

# Aesop: A Semantic System for Autonomic Management of End-User Service Quality

Liam Fallon

Network Management Lab, LM Ericsson,  
Athlone, Co. Westmeath, Ireland  
Email: liam.fallon(at)ericsson.com

Declan O'Sullivan

Knowledge & Data Engineering Group (KDEG)  
Trinity College Dublin, Ireland  
Email: declan.osullivan(at)cs.tcd.ie

**Abstract**—In order to manage the quality of end-user service sessions, the user experience of those sessions and the context in which they are delivered must be captured and understood, and network resource usage must be tuned to ensure user experience matches expectations as closely as possible. Aesop is an autonomic system that uses semantic techniques to assure compliance of service sessions with user expectations. It monitors terminal reports to capture service experience and context knowledge which, when analysed, allows adjustments to be planned and executed that improve service experience. Aesop is inherently model driven; all features are implemented using semantic reasoning, queries and rules, making it applicable to any networking domain. A prototype Aesop system has been applied to monitor and analyse end user services. The prototype has been evaluated on commodity hardware in a real Home Area Network (HAN).

## I. INTRODUCTION

The ability to automatically optimize the quality of the highest priority end user sessions has long been a goal of network management[1]. This goal is difficult to achieve because it is necessary to understand the expectation on each session, gather information on the experience of each session as it runs, and monitor the context in which each session is running [2]. The only definitive source of the service experience and context of user sessions is the client running those sessions [3].

Current centralized management systems do not manage end user sessions [4]. Approaches have been described for central management of user service experience when an authority such as the Next Generation Network (NGN) is controlling user sessions [4][5]. Conventional end-user Internet sessions have no central authority so such approaches cannot be applied [6].

In this paper we introduce Aesop, an autonomic manager for end user session quality. Aesop implements the classic autonomic MAPE loop [7] around a semantic knowledge base. End user sessions are monitored by lifting terminal reports from clients into a knowledge base (*Monitor*). Service session knowledge is analysed to ensure sessions comply with expectations (*Analyse*). A set of optimizations are suggested if the overall compliance of all sessions in a managed domain are not using network resources optimally (*Plan*). Those optimizations are then applied to the network (*Execute*).

Aesop uses a semantic approach because the relationship between network QoS and service experience perceived by customers is complex and dynamic [1][3] and difficult to capture using conventional databases. Aesop models knowledge

in its knowledge base with the End User Service Analysis and Optimization (EUSAO) ontology [8]. In Aesop, we use a novel architecture that partitions the knowledge base to overcome the well known performance constraints of semantic reasoning, querying, and rule execution operations [9][10].

A full prototype of the Aesop system deployed in a Home Area Network test bed, where its performance in monitoring and analyzing end user sessions was evaluated. The evaluation demonstrated that Aesop can efficiently monitor and analyse service sessions in a 20s time window. The performance of Aesop in optimization will be evaluated in future work.

This paper is organised as follows. §II gives a summary of related work. §III describes the Aesop architecture and implementation. §IV presents our evaluation of the Aesop monitoring and analysis tasks, and §V summarises the paper.

## II. BACKGROUND AND RELATED WORK

Management systems model network Quality of Service (QoS) metrics statically [11]. This makes service experience management difficult because the relationship between network QoS and service experience is complex and dynamic [1]. Batteram et al. [12] describe an approach that maps network quality indicators to service quality indicators but a mapping must be built and maintained for each end user service type. End-user service experience management has yet to see large scale deployment in such systems [4].

A central authority can implement experience monitoring because it is aware of ongoing sessions. Information from terminals [5] or service sessions [4] can be used to optimize service delivery. These approaches are difficult to apply to conventional end-user Internet sessions because no central authority controls such sessions [6].

Latré et al. [3] describe an autonomic architecture for experience optimization of video services where service experience was estimated from network metrics using mathematical equations or neural networks. Separate equations or neural networks must be developed for each service type. Aesop uses configurable semantic mappings and queries to handle end user services, allowing it to flexibly handle any given service type.

