An XML-based Model for SLA Definition with Quality and Performance Indicators

Emir Toktar*, Guy Pujolle*, Edgard Jamhour, Manoel Camillo Penna, Mauro Fonseca

Pontifical Catholic University of Paraná, PUCPR, PPGIA, Rua Imaculada Conceição 1155, CEP 80215-901, Curitiba, Brasil.

***University of Paris VI, LIP6 Lab. 8, rue du Capitaine Scott, 75015, Paris.**

{fonseca, jamhour, penna}@ppgia.pucpr.br, emir.toktar@etu.upmc.fr, emir.toktar@computer.org, Guy.Pujolle@lip6.fr

Abstract. *This work proposes a XML-based model for defining service level agreements (SLA). The model has XML elements to define a semantic to represent key performance indicators (KPI) and key quality indicators (KQI) and the relationship between them. Upper and lower thresholds are associated to the indicators in order to indicate warnings or errors conditions. The relationship between the indicators is expressed by reusable functions which are evoked by the XML-based model. An example of reusable function for calculating the KQI service availability based on KPI indicators is also presented in this paper.*

1. Introduction

During the last years there have been a lot of research efforts on Quality of Service (QoS) management. These works have been strongly influenced by the Internet Engineering Task Force (IETF) specifications, from the early definition of management information for TCP/IP-based networks to the definition of a policy based management framework [1], including specifications of integrated service (*IntServ*) [2] and differentiated service (*DiffServ*) [3]. The related concepts are quite oriented to IP infrastructure management, and the underlying objective is to manage the network infrastructure, addressing issues as control of admission conditions and packet traffic monitoring. The ultimate goal is to assure the quality of information flow through IP links.

From a different perspective the objective of QoS management can be seen as assuring the quality of delivered services. The focus is not on defining network performance parameters but on finding quality indicators that reflect a customer-service provider common understanding of what QoS is. Although the involved concepts are the same, they are considered under a different point of view. Service Level Agreement (SLA) provides the basis for QoS management according this second perspective. It establishes a two-way accountability for service that is negotiated and mutually agreed upon by customer and service provider. An SLA defines a set of service level indicators that are not derived directly from the network but from a common view on what service quality means for both customer and service provider. This point of view intends to align QoS with business needs by seeking for service level indicators that reflects in improvement of business transactions.

However, when expressing QoS, service level indicators must reflect what is delivered by a particular network making necessary a relationship between service level indicators and network performance parameters. Moreover, when provisioning IP services with quality guaranties, negotiated service level must be taken into account, that is, resources should be configured at network level accordingly, in order to support QoS requirements. The mapping from SLA needs to network configuration is usually done manually and is a difficult and error prone task. Indeed, automatic translation of service level indicators into network configuration parameters is still an open problem. It requires taking into account proprietary procedures of specific network elements and the construction of a common semantic for multiple heterogeneous network elements across multi domain networks. Beside of this, another issue is how to relate network performance parameters with service level indicators. This mapping is not straightforward because the first evaluates quality of flow from the network perspective while the second reflects a negotiated view of service quality. An SLA information model that takes into account this issue is needed together with the algorithms that perform the necessary mapping.

The main concern of this work is related with mapping SLA indicators with network performance parameters. One objective is to present a XML based SLA model that uses Key Quality Indicator (KQI) and Key Performance Indicator (KPI) concepts for specifying service level indicators and network performance parameters, respectively, which includes the necessary elements to specify the translation from KQI into KPI.

The paper is structured as follows. Section two presents some related works to the SLA modeling issue. Section 3 describes the proposed model. Section 4 presents a function for calculation service availability, illustrating the main concepts presented by the proposed model. Finally, the section 5 presents the conclusions and points to future developments

2. Related Works

Several industrial and research works have been developed regarding Quality of Service (QoS) subject, using the SLA (Service Level Agreement) concept. Such a concept has been used for QoS management, including communication networks and information technology. This concept has been developed by several IETF contributions [3,4,5,6]. There are also proposals from several international projects, as presented below, and has been based on the EGEE publication [7], but we have include some additional works such as the proposals from the TMForum [8] and 3GPP [9].

- AQUILA [5]: The AQUILA project defined SLS (Service Level Specification) templates to standardize the requests of QoS between the customer and service provider, for the support of QoS in IP networks. The idea was to define SLS templates to simplify the process to translating SLS definitions to device configurations. In the AQUILA architecture, see figure 1, a reservation request is sent by the so called "*End-user Application Toolkit*" (EAT) to the "*Admission Control Agent*" (ACA). The reservation request specifies the required resources and QoS level and provides the information needed to identify the flow(s) to which the reservation applies. Resources are managed by the "*Resource Control Agent*" (RCA). These features were defined by semantic content of SLSs composed of the following attributes: SLS type, Scope, Flow identification, traffic description and conformance testing, Performance guarantees and Service schedule.

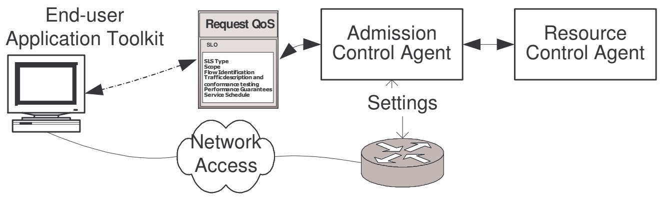

Figure 1. SLA Related Works

One goal of the standardizing SLS, was delivering templates of SLS avoiding mistakes and complexity in requests of QoS. The approach proposed was grouping IP applications by similar QoS behavior and requirements. This idea is similar to our model that defines templates for key performance indicators. The requests of QoS in AQUILA are a mixed of SLA and SLS concepts and the focus in the project was defining SLS templates on the negotiations between client and service provides. There is no common further development of the whole system.

