, 2003 IBM Corp -->
2  <!-- Action Invocation Types -->
3
4  <xsd:complexType name="ActionInvocationType">
5    <xsd:attribute name="actionName" type="xsd:string"/>
6  </xsd:complexType>
7
8  <xsd:complexType name="Notification">
9    <xsd:complexContent>
10     <xsd:extension base="wsla:ActionInvocationType">
11       <xsd:sequence>
12         <xsd:element name="NotificationType" type="wsla:NotificationType"/>
13         <xsd:element name="CausingGuarantee" type="xsd:string"/>
14         <xsd:element name="SLAParameter" type="wsla:StringList"/>
15         <!-- Convention: Can be either:
16           - "NONE"
17           - <enumerated list of SLAParameter IDs>
18           It is understood that only the relevant
19           subset of parameters are actually sent.
20         -->
21       </xsd:sequence>
22     </xsd:extension>
23   </xsd:complexContent>
24 </xsd:complexType>
25
26 <xsd:simpleType name="NotificationType">
27   <xsd:restriction base="xsd:string">
28     <xsd:enumeration value="Violation"/>
29     <xsd:enumeration value="Information"/>
30   </xsd:restriction>
31 </xsd:simpleType>
```

Listing 4.19: XML Schema for the Notification, from [7].

Listing 4.19 shows the construction of the Notification element. Notification extends the type *ActionInvocationType*, line 4 by adding:

- the Notification Type, at line 12, that can be a Notification of Violation, or of Information. See line 26;
- a *CausingGuarantee* period, referring to the SLO that has not been fulfilled;
- a list of SLA Parameters, triggering the Notification.

4.3 Specificities and Limits of the Use of WSLA for WSN Services

To a large extent, the WSLA framework address the IoT Operation problem. Indeed, the WSLA framework provides:

- A structure that fits our assumptions on the SLAs:

  They define (providers, customers, and supporting parties) when we have (WSN Operators, clients, and supporting parties);

  They define complex metrics as SLA parameters, based on functions of other metrics;

  They define SLOs, notifications, guarantees, time intervals and schedules, etc. that are needed in our scope;
- The way to write one's own non ambiguous basic metrics (measurement directives), provided by the Information System;
- The evaluation of parameters over time series, giving a lot of expressiveness;
- The opportunity to specify Actions, at the interface between the system control, and the SLA evaluation;

In a way, Keller and Ludwig proved that the WSLA framework is extensible by defining elements specific to their domains. For example they wrote extensions of the action type and the operation type for their use with WSDL and SOAP: *WSDLSOAPActionDescriptionType* and *WSDLSOAPOperationDescriptionType*.

The mathematical functions are only named in the document, and specified and implemented elsewhere. This fact also gives the liberty to define functions that suits our scope of IoT operation. For example, the function *RateOfChange* [7], defined in the WSLA framework, corresponds to a specific need in Web Services. We extend the model in the same way.

One major difference between the two domains remains in the multiplicity of the devices. The service definitions depends on which devices you are addressing, when, and where. This problem is explained in Section 2: in the Web Services domain, the information system is easy to define and reach. In the IoT domain, the devices and their capacities must appear in the SLAs.

We propose to extend the structure of the SLA document, given in the WSLA framework, by adding a fourth part, *Devices*, allowing us to include the missing information. This method address our problem without loosing the large amount of contributions that suits our scope in the WSLA framework.

Chapter 5

XML Schema : SLAs for the IoT Operation

This chapter provides the guidelines to build SLAs for the IoT Operation. It specifies the XML elements that constitutes the SLA documents. It explains in details the extensions of the WSLA model [7] that we propose in order to answer our problem (presented in Section 2).

5.1 Structure of the Document

5.1.1 The File Structure

Before the negotiation of the SLA, the client must provide:

1. the number and position of each device;
2. the list of features of each node.

This device specification allows to specifically address subsets of sensors with different Quality of Service requirements; for instance, a message delivery ratio for a given zone, or a maximum delay for a given application.

The resulting description of the list of devices is included in the main document (See Section 5.2). However, in cases of numerous devices, we may use a separate file and make reference to it in the main document.

The main document is written in XML. It must correspond to the XML Schema that we present here. Moreover, this specification gives the mathematical functions used to construct unambiguously the complex conditions of the SLA (See Section 5.5).

5.1.2 The SLA Components

The global structure of the SLAs for WSNs remains nearly the same as for WSLA. A WSN SLA document consists in four parts:

1. the Parties (See Section 4.2):
    - the Service Provider: in our case, the WSN Operator:
    - the Service Consumer: the clients;

- the Supporting Parties: hardware providers, companies responsible for the installation, the maintenance, or the monitoring of the WSN.

2. the Devices (new related to [7]): the list of considered devices and their properties. See Section 5.2;

3. the Service Definitions: where the SLA parameters and metrics are defined. See Section 5.3;

4. the Obligations: where the Service Level Objectives and corresponding Actions are defined. See Section 5.4.

```
1  <!-- Version 1.0 Gaillard, Copyright Orange Labs - INRIA, June 2014 -->
2  <!-- -->
3  <!-- Global WSN_SLA structure -->
4  <!-- -->
5  <xsd:complexType name="WSN_SLAType">
6    <xsd:sequence>
7      <xsd:element ref="wsn_sla:Parties"/>
8      <!-- Version 2014 -->
9      <xsd:element ref="wsn_sla:Devices"/>
10     <xsd:element name="ServiceDefinition" type="wsn_sla:ServiceDefinitionType"
       maxOccurs="unbounded"/>
11     <xsd:element name="Obligations" type="wsn_sla:ObligationsType"/>
12   </xsd:sequence>
13   <xsd:attribute name="name" type="xsd:string"/>
14 </xsd:complexType>
15
16 <xsd:element name="SLA" type="wsn_sla:WSN_SLAType"></xsd:element>
```

Listing 5.1: XML Schema for the SLA structure.

Listing 5.1 shows these four parties, including the new one, *Devices*, at line 9. Hence, in our context, a very simple example of SLA could be as follows:

```
1  <!-- Version 1.0 Gaillard, Copyright Orange Labs - INRIA, June 2014 -->
2  <wsn_sla:SLA name="IoT Service Level Agreement 1" >
3    <Parties>
4      <ServiceProvider name="WSN Operator">
5        ...
6      </ServiceProvider>
7      <ServiceConsumer name="Smart Parking Client 1">
8        <Contact>
9          ...
10     </ServiceConsumer>...
11   </Parties>
12   <Devices>
13     <Sensor>
14     ...
15   </Devices>
16   <ServiceDefinition name="Collect_Solution">
17     <OperationGroup name="Client Traffic 1">
18       <Operation name="Client Up Traffic">
19         <SLAParameter name="Packet Count">
20           ...
21   </ServiceDefinition>
22   <Obligations>
23     <ServiceLevelObjective name="Max Client Up Traffic">
24       ...
25     </ServiceLevelObjective>
26     <ActionGuarantee name="Notify Reached Max Client Up Traffic">
27       ...
28     </ActionGuarantee>
29   </Obligations>
30 </wsn_sla:SLA>
```

Listing 5.2: XML simple SLA for the IoT Operation context.

The Listing 5.2 shows that the first item of the SLA structure, <*Parties*>, is unambiguous for the IoT field:

- the Service Provider is the WSN Operator, line 4;
- the Service Consumer is the considered client, line 7;
- there is no Supporting Parties in the present example, but more complex SLAs may define some of them. For example, the SLA may include a company that provides independent performance measurements on the WSN. Besides, actors may be given the role of installing devices and changing faulty ones in case of service degradation.

Then, a collect service is defined at line 16 and corresponding Obligations at line 22.

For more details, see the complete example we provide in Appendix A.

5.2 The List of Devices

This Section describes the constitution of the list of Devices in the SLA document. We specify the sensor node elements, and their associated characteristics.

5.2.1 The WSN Nodes Specification

The Sensor Nodes

The necessary information given by the client about the considered sensor nodes is constituted of:

- An identifier (ID);
- The 3D localization of the node;
- A list of features: they correspond to the possible activities of the node. For example, sensors can have both humidity and temperature features.

Note that to-be-deployed sensors have to be declared at the contract negotiation. For example, a client may deploy its sensors step by step, having previously contracted an SLA for the whole set of devices.

```
1 <!-- Version 1.0 Gaillard, Copyright Orange Labs - INRIA, June 2014 -->
2 <xsd:complexType name="SensorType">
3   <xsd:sequence>
4     <xsd:element name="ID" type="xsd:string"/>
5     <xsd:element name="Position" type="wsn_sla:PositionType"/>
6     <xsd:element name="Feature" type="xsd:string" minOccurs="0" maxOccurs="unbounded"/>
7   </xsd:sequence>
8 </xsd:complexType>
```

Listing 5.3: XML Schema for the Sensor Nodes.

Listing 5.3 shows the XML elements of a node, one *ID*, one *Position*, and from zero to any *Features*.

The position follows a classical geographic coordinate system.

```
1 <!-- Version 1.0 Gaillard, Copyright Orange Labs - INRIA, June 2014 -->
2 <xsd:complexType name="PositionType">
3   <xsd:sequence>
4     <xsd:element name="Latitude" type="xsd:string"/>
5     <xsd:element name="Longitude" type="xsd:string"/>
```

```
6       <xsd:element name="Height" type="xsd:string"/>
7     </xsd:sequence>
8   </xsd:complexType>
```

Listing 5.4: XML Schema for the node localization.

Listing 5.4 shows the XML elements of a Position, a *Latitude*, a *Longitude*, and a *Height*.

Further versions of this specification may restrict the type of the text contents for the latitude, the longitude, and the height in order to force the clients to give enough precision in the values. For example, we could impose 5 decimals for the latitude values.

The considered node elements for an SLA are then listed in the Devices element, just before the Service Definition part.

```
1  <!-- Version 1.0 Gaillard, Copyright Orange Labs - INRIA, June 2014 -->
2  <xsd:element name="Devices" type="wsn_sla:DevicesType"/>
3  <xsd:complexType name="DevicesType">
4    <xsd:sequence>
5      <xsd:element name="Sensor" type="wsn_sla:SensorType" minOccurs="0" maxOccurs="
       unbounded"/>
6    </xsd:sequence>
7  </xsd:complexType>
```

Listing 5.5: XML Schema for the list of Devices.

Listing 5.5 shows the XML Schema of the list of Devices, constituted of sensors.

```
1  <!-- Version 1.0 Gaillard, Copyright Orange Labs - INRIA, June 2014 -->
2  <Devices>
3    <Sensor>
4      <ID>21</ID>
5      <Position>
6        <Latitude>45.15123N</Latitude>
7
8        <Longitude>5.70798E</Longitude>
9        <Height>1.26</Height>
10     </Position>
11     <Feature>GasMeter</Feature>
12     <Feature>GasAlarm</Feature>
13   </Sensor>
14   ...
```

Listing 5.6: XML example of a device.

Listing 5.6 shows the XML example of sensor 21. Sensor 21 has a 3D position, and two features, *GasMeter*, and *GasAlarm*, corresponding to a metering application, and a gas alarm application.

5.3 The Service Definition

The principal novelties brought by this document are in this section. Here are defined all the parameters on which the WSN Operator will concretely commit himself. We allow operations on subsets of the sensor nodes (defined in previous Section), in order to address all the requirements of the clients.

We define the sensor sets and ways to determine them in Section 5.3.1. Then, we introduce basic metrics obtained from the system, that allow to build the whole services. Finally, we describe unambiguously the tools, operators and functions that are used to build the complex parameters that specify the services.

5.3.1 The Sensor Sets

The goal of creating Sensor Sets is to provide the opportunity to define service parameters on a specific collection of sensors. Such sets can be addressed as a list of sensors, as a range of sensors,

or as a combination thereof (logical intersections and unions). For example, one could address the nodes:

- with feature *pollution*,
- that have the IDs in the list (24, 42, 107, or 1457),
- and that have their latitude in the range (10.04N, 10.05N)

Here is presented how to address a Sensor Set, through the XML element *SensorSet*.

```
1  <!-- Version 1.0 Gaillard, Copyright Orange Labs - INRIA, June 2014 -->
2  <xsd:element name="SensorSet" type="wsn_sla:SensorSetType"/>
3  <xsd:complexType name="SensorSetType">
4    <xsd:sequence>
5      <xsd:element name="Range" type="wsn_sla:NodeRangeType" minOccurs="0" maxOccurs="
       unbounded"/>
6      <xsd:element name="List" type="wsn_sla:NodeListType" minOccurs="0" maxOccurs="
       unbounded"/>
7    </xsd:sequence>
8    <xsd:attribute name="name" type="xsd:string"/>
9  </xsd:complexType>
10
11 <xsd:complexType name="NodeRangeType">
12   <xsd:sequence>
13     <xsd:element name="Min" type="xsd:string"/>
14     <xsd:element name="Max" type="xsd:string"/>
15   </xsd:sequence>
16   <xsd:attribute name="ParameterType" type="xsd:string"/>
17 </xsd:complexType>
18
19 <xsd:complexType name="NodeListType">
20   <xsd:simpleContent>
21     <xsd:extension base="wsn_sla:StringList">
22       <xsd:attribute name="ParameterType" type="xsd:string"/>
23     </xsd:extension>
24   </xsd:simpleContent>
25 </xsd:complexType>
```

Listing 5.7: XML Schema for the Sensor Set.

Listing 5.7 shows that Sensor Sets may have a list of two sorts of elements, line 4:

- Ranges, line 11: they define a lower bound and an upper bound for a given ordered node parameter, *ParameterType* (line 16), (e.g. Latitude). Note that if the node IDs constitute an ordered set, then you can define a range of IDs (e.g. *The IDs between 0 and 99*);
- Lists, line 19: they define a list of values of a given parameter, *ParameterType*. This allows to select the union of devices described by one of their parameters in the list. For example, you may have lists of IDs, lists of features, lists of sensor sets, etc.

Hence, the selection of sensors by a *SensorSet* selector is the intersection of the sets defined by its elements of type Range and List.

