

# IPTV Service Modeling in Magneto Networks

Sidath Handurukande  
Network Management Lab  
LM Ericsson, Athlone, Ireland  
Email: sidath.handurukande@ericsson.com

Stefan Wallin and Andreas Jonsson  
Data Ductus Nord AB  
SE-931 31 Skellefteå Sweden  
Email: stefan.wallin@dataductus.se, andreas.jonsson@kreiblo.se

**Abstract**—One of the main steps of service assurance is service monitoring using Key Performance Indicators (KPIs) and Service Level Agreements (SLAs). We show an approach for service modeling, first starting with an abstract service model that depends on the network. And then, we show how a corresponding model can be realized using a domain specific language. This solution is able to condense various sources of service model requirements into a condense formal and executable model including service decomposition and KPI aggregation. We have described this solution in the context of Magneto project and uses IPTV as a service in our description.

## I. INTRODUCTION

Modern day telecom and Internet based services are complex and large in scale. These services are generally built on top of shared network resources and they compete for common network resources. In addition, a new paradigm has emerged where content is provided from one user to another, as opposed to the traditional model, where content is centrally stored and fetched from the users at the edge of the network (e.g., Home Area Networks -HAN). The edge of the network (which includes HANs) has become important in terms of (1) configuration and performance management of the edge to deliver good quality complex services (2) edge being a demanding “content provider” (e.g., delivering high quality live video to other users). The management of outer-edge (HAN and access network) and end-to-end (one edge to another) service assurance has become very important. The Magneto project [1] aims to develop enabling technologies for outer-edge management and end-to-end service assurance. Even though there are many research facets in the Magneto project, in this paper we only consider end-to-end service modeling and its usage for end-to-end service quality monitoring.

Our solution leverage on managing services rather than the network resources which are used to deliver the services. Managing services rather than resources helps to tackle increased competition and handle more complex, distributed service offerings. It also helps to evaluate and fulfil Service Level Agreements [2] elegantly. However, the complexity of understanding and modeling services is a serious obstacle.

Section II describes the Magneto IPTV media sharing scenario. The challenges around service models and our approach to modeling is presented in Section III. We show the value of KPI aggregation and service decomposition in an abstract IPTV model in Section IV. Section V illustrates the expressive power of a domain specific language for service modeling. This approach is able to formalize informal models, and

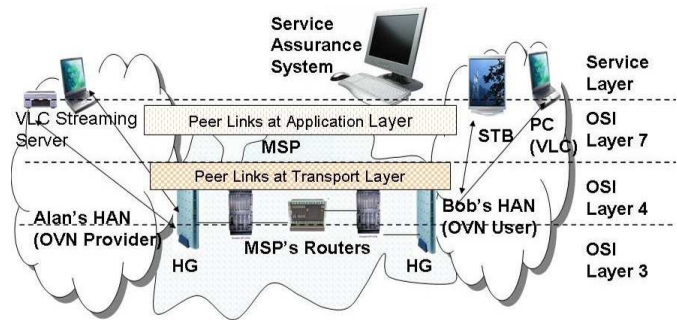


Fig. 1. The Magneto IPTV Delivery Scenario

produce semantically rich models that are directly executable in a dedicated model engine. In this way we close the gap between informal model specifications and the later stages of model semantics embedded in implementations.

## II. THE MAGNETO IPTV SCENARIO

The solutions developed in Magneto project facilitate easy, reliable, secure distribution and access of content available in HANs. Ease of use, reliability and security comes as a value added service on top of normal ISP services. The provider of this value added services is known as Magneto Service Provider (MSP). In general, MSPs could be a “virtual provider” or a given ISP could provide value added Magneto services. Due to lack of space and for descriptive purposes we only show a simplified network topology in Figure 1 that depicts the Magneto media sharing scenario. In this simplified network a given ISP connects two HANs and also acts as the MSP.