- WSLA [10]: It is a XML based web services SLA Language. This model towards web SLAs specification and monitoring through Web Services technology. The Web Service Level Agreement language (WSLA) is defined as an XML Schema and covers the definition of the involved parties, the services guarantees and the service description, it comprises the following major parts: *parties*, *service definition* and *obligations*. ***Parties*** describes the parties involved in the management of the Web Services, including the signatory parties as well as the supporting parties that are brought into SLA to act on behalf of service provider or customer. This possibility structure of monitoring clauses separated from contractual terms for distribution to a third party is an interesting approach that our model not support in this version, but it'll be analyzed to accomplish this function in future. ***Service Definitions*** describe the services the WSLA is applied to, depicted the common understanding of the contract parties, in terms of operations, service's parameters and metrics. WSLA provides the ability to create news metrics defined as functions over existing metrics and this aggregation is helpful to formalize requirements expressed in terms of multiple QoS characteristics, for example, one operation *getQuote* of a Web Service

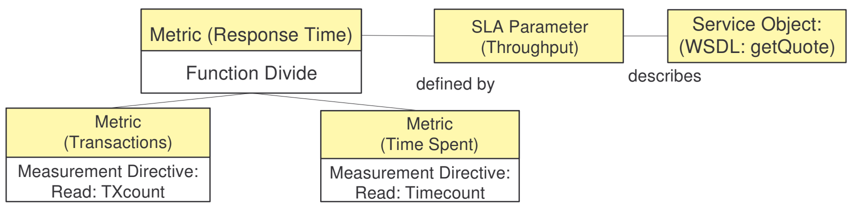

Figure 2. SLA Related Works

is defining by parameter *Throughput*, that has a metric, which, in turn, aggregates more metrics, see figure 2. Finally, ***Obligations*** defines the service level that is guaranteed with respect to *SLAParameters* defined in the service definition. This relation is similar to our relation approach between KPI and JQI. The WSLA can be applied for inter-domains management in business-oriented scenarios for the processes and general applications.

- CADENUS [11]: This project considers a configuration and provisioning integrated solution for end users QoS services. The Service/SLA Model uses specialized services classes and SLS models to compose a SLA. The SLOs (Service Level Objectives) are associated with classes: metric, QoS service and policy rules derived from CIM model. This project is frequently cited by other works and projects, however is not analyzed in [7].
- TEQUILA [12]: Addresses the modeling of DiffServ IP networks with provisioning and admission control. The TEQUILA is focused in intra-domain context where QoS-based IP service offerings are deployed over the whole Internet. This project presents the *DiffServ* specification in a layered model and discuss topics like SLA and SLS, moreover, its define SLS template. The major attributes of SLS template are: *Customer-userId*{Identifies the customer}; *Flow descriptor*{Packet stream (DSCP, IP addresses, etc)}; *Service Scope*{Geographical region (ingress–egress)}; *Service Schedule*{Specifies when the contract is applicable}; *Traffic descriptor*{ Traffic envelope (e.g. a token bucket)}; *QoS Parameters*{QoS guarantees (delay, jitter, packet loss)}; *Excess Treatment* {Traffic conditioning (dropping, remarking)}. Some TEQUILA works attempts to propose IETF standards for defining a SLS model and requirements for a negotiation protocol. This project is frequently referenced by other works and projects, however is not analyzed in [7].

Different approaches have been proposed, generating massive concept diffusion, nevertheless with an elevated divergence in the definitions and models adopted.

In the communication networks domain it is possible to identify two major approaches: (i) The first approach is oriented to the control mechanisms, i.e., provisioning networks elements in order to assure the QoS levels defined by the SLA. Examples of projects that follow this approach are: TEQUILA, QBone [13] and AQUILA. (ii) The second approach is oriented to the service relationship between customer and the service provider. It includes proposals for defining metrics for describing QoS levels and the mechanisms that could be used to guarantee these levels [14,15,16,17]. This is a general classification and does not have the intention to define a new taxonomy, but only to simplify the presentation of the works related to our proposal.

In this work the SLA concept is used in conformity with the second approach, being admitted the definition contained in the SLA Management Handbook of the TeleManagement Forum [18]: “A formal negotiated agreement between two parties, sometimes called a Service Level Guarantee. It is a contract (or part of one) that exists between the service provider and the customer, designed to create a common understanding about services, priorities, responsibilities, etc. (TMF 701 modified). An SLA or Contract is a set of appropriate procedures and targets formally or informally agreed between network operators/service providers (NO/SP) or between network operators/service providers and customers, in order to achieve and maintain specified Quality of Service (QoS) in accordance with ITU (ITU-T and ITU-R) Recommendations. The SLA may be an integral part of the Contract. These procedures and targets are related to specific circuit/service availability (SA), error performance, Ready for Service Date (RFSD), Mean Time Between Failures (MTBF), Mean Time to Restore Service (MTRS), and Mean Time To Repair (MTTR) (ITU-T Rec. M.1340)”. Another definition of SLA can be found in RFC2475 [2] and RFC3198 [6].

As the quality focus is on the service relationship involving client/provider, an important question is raised, in the way to consider adequate metrics for this relation. In this context the concept of key indicator appears to identify the essential metric for the relation client/provider. The concept is extended to facilitate the mapping between specific parameters of service and specific parameters of technology by two new indicators: key quality indicator (KQI) and key performance indicator (KPI).

These new pointers had been introduced by the TeleManagement Forum [19], and supply specific measurements aspects of application performance or service. The KQI is derived from a number of information sources, including metrics for calculating the performance of the service or derived from metric of underlying services as KPI. As a service or application is supported by a service elements number, a KPI different number can be indispensable to determine the calculation of a particular KQI. According to the TeleManegement Forum definition, the mapping between KQI and KPI is application dependent, and can be simple or complex, empirical or formal.