Vergara et al. [9] examined research that applied semantics to network management. They found that an ontology tailored to a management domain must be specified if semantic techniques are to be usefully applied. Keeney et al. [10] carried out experiments on semantic operations, reporting retrieval times

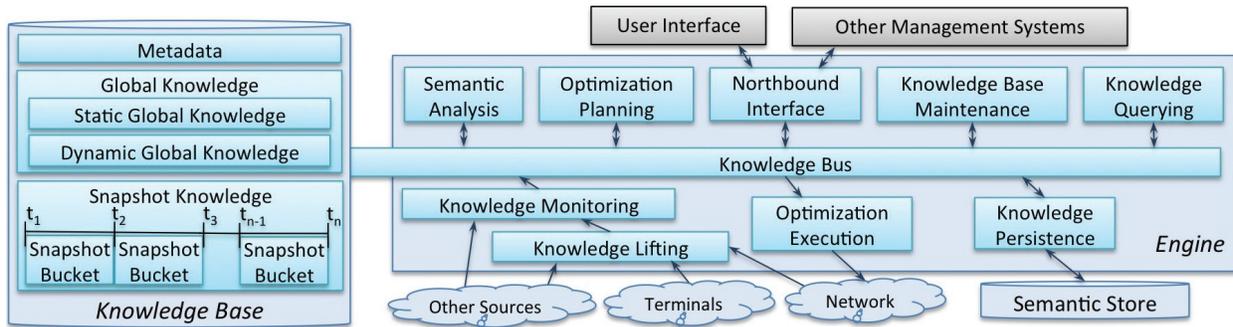


Fig. 1. The Aesop Semantic Service Analysis and Optimization System

of 1 to 40 seconds for reasoners and rules, and between 0.6 and 6 seconds for queries as knowledge base sizes increase.

### III. DESCRIPTION

Aesop, shown in Fig. 1, is an autonomic manager [7] composed of a knowledge base and an engine that executes an autonomic loop. Service expectation, experience, and context knowledge is held semantically using the EUSAO ontology [8]. The knowledge base is held in memory and is partitioned into a number of separate models to increase performance.

The Aesop engine loads *monitored* end user service knowledge into the knowledge base, *analyses* that knowledge to determine how well the set of running service sessions are complying with expectations, *plans* optimizations and *executes* those optimizations on the network.

#### A. The Knowledge Base

The knowledge base, shown in Fig. 1 models the concepts, relationships, and constraints of the end user service domain. It is partitioned into a number of models, thus keeping individual model sizes small. Semantic operations use as few models as possible, thus maximising their performance [10].

The *Metadata* model holds the structure or *Tbox* [13] of the knowledge in the knowledge base and is loaded at system start. A number of models are used to hold individuals or *Abox* [13], the knowledge in the knowledge base, which is partitioned as *Global Knowledge* and *Snapshot Knowledge*.

*Global Knowledge* such as User, Terminal, or Address does not change during execution of a session. Global Knowledge is partitioned as *Static* or *Dynamic*; with a separate model being used to hold each. Individuals of static concepts such as User or Service are explicitly configured and loaded at system start. Dynamic concepts such as Session or Terminal are concepts whose individuals are added and modified by knowledge read from the network or terminals.

*Snapshot Knowledge* such as session metrics or terminal locations is reported during sessions so the number of Snapshot Knowledge individuals is much greater than Global Knowledge individuals. The concept of a snapshot is fundamental to the performance of the Aesop knowledge base. Snapshots represent the state of a concept for a particular time period, all the snapshots for a time period are partitioned into a *Snapshot Bucket*, the knowledge base maintains a model for each snapshot bucket.

#### B. The Engine

The *Knowledge Bus* is a thread safe software interface used by all SSAO engine components to access the knowledge base.

Non-semantic incoming knowledge from terminals is processed into RDF (Resource Description Framework) format by *Knowledge Lifting* [8]. *Knowledge Monitoring* loads that knowledge into the knowledge base, updating the Dynamic Global Knowledge model and snapshot bucket models. Rules are fired to set the priority of each incoming service session and to determine the state of incoming sessions.

*Semantic Analysis* runs periodically, and analyses each snapshot bucket that has been updated. For each snapshot, it calculates the compliance of each service session to its expectations. It uses global knowledge and knowledge from a single snapshot bucket, thus minimising the volume of knowledge semantic analysis operations use.

*Optimization Planning* also runs periodically, checking the compliance of all end user sessions in recent snapshot buckets. If high priority sessions are not conforming to expectations, it uses semantic operations to determine if and what optimizations should be applied to the network. *Optimization Execution* applies those optimizations using a method such as running a script or firing a policy or firewall rule.

*Knowledge Querying* is used to run knowledge base queries for human users or other Aesop components. Queries run over the entire scope of the knowledge base; the Metadata model, the Global Knowledge models and all snapshot models.