```
1  <!-- Version 1.0 Gaillard, Copyright Orange Labs - INRIA, June 2014 -->
2  <SensorSet name="First Sensor Set example : the Urban Pollution Monitoring">
3    <Range ParameterType="Latitude">
4      <Min>10.04N</Min>
5      <Max>10.05N</Max>
6    </Range>
7    <List ParameterType="Feature">
8      pollution
9    </List>
10   <List ParameterType="ID">24 42 107 1457</List>
11 </SensorSet>
```

Listing 5.8: XML example of a Sensor Set.

Listing 5.8 shows the expression of the Sensor Set given in example at the beginning of this Section 5.3.1.

Note that in the SLA, this definition of Sensor Sets is the key point that allows to define performance parameters concerning the broadcasting toward a given sensor set. For example, you may define the ratio of received configuration messages for all the sensors of a building.

Further versions of this document may specify other set properties if needed, e.g. *range of sensor firmware version, distance to a point of interest.*

5.3.2 The Basic Counter Metrics for the WSN SLAs

In these two subsections, we list the basic metrics dedicated to WSNs that we use in the SLA specification. The metrics result of the observation of the network performance of the WSN. The WSN Operator Information System provides them when a party needs to evaluate the SLA fulfillment.

We compose these basic metrics with a set of functions, in order to build the complex metrics and the corresponding SLA parameters.

This subsection specifies the message counters. Subsection 5.3.3 specifies the basic delay metrics. Then, Subsection 5.3.4 will describe available functions and the way to build complex metrics.

In the WSLA framework [7], Keller and Ludwig propose "Standard Measurement Directive Types". We adopt their formalism to specify the basic metrics in our scope.

Device Generated Message Counter

The Device Generated Message Counter (DGMC) metric corresponds to the number of messages generated by each sensor of a set. It is defined:

1. on a set of source sensor nodes of the client;
2. during a given period of time;
3. for a given feature.

Note that a sensor set may consist in a geographic area, a common feature, or even a single sensor. The DGMC could be calculated by timestamping the messages when generated. However, this is out of scope of this document.

```
1  <!-- Version 1.0 Gaillard, Copyright Orange Labs - INRIA, June 2014 -->
2  <!-- Device Generated Message Counter-->
3  <xsd:complexType name="DGMC">
4    <xsd:complexContent>
5      <xsd:extension base="wsn_sla:MeasurementDirectiveType">
6        <xsd:sequence>
7          <xsd:element name="SensorSetName" type="xsd:string"/>
8        </xsd:sequence>
9      </xsd:extension>
10   </xsd:complexContent>
11 </xsd:complexType>
```

Listing 5.9: XML Schema for the DGMC.

Listing 5.9 shows the XML Schema for the DGMC. It is an XML extension of the *Measurement-DirectiveType* [7], defined in Section 4.2.2, Listing 4.11, line 23.

It uses the *ReadingSchedule* element of the *MeasurementDirectiveType* as the given interval of time. It adds an element *SensorSetName*, at line 7, that indicates the set of sensors that is considered.

```
1 <!-- Version 1.0 Gaillard, Copyright Orange Labs - INRIA, June 2014 -->
2 <Metric name="SouthMinuteDGMC" type="integer" unit="messages">
3   <Source>WSNoperator</Source>
4   <MeasurementDirective xsi:type="DGMC" resultType="integer">
5     <ReadingSchedule>Minutely</ReadingSchedule>
6     <SensorSetName>South of Grenoble, France</SensorSetName>
7   </MeasurementDirective>
8 </Metric>
```

Listing 5.10: XML example of the DGMC.

Listing 5.10 shows an example of use of the DGMC. It specifies a Schedule, *Minutely*, at line 5, and a sensor set, *South of Grenoble, France*, at line 6. See the Schedule construction in Section 4.2.2, Listing 4.8.

The other Counters defined in the following have nearly the same XML Schema.

```
1  <!-- Version 1.0 Gaillard, Copyright Orange Labs - INRIA, June 2014 -->
2  <!-- Device Received Message Counter-->
3  <xsd:complexType name="DRMC">
4    ...
5  <!-- System Generated Message Counter-->
6  <xsd:complexType name="SGMC">
7    ...
8  <!-- System Received Message Counter-->
9  <xsd:complexType name="SRMC">
10   ...
```

Listing 5.11: Declaration of the counters in the XML Schema.

Listing 5.11 shows the declaration of these other three counters.

Device Received Message Counter

The Device Received Message Counter (DRMC) metric corresponds to the number of messages received by the Information System from each sensor of a set. It is defined:

1. on a set of source sensor nodes of the client;
2. during a given period of time;
3. for a given feature.

The DRMC could be calculated by counting the messages when received. However, this is out of scope of this document.

System Generated Message Counter

The System Generated Message Counter (SGMC) metric corresponds to the number of messages generated by the Information System toward each sensor of a set. It is defined:

1. on a set of source sensor nodes of the client;
2. during a given period of time;
3. for a given feature.

The SGMC could be calculated by counting the messages when generated. However, this is out of scope of this document.

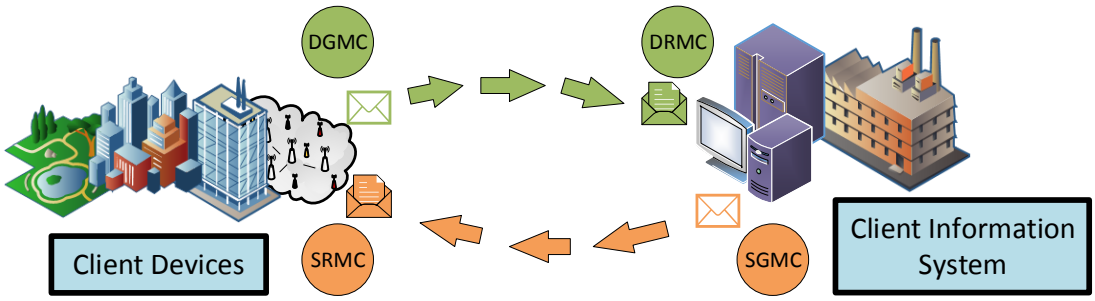


Figure 5.1: The Basic Counter Metrics.

System Received Message Counter

The System Received Message Counter (SRMC) metric corresponds to the number of messages received by each sensor of a set. It is defined:

1. on a set of source sensor nodes of the client;
2. during a given period of time;
3. for a given feature.

The SRMC could be calculated by timestamping the messages when generated. However, this is out of scope of this document.

Note that when addressed to multiple sensors, the messages are considered Multicast. When addressed to a single sensor, the messages are Unicast.

Figure 5.1 illustrates the four introduced counters:

- the DGMC and the SRMC are evaluated at the client devices;
- the DRMC and the SGMC are evaluated at the client Information System.

5.3.3 The Basic Delay Metrics for the WSN SLAs

This subsection deals with the basic evaluation of the delay of transmission of the messages. These two basic metrics allows to build end-to-end delay conditions.

Device to IS Delay

The Device to IS Delay (DID) corresponds to the list of the times between the generation of all the received messages from each sensor of a set of source sensor nodes and their reception by the Information System during a given period of time.

The DID could be calculated by timestamping the messages when generated, under the assumption that the devices are synchronized. However, this is out of scope of this document.

```
1 <!-- Version 1.0 Gaillard, Copyright Orange Labs - INRIA, June 2014 -->
2 <xsd:complexType name="DeviceToISDelay">
3   <xsd:complexContent>
4     <xsd:extension base="wsn_sla:MeasurementDirectiveType">
5       <xsd:sequence>
6         <xsd:element name="SensorSetName" type="xsd:string"/>
7       </xsd:sequence>
```

```
8       </xsd:extension>
9     </xsd:complexContent>
10  </xsd:complexType>
```

Listing 5.12: XML Schema for the Device to IS Delay(DID).

Listing 5.12 shows the XML Schema for the Device to IS Delay. It is the same extension as in Section 5.3.2.

IS To Device Delay

The IS To Device Delay (IDD) corresponds to the list of the times between the generation of all the received messages from the Information System and their reception by each sensor of a set of sensor nodes during a given period of time.

The IDD could be calculated by timestamping the messages when generated, under the assumption that the devices are synchronized. However, this is out of scope of this document.

```
1  <!-- Version 1.0 Gaillard, Copyright Orange Labs - INRIA, June 2014 -->
2  <xsd:complexType name="ISToDeviceDelay">
3     ...
```

Listing 5.13: XML Schema for the IS to Device Delay(IDD).

Listing 5.13 shows the declaration of the IS to Device Delay. It is the same extension as in Section 5.3.2.

5.3.4 Building SLA Parameters

In Section 4.2.2 we explained that functions may be defined in order to give the possibility to create more metric as functions of the basic metrics. We described the *TSConstructor* function in Listing 4.13, that allows to build Time Series of a given parameter.

In the IoT context, we need a function that allows to build a list of values of a parameter on a set of sensors.

In this subsection, we created the corresponding function for our field : the Sensor Series Constructor. It builds Sensor Series, that are lists of values on a set of sensors.

```
1  <!-- Version 1.0 Gaillard, Copyright Orange Labs - INRIA, June 2014 -->
2  <xsd:complexType name="SensorSeriesConstructor">
3    <xsd:complexContent>
4      <xsd:extension base="wsn_sla:FunctionType">
5        <xsd:sequence>
6          <xsd:element name="SensorSetName" type="xsd:string"/>
7          <xsd:choice>
8            <xsd:element name="Metric" type="xsd:string"/>
9            <xsd:element name="Function" type="wsn_sla:FunctionType"/>
10         </xsd:choice>
11       </xsd:sequence>
12     </xsd:extension>
13   </xsd:complexContent>
14 </xsd:complexType>
```

Listing 5.14: XML Schema for the Sensor Series Constructor.

Listing 5.14 shows that the XML Schema of the Sensor Series Constructor is very similar to the TSConstructor (Listing 4.13). The Sensor Series Constructor has a *SensorSetName* element, line 6. This indicates the considered sensor set (for example "Building42Temperature"). The values are calculated for a Metric or a Function on each device.

Typically, it is possible to build a Metric giving the list of results of a Measurement Directive applied on a entire sensor set.

```
1 <!-- Version 1.0 Gaillard, Copyright Orange Labs - INRIA, June 2014 -->
2 <Metric name="SouthMinutelyDGMCSensorSeries" type="SensorSeries" unit="messages">
3   <Source>WSNoperator</Source>
4   <Function xsi:type="SensorSeriesConstructor" resultType="SensorSeries">
5     <SensorSetName>South of Grenoble, France</SensorSetName>
6     <Metric>SouthMinuteDGMC</Metric>
7   </Function>
8 </Metric>
```

Listing 5.15: XML example of construction of a Sensor Series.

Listing 5.15 shows the construction of the Metric *SouthMinutelyDGMCSensorSeries*. It is the result of the function *SensorSeriesConstructor* applied on the metric *SouthMinuteDGMC*, previously defined in Listing 5.10. It gives the list of DGMC values of the sensors in the set *South of Grenoble, France*.

Other functions that have been named and defined in [7] are adopted here. We must specify them accordingly to the domain, respecting the specificities of the IoT operation. We provide a formal description of the most ambiguous notions, in our scope, in Section 5.5.

See the complete example of XML instance we provide in Appendix A.

5.4 The Obligations

Obligations have exactly the same schema for WSLA and for WSN SLA. They are either Service Level Objectives or Action Guarantees. See Section 4.2.3 for the specification of these elements.

Obligations are based on predicates defined in the same way as previously For example, we define the predicate that has:

- the comparator Greater;
- the SLA Parameter ”DGMC of node 56”;
- the value 7.

This predicate is: the DGMC of node 56 is greater than 7.

In WSLA, the authors provided the XML specification of the actions Notification of Violation and Notification of Information. See Listing 4.19 in Section 4.2.5.

We propose another Action type that changes the monitoring granularity for the basic metrics.

```
1  <!-- Version 1.0 Gaillard, Copyright Orange Labs - INRIA, June 2014 -->
2  <xsd:complexType name="SetMonitoringSchedule">
3    <xsd:complexContent>
4      <xsd:extension base="wsn_sla:ActionInvocationType">
5        <xsd:sequence>
6          <xsd:element name="MeasurementDirective" type="xsd:string"/>
7          <xsd:element name="NewSchedule" type="xsd:string"/>
8          <xsd:element name="CausingGuarantee" type="xsd:string"/>
9          <xsd:element name="SLAParameter" type="wsn_sla:StringList" minOccurs="0"/>
10       </xsd:sequence>
11     </xsd:extension>
12   </xsd:complexContent>
13 </xsd:complexType>>
```

Listing 5.16: XML Schema for the monitoring change Action.

Listing 5.16 shows the XML Schema of the action *SetMonitoringSchedule*. It extends the type *ActionInvocationType* by adding three elements:

- the *MeasurementDirective* element, at line 6: the targeted basic metric;

- the *NewSchedule* element: the new schedule to apply;
- the *CausingGuarantee* element: the SLO that has not been fulfilled.

With this type of action, a refinement in the monitoring policy of the WSN may appear in the SLA documents.

Here is an example of application.