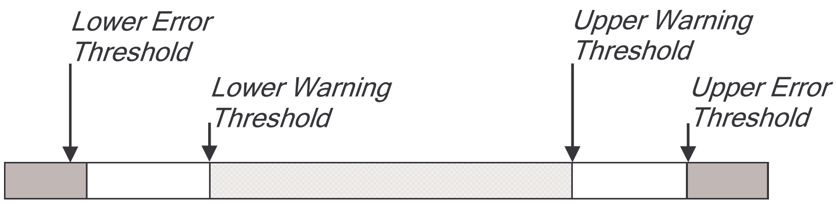

Figure 3. Threshold Parameters of KPI/KQI.

In general way a KQI is defined from a set of KPIs and each KPI or KQI will have upper thresholds and lower thresholds of warning ("Lower Warning/Upper Warning") and error ("Lower Error/Upper Error"), as shown in figure 3.

The KPI then are combined by empirical or theoretical function to calculate the value of the related KQI. Figure 4 illustrates this relationship of some parameters through a function f (P_1, P_2,...,P_n) and its relation with KQI parameters represented for a function F (S_1, S_n). For example, a set of KPIs values signaling warnings can degrade a service until it provokes the interruption, then, it would have to be considered as an error indicating a KQI violation.

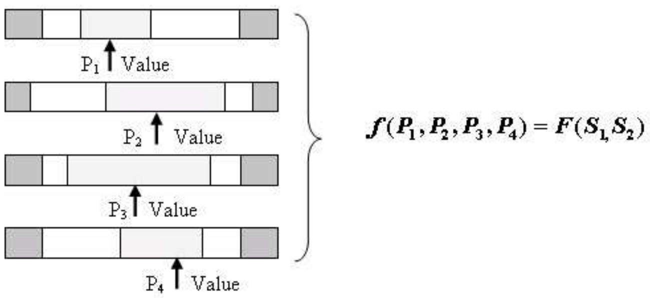

Figure 4. KPI combination for calculating a KQI.

The functions definition of relationship between KQI and KPIs is an important subject of current research. Considerable works in this way have been developed for entities as ITU (R-value) and other commercial entities as NetForecast (www.netforecast.com). In case that the relationships of KPI and KQI cannot be determined, measurements in real environments or laboratories can be developed. In this work we use references to indicators KPI and KQI in scheme XML. A corporative vision for applications, businesses, networks services, management and SLA negotiation can be gotten in the Management Handbook of the TeleManagement Forum [18].

3. Proposed Model with KPI and KQI Indicators

For many years the SLA management issue has been intensely researched and has achieved a reasonably matured state. However, the problem of establishing a generic inter-organizational SLA model remains without a definitive solution, even though one could observe some convergence to the use of XML for most proposal implementations.

The model proposed in this work is called KISLA (Key-Indicator SLA). It adopts the concept of describing service levels by using performance indicators (KPI) and defining functions for evaluating the KPI indicators with respect to the quality indicators (KQI). The proposed model combines elements described by the CADENUS [11] and WSLA [10] projects, and has proposed some additional classes as indicated in the model figures presented in this section.

The KISLA model is implemented in a XML-based language called KISLAML (Key-Indicator SLA Markup Language). In the KISLAML approach, XML-based elements are used for describing the entities and relationships related to a service level agreement. The KPI and KQI definitions and evaluation are preformed through calls to reusable functions embedded in the markup language. The functions itself are not described in XML, but they are just evoked by the KISLAML interpreter. This approach is inspired by the XACML (eXtensible Access Control Markup Language) [20], which is a language for representing policies using a descriptive approach. In our opinion, the XACML approach is more human-readable than adopting a purely object-oriented approach, such as CIM [21]. The idea of defining reusable functions permits to create policies where most semantic definition is defined by the policy language, instead of being implicitly defined by the policy interpretation algorithm.

Considering this aspect, the WSLA [10] approach has several similarities with our proposal. It also adopts UML for representing a SLA model, which is implemented in a XML-based language. The KPI and KQI concepts are not used, but according to the WSLA, a service has SLA parameters, defining metrics which are expressed or evaluated in terms of cascaded functions. The possibility of creating metrics using cascaded functions is not explicitly supported by the current KISLAML version.

The model proposed by CADENUS [11] for service definitions has inspired the reuse of service classes in our proposal. CADENUS has defined reusable SLA and SLO (Service Level Objectives) templates for service model definitions, but has also opened the possibility of creating customized services. The CADENUS model has imported several CIM class definitions in order to describe metrics, QoS services and policy rule definitions. In our proposal, this CIM classes have been replaced the KPI and KQI definitions.

Next, it will be presented a short description of the proposed KISLA model and the relationship between the KPI and KQI indicators. The <SLAContractType> class is the main element of the KISLA model, and it aggregates three main classes: <Parties>, <Services> and <Responsabilities> (see Figure. 5).

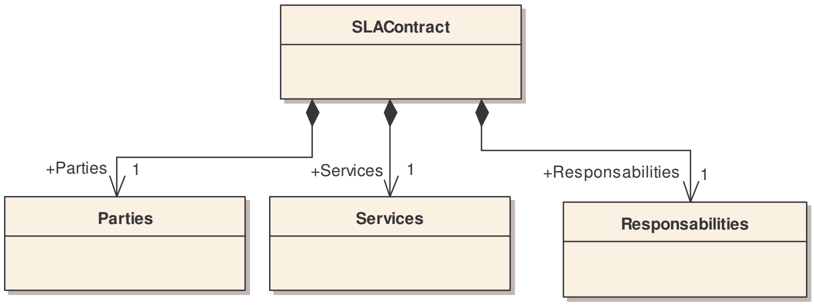

Figure 5. Proposed SLA Model.

In order to explain the model, let's consider the following example: "a provider agrees with a customer to supply a specific audio service (e.g., CODEC G.911) and classifies this service as being GOLD, within a commercial period. It also defines a service availability of 98% within the contracted period".

