

Modeling and Management of Service Level Agreements for Digital Video Broadcasting(DVB) Services

Thapelo Tlhong and Jeff S. Reeve

Electronics and Computer Science
University of Southampton
SO17 1BJ United Kingdom

Abstract. This paper describes a metamodeling strategy of Service Level Agreements for Digital Video Broadcasting services based on Service Level Agreement Language(SLAng). The purpose of the paper is to provide a detailed analysis of SLAs in this domain and provide a motivation for modeling and automating their management. We also discuss why precise and machine readable SLAs can improve the levels of automation in SLA Management thereby reducing potential violations. The meta-modeling approach based on the Model Driven Architecture(MDA) described in this paper also simplifies the integration of a SLA Management systems with other infrastructure that delivers the service to the client.

1 Introduction

Service Level Agreements(SLAs) have traditionally been considered as a legal binding between a service provider and a customer. However, the advent of Service Oriented Architectures(SOA) and service based business models has seen the IT industry move away from considering SLAs only as a legal document but instead as means of enforcing and managing user requirements and expectations. As proposed in [1] a SLA can be used as a basis for the specification and development of a contracted service. Hence, it is necessary to integrate the specification and management of SLAs with other systems that are involved in the development, provision, maintenance and management of the service. A potential benefit of this integration is the ability to monitor the real time conformance of the service performance and related metrics to the SLA requirements. This can enable service reconfiguration and adjustment to minimise SLA violations. Traditionally SLAs are defined in legalese whilst the actual service being constrained by the SLA is specified, designed and implemented in a technical context [2]. This paper presents an analysis of requirements for modeling SLAs in a Digital Video Broadcasting domain and a modeling strategy based on SLAng [3].

2 Related Work

2.1 SLang

Service Level Agreement Language(SLang) [4] forms the basis of the our work on SLA. However, the original SLang was developed for Web (application) Services. The language is based on OMG's Meta Object Facility(MOF)¹ and Object Constraint Language(OCL)². SLang's dependence on MDA standards(XMI, UML, MOF and OCL) means that the extensibility mechanisms offered by such standards can also be utilized in extending SLang. Moreover, the separation of concepts between the domain of the language and the language itself used in specifying the language enables SLang to be extended to different domains for which it was not originally defined. This makes it an appropriate candidate for adapting existing specifications for DVB services.

2.2 X-Contract

In [5] a contract representation scheme is proposed based on the Finite State Machine(FSM) semantics. In this strategy, a contract is interpreted as a set of rights and obligations which are divided into sets R and O of Rights and Obligations respectively for each signatory party in the contract. According to [6] a right is defined as *an action that a signing entity can do if it wishes to* and an obligation is *a duty that an entity is expected to perform*. Consequently failure to perform an obligation may result in a breach of the contract or violation. The designers of the language propose a state machine-like approach implementation of the contracts(SLAs) written in the language, by having one state machine for the R and another for O per party. The violations of or conformance to the SLA can then be inferred from any given state in the FSM. This approach might be sufficient for small scale SLAs but for large systems of SLAs it will lead to complicated state machines that are not maintainable.

2.3 RBSLA

The Rule Based Service Level Agreement Language is a rule based modeling and implementation of SLAs [2] based on Predicate Logic and Event calculus. The language extends RuleML with concepts that enable SLA specification to be written as accurately as possible. These concepts include Event, Condition and Action model, Procedural Attachments, External Data Integration and Typed Logic, which enable assigning logical terms a type. The formal approach taken in defining RBSLA provides accurate type and constraints checking as well as allowing formal analysis of SLA written in the language. The dependence of the language on the use of rules for the specification of SLAs restricts the deployment of such SLAs to rule based systems which is in contradiction implementation independence we propose in Section 4.1.