```
<!-- Version 1.0 Gaillard, Copyright Orange Labs - INRIA, June 2014 -->
<QualifiedAction>
  <Party>WSNMeasurements</Party>
  <Action actionName="ChangeMonitoring" xsi:type="SetMonitoringSchedule">
    <MeasurementDirective>DeviceToISDelay</MeasurementDirective>
    <NewSchedule>Minutely</NewSchedule>
    <CausingGuarantee>SouthDailyDeviceToISDelay</CausingGuarantee>
  </Action>
</QualifiedAction>
```

Listing 5.17: XML example of a Change Monitoring Action.

Listing 5.17 shows the description of an Action. It tells what the supporting party *WSNMeasurements* must do if the SLO *SouthDailyDeviceToISDelay* is not fulfilled. The schedule of the Measurement Directive *DeviceToISDelay* must be changed to a *Minutely* Schedule.

See the complete example of XML instance we provide in Appendix A.

5.5 Mathematical and Formal Definitions

In this Section we describe all the tools used in the XML document of an SLA in the context of IoT applications. We give formal details of the element types and functions, used when building the SLA Parameters. We focus on the tools that may be ambiguous for the common reader. (Further versions may be more exhaustive than this one).

5.5.1 The Predicate Types

Greater

Greater is the boolean corresponding to common mathematical operator $>$. It takes two operands A and B, that belong to the same ordered set. Greater is the boolean $A>B$.

Less

Less is the boolean corresponding to common mathematical operator $<$. It takes two operands A and B, that belong to the same ordered set. Less is the boolean $A<B$.

The Others

- NewValue: this predicate means that *a new value of the considered parameter has been given by the System.* See [7] for more details.
- Violation: a SLO is violated if the corresponding predicate is not true.
- Equal
- GreaterEqual: Greater or Equal.
- LessEqual: Less or Equal.

- True: the boolean True.
- False

5.5.2 The Other Types

The Logic Expressions

Logic expressions are classical predicate operators. They take as parameters one or two predicates A and B. For example, *And* allows to compose predicate *A and B*.

```
<!-- Copyright 2001, 2002, 2003 IBM Corp -->
<xsd:complexType name="LogicExpressionType">
  <xsd:sequence>
    <xsd:choice>
      <xsd:element name="Predicate" type="wsn_sla:PredicateType"/>
      <xsd:element name="And" type="wsn_sla:BinaryLogicOperatorType"/>
      <xsd:element name="Or" type="wsn_sla:BinaryLogicOperatorType"/>
      <xsd:element name="Not" type="wsn_sla:UnaryLogicOperatorType"/>
      <xsd:element name="Implies" type="wsn_sla:BinaryLogicOperatorType"/>
    </xsd:choice>
  </xsd:sequence>
</xsd:complexType>
```

Listing 5.18: XML Schema of the Logic Expression Type, from [7].

Listing 3.1 shows the Logic Operators named in the WSLA framework. The operators *And*, *Or*, *Not*, and *Implies* are unambiguous.

The Others

- NodeRangeType: this type is defined and illustrated Section 5.3.1, in Listing 5.7.
- IntervalType: an interval of time.
- PeriodType: this Type is defined on the base of the RFC 3060.

```
<!-- Copyright 2001, 2002, 2003 IBM Corp -->
<xsd:complexType name="PeriodType">
  <xsd:choice>
    <xsd:sequence>
      <xsd:element name="Start" type="xsd:dateTime"/>
      <xsd:element name="End" type="xsd:dateTime"/>
    </xsd:sequence>
    <xsd:sequence><!-- Version 2003 -->
      <xsd:element name="ConditionTime" type="xsd:string"/>
      <xsd:element name="ConditionMonthOfYearMask" type="xsd:string"/>
      <xsd:element name="ConditionDayOfMonthMask" type="xsd:string"/>
      <xsd:element name="ConditionDayOfWeekMask" type="xsd:string"/>
      <xsd:element name="ConditionTimeOfDayMask" type="xsd:string"/>
      <xsd:element name="ConditionTimeZone" type="xsd:string"/>
      <xsd:element name="ConditionLocalOrUtcTime" type="xsd:string"/>
    </xsd:sequence>
  </xsd:choice>
</xsd:complexType>
```

Listing 5.19: XML Schema of PeriodType, from [7].

Listing 5.19 shows the simple and the complex format of a period. That corresponds to what is defined in the RFC.

- SensorType: sensors are specified in Section 5.2.1.
- PositionType: positions are described in Section 5.2.1, Listing 5.4.

- EvaluationEventType: this type defines when a SLO is evaluated. See [7] for more details.
- ExecutionModalityType: this type defines how often an Action is executed.

```
<!-- Copyright 2001, 2002, 2003 IBM Corp -->
<xsd:simpleType name="ExecutionModalityType">
  <xsd:restriction base="xsd:string">
    <xsd:enumeration value="Always"/>
    <xsd:enumeration value="OnEnteringCondition"/>
    <xsd:enumeration value="OnEnteringAndOnLeavingCondition"/>
    <!-- Version 2003 --><xsd:enumeration value="OnEveryEvaluation"/>
  </xsd:restriction>
</xsd:simpleType>
```

Listing 5.20: XML Schema of ExecutionModalityType, from [7].

Listing 5.20 shows the possible execution modalities of an action. An action can be executed each time the corresponding condition is *True* (*Always*). It can also be executed the first time the corresponding condition is *True*, i.e. (*OnEnteringCondition*). And at the beginning and at the end of the condition state, i.e. (*OnEnteringAndOnLeavingCondition*).

- OperandType:

```
<!-- Copyright 2001, 2002, 2003 IBM Corp -->
<xsd:complexType name="OperandType">
  <xsd:sequence>
    <xsd:choice>
      <xsd:element name="Metric" type="xsd:string"/>
      <xsd:element name="Function" type="wsn_sla:FunctionType"/>
      <!-- Version 2003-->
      <xsd:element name="LongScalar" type="xsd:long"/>
      <xsd:element name="FloatScalar" type="xsd:float"/>
      <xsd:element name="StringScalar" type="xsd:string"/>
      <xsd:element name="Constant" type="xsd:string"/>
    </xsd:choice>
  </xsd:sequence>
</xsd:complexType>
```

Listing 5.21: XML Schema of OperandType, from [7].

Listing 5.21 shows the list of possible Operands:

  - Metrics;
  - Functions;
  - Scalars (integers, floats, strings);
  - Predefined constants (e.g. a packet size, the payload of an applicative message, etc.).

5.5.3 The Standard Functions

Plus

Plus is the Function corresponding to common mathematical operator +. It takes two operands A and B, that have the same type. Plus is the function *A+B*.

Mean

Mean is the Function corresponding to common mathematical function mean. It takes a list of K numerical values *A, B, ..., N*. It has the result *(A+B+...+N)/K*.

The Others

- QConstructor: this leaves the flexibility to obtain a list of values independently of a periodic interval of time. See [7] for more details.
- TSConstructor: specified in Section 4.2.2, Listing 4.13.
- SensorSeriesConstructor: specified in Section 5.3.4, Listing 5.14.
- Size: *Size* gives the number of elements of a Queue, a Time Series, or a Sensor Series.
- TSSelect: this allows to select a specific element in a Time Series
- Plus: specified in Section 4.2.2, Listing 4.12.
- Minus, Divide, Times, Median, Mode, Round, Max, Sum:
  These functions are either unambiguous or well defined in [7].
- ValueOccurs: the occurrences of a value in a list.
- PercentageGreaterThanThreshold: specified in Section 4.2.2, Listing 4.14.
- PercentageLessThanThreshold, NumberGreaterThanThreshold, and NumberLessThanThreshold have similar definitions as *PercentageGreaterThanThreshold*.

Chapter 6

XML Instance : SLA for the Telemetering Use Case

6.1 Description of the Use Case

In order to illustrate by an example the XML Schema, we give the following complete SLA document. It is the result of the negotiation between three parties:

- the WSN Operator *WSNOperator*, in charge of the network infrastructure, and the handling of the messages;
- the client, a gas company, named GasCompany23, which owns sensors generating messages related to gas usages;
- *WSNMeasurements*, a supporting party that is in charge of the delays measurements.

The SLA specifies the traffic bounds that the client must not exceed, and the expected correct transmission ratio, and delay, expected by the client applications.

6.2 Scope and Contents of the SLA

The SLA specifies the terms corresponding to the two services needed by GasCompany23:

- the telemetering service, including:
  - the collection of gas indexes from gas meters;
  - the transmission of configuration messages from the client Information System.
- the collection of gas alarm messages from a specific building.

The first service (metering) is divided into two Operation Groups:

- the SLA parameters and metrics that correspond to the collection of the messages from the devices to the client Information System;
- the SLA parameters and metrics that correspond to the transmission of the messages from the client Information System to the devices.

The second service (gas alarms) is only defined by the collection of messages. For simplicity, we don't define clauses for the configuration of these sensors.

For the illustration, we defined twelve sensors and two sensor sets. Note that a real SLA would consider a city scale, for example, with hundreds of sensors. Some gas meters can send alarm messages, others can't.The sensors are situated in the south of Grenoble, France. A first set covers the zone of interest for GasCompany23: the south of the city. A second set identifies the sensors of a building (not existing in reality) by their list of ID.

Each SLA Parameter is associated with a Service Level Objective (SLO), and the corresponding action in case the SLO is not fulfilled.

But let's go into details of the document:

6.3 The Telemetering Message Collection

Firstly, we have three SLA parameters that are used to evaluate if the client generate more traffic than permitted:

- "South Daily Input Rates Overflow Percentage": this is the percentage of sensors that have generated more than 5 messages during the day. It is based on the metric "SouthDaily-DGMCOverflow".

- "South Hourly Input Rates Overflow Percentage": this is the percentage of sensors that have generated more than 2 messages in one hour during the day. It is based on the metric "SouthHourlyDGMCOverflow".

- "South Minutely Input Rates Overflow Percentage": this is the percentage of sensors that have generated more than 1 message in 2 minutes during the day. It is based on the metric "SouthMinutelyDGMCOverflow".

Secondly, we have four SLA parameters that force the WSN Operator to answer the application requirements, in terms of delay and loss rate of the messages:

1. *South Daily Global Collection Percentage*: this is the percentage of sensors from which the client Information System has received more than 50% of the generated messages during a day. It is based on the metric *SouthDailyCollectionPercentageMetric*.

2. *South Consecutive Days Global Collection Counter*: this is the percentage of sensors from which the client Information System has received less than 50% of the generated messages during each day, 4 consecutive days. It is based on the metric *SouthConsecutiveDaysCollectionCounterPercentage*.

3. *South Sliding Monthly Global Collection Counter*: this is the percentage of sensors from which the client Information System has received less than 50% of the generated messages during a day, more than 5 days in a sliding month (30 days). It is based on the metric *SouthSlidingMonthlyCollectionCounterPercentage*.

4. *South Daily Global Collection In Time Percentage*: this is the percentage of sensors from which the client generated messages suffered a collection delay of more than 1 hour. It is evaluated each hour. It is based on the metric *CollectionInTimePercentageMetric*.

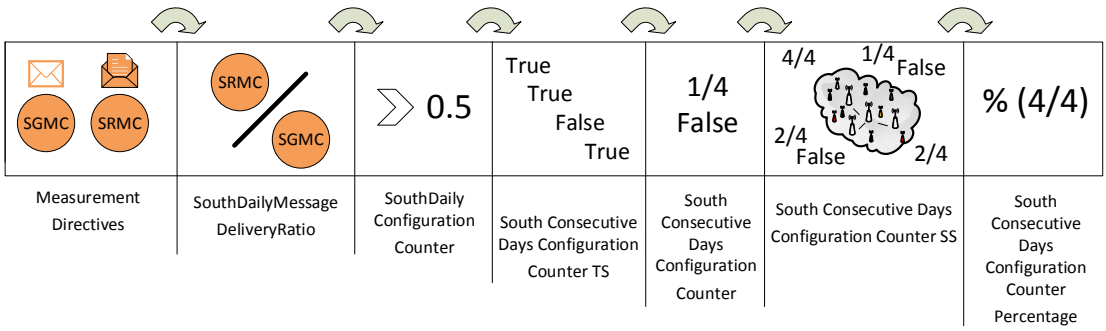


Figure 6.1: The Telemetering Configuration Sequence of Metrics.

6.4 The Telemetering Configuration Messages

This part concerns the traffic flows from the client Information System toward its devices.

The SLA Parameters and metrics are the same as previously. Their names change (configuration instead of collection) and the measurement directives too.

Let's detail as an example the construction of the SLA Parameter *South Consecutive Days Global Configuration Counter*. Figure 6.1 illustrates the sequence of construction of the corresponding complex metric *SouthConsecutiveDaysConfigurationCounterPercentage*.

1. Everything starts with two measurement directives given by the WSNoperator Information System: the System Generated Message Counter (SGMC) which is the number of messages generated by the client Information System, and the System Received Message Counter (SRMC), which is the number of these messages that have been received by the devices.

   We build the two corresponding metrics, giving the SGMC and SRMC, daily, of each sensor concerned in the south of Grenoble: *SouthDailySRMC* and *SouthDailySGMC*.

2. We build the ratio of them (SRMC/SGMC) in the metric *SouthDailyMessageDeliveryRatio*.

3. We build a boolean metric, *SouthDailyConfigurationCounter*, that is true if the value of the metric *SouthDailyMessageDeliveryRatio* is greater than 0.5 (more than half of the messages has been received in each day by each sensor).

4. We create Time Series of these boolean values, keeping the last 4 days values, in the *SouthConsecutiveDaysConfigurationCounterTS* metric. So we have a list of 4 boolean values for each sensor (e.g. {True, True, False, True}).

5. We count the False values of the Time Series in the *SouthConsecutiveDaysConfigurationCounter* metric. Previous example would result in 1 False value.

6. We build the Sensor Series corresponding to the considered Sensor Set (the gas meters of the south of Grenoble). So the metric *SouthConsecutiveDaysConfigurationCounterSS* lists the number of "False" days for all the Sensor Set during last 4 days.

7. Finally, the metric *SouthConsecutiveDaysConfigurationCounterPercentage* gives the percentage of sensors that have more than 3 (i.e 4) False in a 4 days period. And this metric is used in the final SLA Parameter.

6.5 The Gas Alarm Message Collection

For simplicity, the contracted service is restrained to the collection of alarm data. We have two SLA Parameters :

- Building46 Daily Global Collection In Time Percentage Metric: It is the same as in Section6.3. It is the percentage of sensors from which the client generated messages suffered a collection delay of more than 10 minutes. It is evaluated each 2 minutes. It is based on the metric *CollectionInTimePercentageMetric*.
- Building46 Collection In Time Degraded Months: From the first SLA paramater, we build a second complex metric that counts the numbers of days when half of the sensors suffered delays. We then evaluate on fixed date to date months if the number of degraded days is upper than 5.

Chapter 7

Conclusion

In this document we have addressed the problem of having Operated IoT solutions. A WSN Operator can specify a QoS contract with several clients, that takes the form of a Service Level Agreement. SLAs force the operator to fulfill the requirements specified by the clients and negotiated according to its business model. It express the limits of the guarantees each party provides, and the actions that must be executed if Service Level Objectives are violated.

In this purpose, we extended the WSLA framework to make it useful in our scope. We defined sensor sets (and the way to address a given list of devices). We defined basic metrics that we consider given by the network operation center: message counters and the way to use them, particularly on a given sensor set. We also defined basic delay metrics for the received messages. These may be evaluated by supporting parties analyzing the network performance, or by the operator itself.

We provided a use case build around a gas utility company, with a telemetering service and a gas alarm collection service. We showed in this example how one can build the SLA Parameters that define in details the offered services. Starting from basic metrics, we build complex conditions that cover a spatial and timely distribution of the performance values of the devices. The XML code of this instance is provided in Appendix A.

We specified all the functions and mathematical objects we use, in a non ambiguous way. This version of the document constitutes a guideline for real companies to build Service Level Agreements. Further improvements in future versions will complete this first proposal.

Appendix A

Main SLA Document

This Appendix presents the whole SLA document addressing the use case described in Section 6. We have divided the code into logical parts in order to describe them and refer to them more easily.

So, it begins here:

```
<?xml version="1.0" encoding="UTF-8"?>

<!--
SLA for the Telemetering Use Case
Authors: Guillaume Gaillard (Contact), guillaume.gaillard@orange.com
       Dominique Barthel
       Fabrice Theoleyre
       Fabrice Valois
    Copyright Orange Labs - INRIA, June 2014
Date: June 30, 2014
-->
```

Listing A.1: XML Instance Header for the telemetering use case.

Listing A.1 shows the header of the document. It includes the authors, the version of the document, the rights, etc.

A.1 The Definition of the Parties

This section corresponds to the instantiation of the elements described in Section 4.2.1.

```
<!-- -->
<!-- SLA for the Telemetering Use Case -->
<!-- -->

<SLA xmlns="http://www.orange.com/wsn_sla"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.orange.com/wsn_sla SLA_WSN.xsd"
name="SLA for GasCompany23" >

<!-- -->
<!-- Party Definitions -->
<!-- -->
<Parties>

  <!-- Signatory Parties -->
  <ServiceProvider name="WSNOperator">
    <Contact>
      <Street>28, Rue des Pins</Street>
      <City>Paris, France</City>
    </Contact>
```

```
21      <Action>Handle the messages of the clients</Action>
22    </ServiceProvider>
23
24    <ServiceConsumer name="GasCompany23">
25      <Contact>
26        <Street>688 Avenue de la Republique</Street>
27        <City>Lyon, France</City>
28      </Contact>
29      <Action>Generate messages for the GasMetering application</Action>
30    </ServiceConsumer>
31
32    <!-- Supporting Parties -->
33    <SupportingParty name="WSNMeasurements">
34      <Contact>
35        <Street>184 Avenue de l'Europe</Street>
36        <City>Grenoble, France</City>
37      </Contact>
38      <Action>Measure the performance of the WSN Network</Action>
39      <Sponsor>WSNOperator</Sponsor>
40      <Role>MeasurementService</Role>
41    </SupportingParty>
42
43
44  </Parties>
```

Listing A.2: XML Instance : The Definition of the Parties.

Listing A.2 shows the definition of the parties. The WSN Operator is described at line 16. The client is described at line 24. The Supporting Party is described at line 33.

A.2 The List of Devices

This section corresponds to the instantiation of the list of Devices described in Section 5.2.

```
1   <!-- -->
2   <!-- Devices -->
3   <!-- -->
4
5
6   <Devices>
7     <Sensor>
8       <ID>21</ID>
9       <Position>
10        <Latitude>45.15123N</Latitude>
11        <Longitude>5.70798E</Longitude>
12        <Height>1.26</Height>
13      </Position>
14      <Feature>GasMeter</Feature>
15      <Feature>GasAlarm</Feature>
16    </Sensor>
17
18    <Sensor>
19      <ID>22</ID>
20      <Position>
21        <Latitude>45.15245N</Latitude>
22        <Longitude>5.71342E</Longitude>
23        <Height>1.50</Height>
24      </Position>
25      <Feature>GasMeter</Feature>
26    </Sensor>
27
28    <Sensor>
29      <ID>23</ID>
30      <Position>
31        <Latitude>45.15367N</Latitude>
32        <Longitude>5.70587E</Longitude>
33        <Height>1.23</Height>
34      </Position>
35      <Feature>GasAlarm</Feature>
```

```
36    </Sensor>
37
38    <Sensor>
39      <ID>24</ID>
40      <Position>
41        <Latitude>45.15429N</Latitude>
42        <Longitude>5.70995E</Longitude>
43        <Height>0.57</Height>
44      </Position>
45      <Feature>GasMeter</Feature>
46      <Feature>GasAlarm</Feature>
47    </Sensor>
48
49    <Sensor>
50      <ID>25</ID>
51      <Position>
52        <Latitude>45.15578N</Latitude>
53        <Longitude>5.70323E</Longitude>
54        <Height>1.34</Height>
55      </Position>
56      <Feature>GasMeter</Feature>
57    </Sensor>
58
59    <Sensor>
60      <ID>26</ID>
61      <Position>
62        <Latitude>45.15622N</Latitude>
63        <Longitude>5.70798E</Longitude>
64        <Height>-2.33</Height>
65      </Position>
66      <Feature>GasMeter</Feature>
67    </Sensor>
68
69    <Sensor>
70      <ID>27</ID>
71      <Position>
72        <Latitude>45.15789N</Latitude>
73        <Longitude>5.70222E</Longitude>
74        <Height>3.09</Height>
75      </Position>
76      <Feature>GasMeter</Feature>
77      <Feature>GasAlarm</Feature>
78    </Sensor>
79
80    <Sensor>
81      <ID>28</ID>
82      <Position>
83        <Latitude>45.15832N</Latitude>
84        <Longitude>5.70876E</Longitude>
85        <Height>5.87</Height>
86      </Position>
87      <Feature>GasAlarm</Feature>
88    </Sensor>
89
90    <Sensor>
91      <ID>75</ID>
92      <Position>
93        <Latitude>45.15923N</Latitude>
94        <Longitude>5.71242E</Longitude>
95        <Height>-1.07</Height>
96      </Position>
97      <Feature>GasMeter</Feature>
98      <Feature>GasAlarm</Feature>
99    </Sensor>
100
101   <Sensor>
102     <ID>AZ24</ID>
103     <Position>
104       <Latitude>45.16432N</Latitude>
105       <Longitude>5.72444E</Longitude>
106       <Height>-5.32</Height>
107     </Position>
108     <Feature>GasAlarm</Feature>
```

```
109   </Sensor>
110
111   <Sensor>
112     <ID>T32</ID>
113     <Position>
114       <Latitude>45.16523N</Latitude>
115       <Longitude>5.72327E</Longitude>
116       <Height>9.23</Height>
117     </Position>
118     <Feature>GasAlarm</Feature>
119   </Sensor>
120
121   <Sensor>
122     <ID>452</ID>
123     <Position>
124       <Latitude>45.16729N</Latitude>
125       <Longitude>5.72761E</Longitude>
126       <Height>10</Height>
127     </Position>
128     <Feature>GasMeter</Feature>
129     <Feature>GasAlarm</Feature>
130   </Sensor>
131 </Devices>
```

Listing A.3: XML Code for the List of Devices.

Listing A.3 shows the list of sensors that have been declared by the client before the negotiation of the SLA. They are described with their IDs, positions, and features. Here, two features are presented: *GasMeter* for the meters and *GasAlarm* for the devices that are capable of sending alarms.

A.3 The Service Definition

This section corresponds to the instantiation of the Service Definition as described in Sections 4.2.2, 5.3, and 6.3.

```
1  <!-- -->
2  <!-- Service Definitions -->
3  <!-- -->
4
5  <ServiceDefinition name="Daily Metering">
6
7    <Schedule name="Daily">
8      <Period>
9        <Start>2014-07-31T14:00:00.000-05:00</Start>
10       <End>2020-12-31T14:00:00.000-05:00</End>
11     </Period>
12     <Interval>
13       <Days>1</Days>
14     </Interval>
15   </Schedule>
16   <Schedule name="Hourly">
17     <Period>
18       <Start>2014-07-31T14:00:00.000-05:00</Start>
19       <End>2020-12-31T14:00:00.000-05:00</End>
20     </Period>
21     <Interval>
22       <Hours>1</Hours>
23     </Interval>
24   </Schedule>
25   <Schedule name="Minutely">
26     <Period>
27       <Start>2014-07-31T14:00:00.000-05:00</Start>
28       <End>2020-12-31T14:00:00.000-05:00</End>
29     </Period>
30     <Interval>
31       <Minutes>2</Minutes>
32     </Interval>
```

```
33  </Schedule>
34
35  <SensorSet name="South of Grenoble, France">
36    <Range ParameterType="Latitude">
37      <Min>45.15N</Min>
38      <Max>45.18N</Max>
39    </Range>
40    <Range ParameterType="Longitude">
41      <Min>5.70E</Min>
42      <Max>5.76E</Max>
43    </Range>
44    <List ParameterType="Feature">
45      GasMeter
46    </List>
47  </SensorSet>
48
49  <OperationGroup name="Gas Daily Metering Collection">
50
51    <Metric name="SouthDailyDGMCOverflow" type="float" unit="Percentage">
52      <Source>WSNoperator</Source>
53      <Function xsi:type="PercentageGreaterThanThreshold" resultType="float">
54        <Metric>SouthDailyDGMCSensorSeries</Metric>
55        <Value>
56          <LongScalar>5</LongScalar>
57          <!-- max 5 mess a day -->
58        </Value>
59      </Function>
60    </Metric>
61    <Metric name="SouthHourlyDGMCOverflow" type="float" unit="Percentage">
62      <Source>WSNoperator</Source>
63      <Function xsi:type="PercentageGreaterThanThreshold" resultType="float">
64        <Metric>SouthHourlyDGMCSensorSeries</Metric>
65        <Value>
66          <LongScalar>2</LongScalar>
67          <!-- max 2 mess by hour -->
68        </Value>
69      </Function>
70    </Metric>
71    <Metric name="SouthMinutelyDGMCOverflow" type="float" unit="Percentage">
72      <Source>WSNoperator</Source>
73      <Function xsi:type="PercentageGreaterThanThreshold" resultType="float">
74        <Metric>SouthMinutelyDGMCSensorSeries</Metric>
75        <Value>
76          <LongScalar>1</LongScalar>
77          <!-- max 1 mess a min -->
78        </Value>
79      </Function>
80    </Metric>
81    <Metric name="SouthDailyDGMCSensorSeries" type="SensorSeries" unit="messages">
82      <Source>WSNoperator</Source>
83      <Function xsi:type="SensorSeriesConstructor" resultType="SensorSeries">
84        <SensorSetName>South of Grenoble, France</SensorSetName>
85        <Metric>SouthDailyDGMC</Metric>
86      </Function>
87    </Metric>
88    <Metric name="SouthHourlyDGMCSensorSeries" type="SensorSeries" unit="messages">
89      <Source>WSNoperator</Source>
90      <Function xsi:type="SensorSeriesConstructor" resultType="SensorSeries">
91        <SensorSetName>South of Grenoble, France</SensorSetName>
92        <Metric>SouthHourlyDGMC</Metric>
93      </Function>
94    </Metric>
95    <Metric name="SouthMinutelyDGMCSensorSeries" type="SensorSeries" unit="messages">
96      <Source>WSNoperator</Source>
97      <Function xsi:type="SensorSeriesConstructor" resultType="SensorSeries">
98        <SensorSetName>South of Grenoble, France</SensorSetName>
99        <Metric>SouthMinuteDGMC</Metric>
100     </Function>
101   </Metric>
102   <Metric name="SouthDailyDGMC" type="integer" unit="messages">
103     <Source>WSNoperator</Source>
104     <MeasurementDirective xsi:type="DGMC" resultType="integer">
105       <ReadingSchedule>Daily</ReadingSchedule>
```