The <Parties> class describes the entities involved in the SLA agreement, i.e., the customer and the provider. It aggregates other classes used for describing information such as name, phone numbers, addresses, email and other data related to the parties in the agreement. In a typical SLA model, parties establish an agreement, describing the responsibilities for the engaged services. There are several proposals for modeling parties, services and other elements of a SLA agreement. In particular, the WSLA version 1 has no elements for defining the contact information for the SLA parties, but it suggests, for the following versions, the use of international standards for representing this information. In our proposal, SLA parties information is expressed using the classes proposed by the CIM version 2.9 [21].

The <Services> class represents the information about the offered service, in this case CODEC G.911, with service level GOLD. It defines the topology of the service (i.e., end points of the provider domain) and the validity period (e.g., commercial period) of the SLA contract. The service level GOLD is defined in terms of the key performance indicators (KPIs) required for assuring the performance of the offered service. Delay, jitter, packet loss and average bit rate are examples of KPIs.

The <Responsabilities> class represents the conditions that must be respected by the provider with respect to the offered service. These conditions are expressed in terms of quality key indicators (KQI), which are used for defining the terms under which the offered service will be monitored and evaluated. The KQI are expressed, preferentially, but not exclusively, as a function of the performance key indicators (KPIs). Availability is an example of KQI, and can be defined in terms of the KPIs delay, packet loss and average bit rate. Other information such as penalties for violating a KQI is also represented by the <Responsabilities> class.

Figure 6 presents the classes related to the <Parties> model. As shown in the figure, most elements have been imported from the CADENUS, WLSA and CIM definitions. The <Parties> class aggregates elements for representing the entities that negotiate a service level agreement: provider and customer. Both parties are represented by the <SignatoryParty> class, which is a specialization of the abstract class <Party>. As shown in the figure, the party information is represented by the CIM version 2.9 classes [21].

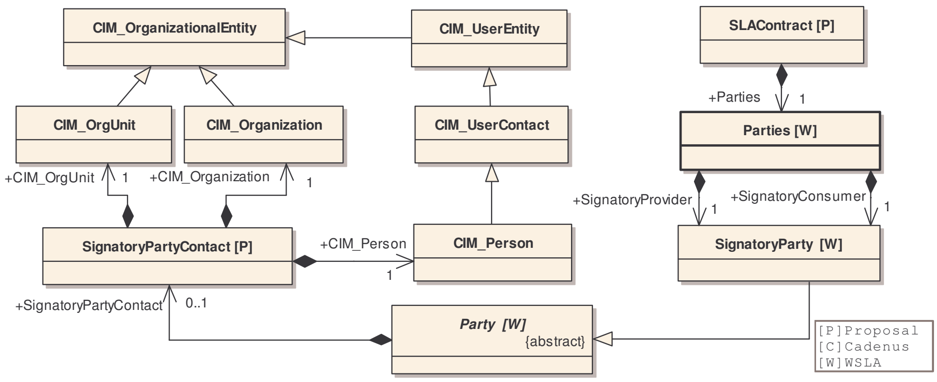

Figure 6. "Parties" Definition.

Figure 7 presents the classes related to the <Services> model. A SLA can offer several services through the <Service> cardinality. Examples of services are "CODEC G729", "CODEC H263", “generic audio”, etc. A service definition includes the Service Access Point (SAP) and the Key Performance Indicators (KPI) definitions. Each service level defined by the provider, e.g., audio GOLD or audio SILVER, is defined in terms of a KPI set. The KPI set indicates the performance levels that must be satisfied by the provider, and are constrained to a validity period defined by the <Schedule> class. Note that GOLD has no meaning if not associated to the service "audio". Therefore, a KPI set is always related to a service. An example of KPI set is {delay, packet loss and bit rate}.

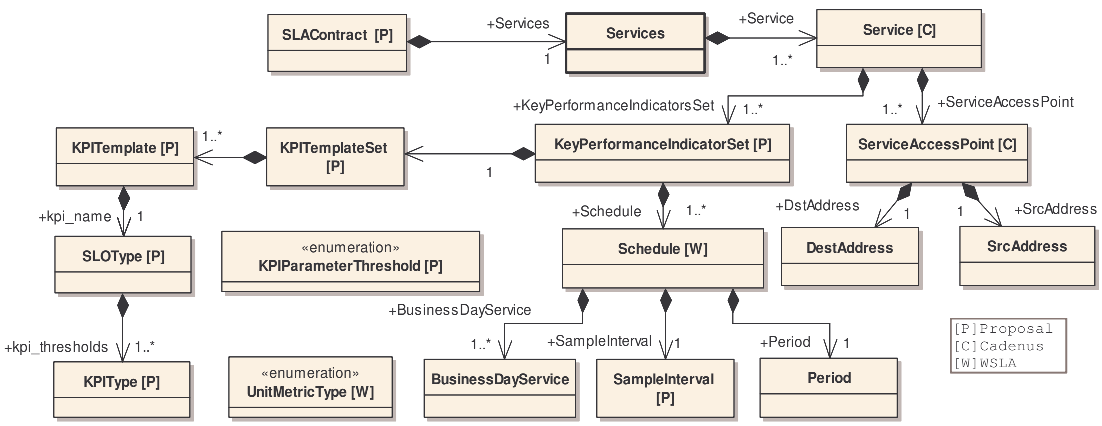

Figura 7. “Services” Definitions.

In order to simplify the process of defining KPI sets for services offered by a provider, the KISLA model proposes the use of XML schemas for automatic validating the KPI definitions. The KPITemplateSet is associated to a XML schema that can be selected from a library, when a service is instantiated. KPITemplateSet defines a set of reusable KPIs that can be used for describing similar services. For example, the family CODEC G defined by the ITU-T, specifies a set of CODEC specifications with similar parameter definitions. Therefore, CODECG is an example of KPITemplateSet and

CODEC G729 is an example of KPITemplateSet instantiation. The idea of creating KPI templates has been adapted from the concept proposed by CADENUS.