The *Northbound Interface* supplies knowledge to external systems over interfaces. It may retrieve knowledge from the knowledge base directly or may use the Knowledge Querying component to query the knowledge base indirectly.

*Knowledge Persistence* is used to load static global knowledge from persistent storage at system start up and to store knowledge that has been identified as being useful for off-line reporting and analysis to persistent storage.

*Knowledge Base Maintenance* performs memory management of the knowledge base. It periodically removes expired snapshot buckets older than a configured retention time. It also removes Dynamic Global Knowledge individuals no longer referenced by knowledge in snapshot buckets.

```

PREFIX sa: <http://aesop/ServiceAnalysis.owl#>
PREFIX sid: <http://tmforum.org/SID_Phase_VII_Models#>

SELECT ?service ?expectation ?session ?snapshot WHERE {
?snapshot rdfs:type sa:ServiceExperienceSnapshot.
?snapshot sa:experienceSnapshotOfSession ?session .
?snapshot sa:experienceSnapshotOfService ?service .
?session sa:runsOnConnection ?connection .
?connection sa:usesSubscription ?subscriber .
?subscriber sid:PartyRole_party ?subscriber.
?expectation sa:expectationSetBy ?subscriber.
?expectation sa:expectationOnService ?service.
}

```

Listing 1. Query for Snapshot Expectation, Session, and Service

### C. Implementation

We have developed an Aesop prototype in Java. The knowledge base is held in Jena<sup>1</sup> OWL<sup>2</sup> models. Pellet<sup>3</sup> is used for reasoning and to execute rules. Aesop operations such as monitoring, analysis, and optimization run in parallel, using the thread safe knowledge bus to access the knowledge base.

The Metadata, which consists of the EUSAO ontology and the rules, was specified using Protégé<sup>4</sup> and loaded at startup from OWL files. In the prototype, Static Global Knowledge such as users and service expectations were also defined in Protégé and loaded at startup. In a deployed system, expectations should be set using a client interface such as in [6].

The prototype monitors knowledge reported by clients in XML terminal reports that contain metrics on connections and end user service sessions running on each client. Semantic rules were used to calculate the priority and status of sessions.

Analysis, Optimization, and Knowledge Base Maintenance operation run periodically and are implemented using SPARQL (SPARQL RDF Query Language) queries. The query shown in Listing 1 is used in analysis operations to find the service, session, and expectation individuals for a service experience snapshot individual.

Knowledge Base maintenance operations run every minute deleting snapshot knowledge older than 10 minutes and un-referenced Dynamic Global individuals. The Northbound Interface is implemented as a Java Remote Method Invocation (RMI) service that uses SPARQL queries to fetch the requested knowledge. A prototype UI was implemented in Java.

In the prototype an interval of 10s is used for Semantic Analysis, Optimization, and Knowledge Base Maintenance periodic operations. The snapshot bucket interval is also set at 10s. A 10s interval is used because it is sufficiently long to mitigate momentary network fluctuations, is frequent enough to be perceived as being near real time [14], and is a reasonable compromise between speed of operation and resource consumption by Aesop. The time from reception of a terminal report to the report being analysed is at most 20s and the time from reception to that knowledge being considered in optimization is at most 30s.

## IV. EVALUATION

We evaluated Aesop in a HAN test bed. The evaluation goal was to determine if Aesop can efficiently monitor and analyse

<sup>1</sup>Apache Jena, <http://jena.apache.org>

<sup>2</sup>Web Ontology Language (OWL), <http://www.w3.org/TR/owl2-overview>

<sup>3</sup>Pellet OWL Reasoner for Java, <http://clarkparsia.com/pellet>

<sup>4</sup>Protégé 4.1, <http://protege.stanford.edu>

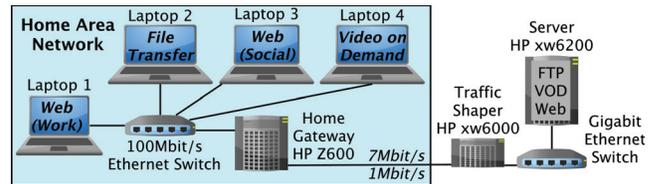


Fig. 2. The Aesop Test Bed

TABLE I. SESSION TIMINGS, PRIORITIES AND EXPECTATIONS

Laptop	Min	Session	Priority	Expectation
2	t	File Transfer	10	Speed $\geq 4.2\text{Mb/s}$
3	t+1m	Web (Social)	3	PageLoad $\leq 40\text{s}$
4	t+2m	VOD	2	MOS $\geq 3$
1	t+3m	Web (Work)	1	PageLoad $\leq 4\text{s}$
2	t+8m	File Transfer	10	Stop
3	t+10m	Web (Social)	3	Stop
4	t+12m	VOD	2	Stop
1	t+14m	Web (Work)	1	Stop

