Towards an Autonomic Architecture for Optimising the QoE in Access Networks

Steven Latré

Supervisor(s): Filip De Turck

I. Introduction

Multimedia services over broadband DSL access and aggregation networks such as Broadcast TV and Video on Demand have gained a lot of popularity in the last few years. Operators who want to maximise their revenue try to manage the service quality as perceived by the end user, commonly described as the Quality of Experience (QoE). This QoE management is further complicated by the heterogeneity of today's access and aggregation networks, triggering a QoE management on a per service or per subscriber basis. This introduces the need for detailed knowledge about users and services. In our research, we are focusing on an architecture for effectively partitioning this knowledge in an autonomic management environment.

The problem of organising knowledge in an autonomic network has been addressed through the Knowledge Based Network paradigm [1]. In a Knowledge Based Network, producers of information describe the available information through ontologies. Consumers subscribe to this information through semantic queries. The work presented in this paper complements this approach: while the KBN work focuses on semantic clustering of information [2] and augmenting the semantic capabilities of existing solutions [3], we focus on the automatic generation of the semantic queries, which we call filter queries, through a cognitive model.

S. Latré is with the Department of Information Technology, Ghent University (UGent), Gent, Belgium. He is funded by Ph.D grant of the Fund for Scientific Research, Flanders (FWO-V). E-mail: Steven.Latre@UGent.be .

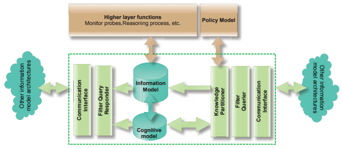

Figure 1. Architecture for effective knowledge communication.

The remainder of this paper is structured as follows. Section II discusses the integration of the cognitive model within the KBN architecture. In Section III, the automatic generation of filter queries based on information in the cognitive model is discussed for the employed implementation. Section IV concludes this paper.

II. Architecture Description

An overview of the employed architecture, which is responsible for collecting the necessary knowledge for each node by querying information present in other remote nodes, is given in Figure 1. At the heart of the architecture lies the information model which represents all knowledge needed for the higher layer functions. Conceptually, the architecture splits the information model of every node into different sub models. Every node X has a dedicated information model and several derived information models, containing parts of the information model of other nodes. The dedicated information model consists of knowledge which is local to node X. The derived information models contain parts of the information models of other nodes and typically contain knowledge

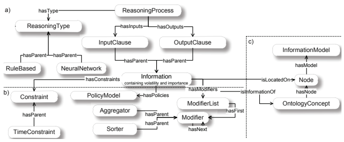

Figure 2. Schematic overview of the concepts and relationships introduced in the cognitive model.

which can aid in the reasoning process of node X. The derived models are constructed through filter queries, defining which part of the knowledge is requested from other nodes.

The cognitive model comprises information about the capabilities of each node and describes which kind of knowledge is generated by each node. Additionally, the cognitive model provides information about the requirements of the local reasoning process by stating what kind of information is needed to reason upon.

The information available in the cognitive model is used to select the data that needs to be extracted from other information models. This is done in the Knowledge Partitioner, a third component in the architecture, which automatically generates the filter queries.

III. Automatic generation of queries

To automatically generate the necessary queries, the Knowledge Partitioner relies on knowledge about the reasoning process defined through the cognitive model. The cognitive model used in our implementation is illustrated in Figure 2. The illustrated concepts can be divided into three groups. A first group (a), provides information about the reasoning process itself. The cognitive model describes how reasoning is performed on this node and what knowledge is needed. In the second group (b), the cognitive model describes how this information is treated by the reasoning process. The cognitive model allows for applying policies on the generation process and supports that information can be modified or constrained. The third part of the cognitive model (c) describes which concepts are modelled on remote information models. This information is needed to know where the filter queries need to be redirected to and if the requested knowledge is available or not.

The knowledge available in the cognitive model can be easily added by investigating the reasoning process of each entity. For example, for a rule based engine, the facts needed to trigger rules can be seen as the *InputClauses* of the cognitive model. Depending on how this information occurs in the rules, they may or may not be aggregated. Based on this cognitive model, constructing the filter queries is straightforward. Each individual of the *InputClause* concept is mapped to a filter query that and through the is *isLocatedOn* and *hasModel* relationships the Knowledge Partitioner knows which information model to contact.

IV. Conclusions

We discussed how filter queries in Knowledge Based Networks can be automatically generated through the use of a cognitive model. The cognitive model defines the knowledge requirements of each node by identifying the reasoning processes running on each node and describing the information they need to perform their task. The structure of the cognitive model allows to easily map the information onto filter queries.

References

[1] D. Lewis, D. O' Sullivan, and J. Keeney, "Towards the knowledge-driven benchmarking of autonomic communications," in *World of Wireless, Mobile and Multimedia Networks, 2006. WoWMoM 2006. International Symposium on a*, 2006, pp. 6 pp.–505.

[2] John Keeney, Dominic Jones, Dominik Roblek, David Lewis, and Declan O'Sullivan, "Knowledge-based semantic clustering," in *SAC '08: Proceedings of the 2008 ACM symposium on Applied computing*, New York, NY, USA, 2008, pp. 460–467, ACM.

[3] John Keeney, Dominik Roblek, Dominic Jones, David Lewis, and Declan O'Sullivan, "Extending siena to support more expressive and flexible subscriptions," in *DEBS '08: Proceedings of the second international conference on Distributed event-based systems*, New York, NY, USA, 2008, pp. 35–46, ACM.