Figure 6 illustrates the structure of a KPITemplateSet for the ITU-T CODEC G series. Note that template define three KPIs: <Packet_Loss>, <Jitter> and <Delay>, which are used to assure the "audio" quality, according to the TMForum definition for some CODECs types [22]. To each element is associated a <SLOType>, which defines the thresholds associated to the KPI. As explained in the section two, each KPI can be associated up to 4 thresholds, representing lower and upper levels for warning and error events. The acceptable metric units for each threshold are also defined in the schema.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:kpi="urn:kpi:template:set:1.0"
targetNamespace="urn:kpi:template:set:1.0" elementFormDefault="qualified"
attributeFormDefault="unqualified">
  <xs:complexType name="KPITemplateType">
    <xs:sequence>
      <xs:element name="Packet_Loss" type="kpi:SLOType"/>
      <xs:element name="Jitter" type="kpi:SLOType"/>
      <xs:element name="Delay" type="kpi:SLOType"/>
    </xs:sequence>
    <xs:attribute name="KPITemplate_Id" type="xs:anyURI" use="required"/>
  </xs:complexType>
  <xs:complexType name="SLOType">
    <xs:sequence>
      <xs:element name="unit" type="kpi:UnitMetricType"/>
      <xs:element name="kpi_thresholds" type="kpi:KPIType" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="KPIType">
    <xs:sequence>
      <xs:element name="threshold-parameter" type="kpi:KPIParameterThresholdType"/>
      <xs:element name="threshold-value" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
  <xs:simpleType name="KPIParameterThresholdType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="LowerErrorThreshold"/>
      <xs:enumeration value="LowerWarningThreshold"/>
      <xs:enumeration value="UpperWarningThreshold"/>
      <xs:enumeration value="UpperErrorThreshold"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:simpleType name="UnitMetricType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="urn:kpi:template:set:1.0:unit:none"/>
      <xs:enumeration value="urn:kpi:template:set:1.0:unit:milliseconds"/>
      <xs:enumeration value="urn:kpi:template:set:1.0:unit:seconds"/>
      <xs:enumeration value="urn:kpi:template:set:1.0:unit:percent"/>
      <xs:enumeration value="urn:kpi:template:set:1.0:unit:bps"/>
      <xs:enumeration value="urn:kpi:template:set:1.0:unit:packets"/>
      <xs:enumeration value="urn:kpi:template:set:1.0:unit:units"/>
    </xs:restriction>
  </xs:simpleType>
```

Figure 8. KPITemplate schema for the ITU-T CODEC G series definition.

As an example, the template illustrated in Figure 8 could be used for defining the KPI set for audio applications according to the ITU-T standard CODEC G.729. The standard specifies delay limited to 150 ms, jitter limited to 50 ms and packet loss limited to 0.5% of the transmitted packets. These KPI values specify the conditions the provider must assure for the customer. In order to satisfy these conditions, the provider must translate these parameters into SLS configurations, and apply them to the network devices that are affected by the ServiceAccessPoint associated to the service. For example, the SLS configuration for the G.729 service correspond to a “token bucket” [23] with the following parameters (rate=2000, bucket=80, peak=4000). The translation process for calculating the SLSs form the SLA definitions is out of the scope of this paper.

Figure 9 illustrates the instantiation of the "Delay KPI", which corresponds to one of the KPIs illustrated in the schema in the figure 6. In the example, there are only upper threshold, since lower threshold make no sense for delay. A warning is defined for delay measures between 130 and 150 milliseconds. All delay measures above 150 milliseconds are considered error events.