```
106         <SensorSetName>South of Grenoble, France</SensorSetName>
107       </MeasurementDirective>
108     </Metric>
109     <Metric name="SouthHourlyDGMC" type="integer" unit="messages">
110       <Source>WSNoperator</Source>
111       <MeasurementDirective xsi:type="DGMC" resultType="integer">
112         <ReadingSchedule>Hourly</ReadingSchedule>
113         <SensorSetName>South of Grenoble, France</SensorSetName>
114       </MeasurementDirective>
115     </Metric>
116     <Metric name="SouthMinuteDGMC" type="integer" unit="messages">
117       <Source>WSNoperator</Source>
118       <MeasurementDirective xsi:type="DGMC" resultType="integer">
119         <ReadingSchedule>Minutely</ReadingSchedule>
120         <SensorSetName>South of Grenoble, France</SensorSetName>
121       </MeasurementDirective>
122     </Metric>
123
124     <Metric name="SouthConsecutiveDaysCollectionCounterPercentage" type="float" unit="
        Percentage">
125       <Source>WSNoperator</Source>
126       <Function xsi:type="PercentageGreaterThanThreshold" resultType="float">
127         <Metric>SouthConsecutiveDaysCollectionCounterSS</Metric>
128         <Value>
129           <LongScalar>4</LongScalar>
130           <!-- percentage of sensors with 4 consecutive degraded days -->
131         </Value>
132       </Function>
133     </Metric>
134     <Metric name="SouthConsecutiveDaysCollectionCounterSS" type="SensorSeries" unit="
        Degradation Ocurrences">
135       <Source>WSNoperator</Source>
136       <Function xsi:type="SensorSeriesConstructor" resultType="SensorSeries">
137         <SensorSetName>South of Grenoble, France</SensorSetName>
138         <Metric>SouthConsecutiveDaysCollectionCounter</Metric>
139       </Function>
140     </Metric>
141     <Metric name="SouthConsecutiveDaysCollectionCounter" type="float" unit="Days">
142       <Source>WSNoperator</Source>
143       <Function xsi:type="ValueOccurs" resultType="integer">
144         <Metric>SouthConsecutiveDaysCollectionCounterTS</Metric>
145         <Value>
146           <Boolean>False</Boolean>
147           <!-- how many false in last four days for each sensor -->
148         </Value>
149       </Function>
150     </Metric>
151     <Metric name="SouthConsecutiveDaysCollectionCounterTS" type="TS" unit="Percentage">
152       <Source>WSNoperator</Source>
153       <Function xsi:type="TSConstructor" resultType="TS">
154         <Schedule>Daily</Schedule>
155         <Metric>SouthDailyCollectionCounter</Metric>
156         <Window>4</Window>
157         <!-- last four days -->
158       </Function>
159     </Metric>
160     <Metric name="SouthSlidingMonthlyCollectionCounterPercentage" type="float" unit="
        Percentage">
161       <Source>WSNoperator</Source>
162       <Function xsi:type="PercentageGreaterThanThreshold" resultType="float">
163         <Metric>SouthSlidingMonthlyCollectionCounterSS</Metric>
164         <Value>
165           <LongScalar>5</LongScalar>
166           <!-- percentage of sensors with more than 5 degraded in a sliding month -->
167         </Value>
168       </Function>
169     </Metric>
170     <Metric name="SouthSlidingMonthlyCollectionCounterSS" type="SensorSeries" unit="
        Degradation Ocurrences">
171       <Source>WSNoperator</Source>
172       <Function xsi:type="SensorSeriesConstructor" resultType="SensorSeries">
173         <SensorSetName>South of Grenoble, France</SensorSetName>
174         <Metric>SouthSlidingMonthlyCollectionCounter</Metric>
```

```
175         </Function>
176       </Metric>
177       <Metric name="SouthSlidingMonthlyCollectionCounter" type="float" unit="Days">
178         <Source>WSNoperator</Source>
179         <Function xsi:type="ValueOccurs" resultType="integer">
180           <Metric>SouthSlidingMonthlyCollectionCounterTS</Metric>
181           <Value>
182             <Boolean>False</Boolean>
183             <!-- how many false in a month for each sensor -->
184           </Value>
185         </Function>
186       </Metric>
187       <Metric name="SouthSlidingMonthlyCollectionCounterTS" type="TS" unit="Percentage">
188         <Source>WSNoperator</Source>
189         <Function xsi:type="TSConstructor" resultType="TS">
190           <Schedule>Daily</Schedule>
191           <Metric>SouthDailyCollectionCounter</Metric>
192           <Window>30</Window>
193         </Function>
194       </Metric>
195       <Metric name="SouthDailyCollectionCounter" type="boolean" unit="Percentage">
196         <Source>WSNoperator</Source>
197         <Function xsi:type="ValueGreaterThanThreshold" resultType="boolean">
198           <Metric>SouthDailyMessageDeliveryRatio</Metric>
199           <Value>
200             <FloatScalar>0.5</FloatScalar>
201             <!-- One of two messages -->
202           </Value>
203         </Function>
204       </Metric>
205
206       <Metric name="SouthDailyCollectionPercentageMetric" type="float" unit="Percentage">
207         <Source>WSNoperator</Source>
208         <Function xsi:type="PercentageGreaterThanThreshold" resultType="float">
209           <Metric>SouthDailyMessageDeliveryRatioSS</Metric>
210           <Value>
211             <FloatScalar>0.5</FloatScalar>
212             <!-- One of two messages -->
213           </Value>
214         </Function>
215       </Metric>
216       <Metric name="SouthDailyMessageDeliveryRatioSS" type="SensorSeries" unit="
        Percentage">
217         <Source>WSNoperator</Source>
218         <Function xsi:type="SensorSeriesConstructor" resultType="SensorSeries">
219           <SensorSetName>South of Grenoble, France</SensorSetName>
220           <Metric>SouthDailyMessageDeliveryRatio</Metric>
221         </Function>
222       </Metric>
223       <Metric name="SouthDailyMessageDeliveryRatio" type="float" unit="Percentage">
224         <Source>WSNoperator</Source>
225         <Function xsi:type="Divide" resultType="float">
226           <Operand>
227             <Metric>SouthDailyDRMC</Metric>
228           </Operand>
229           <Operand>
230             <Metric>SouthDailyDGMC</Metric>
231           </Operand>
232         </Function>
233       </Metric>
234       <Metric name="SouthDailyDRMC" type="integer" unit="messages">
235         <Source>WSNoperator</Source>
236         <MeasurementDirective xsi:type="DRMC" resultType="integer">
237           <ReadingSchedule>Daily</ReadingSchedule>
238           <SensorSetName>South of Grenoble, France</SensorSetName>
239         </MeasurementDirective>
240       </Metric>
241
242       <Metric name="CollectionInTimePercentageMetric" type="float" unit="Percentage">
243         <Source>WSNMeasurements</Source>
244         <Function xsi:type="PercentageGreaterThanThreshold" resultType="float">
245           <Metric>SouthHourlyDeviceToISDelaySensorSeries</Metric>
246           <Value>
```

```
247             <LongScalar>3600</LongScalar>
248             <!-- More than 1h -->
249           </Value>
250         </Function>
251       </Metric>
252       <Metric name="SouthHourlyDeviceToISDelaySensorSeries" type="SensorSeries" unit="
          seconds">
253         <Source>WSNMeasurements</Source>
254         <Function xsi:type="SensorSeriesConstructor" resultType="SensorSeries">
255           <SensorSetName>South of Grenoble, France</SensorSetName>
256           <Metric>SouthHourlyDeviceToISDelay</Metric>
257         </Function>
258       </Metric>
259       <Metric name="SouthHourlyDeviceToISDelay" type="integer" unit="seconds">
260         <Source>WSNMeasurements</Source>
261         <Function xsi:type="Max" resultType="integer">
262           <Metric>SouthHourlyDeviceToISDelayList</Metric>
263         </Function>
264       </Metric>
265       <Metric name="SouthHourlyDeviceToISDelayList" type="integer" unit="seconds">
266         <Source>WSNMeasurements</Source>
267         <MeasurementDirective xsi:type="DeviceToISDelay" resultType="integer">
268           <ReadingSchedule>Hourly</ReadingSchedule>
269           <SensorSetName>South of Grenoble, France</SensorSetName>
270         </MeasurementDirective>
271       </Metric>
272
273       <Operation name="Client traffic bounds">
274         <SLAParameter name="South Daily Input Rates Overflow Percentage" type="float"
          unit="Percentage">
275           <Metric>SouthDailyDGMCOverflow</Metric>
276           <Communication>
277             <Source>WSNoperator</Source>
278           </Communication>
279         </SLAParameter>
280         <SLAParameter name="South Hourly Input Rates Overflow Percentage" type="float"
          unit="Percentage">
281           <Metric>SouthHourlyDGMCOverflow</Metric>
282           <Communication>
283             <Source>WSNoperator</Source>
284           </Communication>
285         </SLAParameter>
286         <SLAParameter name="South Minutely Input Rates Overflow Percentage" type="float"
          unit="Percentage">
287           <Metric>SouthMinutelyDGMCOverflow</Metric>
288           <Communication>
289             <Source>WSNoperator</Source>
290           </Communication>
291         </SLAParameter>
292
293         <Constant name="MaxMessagePayload">
294             <Integer>48</Integer>
295         </Constant>
296       </Operation>
297       <Operation name="WSN Operator Traffic Management">
298         <SLAParameter name="South Daily Global Collection Percentage" type="float" unit="
          Percentage">
299           <Metric>SouthDailyCollectionPercentageMetric</Metric>
300           <Communication>
301             <Source>WSNoperator</Source>
302           </Communication>
303         </SLAParameter>
304
305         <SLAParameter name="South Consecutive Days Global Collection Counter" type="float
          " unit="Percentage">
306           <Metric>SouthConsecutiveDaysCollectionCounterPercentage</Metric>
307           <Communication>
308             <Source>WSNoperator</Source>
309           </Communication>
310         </SLAParameter>
311         <SLAParameter name="South Sliding Monthly Global Collection Counter" type="float"
          unit="Percentage">
312           <Metric>SouthSlidingMonthlyCollectionCounterPercentage</Metric>
```

```
313           <Communication>
314             <Source>WSNoperator</Source>
315           </Communication>
316         </SLAParameter>
317
318         <SLAParameter name="South Daily Global Collection In Time Percentage" type="float
      " unit="Percentage">
319           <Metric>CollectionInTimePercentageMetric</Metric>
320           <Communication>
321             <Source>WSNMeasurements</Source>
322           </Communication>
323         </SLAParameter>
324
325       </Operation>
326     </OperationGroup>
```

Listing A.4: XML Instance of the Service Definition.

Listing A.4 shows the instantiation of the fist part of the daily metering service: the daily collection of meter indices. We have:

- Schedules at line 7, 16, and 25;
- A Sensor Set at line 35;
- An Operation Group at line 49;
- All the Metrics at line 51;
- A corresponding Operation from line 297 onward.

A.4 The Gas Daily Metering Sensor Configuration

This section corresponds to the instantiation of the use case, regarding the service definition as described in Section 6.4.

```
1   <OperationGroup name="Gas Daily Metering Sensor Configuration">
2
3     <Metric name="SouthDailySGMCOverflow" type="float" unit="Percentage">
4       <Source>WSNoperator</Source>
5       <Function xsi:type="PercentageGreaterThanThreshold" resultType="float">
6         <Metric>SouthDailySGMCSensorSeries</Metric>
7         <Value>
8           <LongScalar>5</LongScalar>
9           <!-- max 5 mess a day -->
10        </Value>
11      </Function>
12    </Metric>
13    <Metric name="SouthHourlySGMCOverflow" type="float" unit="Percentage">
14      <Source>WSNoperator</Source>
15      <Function xsi:type="PercentageGreaterThanThreshold" resultType="float">
16        <Metric>SouthHourlySGMCSensorSeries</Metric>
17        <Value>
18          <LongScalar>2</LongScalar>
19          <!-- max 2 mess by hour -->
20        </Value>
21      </Function>
22    </Metric>
23    <Metric name="SouthMinutelySGMCOverflow" type="float" unit="Percentage">
24      <Source>WSNoperator</Source>
25      <Function xsi:type="PercentageGreaterThanThreshold" resultType="float">
26        <Metric>SouthMinutelySGMCSensorSeries</Metric>
27        <Value>
28          <LongScalar>1</LongScalar>
29          <!-- max 1 mess a min -->
30        </Value>
```

```
31        </Function>
32      </Metric>
33      <Metric name="SouthDailySGMCSensorSeries" type="SensorSeries" unit="messages">
34        <Source>WSNoperator</Source>
35        <Function xsi:type="SensorSeriesConstructor" resultType="SensorSeries">
36          <SensorSetName>South of Grenoble, France</SensorSetName>
37          <Metric>SouthDailySGMC</Metric>
38        </Function>
39      </Metric>
40      <Metric name="SouthHourlySGMCSensorSeries" type="SensorSeries" unit="messages">
41        <Source>WSNoperator</Source>
42        <Function xsi:type="SensorSeriesConstructor" resultType="SensorSeries">
43          <SensorSetName>South of Grenoble, France</SensorSetName>
44          <Metric>SouthHourlySGMC</Metric>
45        </Function>
46      </Metric>
47      <Metric name="SouthMinutelySGMCSensorSeries" type="SensorSeries" unit="messages">
48        <Source>WSNoperator</Source>
49        <Function xsi:type="SensorSeriesConstructor" resultType="SensorSeries">
50          <SensorSetName>South of Grenoble, France</SensorSetName>
51          <Metric>SouthMinutelySGMC</Metric>
52        </Function>
53      </Metric>
54      <Metric name="SouthDailySGMC" type="integer" unit="messages">
55        <Source>WSNoperator</Source>
56        <MeasurementDirective xsi:type="SGMC" resultType="integer">
57          <ReadingSchedule>Daily</ReadingSchedule>
58          <SensorSetName>South of Grenoble, France</SensorSetName>
59        </MeasurementDirective>
60      </Metric>
61      <Metric name="SouthHourlySGMC" type="integer" unit="messages">
62        <Source>WSNoperator</Source>
63        <MeasurementDirective xsi:type="SGMC" resultType="integer">
64          <ReadingSchedule>Hourly</ReadingSchedule>
65          <SensorSetName>South of Grenoble, France</SensorSetName>
66        </MeasurementDirective>
67      </Metric>
68      <Metric name="SouthMinutelySGMC" type="integer" unit="messages">
69        <Source>WSNoperator</Source>
70        <MeasurementDirective xsi:type="SGMC" resultType="integer">
71          <ReadingSchedule>Minutely</ReadingSchedule>
72          <SensorSetName>South of Grenoble, France</SensorSetName>
73        </MeasurementDirective>
74      </Metric>
75
76      <Metric name="SouthConsecutiveDaysConfigurationCounterPercentage" type="float" unit
        ="Percentage">
77        <Source>WSNoperator</Source>
78        <Function xsi:type="PercentageGreaterThanThreshold" resultType="float">
79          <Metric>SouthConsecutiveDaysConfigurationCounterSS</Metric>
80          <Value>
81            <LongScalar>3</LongScalar>
82            <!-- percentage of sensors with 4 degraded days -->
83          </Value>
84        </Function>
85      </Metric>
86      <Metric name="SouthConsecutiveDaysConfigurationCounterSS" type="SensorSeries" unit=
        "Degradation Ocurrences">
87        <Source>WSNoperator</Source>
88        <Function xsi:type="SensorSeriesConstructor" resultType="SensorSeries">
89          <SensorSetName>South of Grenoble, France</SensorSetName>
90          <Metric>SouthConsecutiveDaysConfigurationCounter</Metric>
91        </Function>
92      </Metric>
93      <Metric name="SouthConsecutiveDaysConfigurationCounter" type="float" unit="Days">
94        <Source>WSNoperator</Source>
95        <Function xsi:type="ValueOccurs" resultType="integer">
96          <Metric>SouthConsecutiveDaysConfigurationCounterTS</Metric>
97          <Value>
98            <Boolean>False</Boolean>
99            <!-- how many false in last four days for each sensor -->
100         </Value>
101       </Function>
```