As shown in Figure 1 one HAN (Alan's) streams live video to another (Bob's). The “virtual” network that spans from Alan's HAN to Bob is the Magneto network and is known as Omnipresent Virtual Network (OVN). The OVN is a value added service provided by MSP and it is a virtual network similar to the concept of VPN but provides additional functionalities like ease of use, high reliability and security. Based on service modeling approaches, MSP will monitor and analyze the end-to-end overall performance of the service, for example in this case the IPTV service. The Figure 1 shows how a VideoLAN media player (VLC) [3] server streams IPTV content to a set-top-box (STB) and to a computer with VLC

(PC) in another HAN. Alan's HAN which provides media content is known as OVN provider HAN and content consumer HAN is known as OVN user HAN. These HANs are connected to MSP using Home-Gateways (HG) such as ADSL modems.

### III. SERVICE MODELING AND CHALLENGES

There are a number of goals of service modeling [4], [5]; the main goals are to (1) express what a service can do (2) monitor and analyze the quality of the service (QoS) and its overall performance (3) understand, evaluate services and for agreement of contracts by customers of services (4) design and compose services (5) use as a blueprint for deploying and managing services. In our solution, we use service models to monitor and analyze QoS and end-to-end performance of a service.

A form of service models, known as service topology, represents dependencies between resources and services. Basically, these models do service decomposition and depict how a service is built using lower level service components. Additionally, these service models will also include (1) higher level service-KPIs [6] (S-KPI) which represents how a higher level service perform (2) lower level network/resource KPIs which indicate how a lower level network service perform (3) service levels (SL) (service goals/objectives) (4) SLAs and (5) various relationships between network resources, KPIs, end-user service, S-KPIs, users and SLAs. This representation is very useful and important for operating and managing services, especially when there are a large number of inter-dependant services and resources. However the challenge is to formalize informal models and then make these formal models machine executable.

### IV. THE MAGNETO APPROACH TO SERVICE MODELS

Figure 1 illustrates the topology which we now will formalize into a service model. We will focus on the parts of the model that are close to the user. Our model is primarily based on the DSL Forum report on Triple-play Services QoE Requirements [7]. The report structures quality of service into three layers:

- *Service Layer*: The layer exposed to the user, where Quality of Experience (QoE) is measured.
- *Application Layer*: Where various application parameters are managed, for example media resolution and codec errors.
- *Transport Layer*: where network impairments such as loss, delay, and jitter may occur.

Figure 2 illustrates the classes in our IPTV model. We have two major classes in the *Service Layer*: IPTVUserService and IPTVProviderService. The IPTVUserService represents the status and QoS/QoE for an individual end-user and the latter aggregates user services into an overall IPTV service quality from a virtual provider perspective. The IPTVUserService is decomposed into its application and network components by the RTSPSession, (Realtime Streaming Protocol [8]), and IPConnectivity classes.

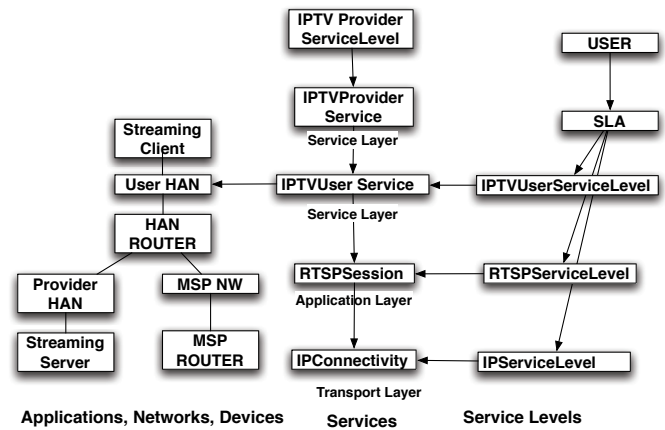


Fig. 2. Sample Model

It is important to formulate what is “good or bad” Quality Of Service. This is needed both to be able to write SLAs between users and providers and also to detect degraded service performance in an overall service assurance perspective. Whether a service is good or bad depends on contexts like business agreements, location, and network access. Therefore it is necessary to separate the criteria whether a service is good or bad from the actual service class. This is the purpose of Service Levels classes to the right in Figure 2. Service levels verify that each service performs within configured thresholds.