```
<kpi:Delay>
  <kpi:unit>urn:kpi:template:set:1.0:unit:milliseconds</kpi:unit>
  <kpi:kpi_thresholds>
    <kpi:threshold-parameter>UpperErrorThreshold</kpi:threshold-parameter>
    <kpi:threshold-value>150</kpi:threshold-value>
  </kpi:kpi_thresholds>
  <kpi:kpi_thresholds>
    <kpi:threshold-parameter>UpperWarningThreshold</kpi:threshold-parameter>
    <kpi:threshold-value>130</kpi:threshold-value>
  </kpi:kpi_thresholds>
</kpi:Delay>
```

Figure 9. Example of thresholds for "*Delay*" KPI.

Figure 10 presents the classes related to the <Responsabilities> model. This model defines the quality indicators (KQIs) that must be satisfied by the provider, with respect to the accorded service, and the respective penalties applied when these KQIs are not respected. The two main classes of this model are: <KeyQualityIndicator> and <Duties>.

The <KeyQualityIndicator> class defines how a KQI is calculated in terms of the KPIs indicated by the <KeyPerformanceIndicatorSet> class. The <Schedule> class defines the periodicity that the KQI is evaluated (e.g., daily, weekly, etc.). The KQI evaluation function <KQIFunction> is defined in a library and evoked by the KISLA markup language interpreter. As mentioned early in this section, the strategy of creating conditions based on reusable expressions was inspired by the XACML approach.

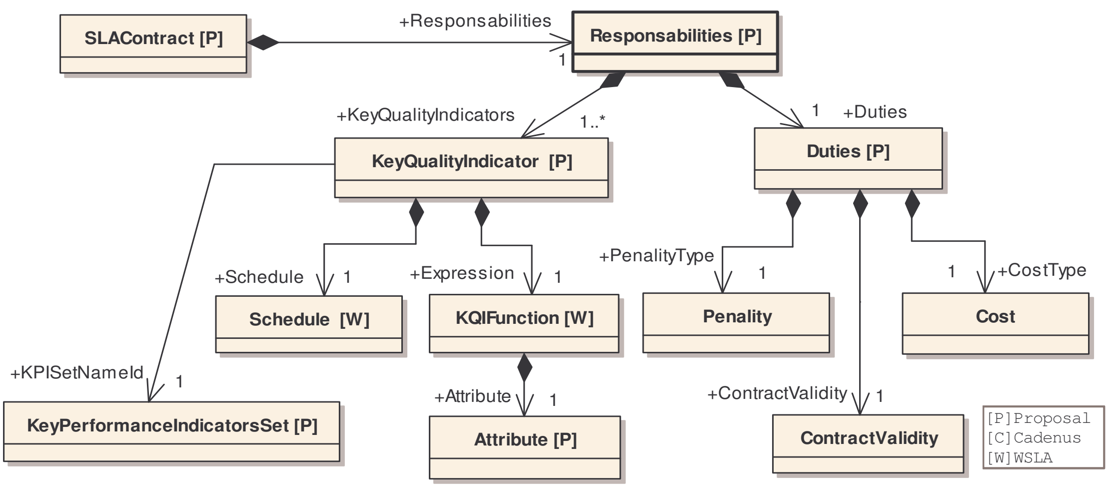

Figura 10. KQI and KPI relationship in the proposed model.

The <Duties> class defines the penalties applied to the provider when the <Responsabilities> are violated. The penalties are described by the <Penalty> class, and can be calculated as a function of the fees paid by the customer for the service, as defined by the <Cost> class.

The definition of KQI functions is an important step for evaluating a SLA. In the next section it will be presented an example of KQI function for calculating the availability of a service.

4. An example: assessment of service availability

As discussed in section 2, KQI are suitable metrics for defining service level under a perspective that takes into account an accorded view of service quality established by customer and service provider. Examples of KQI are Service Availability (SA), Mean Time Between Failures (MTBF) and Response Time (RT) [18]. On the other hand, KPI are technical indicators adequate for measuring the performance of underlying network. Examples of KPI are packet loss, delay and jitter. In general KPI are evaluated directly from performance data collected from the network (KPI data), and KQI are computed from specific combination of KPI data values.

QoS expressed by a KQI must be clearly understood by both parties, but misunderstanding is very common. For example, SA is a well-known indicator that has an understood formula, which indicates the percentage of time the service is operational:

$$SA\% = 100\% - \left(\frac{\sum OutageInterval}{ActivityTime} x100\% \right) \quad \textbf{(equation 1)}$$

Outage period can be computed by summing up all intervals when service is disrupted. Customer tends to consider that all periods having any kind of problem should be considered as an outage interval, wherever service provider tends to do not consider the periods when there is just some level of service degradation. If both customer and service provider can agree on what is service disruption and what is service degradation, degradation period can be similarly computed by summing up all degradation intervals. Additionally service degradation factor (SDF) complements SA formula, where SDF varies from 0 to 1, depending on the weight given to degradation period when computing SA, as follows:

$$SA\% = 100\% - \left[\left(\frac{\sum OutageInterval + \left(SDF \times \sum DegradationInterval \right)}{ActivityTime} \right) x100\% \right] \quad \text{(equation 2)}$$

Outage and degradation intervals are defined by the value of a KPI data sample. Let Pt be a KPI sample at instant t. As mentioned in section 2 a KPI has four thresholds, lower and upper error, and lower and upper warning. We have an outage interval when Pt is less then under lower error threshold and greater then upper error threshold. Similarly, we have a warning interval when Pt is between lower error and lower warning, or between upper warning and upper error. We can then calculate the outage period by summing up all error intervals and the degradation period by summing up all warning intervals (see Figure 3).

We have to consider multiple KPIs, possibly with different sampling rates, when computing a KQI. Thus, the first step is to get a common sampling period, that is, their greatest common divisor. Let $\alpha = \{ KPI_i \mid i = 1 \ldots N \}$ be the set of KPIs related to SA% KQI. Figure 11 illustrates the procedure for getting a common sampling interval. Tstart and Tend are the initial and final instant of activity time. D is the common sampling interval, that is, D = GDC(di) I = 1…N, where di is the duration of sampling interval of KPIi. Pnk is the value of a sample for KPIn in its k-th sampling interval. Pn is the set of all KPIn samples.

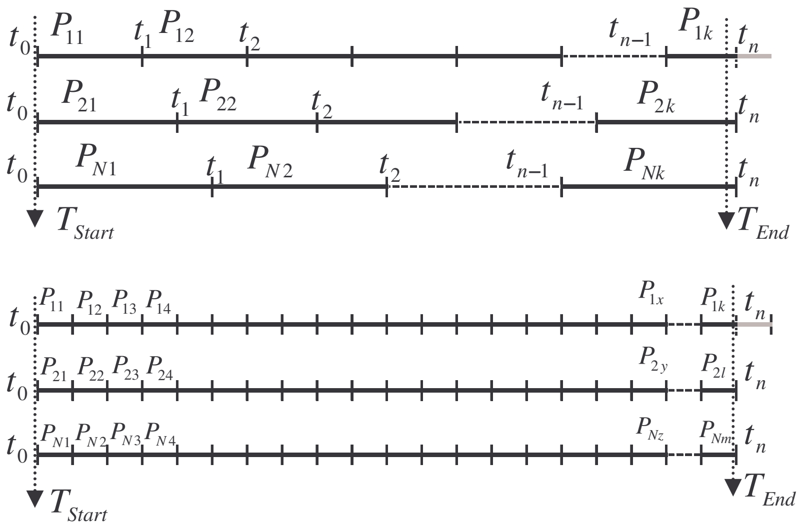

Figure 11. Sample terms of KPIs.

KPIs are measured in different scales. The second step is to get a common scale by normalizing sample values (P_{nk}) with respect to KPI thresholds. The values between *Lower Warning*$_n$ and *Upper Warning*$_n$ are set to 0 (zero); those between *Lower Error* and *Lower Warning* are set to –1 (minus one); those between *Upper Warning* and *Upper Error* are set to 1 (one); those under *Lower Error* are set to –2 (minus two); and those above *Upper Error* are set to 2 (two). P'_{nk} is the normalized value of a sample for KPI_n in its k-th sampling interval.