¹ <http://www.omg.org/cgi-bin/doc?formal/2006-01-01>

² <http://www.omg.org/docs/ptc/03-10-14.pdf>

3 DVB over Terrestrial Services

DVB³ is a standard based on the MPEG-2 video compression scheme for the provision of digital video services [7]. Since DVB has several flavours and their associated standards - Satellite(DVB-S), Terrestrial(DVB-T), Cable(DVB-C) and Handheld(DVB-H), the scope of this paper is on Digital Terrestrial Television (DVB-T) services. Traditionally the broadcaster or content producer was also the transmission provider. However, this operational model has changed over recent years. This has led to a fragmentation of the broadcast transmission chain as shown in Figure 1. Compared to traditional analogue video broadcasting,

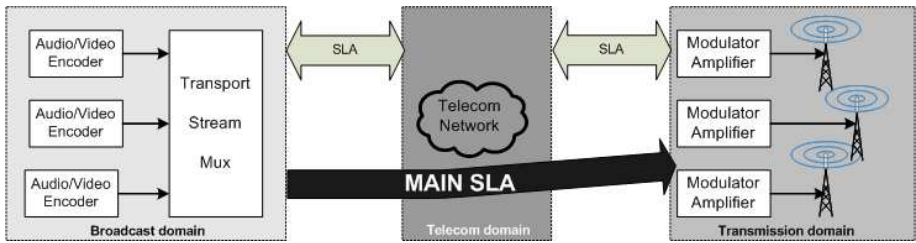


Fig. 1. DTT transmission chain showing inter-domain and intra-domain SLA interfaces

DVB offers the ability to deliver high quality video and audio, subtitles, audio descriptions, wide screen signalling, MHEG data services, Conditional Access Control and Schedule Management in the form of Electronic Programme Guide(EPG). The multipartite service composition means that the potential for errors occurring in the service increases since error probability varies linearly with the number of service components. Moreover, some broadcast services are time-shared or time-exclusive, which means that multiple services can share a single service slot within the service multiplexer and the switching between the services is performed on time based schedules. As the signal traverses the chain from one domain to the other it is necessary to have well defined metrics and parameters that describe the service at each domain interface. This requires well specified and precise SLAs that explicitly state the roles and responsibilities of each party at each service interface point.

4 DVB SLAs

A DVB Service traverses at least three domains - Broadcast, Telecom and Transmission, before reception by the viewer(Figure 1). The interfaces between these domains represent inter-domain SLAs and vertically within each domain there is also intra-domain SLAs. In the context of this paper the Broadcast(DVB) SLA refers to a bilateral agreement between a transmission service provider and

³ <http://www.dvb.org>

a broadcaster that controls the management, performance, quality and monitoring of the service (denoted MAIN-SLA in Figure 1). We informally define such SLA as a set of contract clauses that specify and constrain the behaviour of entities along the transmission chain to ensure reception of a decodable and viewable service.

Several languages and Information Models for SLAs have been proposed. Instead of proposing a new model, we chose to build on an existing model (SLAng) and use such to evaluate the specification and management of DVB SLAs. The choice of the relevant strategy is based on the following requirements that we impose on the existing strategies.

4.1 Requirements for SLA Modeling and Specification

Integration - SLA specification language should ease the integration of the SLA management framework with other subsystems that cooperate to provide the service to the customer.

Machine Readability- is the basis for automating the management of service level agreements and enabling exchange of SLA specific information with other systems involved in the provision of the service e.g. Fault Management and Correlation systems.

Automatability - is meant to ensure a more proactive monitoring of the runtime system that provides the service to the SLA requirements. Automatability is necessary for a system of SLAs to self manage and adapt itself to minimise violations.

Implementation Independence - It is necessary to describe SLAs in a language that is technology neutral and platform independent to cohere with the Integration requirement described above.

Reusability - is necessary to allow SLAs for different domains(e.g. Web Services, Broadcast, Telecoms) to be expressed in the specified language without requiring extensive modification to the language.

Extensibility - an SLA language/specification should provide extension mechanisms that allows language users to extend it for other peculiar service domains and service provision scenarios for which it was not originally defined. This requirement is an extension to the Reusability requirement discussed above.

Other requirements for SLA specifications are described in detail in [3] and these include Precision, Expressiveness, Understandability and Analysability.