```
102     </Metric>
103     <Metric name="SouthConsecutiveDaysConfigurationCounterTS" type="TS" unit="
        Percentage">
104       <Source>WSNoperator</Source>
105       <Function xsi:type="TSConstructor" resultType="TS">
106         <Schedule>Daily</Schedule>
107         <Metric>SouthDailyConfigurationCounter</Metric>
108         <Window>4</Window>
109         <!-- last four days -->
110       </Function>
111     </Metric>
112     <Metric name="SouthSlidingMonthlyConfigurationCounterPercentage" type="float" unit=
        "Percentage">
113       <Source>WSNoperator</Source>
114       <Function xsi:type="PercentageGreaterThanThreshold" resultType="float">
115         <Metric>SouthSlidingMonthlyConfigurationCounterSS</Metric>
116         <Value>
117           <LongScalar>5</LongScalar>
118           <!-- percentage of sensors with more than 5 degraded in a sliding month -->
119         </Value>
120       </Function>
121     </Metric>
122     <Metric name="SouthSlidingMonthlyConfigurationCounterSS" type="SensorSeries" unit="
        Degradation Ocurrences">
123       <Source>WSNoperator</Source>
124       <Function xsi:type="SensorSeriesConstructor" resultType="SensorSeries">
125         <SensorSetName>South of Grenoble, France</SensorSetName>
126         <Metric>SouthSlidingMonthlyConfigurationCounter</Metric>
127       </Function>
128     </Metric>
129     <Metric name="SouthSlidingMonthlyConfigurationCounter" type="float" unit="Days">
130       <Source>WSNoperator</Source>
131       <Function xsi:type="ValueOccurs" resultType="integer">
132         <Metric>SouthSlidingMonthlyConfigurationCounterTS</Metric>
133         <Value>
134           <Boolean>False</Boolean>
135           <!-- how many false in a month for each sensor -->
136         </Value>
137       </Function>
138     </Metric>
139     <Metric name="SouthSlidingMonthlyConfigurationCounterTS" type="TS" unit="Percentage
        ">
140       <Source>WSNoperator</Source>
141       <Function xsi:type="TSConstructor" resultType="TS">
142         <Schedule>Daily</Schedule>
143         <Metric>SouthDailyConfigurationCounter</Metric>
144         <Window>30</Window>
145       </Function>
146     </Metric>
147     <Metric name="SouthDailyConfigurationCounter" type="boolean" unit="Percentage">
148       <Source>WSNoperator</Source>
149       <Function xsi:type="ValueGreaterThanThreshold" resultType="boolean">
150         <Metric>SouthDailyMessageDeliveryRatio</Metric>
151         <Value>
152           <FloatScalar>0.5</FloatScalar>
153           <!-- One of two messages -->
154         </Value>
155       </Function>
156     </Metric>
157
158     <Metric name="SouthDailyConfigurationPercentageMetric" type="float" unit="
        Percentage">
159       <Source>WSNoperator</Source>
160       <Function xsi:type="PercentageGreaterThanThreshold" resultType="float">
161         <Metric>SouthDailyMessageDeliveryRatioSS</Metric>
162         <Value>
163           <FloatScalar>0.5</FloatScalar>
164           <!-- One of two messages -->
165         </Value>
166       </Function>
167     </Metric>
168     <Metric name="SouthDailyMessageDeliveryRatioSS" type="SensorSeries" unit="
        Percentage">
```

```
169         <Source>WSNoperator</Source>
170         <Function xsi:type="SensorSeriesConstructor" resultType="SensorSeries">
171           <SensorSetName>South of Grenoble, France</SensorSetName>
172           <Metric>SouthDailyMessageDeliveryRatio</Metric>
173         </Function>
174       </Metric>
175       <Metric name="SouthDailyMessageDeliveryRatio" type="float" unit="Percentage">
176         <Source>WSNoperator</Source>
177         <Function xsi:type="Divide" resultType="float">
178           <Operand>
179             <Metric>SouthDailySRMC</Metric>
180           </Operand>
181           <Operand>
182             <Metric>SouthDailySGMC</Metric>
183           </Operand>
184         </Function>
185       </Metric>
186       <Metric name="SouthDailySRMC" type="integer" unit="messages">
187         <Source>WSNoperator</Source>
188         <MeasurementDirective xsi:type="SRMC" resultType="integer">
189           <ReadingSchedule>Daily</ReadingSchedule>
190           <SensorSetName>South of Grenoble, France</SensorSetName>
191         </MeasurementDirective>
192       </Metric>
193
194       <Metric name="ConfigurationInTimePercentageMetric" type="float" unit="Percentage">
195         <Source>WSNMeasurements</Source>
196         <Function xsi:type="PercentageGreaterThanThreshold" resultType="float">
197           <Metric>SouthHourlyISToDeviceDelaySensorSeries</Metric>
198           <Value>
199             <LongScalar>3600</LongScalar>
200             <!-- More than 1h -->
201           </Value>
202         </Function>
203       </Metric>
204       <Metric name="SouthHourlyISToDeviceDelaySensorSeries" type="SensorSeries" unit="
          seconds">
205         <Source>WSNMeasurements</Source>
206         <Function xsi:type="SensorSeriesConstructor" resultType="SensorSeries">
207           <SensorSetName>South of Grenoble, France</SensorSetName>
208           <Metric>SouthHourlyISToDeviceDelay</Metric>
209         </Function>
210       </Metric>
211       <Metric name="SouthHourlyISToDeviceDelay" type="integer" unit="seconds">
212         <Source>WSNMeasurements</Source>
213         <Function xsi:type="Max" resultType="integer">
214           <Metric>SouthHourlyISToDeviceDelayList</Metric>
215         </Function>
216       </Metric>
217       <Metric name="SouthHourlyISToDeviceDelayList" type="integer" unit="seconds">
218         <Source>WSNMeasurements</Source>
219         <MeasurementDirective xsi:type="ISToDeviceDelay" resultType="integer">
220           <ReadingSchedule>Hourly</ReadingSchedule>
221           <SensorSetName>South of Grenoble, France</SensorSetName>
222         </MeasurementDirective>
223       </Metric>
224
225       <Operation name="Client traffic bounds">
226         <SLAParameter name="South Daily Input Rates Overflow Percentage" type="float"
          unit="Percentage">
227           <Metric>SouthDailySGMCOverflow</Metric>
228           <Communication>
229             <Source>WSNoperator</Source>
230           </Communication>
231         </SLAParameter>
232         <SLAParameter name="South Hourly Input Rates Overflow Percentage" type="float"
          unit="Percentage">
233           <Metric>SouthHourlySGMCOverflow</Metric>
234           <Communication>
235             <Source>WSNoperator</Source>
236           </Communication>
237         </SLAParameter>
238         <SLAParameter name="South Minutely Input Rates Overflow Percentage" type="float"
```

```
    unit="Percentage">
239         <Metric>SouthMinutelySGMCOverflow</Metric>
240         <Communication>
241           <Source>WSNoperator</Source>
242         </Communication>
243       </SLAParameter>
244       <Constant name="MaxMessagePayload">
245           <Integer>148</Integer>
246       </Constant>
247     </Operation>
248
249     <Operation name="WSN Operator Traffic Management">
250       <SLAParameter name="South Daily Global Configuration Percentage" type="float"
    unit="Percentage">
251         <Metric>SouthDailyConfigurationPercentageMetric</Metric>
252         <Communication>
253           <Source>WSNoperator</Source>
254         </Communication>
255       </SLAParameter>
256
257       <SLAParameter name="South Consecutive Days Global Configuration Counter" type="
    float" unit="Percentage">
258         <Metric>SouthConsecutiveDaysConfigurationCounterPercentage</Metric>
259         <Communication>
260           <Source>WSNoperator</Source>
261         </Communication>
262       </SLAParameter>
263       <SLAParameter name="South Sliding Monthly Global Configuration Counter" type="
    float" unit="Percentage">
264         <Metric>SouthSlidingMonthlyConfigurationCounterPercentage</Metric>
265         <Communication>
266           <Source>WSNoperator</Source>
267         </Communication>
268       </SLAParameter>
269
270       <SLAParameter name="South Daily Global Configuration In Time Percentage" type="
    float" unit="Percentage">
271         <Metric>ConfigurationInTimePercentageMetric</Metric>
272         <Communication>
273           <Source>WSNMeasurements</Source>
274         </Communication>
275       </SLAParameter>
276
277     </Operation>
278   </OperationGroup>
279 </ServiceDefinition>
```

Listing A.5: XML Instance of the Gas Daily Metering Sensor Configuration Service Definition.

Listing A.5 includes the Operations, Operation Groups and Metrics corresponding to the telemetering configuration service. The metrics are based on the measurement directives, specified at line 54, 61, 68, 186, and 217.

A.5 The Gas Alarm Message Collection

This section corresponds to the instantiation of the service definition for the Gas Alarm Service, described in Section 6.5.

```
1  <ServiceDefinition name="Building Gas Alarm Collection">
2
3    <Schedule name="Monthly">
4      <Period>
5        <Start>2014-07-31T14:00:00.000-05:00</Start>
6        <End>2020-12-31T14:00:00.000-05:00</End>
7      </Period>
8      <Interval>
9        <Months>1</Months>
10     </Interval>
```

```
11  </Schedule>
12
13  <Schedule name="Daily">
14    <Period>
15      <Start>2014-07-31T14:00:00.000-05:00</Start>
16      <End>2020-12-31T14:00:00.000-05:00</End>
17    </Period>
18    <Interval>
19      <Days>1</Days>
20    </Interval>
21  </Schedule>
22
23  <Schedule name="Minutely">
24    <Period>
25      <Start>2014-07-31T14:00:00.000-05:00</Start>
26      <End>2020-12-31T14:00:00.000-05:00</End>
27    </Period>
28    <Interval>
29      <Minutes>2</Minutes>
30    </Interval>
31  </Schedule>
32
33  <SensorSet name="Building 46, Grenoble, France">
34    <List ParameterType="ID">
35      23 24 28 75 AZ24 T32
36    </List>
37  </SensorSet>
38
39  <SLAParameter name="Building46 Daily Global Collection In Time Percentage Metric"
      type="float" unit="Percentage">
40    <Metric>CollectionInTimePercentageMetric</Metric>
41    <Communication>
42      <Source>WSNMeasurements</Source>
43    </Communication>
44  </SLAParameter>
45
46  <SLAParameter name="Building46 Collection In Time Degraded Months" type="float" unit=
      "Percentage">
47    <Metric>CollectionInTimeDegradedMonths</Metric>
48    <Communication>
49      <Source>WSNMeasurements</Source>
50    </Communication>
51  </SLAParameter>
52
53
54  <Metric name="CollectionInTimeDegradedMonths" type="float" unit="Months">
55    <Source>WSNMeasurements</Source>
56    <Function xsi:type="NumberGreaterThanThreshold" resultType="integer">
57      <Metric>MonthCollectionInTimePercentageTS</Metric>
58      <Value>
59        <LongScalar>5</LongScalar>
60        <!-- Number of month whith more than 5 degraded days -->
61      </Value>
62    </Function>
63  </Metric>
64
65  <Metric name="MonthCollectionInTimePercentageTS" type="TS" unit="Days">
66    <Source>WSNMeasurements</Source>
67    <Function xsi:type="TSConstructor" resultType="TS">
68      <Schedule>Monthly</Schedule>
69      <Metric>MonthCollectionInTimePercentage</Metric>
70      <Window>1</Window>
71    </Function>
72  </Metric>
73
74  <Metric name="MonthCollectionInTimePercentage" type="integer" unit="Days">
75    <Source>WSNMeasurements</Source>
76    <Function xsi:type="NumberGreaterThanThreshold" resultType="integer">
77      <Metric>CollectionInTimePercentageMetricTS</Metric>
78      <Value>
79        <FloatScalar>0.5</FloatScalar>
80        <!-- Number of days when more than 50 percent of sensors have a more than 10min
        delay-->
```

```
81          </Value>
82        </Function>
83    </Metric>
84
85    <Metric name="CollectionInTimePercentageMetricTS" type="TS" unit="Percentage">
86        <Source>WSNMeasurements</Source>
87        <Function xsi:type="TSConstructor" resultType="TS">
88          <Schedule>Daily</Schedule>
89          <Metric>CollectionInTimePercentageMetric</Metric>
90        </Function>
91    </Metric>
92
93    <Metric name="CollectionInTimePercentageMetric" type="float" unit="Percentage">
94        <Source>WSNMeasurements</Source>
95        <Function xsi:type="PercentageGreaterThanThreshold" resultType="float">
96          <Metric>Building46MinutelyDeviceToISDelaySensorSeries</Metric>
97          <Value>
98            <LongScalar>600</LongScalar>
99            <!-- More than 10min -->
100         </Value>
101       </Function>
102   </Metric>
103
104   <Metric name="Building46MinutelyDeviceToISDelaySensorSeries" type="SensorSeries" unit
        ="seconds">
105       <Source>WSNMeasurements</Source>
106       <Function xsi:type="SensorSeriesConstructor" resultType="SensorSeries">
107         <SensorSetName>Building46, Grenoble, France</SensorSetName>
108         <Metric>Building46MinutelyDeviceToISDelay</Metric>
109       </Function>
110   </Metric>
111   <Metric name="Building46MinutelyDeviceToISDelay" type="integer" unit="seconds">
112       <Source>WSNMeasurements</Source>
113       <Function xsi:type="Max" resultType="integer">
114         <Metric>Building46MinutelyDeviceToISDelayList</Metric>
115       </Function>
116   </Metric>
117   <Metric name="Building46MinutelyDeviceToISDelayList" type="integer" unit="seconds">
118       <Source>WSNMeasurements</Source>
119       <MeasurementDirective xsi:type="DeviceToISDelay" resultType="integer">
120         <ReadingSchedule>Minutely</ReadingSchedule>
121         <SensorSetName>Building46, Grenoble, France</SensorSetName>
122       </MeasurementDirective>
123   </Metric>
124
125   <Constant name="MaxMessagePayload">
126       <Integer>48</Integer>
127   </Constant>
128
129 </ServiceDefinition>
```

Listing A.6: XML Instance of The Gas Alarm Message Collection Service Definition.

Listing A.6 shows the construction of the Alarm Service. We have:

- Schedules between lines 3 and 31;
- A Sensor Set at line 33;
- Two SLA Parameters at line 39 and 46;
- All the corresponding Metrics from line 54;

A.6 The Obligations

This section corresponds to the instantiation of the Obligations as described in Sections 4.2.3, 5.4, and 6.3.