```
for (n ← 1  to  N) do
    for (k ← 1 to | Pn |) do
        P´nk ← (Pnk > UpperErrorn)?2 : ((Pnk > UpperWarningn)?1:Pnk));
        P´nk← (Pnk< LowerErrorn)? -2 :  ((Pnk < LowerWarningn)?-1 : Pnk));
        P´nk ← (Pnk > LowerWarningn) and (Pnk < UpperWarningn) ? 0;
    end-for ;
end-for ;
```

Figure 12 illustrates the normalization procedure. It shows in the left, samples before normalization procedure, and in the right, after normalization. In this example, P_n has only values greater then *Lower Warning*.

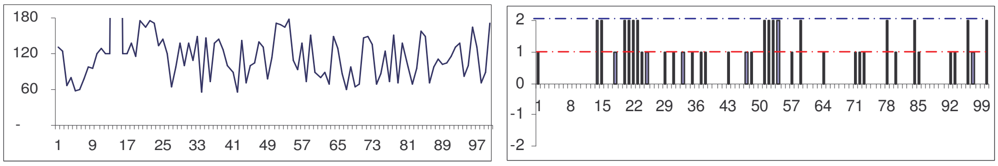

Figure 12.KPIs samples qualified.

Lets Q_k be the value of a KQI sample at interval I_k. Q_k should be calculated from P_{ik}. The combination of multiple P_{ik} into Q_k can be done according to two principles: (i) Q_k will express an exception condition (warning or error) when **all** P_{ik} under consideration are in exception condition, prevailing the less severe one. (ii) Q_k will express an exception condition (warning or error) when **at least one** P_{ik} is in exception condition, prevailing the severest one. In order to combine P_{ik} to compute Q_k we define two operators: $\oplus$ and $\otimes$, according to the following decision table, see table 1:

Table 1 – Decision table for ⊗ and ⊕ operators

$\lvert P_{ik}\rvert$	$\lvert P_{jk}\rvert$	$P_{ik} \oplus P_{jk}$	$P_{ik} \otimes P_{jk}$
0	0	0	0
0	1	0	1
0	2	0	2
1	0	0	1
1	1	1	1
1	2	1	2
2	0	0	2
2	1	1	2
2	2	2	2

It is possible to show that $\oplus$ and $\otimes$ have the same properties than logical operators $\vee$ and $\wedge$. Thus, a generic combination of P'_{ik} can be given by an expression evolving $\otimes$ and $\oplus$ in the Disjunctive Normal Form (DNF). Let's consider the following definitions:

- $\alpha = \{ KPI_n \mid n = 1 \ldots N \}$ the set of KPIs related to SA% KQI.
- $\alpha_1 \ldots \alpha_W$: the subsets of α, such that, $\{\alpha_1 \ldots \alpha_W\}$ is a partition of α, that is $\bigcup_{i=1\ldots W} \alpha_i = \alpha$ and $\bigcap_{1=1\ldots W} \alpha_i = \varnothing$.
- $N_i = \lvert \alpha_i \rvert$ (cardinality of α_i).

$$Q_k = \underset{i=1..W}{\oplus} \left(\underset{j=1\ldots Ni}{\otimes} \lvert P'_{jk} \rvert \right) \text{ where } KPI_j \in \alpha_i \text{ (equation 3)}$$

Q_k is computed by equation 3, outage period can be computed by summing up errors samples ($Q_k = 2$) and degradation interval by summing up warning samples ($Q_k = 1$). SA% is computed by equation 2. We can come back to the model and use a XML schema to assess the SLA. Figure 13 shows an example defining a service level of 98% for SA%, with a SDF of 0.7.

As said before, outage interval can be computed by summing up errors samples and degradation interval from warning samples. SA is computed by equation 2. Figure 13 presents an XML schema that assesses a service level of 98% for a 0.7 SDF.