A service model is functional in the sense that it takes inputs from the external world and calculates KPIs. So the availability and quality of inputs to the model is essential. In the Magneto context we are using a mix of KPI inputs from sources:

- A distributed event processing component [9] aggregates low level events into high-level events like quality impairment counters.
- Probabilistic reasoning [10] performs root-cause analysis and is able to report service component problems.
- Data from the devices for example using DSL Forum TR-069 [11]
- IP network performance using active monitoring methods and passive monitoring.)

### V. MODEL IMPLEMENTATION WITH SALMON

We express our model in the SALmon environment [12] which is a tailor-made programming for service models. The language is a simple functional language for defining KPI calculations and supports object-oriented structuring of the service model. Due to the nature of service modeling, the programming language must be able to treat time as part of the normal syntax: all variables are seen as arrays indexed by a time stamp.

The language has two fundamental layers: the Definition Layer and the Instantiation Layer. The *definition layer* defines the classes and calculations in the model. Classes have *inputs*; time-indexed variables mapped to external data sources, *anchors*; connections to other class instances, *attributes*; calcula-

tion rules for parameters (KPIs) in a functional language and *properties*; values assigned at instantiation. The *instantiation layer* creates instances of the service classes, assigns properties and establishes connections between instances through anchors.

#### A. SALmon IPTV Definition Layer

We will now illustrate how service models can be specified regarding service decomposition and KPI expressions using SALmon. Listing 1 shows fragments of the `IPConnectivity`, and `IPServiceLevel` classes. The `IPConnectivity` class represents a transport layer link, which is fairly straightforward from a model point of view, it has a set of inputs representing lower level KPIs from probes.

We then define an abstract `ServiceLevel` class which enforces subclasses to implement `jeopardized` and `violated` statuses. The `Service Level` class also defines an expression that refers to the current measurement interval:

```
I = (NOW, NOW-time_interval)
```

This is a first illustration of SALmon's time-awareness and its usefulness in the context of service monitoring. `NOW` is a keyword referring to the current time. The `time_interval` property is defined when the `Service Level` is instantiated and should correspond to the SLA measurement period, for example be hourly or daily. In Listing 1 we show the `Service Level` for an IP Link, `IPServiceLevel`. The anchor `ip_connection` refers to all IP Links that is "monitored" by this `Service Level`. The properties `max_jitter` etc, represents "promises" that shall be monitored according to the SLA. The `violated` and `jeopardized` statuses verifies if jitter, loss-rate or delay has exceeded the thresholds within the time interval.

Finally we show the `IPTVUserService` in Listing 2. The user `IPTV` class calculates various KPIs based on attributes from the anchored video clients and servers. We show the expressions for channel change time and startup-time in line 9-10. The service level can then express thresholds on the KPIs for the control, dependability, and data planes as shown in Listing 2. Note that the service level defines an overall `violated` attribute to check the overall status for the end user service.

With this general definition of a `IPTV Service Level` we can create templates for different service levels where we assign values to the properties (thresholds). Listing 3 shows a service level for `IPTV` based on the requirements defined by DSL Forum [7]. The service level is measured daily by setting `time_interval` to 24h and the other properties defines the thresholds, for example it shall take maximum 2 seconds to change channel.

## VI. RELATED WORK

Kerpez et. al. [13] gives an excellent overview of the `IPTV Assurance` modeling problem. They emphasize the purpose of service models – "... use of service models is the combination of multilayer metrics to evaluate key quality indicators and key performance indicators (KQIs and KPIs)". The model and

Listing 3. Service Level Template

```
def IPTVServiceLevel = IPTVUserServiceLevel(
    max_channel_change_time => 2s,
    max_startup_time => 10s,
    min_availability => 0.9994,
    max_video_quality_impairment => 5,
    time_interval => 24h)
```

KPI aggregation we have shown in this paper is an attempt to realize this vision.