```
<Obligations>
  <ObligationGroup name="Gas Daily Metering Collection">
    <ServiceLevelObjective name="Client traffic bounds">
      <Obliged>GasCompany23</Obliged>
      <Validity>
        <Start>2014-07-31T14:00:00.000-05:00</Start>
        <End>2020-12-31T14:00:00.000-05:00</End>
      </Validity>
      <Expression>
        <And>
          <Expression>
            <Predicate xsi:type="Less">
              <SLAParameter>South Daily Input Rates Overflow Percentage</SLAParameter>
              <Value>0.05</Value>
              <!--less than 5 percent of overflow-->
            </Predicate>
          </Expression>
          <Expression>
            <And>
              <Expression>
                <Predicate xsi:type="Less">
                  <SLAParameter>South Hourly Input Rates Overflow Percentage</
    SLAParameter>
                  <Value>0.02</Value>
                </Predicate>
              </Expression>
              <Expression>
                <Predicate xsi:type="Less">
                  <SLAParameter>South Minutely Input Rates Overflow Percentage</
    SLAParameter>
                  <Value>0.01</Value>
                </Predicate>
              </Expression>
            </And>
          </Expression>
        </And>
      </Expression>
      <Schedule>Hourly</Schedule>
    </ServiceLevelObjective>

    <ServiceLevelObjective name="SouthDailyMessageDeliveryRatio">
      <Obliged>WSNOperator</Obliged>
      <Validity>
        <Start>2014-07-31T14:00:00.000-05:00</Start>
        <End>2020-12-31T14:00:00.000-05:00</End>
      </Validity>
      <Expression>
        <Predicate xsi:type="Greater">
          <SLAParameter>South Daily Global Collection Percentage</SLAParameter>
          <Value>0.95</Value>
          <!--more than 95 percent of sensors collected-->
        </Predicate>
      </Expression>
      <Schedule>Daily</Schedule>
    </ServiceLevelObjective>

    <ServiceLevelObjective name="SouthDailyDeviceToISDelay">
      <Obliged>WSNOperator</Obliged>
      <Validity>
        <Start>2014-07-31T14:00:00.000-05:00</Start>
        <End>2020-12-31T14:00:00.000-05:00</End>
      </Validity>
      <Expression>
        <Predicate xsi:type="Greater">
          <SLAParameter>South Daily Global Collection In Time Percentage</SLAParameter>
          <Value>0.80</Value>
          <!--more than 80 percent of sensors collected in time-->
        </Predicate>
      </Expression>
      <Schedule>Daily</Schedule>
    </ServiceLevelObjective>

```

```
71      <ServiceLevelObjective name="South Four Consecutive Days Global Collection Fault
        Counter">
72        <Obliged>WSNOperator</Obliged>
73        <Validity>
74          <Start>2014-07-31T14:00:00.000-05:00</Start>
75          <End>2020-12-31T14:00:00.000-05:00</End>
76        </Validity>
77        <Expression>
78          <Predicate xsi:type="Greater">
79            <SLAParameter>South Consecutive Days Global Collection Counter</SLAParameter>
80            <Value>0.98</Value>
81            <!--more than 98 percent of sensors are not faulty 4 times in the last 4 days
        -->
82          </Predicate>
83        </Expression>
84        <Schedule>Daily</Schedule>
85      </ServiceLevelObjective>
86
87      <ServiceLevelObjective name="South Sliding Monthly Global Collection Fault Counter"
        >
88        <Obliged>WSNOperator</Obliged>
89        <Validity>
90          <Start>2014-07-31T14:00:00.000-05:00</Start>
91          <End>2020-12-31T14:00:00.000-05:00</End>
92        </Validity>
93        <Expression>
94          <Predicate xsi:type="Greater">
95            <SLAParameter>South Sliding Monthly Global Collection Counter</SLAParameter>
96            <Value>0.99</Value>
97            <!--more than 99 percent of sensors are not faulty 5 times in the last 30
        days-->
98          </Predicate>
99        </Expression>
100       <Schedule>Daily</Schedule>
101     </ServiceLevelObjective>
102
103     <ActionGuarantee name="Client traffic bounds">
104       <Obliged>WSNoperator</Obliged>
105       <Expression>
106         <Predicate xsi:type="Violation">
107           <ServiceLevelObjective>Client traffic bounds</ServiceLevelObjective>
108         </Predicate>
109       </Expression>
110       <EvaluationEvent>NewValue</EvaluationEvent>
111       <QualifiedAction>
112         <Party>WSNoperator</Party>
113         <Action actionName="notification" xsi:type="Notification">
114           <NotificationType>Information</NotificationType>
115           <CausingGuarantee>Client traffic bounds</CausingGuarantee>
116         </Action>
117       </QualifiedAction>
118       <ExecutionModality>Always</ExecutionModality>
119     </ActionGuarantee>
120
121     <ActionGuarantee name="SouthDailyMessageDeliveryRatio">
122       <Obliged>WSNoperator</Obliged>
123       <Expression>
124         <Predicate xsi:type="Violation">
125           <ServiceLevelObjective>SouthDailyMessageDeliveryRatio</ServiceLevelObjective>
126         </Predicate>
127       </Expression>
128       <EvaluationEvent>NewValue</EvaluationEvent>
129       <QualifiedAction>
130         <Party>WSNoperator</Party>
131         <Action actionName="notification" xsi:type="Notification">
132           <NotificationType>Violation</NotificationType>
133           <CausingGuarantee>SouthDailyMessageDeliveryRatio</CausingGuarantee>
134         </Action>
135       </QualifiedAction>
136       <QualifiedAction>
137         <Party>WSNoperator</Party>
138         <Action actionName="ChangeMonitoring" xsi:type="SetMonitoringSchedule">
139           <MeasurementDirective>DGMC</MeasurementDirective>
```

```
140             <MeasurementDirective>DRMC</MeasurementDirective>
141             <NewSchedule>Minutely</NewSchedule>
142             <CausingGuarantee>SouthDailyMessageDeliveryRatio</CausingGuarantee>
143           </Action>
144         </QualifiedAction>
145         <ExecutionModality>Always</ExecutionModality>
146       </ActionGuarantee>
147
148       <ActionGuarantee name="SouthDailyDeviceToISDelay">
149         <Obliged>WSNMeasurements</Obliged>
150         <Expression>
151           <Predicate xsi:type="Violation">
152             <ServiceLevelObjective>SouthDailyDeviceToISDelay</ServiceLevelObjective>
153           </Predicate>
154         </Expression>
155         <EvaluationEvent>NewValue</EvaluationEvent>
156         <QualifiedAction>
157           <Party>WSNMeasurements</Party>
158           <Action actionName="notification" xsi:type="Notification">
159             <NotificationType>Violation</NotificationType>
160             <CausingGuarantee>SouthDailyDeviceToISDelay</CausingGuarantee>
161           </Action>
162         </QualifiedAction>
163         <QualifiedAction>
164           <Party>WSNMeasurements</Party>
165           <Action actionName="ChangeMonitoring" xsi:type="SetMonitoringSchedule">
166             <MeasurementDirective>DeviceToISDelay</MeasurementDirective>
167             <NewSchedule>Minutely</NewSchedule>
168             <CausingGuarantee>SouthDailyDeviceToISDelay</CausingGuarantee>
169           </Action>
170         </QualifiedAction>
171         <ExecutionModality>Always</ExecutionModality>
172       </ActionGuarantee>
173
174       <ActionGuarantee name="South Four Consecutive Days Global Collection Fault Counter"
          >
175         <Obliged>WSNoperator</Obliged>
176         <Expression>
177           <Predicate xsi:type="Violation">
178             <ServiceLevelObjective>South Four Consecutive Days Global Collection Fault
          Counter</ServiceLevelObjective>
179           </Predicate>
180         </Expression>
181         <EvaluationEvent>NewValue</EvaluationEvent>
182         <QualifiedAction>
183           <Party>WSNoperator</Party>
184           <Action actionName="notification" xsi:type="Notification">
185             <NotificationType>Violation</NotificationType>
186             <CausingGuarantee>South Four Consecutive Days Global Collection Fault Counter
          </CausingGuarantee>
187           </Action>
188         </QualifiedAction>
189         <ExecutionModality>Always</ExecutionModality>
190       </ActionGuarantee>
191
192       <ActionGuarantee name="South Sliding Monthly Global Collection Fault Counter">
193         <Obliged>WSNoperator</Obliged>
194         <Expression>
195           <Predicate xsi:type="Violation">
196             <ServiceLevelObjective>South Sliding Monthly Global Collection Fault Counter<
          /ServiceLevelObjective>
197           </Predicate>
198         </Expression>
199         <EvaluationEvent>NewValue</EvaluationEvent>
200         <QualifiedAction>
201           <Party>WSNoperator</Party>
202           <Action actionName="notification" xsi:type="Notification">
203             <NotificationType>Violation</NotificationType>
204             <CausingGuarantee>South Sliding Monthly Global Collection Fault Counter</
          CausingGuarantee>
205           </Action>
206         </QualifiedAction>
207         <ExecutionModality>Always</ExecutionModality>
```

```
208     </ActionGuarantee>
209   </ObligationGroup>
```

Listing A.7: XML Instance of the Obligations.

Listing A.7 shows the instantiation of the fist part of the daily metering obligations: the obligations for the daily collection of meter indexes. We have five Service Level Objectives (SLOs) at line 3, 39, 55, 71, and 87. And the five corresponding Action Guarantees at line 103, 121, 148, 174, and 192.

A.7 The Obligations for the Gas Configuration Messages

This section corresponds to the instantiation of the use case, regarding the obligations as described in Section 6.4.

```
1   <ObligationGroup name="Gas Daily Metering Sensor Configuration">
2     <ServiceLevelObjective name="Client traffic bounds">
3       <Obliged>GasCompany23</Obliged>
4       <Validity>
5         <Start>2014-07-31T14:00:00.000-05:00</Start>
6         <End>2020-12-31T14:00:00.000-05:00</End>
7       </Validity>
8       <Expression>
9         <And>
10          <Expression>
11            <Predicate xsi:type="Less">
12              <SLAParameter>South Daily Input Rates Overflow Percentage</SLAParameter>
13              <Value>0.05</Value>
14              <!--less than 5 percent of overflow-->
15            </Predicate>
16          </Expression>
17          <Expression>
18            <And>
19              <Expression>
20                <Predicate xsi:type="Less">
21                  <SLAParameter>South Hourly Input Rates Overflow Percentage</
    SLAParameter>
22                  <Value>0.02</Value>
23                </Predicate>
24              </Expression>
25              <Expression>
26                <Predicate xsi:type="Less">
27                  <SLAParameter>South Minutely Input Rates Overflow Percentage</
    SLAParameter>
28                  <Value>0.01</Value>
29                </Predicate>
30              </Expression>
31            </And>
32          </Expression>
33        </And>
34      </Expression>
35      <Schedule>Hourly</Schedule>
36    </ServiceLevelObjective>
37
38    <ServiceLevelObjective name="SouthDailyMessageDeliveryRatio">
39      <Obliged>WSNOperator</Obliged>
40      <Validity>
41        <Start>2014-07-31T14:00:00.000-05:00</Start>
42        <End>2020-12-31T14:00:00.000-05:00</End>
43      </Validity>
44      <Expression>
45        <Predicate xsi:type="Greater">
46          <SLAParameter>South Daily Global Configuration Percentage</SLAParameter>
47          <Value>0.95</Value>
48          <!--more than 95 percent of sensors Configurated-->
49        </Predicate>
50      </Expression>
51      <Schedule>Daily</Schedule>
```