```xml
<KeyQualityIndicators kqi_Id="Availability">
  <KPI_NameId>KPIGoldCodec</KPI_NameId>
  <KQIFunction FunctionId="urn:kisla:1.0:policy:operator:greater-than-or-equal">
    <MetricSourceValue DataType="http://www.w3.org/TR/2005/WD-xpath20-
                                              20050211#SimpleFullXPointer">
      <SourceValue>uri-reference#xpointer(expr)</SourceValue>
    </MetricSourceValue>
    <Attribute DataType="http://www.w3.org/2001/XMLSchema#integer">
      <AttributeValue UnitType="urn:kisla:1.0:policy:unit:percent">98</AttributeValue>
    </Attribute>
    <Schedule name="AvailabilityMonthly">
      <Period>
        <Start>2005-10-01T09:30:47.0Z</Start>
        <End>2005-12-31T18:00:00.0Z</End>
    </Period>
    </Schedule>
    <GroupSet>
      <Group FunctionId="urn:kisla:1.0:policy:operator:and">
        <Group FunctionId="urn:kisla:1.0:policy:operator:and">
          <Group FunctionId="urn:kisla:1.0:policy:operator:or">
            <AttributeValueKPI>Delay</AttributeValueKPI>
            <AttributeValueKPI>Jitter</AttributeValueKPI>
          </Group> <!-- Group1(P1 or P2) -->
          <Group FunctionId="urn:kisla:1.0:policy:operator:or">
            <AttributeValueKPI>Packet_Loss</AttributeValueKPI>
            <AttributeValueKPI>PacketErrorRate</AttributeValueKPI>
          </Group> <!-- Group2(P3 or P4) -->
        </Group>
      <Group FunctionId="urn:kisla:1.0:policy:operator:and">
        <AttributeValueKPI>Throughput</AttributeValueKPI>
        <AttributeValueKPI>Delay</AttributeValueKPI>
      </Group>   <!-- Group2(P3 or P4) -->
    </Group> <!-- (Group3)and(Group4) -->
  </GroupSet> <!-- GroupSet= [ G3(G1(P1 or P2) and G2(P3 or P4) ) and G4(P5 or P6) ] -->
</KQIFunction>
```

Figure 13. SAL assessment with XML schema.

5. Conclusion

Defining a model for represent SLA information is an important step for creating management tools for automating the process of provisioning and monitoring DiffServ networks. The work in this paper presents a proposal towards a unified model for representing SLAs called KISLA. The model adopts the concept of representing SLAs by using the KQI and KPI indicators. This approach shows a great flexibility for describing a large number of services negotiated in a DiffServ domain. It provides means for defining a common nomenclature for negotiations between customers and providers.

This paper also presents a markup language for implementing the KISLA model. The KISLA markup language was inspired by the experience acquired by building policy-based management systems based on both, purely object-oriented approaches, such as CIM, and hybrid-approaches, capable of evocating external reusable functions, such as XACML. We have seen advantages in the hybrid approach, because it permits to create policies where most semantic definition is defined by the policy language, instead of being implicitly defined by the policy interpretation algorithm. This approach was particularly useful for adapting the model to the KPI and KQI definitions.

The work in this paper has developed only the “availability” KQI function, applicable to a limited set of KPI indicators. Future works will explore the development of other KQI functions. The idea is to create a library of quality indicator functions that could be useful for creating SLA contracts for a large number of applications.

6. References

[1] Yavatkar, R., Pendarakis, D.; Guerin, R. A Framework for Policy-Based Admission Control, RFC2753, Jan. 2000.

[2] Blake, S.; et al. “An Architecture for Differentiated Services”, IETF Proposed Standard, RFC2475, Dec 1998.

[3] Braden, R.; Clark, D.; S. Shenker, Integrated Services in the Internet Architecture: An Overview, RFC1633, Jul. 1994.

[4] Rajan, R.; Celenti, E.; Dutta, S.; “Service Level Specification for Inter-domain QoS Negotiation”, draft-somefolks-sls-00.txt, Internet Draft, Nov. 2000

[5] Salsano, S.; Ricciato, F.; Winter, M.; Eichler, G.; Thomas, A.; Fuenfstueck, F.; Ziegler, T.; Brandauer, C.; “Definition and usage of SLSs in the AQUILA consortium”, draft-salsano-aquila-sls-00.txt, Internet Draft, Nov., 2000. AQUILA project website. URL: http://www.ist-aquila.org/

[6] Westerinen, A.; et al. “Terminology for Policy-Based Management”, IETF, RFC 3198, Nov. 2001.

[7] EGEE, “Enabling Grids for E-science in Europe”. URL http://www.eu-egee.org

[8] TeleManagement Forum. URL http://www.tmforum.org

[9] 3GPP. 3rd Generation Partnership Project (3GPP). URL: http://www.3gpp.org

[10] Ludwig, H.; Keller, A.; Dan, A.; Franck, R.; King, R.P. “Web Service Level Agreement (WSLA) Language Specification”. IBM Corporation, July 2002.

[11] CADENUS project. References at URL: http://wwwcadenus.fokus.fraunhofer.de

[12] Goderis, D.; et al. “D1.1: Functional Architecture Definition and Top Level Design”, TEQUILA Project “Traffic Engineering for Quality of Service in the Internet, at Large Scale” (IST-1999-11253), 11 Sep. 2000.

[13] QBone. "QBone Bandwidth Broker Architecture”. http://qbone.internet2.edu/bb/bboutline2.html.

[14] Tosic, V.; Pagurek, B.; Esfandiari, B.; Patel, K. “Management of Compositions of E- and M-Business Web Services with multiple Classes of Service”. Proceedings of the 8th IEEE/IFIP NOMS 2002, Florence, Italy, Apr. 2002.

[15] Bhoj, P.; Singhal, S.; Chutani, S. “SLA Management in Federated Environments”. Proceedings of the 6th IFIP/IEEE Symposium on IM’99, Boston, MA, USA, May 1999.

[16] Keller, A.; Kar, G.; Ludwig, H.; Dan, A.; Hellerstein. J.L. “Managing Dynamic Services: A Contract based Approach to a Conceptual Architecture”. Proceedings of the 8th IEEE/IFIP NOMS 2002, Florence, Italy, Apr. 2002.

[17] Sahai, A.; Machiraju, V.; Sayal, M.; Moorsel, A.; Casati, F. “Automated SLA Monitoring for Web Services”, 8th IFIP/IEEE Network Operations and Management Symposium, NOEMS 2002 . Italy, Apr. 2002.

[18] TeleManagement Forum. “SLA Management Handbook, Volume 4, Enterprise Perspective”, G045. The Open Group. Oct 2004.

[19] TeleManagement Forum. “Wireless Service Measurements Handbook”, GB923. Last release - GB923v3, Mar. 2004.

[20] OASIS. “eXtensible Access Control Markup Language (XACML) Version 1.0”. OASIS, Feb. 2003. URL http://www.oasis-open.org/committees/xacml/

[21] DMTF. “Common Information Model (CIM) Specification v 2.9”. Distributed Management Task Force, Inc. Jun. 1999.

[22] EGEE, “Specification of Interfaces for Bandwidth Reservation Service”, EU Deliverable: DJRA4.1, JRA4: Development of Network Services. URL https://edms.cern.ch/document/501154/1.

[23] Shenker, S.; Partridge, C.; Guerin, R. Specification of Guaranteed Quality of Service, RFC 2212, Sep. 1997.