We shall now look at two streams of service modeling approaches: first, industry and standard initiatives and then research initiatives.

**Industry and Standard Initiatives.** TM Forum has defined a rich information model, `SID`, "System Information Model" [14]. `SID` is comparatively high level and models entities in telecom operators' processes. However, `SID` is being refined and moving closer to the resources by incorporating `CIM` [15]. We use these models as a library in our work. As opposed to these approaches, we have tied a run-time engine that is able to execute the models directly whereas `CIM` and `SID` models need to be re-casted into an execution environment. "Service Modeling Language"(SML) [16] adds important modeling semantics such as constraints and policies and support for model instantiation. However the actual KPI expressions is not part of SML.

**Research Initiatives.** A simple and pragmatic model for a general service model is given by Garschhammer et al. [5] and this work identifies several important research areas. Basic requirements for a service modeling language such as aggregation of components and calculations, flexible identification of instances as well as list-valued attributes are identified by Gopal [17]. `SLang` [18] is a language focused on defining formal SLAs in the context of server applications such as web services. `SLang` differs from our current effort in that it "focuses primarily on SLAs, not service models in general".

## VII. CONCLUSION AND FUTURE WORK

We have discussed the importance of service modeling to overcome difficulties in managing large scale, complex set of telecom/Internet services. We also presented how to create an abstract service model based on the `OSI` reference architecture. Creating the abstract service model is one of the first steps in service modeling. Once the abstract model is available, that model can be implemented using a domain specific language such as `SALmon`. We have shown that this domain specific language with built-in time-awareness, functional expressions, list apprehension, and object-oriented structuring is a novel approach to formalize service models. Using this language key requirements as service decomposition and KPI expressions can easily be expressed. The `IPTV` service model will be prototyped and evaluated as part of the ongoing `Magneto` work.

**Acknowledgements.** This paper describes work undertaken in the Context of the `Magneto-Celtic` Telecommunication So-

Listing 1. IP Connectivity

```

class IPConnectivity
    input jitter, packet_losses, packets, delay, datarate
class ServiceLevel
    violated = false, jeopardized = false
    property jeopardized_th = 0.8, time_interval
    I = (NOW, NOW-time_interval)
class IPServiceLevel inherits ServiceLevel
    anchor ip_connection
    property max_jitter, max_loss_rate, max_delay, min_datarate
    losses = ip_connection.losses
    packets = ip_connection.packets
    // Lambda expression for calculating loss rate in the s,e interval
    loss_rate = (s, e) -> (losses@e - losses@s) / (packets@e - packets@s)
    violated = (max jitter@I) > max_jitter or (loss_rate I) > max_loss_rate or
               (max delay@I) > max_delay
    jeopardized = (max jitter@I) > max_jitter*jeopardized_th or ...

```

Listing 2. User IPTV Service

```

class IPTVUserService
    anchor server, client
    channel_change_time = server.response_time + client.jitter_buffer_delay
    + client.rtp_session.latency + client.decoder_delay
    startup_time = client.startup_time + channel_change_time
class IPTVUserServiceLevel inherits ServiceLevel
    anchor iptv_service
    property max_channel_change_time, max_startup_time
    channel_change_time = max iptv_service.channel_change_time@I
    startup_time = max iptv_service.startup_time@I
    control_violated = channel_change_time > max_channel_change_time
    or startup_time > max_startup_time
    violated = control_violated or availability_violated
    jeopardized = control_jeopardized or availability_jeopardized

```

lutions Project, which is partially funded by the Commission of the European Union.