5 DVB SLA Model

This section describes the adaptations and extensions we made to SLaNg to make it appropriate for modeling SLAs for DVB services. As per the SLaNg paradigm, we have two sets of models, the Domain and the Service models as described in the next section.

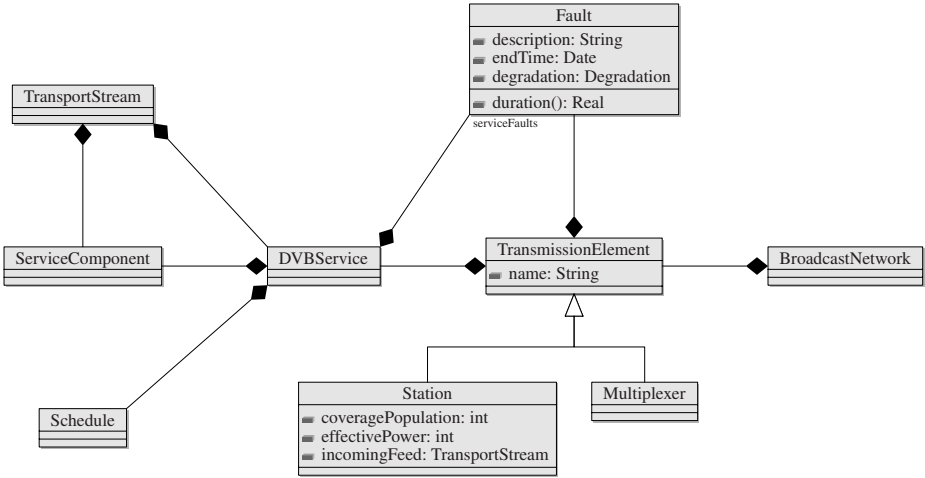


Fig. 2. Domain model for which Transmission service SLAs apply

5.1 Broadcast Domain Model

A broadcast SLA imposes constraints on the behaviour of entities and services within the broadcast domain, as such it is imperative to first model the domain on which the SLA constraints apply. The structure and relationship between the entities within the domain are shown in Figure 2. A DVB service is contained within an MPEG2 Transport Stream which also contains service components and transport level tables that enable the services in the stream to be decoded. The service is carried over a transmission network by different types of transmission elements. The domain model provides the context for the constraints and enables us to attach semantic definitions to the SLA elements in Figure 3. For instance, a *StationClause* and *StationDefinition* are used to express constraints in OCL on the behaviour of Station objects.

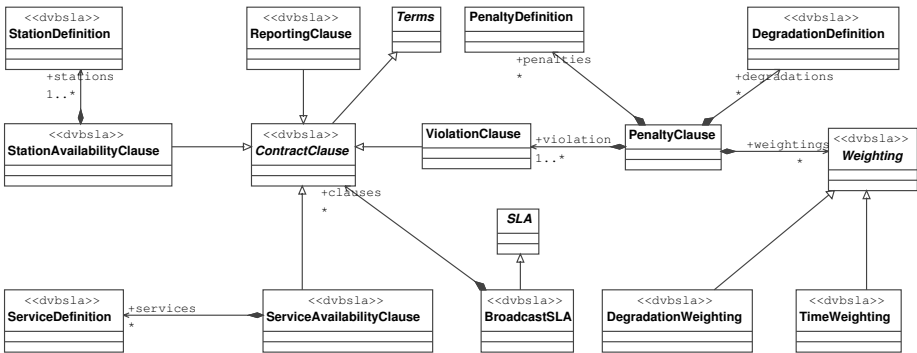


Fig. 3. Service(SLA) model for a broadcast SLA

In the original SLA model, parties to the SLA are defined as entities that are signatory to the contract. However, we found this approach to be limiting in terms of modeling other relationships that exist in the SLA. Instead of this approach, we model a SLA Party as an individual or enterprise whose “behaviour” can affect the conformance of the service to the SLA. Roles are then attached e.g. ProviderRole, MonitoringRole, ClientRole to these parties as required. This approach enables us to model complex interactions of parties involved in the SLA and also enables Accountability for observed service behaviour to be assigned to the appropriate party based on their role.

```

abstract class services::BroadcastService extends
    ::slang::ServiceDefinition
{
    serviceFaults : Fault[0,*] opposite sLA unique
    serviceFaults : Fault[0,*] unique
    components : ServiceComponent[1,*]
    status : ::types::ServiceStatus

    invariant {
        serviceFaults->forAll(f1 : Fault, f2 : Fault |
            f1.degradation = f2.degradation
            and f1.startTime = f2.startTime
            and f1.endTime = f2.endTime
            implies f1 = f2
        )
    }
}