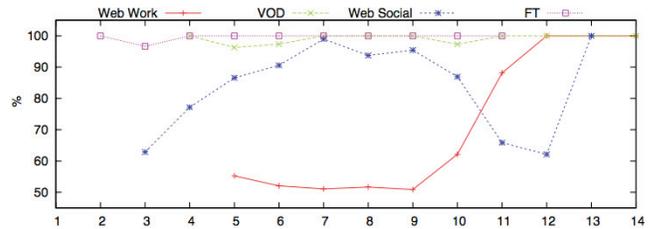


Fig. 3. Average Compliance of Sessions my Minute

the experience of a set of services periodically to deduce their compliance with previously set service expectations using a time window of 20s. A scenario where File Transfer, Video on Demand, Work Web Browsing, and Social Networking Web Browsing sessions were running was used. Service expectations and priorities were set in Protégé as shown in Table I.

The test bed shown in Fig. 2 was used. Aesop ran on the Home Gateway<sup>5</sup>; four laptops<sup>6</sup> ran end user sessions. The Home Gateway and Traffic Shaper hosts each had two bridged ethernet interfaces. Maximum download and upload speeds of 7Mbit/s and 1Mbit/s were set on the Traffic Shaper host bridged interface.

End user sessions were executed with the timing sequence shown in Table I in six experimental runs. Sessions were started at 1-minute intervals allow effects of a starting session on already running sessions to be observed. All sessions were run together for a period of 5 minutes to allow the steady state performance of Aesop to be monitored. Each session was then terminated at 2 minute intervals, allowing time for effects of a terminating session on running sessions to be observed. The total time was 14 minutes for each run.

### A. Results

The plots of average compliance in Fig. 3 show that Aesop did semantically monitor and analyse the experience of sessions against their expectations.

<sup>5</sup>HP Z600 workstation, 4-core Intel Xeon@2.4GHz, 24GB memory

<sup>6</sup>HP tc4200 laptops, Intel Pentium M@1.86GHz, 1.5GB memory

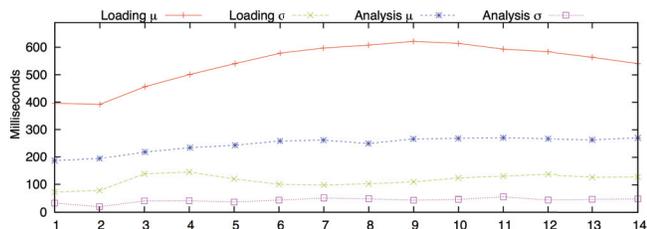


Fig. 4. Average and Standard Deviation of Loading and Analysis Times

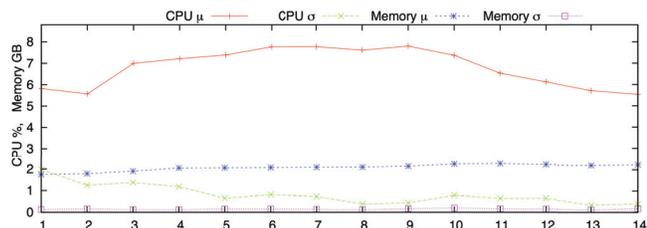


Fig. 5. Average and Standard Deviation of CPU and Memory Utilization

The goal for Aesop to deduce compliance was 20s; 10s for loading and 10s for analysis. Both loading and analysis perform well within that goal, as shown in Fig. 4. Average load times, the time to load knowledge from a single terminal report, peak below 700ms when all sessions are running, and drop off as sessions terminate. Average analysis times increase from 200ms to 300ms as runs progress.

Fig. 5 shows average CPU and Memory usage on the Home Gateway. Average CPU usage plateaus at 8% after 3 minutes when all sessions are running. After 9 minutes average CPU usage decreases in steps of around 1% as sessions terminate. Average Memory usage rises from 1.8GB to 2GB after 3 minutes when all sessions are running, and continues to increase gradually to 2.3GB at the end of runs.