## REFERENCES

- [1] G. Hogan, "Magnetio: Management of the outer edge," 2008. [Online]. Available: <http://www.celtic-initiative.org/Projects/MAGNETIO>
- [2] S. Wallin and V. Leijon, "Multi-Purpose Models for QoS Monitoring," in *21st International Conference on Advanced Information Networking and Applications Workshops (AINAW'07)*. IEEE Computer Society, 2007, pp. 900–905.
- [3] Ecole Centrale Paris, "VideoLAN," 1996, <http://www.videolan.org>.
- [4] V. Raisanen, *Service Modelling*, isbn 9780470018071 ed. Wiley, 2006.
- [5] M. Garschhammer, R. Hauck, H. Hegering, B. Kempter, I. Radisic, H. Rolle, H. Schmidt, M. Langer, and M. Nerb, "Towards generic service management concepts a service model based approach," *Integrated Network Management Proceedings, 2001 IEEE/IFIP International Symposium on*, pp. 719–732, 2001.
- [6] G. Holm-Oste and M. Norling, "Ericsson's user service performance framework," 2008.
- [7] Tim Rahrer and Riccardo Fiandra and Steven Wright, "Technical Report TR-126 Triple-play Services Quality of Experience (QoE) Requirements," DSL Forum, Tech. Rep., 2006.
- [8] H. Schulzrinne, A. Rao, and R. Lanphier, "Real Time Streaming Protocol (RTSP)," RFC 2326 (Proposed Standard), Internet Engineering Task Force, Apr. 1998. [Online]. Available: <http://www.ietf.org/rfc/rfc2326.txt>
- [9] M. Leitner, P. Leitner, M. Zach, S. Collins, and C. Fahy, "Fault Management based on peer-to-peer paradigms; A case study report from the CELTIC project Madeira," in *Integrated Network Management, 2007. IM'07. 10th IFIP/IEEE International Symposium on*, 2007, pp. 697–700.
- [10] Javier Garcia-Algarra, Pablo Arozarena-Liapis, Sergio Garcia-Gomez and Alvaro Carrera-Barroso, "A Lightweight Approach to Distributed Network Diagnosis under Uncertainty," in *International Conference on Intelligent Networking and Collaborative Systems (INCoS 2009)*, 2009.
- [11] J. Bernstein, T. Spets, G. Bathrick, and G. Pitsoulakis, "DSL Forum TR-069, CPE WAN Management Protocol," DSL Forum, Tech. Rep. TR-069, 2007. [Online]. Available: [http://www.broadband-forum.org/technical/download/TR-069\\_Amendment-2.pdf](http://www.broadband-forum.org/technical/download/TR-069_Amendment-2.pdf)
- [12] S. Wallin, V. Leijon, and J. Ehnmark, "SALmon - A Service Modeling Language and Monitoring Engine," in *Proceedings of the Fourth IEEE International Symposium on Service-Oriented System Engineering (SOSE)*. IEEE Computer Society, December 2008, pp. 202 – 207.
- [13] K. Kerpez, D. Waring, G. Lapiotis, J. Lyles, and R. Vaidyanathan, "IPTV service assurance," *IEEE Communications Magazine*, vol. 44, no. 9, p. 166, 2006.
- [14] TM Forum, "Shared information data model," TM Forum, Tech. Rep., 2005.
- [15] DMTF, *Common Information Model Specification v2.23.0*, Distributed Management Task Force Std., Rev. 2.23.0, 2009, <http://www.dmtf.org/standards/cim/>.
- [16] W3C, "Service modeling language," <http://www.w3.org/TR/sml/>, May 2008.
- [17] R. Gopal, "Unifying network configuration and service assurance with a service modeling language," *Network Operations and Management Symposium, 2002. NOMS 2002. 2002 IEEE/IFIP*, pp. 711–725, 2002.
- [18] D. Lamanna, J. Skene, and W. Emmerich, "SLang: A Language for Defining Service Level Agreements," *Proc. of the 9th IEEE Workshop on Future Trends in Distributed Computing Systems-FTDCS*, pp. 100–106, 2003.