```

Listing 1.1. Sample EMOF/OCL Specification for a BroadcastService

5.2 Constraints Checking

The textual syntax of the SLA is in EMOF and OCL. Listing 1.1 shows a sample specification for *DVBService* and an invariant that specifies that any accountable fault affecting the service must not be duplicated. This is to restrain from applying penalties for a single fault occurrence multiple times. The specification is converted into a JMI⁴ repository using the UCLMDA Tools [8]. Constraints checking is performed by creating instances of the SLA from the repository and populating the instance fields with SLA requirements e.g. availability values for a given time period. Each entity in the specification has an embedded constraint checking. For example, an instance of the *BroadcastService* shown in Listing 1.1 will have an methods, *verifyConstraints()* and *verifyConstraintsDeep()* that when called will verify that there are no duplicates for all faults affecting this service.

⁴ <http://java.sun.com/products/jmi/index.jsp>

6 Conclusion and Further Work

In this paper we have presented the an detailed analysis of service level agreements in the broadcast domain and also the motivation for focusing on broadcast services, one of which is the lack of research or publications concerning management of SLAs in this domain. We also discussed the challenges posed by the transition from analogue to digital video broadcasting. The metamodeling approach for specifying broadcast SLAs adopted in this paper based on SLANG language provides a basis for developing an SLA Management framework that we can use to study the effects and level of automation SLA management that can be achieved based on this approach. Further work will focus on the integration of a machine readable broadcast SLA and predictive modeling techniques to determine how previous SLA impact information can be used to predict the potential of SLA violation based on classification of service impact. This can enable the system to reconfigured adaptively against potential violations instead of reconfiguring the system only based on the current status of different SLAs.

References

- [1] Bournan, J.: Specification of service level agreements clarifying concepts on the basis of practical research. *Software Technology and Engineering Practice*, 1999, pp. 169–178 (1999)
- [2] Paschke, A.: A declarative rule-based service level agreement language based on ruleml. In: *International Conference on Intelligent Agents, Web Technology and Internet Commerce*, University of Munich, November 2005, vol. 2, pp. 308–314 (2005)
- [3] James, S.: *Slang - language for service level agreements*. Ph.D. dissertation, Department of Computer Science, Univerity College London, Malet Place, London WC1E 6BT, UK (2006)
- [4] Lamanna, D.D., Skene, J., Emmerich, W.: *Slang: a language for defining service level agreements*. In: *FTDCS 2003. Proceedings the Ninth IEEE Workshop on Future Trends of Distributed Computing Systems*, pp. 100–106. IEEE Computer Society Press, Los Alamitos (2003)
- [5] Molina-Jimenez, C., Shrivastava, S., Solaiman, Warne, J.: Contract representation for run-time monitoring and enforcement. In: *Proceedings IEEE International Conference on E-Commerce. CEC 2003*, pp. 103 – 110 (2003), <http://dx.doi.org/10.1109/COEC.2003.1210239>
- [6] Molina-Jimenez, C., Shrivastava, S., Solaiman, E., Warne, J.: Run-time monitoring and enforcement of electronic contracts. *Electronic Commerce Research and Applications* 3(2), 108–125 (2004), <http://dx.doi.org/10.1016/j.elerap.2004.02.003>
- [7] Dubery, P., Wilson, D.: Policing slas for digital video. *Cable and Satellite International*, Tech. Rep. (May 2003) [Online]. Available: http://www.cable-satellite.com/features/may-jun%2003/csi%2029_30.pdf
- [8] Skene, J.: The uclmda tools, robust implementation of omg standards for research and development. Open Source Project