### B. Experiences, Findings and Interpretation of Results

All knowledge handling in the system is configurable, either by changing relations or constraints in the EUSAO ontology, or by changing rules or SPARQL queries. The use of SPARQL queries is flexible; a short query can achieve complex determinations. Rules used were also straightforward to specify. The domain complexity is encapsulated in the EUSAO ontology.

The Work web sessions were starved of bandwidth until sessions start to terminate, after which compliance increases to 100%. This indicates that restricting lower priority sessions would increase the compliance of Work web sessions.

The monitoring and analysis times in Fig. 4 and resource utilization results in Fig. 5 illustrate the performance of the Aesop system. These results indicate that the Aesop architecture is an efficient way of applying semantic techniques to analyse end user service quality. The CPU and memory usage is in proportion to the load on the Aesop system.

## V. SUMMARY AND FUTURE WORK

The evaluation shows that Aesop can autonomically monitor, analyze, and present the experience and context of realistic

end user service sessions running in a real Home Area network, using a working semantic knowledge base of the domain that has been primed with expectations. The performance results show that the Aesop system architecture is effective in applying semantic techniques in a timely and resource efficient way when running on commodity hardware.

The ontology structure is key to the efficiency of the Aesop architecture. It captures the domain knowledge and is partitioned in a way that allows efficient access. Once the correct ontology structure was established, powerful queries and rules could be specified on the ontology relatively easily.

Evaluation of optimization operations is currently ongoing and will be the subject of future publication submissions.

### ACKNOWLEDGMENT

This work was partially supported by Science Foundation Ireland under the FAME SRC Grant No 08/SRC/I1403.

### REFERENCES

- [1] R. Stankiewicz, P. Cholda, and A. Jajszczyk, "QoX: What is it Really?" *IEEE Communications Magazine*, vol. 49, no. 4, pp. 148–158, Apr 2011.
- [2] F. Toutain, A. Bouabdallah, R. Zemek, and C. Daloz, "Interpersonal Context-Aware Communication Services," *IEEE Communications Magazine*, vol. 49, no. 1, pp. 68–74, Jan 2011.
- [3] S. Latré *et al.*, "An Autonomic Architecture for Optimizing QoE in Multimedia Access Networks," *Computer Networks*, vol. 53, no. 10, pp. 1587–1602, 2009, autonomic and Self-Organising Systems.
- [4] J. Sterle, M. Volk, U. Sedlar, J. Bester, and A. Kos, "Application-Based NGN QoE Controller," *IEEE Communications Magazine*, vol. 49, no. 1, pp. 92–101, Jan. 2011.
- [5] Oyman, O. and Singh, S., "Quality of Experience for HTTP Adaptive Streaming Services," *IEEE Communications Magazine*, vol. 50, no. 4, pp. 20–27, Apr 2012.
- [6] J. Sventek *et al.*, "An Information Plane Architecture Supporting Home Network Management (IM)," in *Integrated Network Management Symposium (IM)*. IEEE, May 2011, pp. 1–8.
- [7] J. O. Kephart and D. M. Chess, "The Vision of Autonomic Computing," *IEEE Computer Magazine*, Jan 2003.
- [8] L. Fallon and D. O'Sullivan, "Using a Semantic Knowledge Base for Communication Service Quality Management in Home Area Networks," in *Network Operations and Management Symposium (NOMS)*, Apr 2012.
- [9] J. López de Vergara, A. Guerrero, V. Villagrà, and J. Berrocal, "Ontology-Based Network Management: Study Cases and Lessons Learned," *Journal of Network and Systems Management*, vol. 17, pp. 234–254, 2009.
- [10] J. Keeney, A. Boran, I. Bedini, C. Matheus, and P. Patel-Schneider, "Approaches to Relating and Integrating Semantic Data from Heterogeneous Sources," in *Web Intelligence and Intelligent Agent Technology Conference (WI-IAT)*, vol. 1, Aug. 2011, pp. 170–177.
- [11] TM-Forum, "Information Framework (SID)," *GB 922*, Sep 2011.
- [12] H. Batteram *et al.*, "Delivering Quality of Experience in Multimedia Networks," *Bell Labs Tech. Journal*, vol. 15, no. 1, pp. 175–193, 2010.
- [13] Nardi, D. and Brachman, R.J., *The Description Logic Handbook*, F. Baader *et al.*, Eds. Cambridge University Press, 2007, no. ISBN 978-0-521-15011-8.
- [14] NGMN, "NGMN Informative List of SON Use Cases," Deliverable, NGMN, Tech. Rep., Apr 2007.