```
52      </ServiceLevelObjective>
53
54      <ServiceLevelObjective name="SouthDailyISToDeviceDelay">
55        <Obliged>WSNOperator</Obliged>
56        <Validity>
57          <Start>2014-07-31T14:00:00.000-05:00</Start>
58          <End>2020-12-31T14:00:00.000-05:00</End>
59        </Validity>
60        <Expression>
61          <Predicate xsi:type="Greater">
62            <SLAParameter>South Daily Global Configuration In Time Percentage</
      SLAParameter>
63            <Value>0.80</Value>
64            <!--more than 80 percent of sensors Configurated in time-->
65          </Predicate>
66        </Expression>
67        <Schedule>Daily</Schedule>
68      </ServiceLevelObjective>
69
70      <ServiceLevelObjective name="South Four Consecutive Days Global Configuration Fault
        Counter">
71        <Obliged>WSNOperator</Obliged>
72        <Validity>
73          <Start>2014-07-31T14:00:00.000-05:00</Start>
74          <End>2020-12-31T14:00:00.000-05:00</End>
75        </Validity>
76        <Expression>
77          <Predicate xsi:type="Greater">
78            <SLAParameter>South Consecutive Days Global Configuration Counter</
      SLAParameter>
79            <Value>0.98</Value>
80            <!--more than 98 percent of sensors configurations are not faulty 4 times in
      the last 4 days-->
81          </Predicate>
82        </Expression>
83        <Schedule>Daily</Schedule>
84      </ServiceLevelObjective>
85
86      <ServiceLevelObjective name="South Sliding Monthly Global Configuration Fault
        Counter">
87        <Obliged>WSNOperator</Obliged>
88        <Validity>
89          <Start>2014-07-31T14:00:00.000-05:00</Start>
90          <End>2020-12-31T14:00:00.000-05:00</End>
91        </Validity>
92        <Expression>
93          <Predicate xsi:type="Greater">
94            <SLAParameter>South Sliding Monthly Global Configuration Counter</
      SLAParameter>
95            <Value>0.99</Value>
96            <!--more than 99 percent of sensors configuration are not faulty 5 times in
      the last 30 days-->
97          </Predicate>
98        </Expression>
99        <Schedule>Daily</Schedule>
100     </ServiceLevelObjective>
101
102     <ActionGuarantee name="Client traffic bounds">
103       <Obliged>WSNoperator</Obliged>
104       <Expression>
105         <Predicate xsi:type="Violation">
106           <ServiceLevelObjective>Client traffic bounds</ServiceLevelObjective>
107         </Predicate>
108       </Expression>
109       <EvaluationEvent>NewValue</EvaluationEvent>
110       <QualifiedAction>
111         <Party>WSNoperator</Party>
112         <Action actionName="notification" xsi:type="Notification">
113           <NotificationType>Information</NotificationType>
114           <CausingGuarantee>Client traffic bounds</CausingGuarantee>
115         </Action>
116       </QualifiedAction>
117       <ExecutionModality>Always</ExecutionModality>
```

```
118     </ActionGuarantee>
119
120     <ActionGuarantee name="SouthDailyMessageDeliveryRatio">
121       <Obliged>WSNoperator</Obliged>
122       <Expression>
123         <Predicate xsi:type="Violation">
124           <ServiceLevelObjective>SouthDailyMessageDeliveryRatio</ServiceLevelObjective>
125         </Predicate>
126       </Expression>
127       <EvaluationEvent>NewValue</EvaluationEvent>
128       <QualifiedAction>
129         <Party>WSNoperator</Party>
130         <Action actionName="notification" xsi:type="Notification">
131           <NotificationType>Violation</NotificationType>
132           <CausingGuarantee>SouthDailyMessageDeliveryRatio</CausingGuarantee>
133         </Action>
134       </QualifiedAction>
135       <QualifiedAction>
136         <Party>WSNoperator</Party>
137         <Action actionName="ChangeMonitoring" xsi:type="SetMonitoringSchedule">
138           <MeasurementDirective>SGMC</MeasurementDirective>
139           <MeasurementDirective>SRMC</MeasurementDirective>
140           <NewSchedule>Minutely</NewSchedule>
141           <CausingGuarantee>SouthDailyMessageDeliveryRatio</CausingGuarantee>
142         </Action>
143       </QualifiedAction>
144       <ExecutionModality>Always</ExecutionModality>
145     </ActionGuarantee>
146
147     <ActionGuarantee name="SouthDailyDeviceToISDelay">
148       <Obliged>WSNMeasurements</Obliged>
149       <Expression>
150         <Predicate xsi:type="Violation">
151           <ServiceLevelObjective>SouthDailyISToDeviceDelay</ServiceLevelObjective>
152         </Predicate>
153       </Expression>
154       <EvaluationEvent>NewValue</EvaluationEvent>
155       <QualifiedAction>
156         <Party>WSNMeasurements</Party>
157         <Action actionName="notification" xsi:type="Notification">
158           <NotificationType>Violation</NotificationType>
159           <CausingGuarantee>SouthDailyISToDeviceDelay</CausingGuarantee>
160         </Action>
161       </QualifiedAction>
162       <QualifiedAction>
163         <Party>WSNMeasurements</Party>
164         <Action actionName="ChangeMonitoring" xsi:type="SetMonitoringSchedule">
165           <MeasurementDirective>ISToDeviceDelay</MeasurementDirective>
166           <NewSchedule>Minutely</NewSchedule>
167           <CausingGuarantee>SouthDailyISToDeviceDelay</CausingGuarantee>
168         </Action>
169       </QualifiedAction>
170       <ExecutionModality>Always</ExecutionModality>
171     </ActionGuarantee>
172
173     <ActionGuarantee name="South Four Consecutive Days Global Configuration Fault
      Counter">
174       <Obliged>WSNoperator</Obliged>
175       <Expression>
176         <Predicate xsi:type="Violation">
177           <ServiceLevelObjective>South Four Consecutive Days Global Configuration Fault
        Counter</ServiceLevelObjective>
178         </Predicate>
179       </Expression>
180       <EvaluationEvent>NewValue</EvaluationEvent>
181       <QualifiedAction>
182         <Party>WSNoperator</Party>
183         <Action actionName="notification" xsi:type="Notification">
184           <NotificationType>Violation</NotificationType>
185           <CausingGuarantee>South Four Consecutive Days Global Configuration Fault
       Counter</CausingGuarantee>
186         </Action>
187       </QualifiedAction>
```

```
188       <ExecutionModality>Always</ExecutionModality>
189     </ActionGuarantee>
190
191     <ActionGuarantee name="South Sliding Monthly Global Configuration Fault Counter">
192       <Obliged>WSNoperator</Obliged>
193       <Expression>
194         <Predicate xsi:type="Violation">
195           <ServiceLevelObjective>South Sliding Monthly Global Configuration Fault
      Counter</ServiceLevelObjective>
196         </Predicate>
197       </Expression>
198       <EvaluationEvent>NewValue</EvaluationEvent>
199       <QualifiedAction>
200         <Party>WSNoperator</Party>
201         <Action actionName="notification" xsi:type="Notification">
202           <NotificationType>Violation</NotificationType>
203           <CausingGuarantee>South Sliding Monthly Global Configuration Fault Counter</
      CausingGuarantee>
204         </Action>
205       </QualifiedAction>
206       <ExecutionModality>Always</ExecutionModality>
207     </ActionGuarantee>
208   </ObligationGroup>
```

Listing A.8: XML Instance of the Gas Daily Metering Sensor Configuration Obligations.

Listing A.8 includes the Obligations corresponding to the tele-metering configuration service. We have five Service Level Objectives (SLOs) at line 2, 38, 54, 70, and 86. And the five corresponding Action Guarantees at line 102, 120, 147, 173, and 191.

A.8 The Gas Alarm Collection Obligations

This section corresponds to the instantiation of the obligations for the Gas Alarm Service, described in Section 6.5.

```
1   <ObligationGroup name="Building Gas Alarm Collection">
2     <ServiceLevelObjective name="Building46 Daily Global Collection In Time Percentage
      Metric">
3       <Obliged>WSNOperator</Obliged>
4       <Validity>
5         <Start>2014-07-31T14:00:00.000-05:00</Start>
6         <End>2020-12-31T14:00:00.000-05:00</End>
7       </Validity>
8       <Expression>
9         <Predicate xsi:type="Greater">
10          <SLAParameter>Building46 Daily Global Collection In Time Percentage Metric</
      SLAParameter>
11          <Value>0.95</Value>
12          <!--more than 95 percent of sensors collected-->
13        </Predicate>
14      </Expression>
15      <Schedule>Daily</Schedule>
16    </ServiceLevelObjective>
17
18    <ServiceLevelObjective name="Building46 Collection In Time Degraded Months">
19      <Obliged>WSNOperator</Obliged>
20      <Validity>
21        <Start>2014-07-31T14:00:00.000-05:00</Start>
22        <End>2020-12-31T14:00:00.000-05:00</End>
23      </Validity>
24      <Expression>
25        <Predicate xsi:type="Less">
26          <SLAParameter>Building46 Collection In Time Degraded Months</SLAParameter>
27          <Value>2</Value>
28          <!--Less than 2 degraded months-->
29        </Predicate>
30      </Expression>
31      <Schedule>Daily</Schedule>
```

```
32      </ServiceLevelObjective>
33
34      <ActionGuarantee name="Building46 Daily Global Collection In Time Percentage Metric
        ">
35        <Obliged>WSNMeasurements</Obliged>
36        <Expression>
37          <Predicate xsi:type="Violation">
38            <ServiceLevelObjective>Building46 Daily Global Collection In Time Percentage
        Metric</ServiceLevelObjective>
39          </Predicate>
40        </Expression>
41        <EvaluationEvent>NewValue</EvaluationEvent>
42        <QualifiedAction>
43          <Party>WSNMeasurements</Party>
44          <Action actionName="notification" xsi:type="Notification">
45            <NotificationType>Violation</NotificationType>
46            <CausingGuarantee>Building46 Daily Global Collection In Time Percentage
        Metric</CausingGuarantee>
47          </Action>
48        </QualifiedAction>
49        <QualifiedAction>
50          <Party>WSNMeasurements</Party>
51          <Action actionName="ChangeMonitoring" xsi:type="SetMonitoringSchedule">
52            <MeasurementDirective>ISToDeviceDelay</MeasurementDirective>
53            <NewSchedule>Minutely</NewSchedule>
54            <CausingGuarantee>SouthDailyISToDeviceDelay</CausingGuarantee>
55          </Action>
56        </QualifiedAction>
57        <ExecutionModality>Always</ExecutionModality>
58      </ActionGuarantee>
59
60      <ActionGuarantee name="Building46 Collection In Time Degraded Months">
61        <Obliged>WSNMeasurements</Obliged>
62        <Expression>
63          <Predicate xsi:type="Violation">
64            <ServiceLevelObjective>Building46 Collection In Time Degraded Months</
        ServiceLevelObjective>
65          </Predicate>
66        </Expression>
67        <EvaluationEvent>NewValue</EvaluationEvent>
68        <QualifiedAction>
69          <Party>WSNMeasurements</Party>
70          <Action actionName="notification" xsi:type="Notification">
71            <NotificationType>Violation</NotificationType>
72            <CausingGuarantee>Building46 Collection In Time Degraded Months</
        CausingGuarantee>
73          </Action>
74        </QualifiedAction>
75        <ExecutionModality>Always</ExecutionModality>
76      </ActionGuarantee>
77
78    </ObligationGroup>
79
80  </Obligations>
81
82  </SLA>
```

Listing A.9: XML Instance of The Gas Alarm Message Collection Obligations.

Listing A.9 shows the construction of the Alarm Service Obligations. We have two Service Level Objectives (SLOs) at line 2 and 18. And the two corresponding Action Guarantees at line 34 and 60.

This finishes the presented SLA Document.

Bibliography

[1] Spectrum Requirements for Short Range Device, Metropolitan Mesh Machine Networks (M3N) and Smart Metering (SM) applications. *ETSI TC ERM, TR 103 055, v1.1.1*, September 2011.

[2] Alain Andrieux, Karl Czajkowski, Asit Dan, Kate Keahey, Heiko Ludwig, Toshiyuki Nakata, Jim Pruyne, John Rofrano, Steve Tuecke, and Ming Xu. Web services agreement specification (ws-agreement). In *Open Grid Forum*, volume 128, 2007.

[3] Tim Bray, Jean Paoli, C Michael Sperberg-McQueen, Eve Maler, and François Yergeau. Extensible markup language (xml). *World Wide Web Consortium Recommendation REC-xml-19980210. http://www. w3. org/TR/1998/REC-xml-19980210*, 1998.

[4] Guillaume Gaillard, Dominique Barthel, Fabrice Theoleyre, and Fabrice Valois. Service Level Agreement Architecture for Wireless Sensor Networks: a WSN Operator's Point of View. In *IEEE/IFIP Network Operations and Management Symposium (NOMS 2014)*.

[5] Alexander Keller and Heiko Ludwig. The WSLA Framework: Specifying and Monitoring Service Level Agreements for Web Services. *J. Netw. Syst. Manage.*, 11(1):57–81, March 2003.

[6] D Davide Lamanna, James Skene, and Wolfgang Emmerich. Slang: a language for service level agreements. 2003.

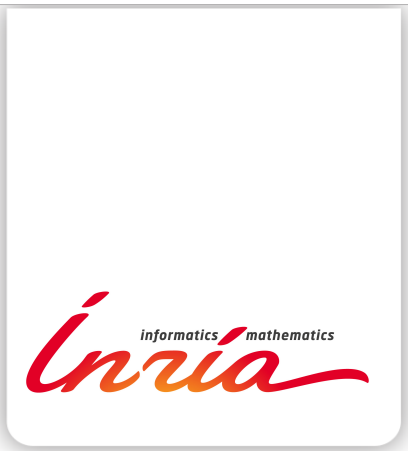
[7] Heiko Ludwig, Alexander Keller, Asit Dan, Richard P King, and Richard Franck. Web service level agreement (wsla) language specification. *IBM Corporation*, pages 815–824, 2003.

[8] Pascal Thubert et al. An Architecture for IPv6 Over Time Slotted Channel Hopping. IETF ID draft-thubert-6tsch-architecture (Work In Progress), 2013.

[9] D.C. Verma. Service level agreements on IP networks. *Proceedings of the IEEE*, 92(9):1382–1388, 2004.

[10] T. Winter, P. Thubert, and al. RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks. RFC 6550, IETF, March 2012.

[11] M. Aykut Yigitel, Ozlem Durmaz Incel, and Cem Ersoy. QoS-aware MAC protocols for wireless sensor networks: A survey. *Computer Networks*, 55(8):1982–2004, June 2011.



RESEARCH CENTRE
GRENOBLE – RHÔNE-ALPES

Inovallée
655 avenue de l'Europe Montbonnot
38334 Saint Ismier Cedex

Publisher
Inria
Domaine de Voluceau - Rocquencourt
BP 105 - 78153 Le Chesnay Cedex
inria.fr

ISSN 0249-6399