
NGSON Service Composition Ontology

Reinhard Schrage

SchrageConsult Seelze, Germany reinhard.schrage.de@ieee.org

Received 5 February 2014; Accepted 28 February 2014;
Publication 8 April 2014

Abstract

The paradigm of Next Generation Service Overlay Networks (NGSON) strives to provide a unified and standardized framework of IP-based service overlay networks, creating an ecosystem of context-aware, dynamically adaptive, and self-organizing networking capabilities, including advanced routing and forwarding schemes. Service composition, i.e. the facility to combine certain atomic services into an aggregated service is considered a vital part of NGSON. As a future directed, next generation oriented paradigm NGSON must enable an intelligent, automated service composition platform. In order to also meet the objectives of being context-aware, dynamically adaptive, and self-organizing this platform needs to know and understand the semantics of its underlying functional entities. Furthermore, to be accepted by users, enterprises and service developers existing, proven, but likewise extendable standards need to be utilized as much as possible.

The W3C consortium has released OWL2 for building ontologies that serve to provide machine-understandable semantics. In order to be feasible for NGSON a service composition ontology also needs to include concepts from deontic logic, i.e. needs to be able to differentiate between omissible and permissible classes, or -in finer granularity- prohibited, obligatory and optional components when composing a service from atoms.

This paper aims to underline the need for an OWL2 ontology, make suggestions on its structure and required interfaces to other network entities as e.g. Software Defined Networks and Network Virtualization Functions.

Keywords: NGSON; Service Composition; OWL2 Ontology.

Journal of Cyber Security, Vol. 2 No. 3 & 4 , 351–358.

doi:10.13052/jcsm2245-1439.238

© 2014 River Publishers. All rights reserved.

1 Introduction

Already, there is a plethora of networking applications available. Yet, very often, these applications form independent silos unaware of each other's functionalities and unable to communicate and benefit from already existing and implemented algorithms. Users, on the other hand, find that they need to invoke several applications in order to retrieve all relevant information required for a particular query. Output from one application may have to be manually re-entered into another application as input, probably also in a different format. Software developers are faced with the necessity of adapting their applications due to minor changes in the rendering of a webpage used to collect input data.

It is evident that users and suppliers alike will benefit from a capability of data integration across individual applications. Semantic Web Services are addressing this horizontal data integration.

At the same time there are new developments in the design and implementation of transport networks that carry user applications: Network Functions Virtualization (NFV) aims at reducing CAPEX, OPEX, space and power consumption by hosting, i.e. virtualizing, dedicated single purpose components into general purpose servers, while Software Defined Networking decouples the control and data planes, centralizing network intelligence and state, and abstracting the underlying network infrastructure from the applications to facilitate faster innovation.

Amongst other issues a platform that is to facilitate machine-automated, context-aware and dynamically adaptive composition of service applications and provide rule-based optimization of transport streams needs to:

1. Know what services/applications are available at request time
2. Know how these services, if at all, can be composed

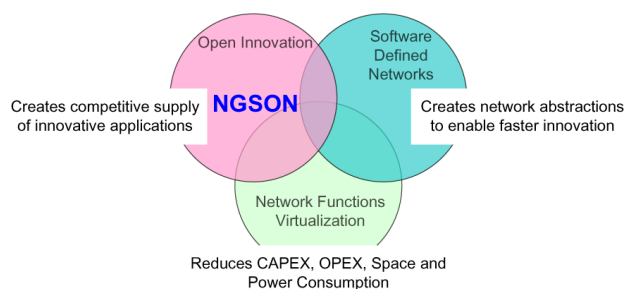


Figure 1 NGSON Interfacing SDN and NFV(adapted from ETSI NFV White Paper)

3. Know how to select best fit routes depending on a number of rules and policies
4. Preserve once gained knowledge

Any form of manual upfront provisioning or commissioning these data into a database would not meet the objectives of being truly context-aware and dynamically adaptive. NGSON needs to be able make decisions on the fly based on a set of rules/policies and previously gained knowledge.

An ontology with a descriptive logic reasoner is able to provide this functionality.

The following standards and open platforms are available to support this NGSON ontology:

1. OWL2 - An ontology descriptive language standardized by the W3C consortium, based on XML syntax.
2. Protégé - An open software platform for designing ontologies in OWL2, also providing a descriptive logic based reasoner released by Stanford University.
3. Protégé - An Descriptive Ontology for Linguistic and Cognitive Engineering An upper level ontology released by the Institute of Cognitive Science and Technology of the Italian National Research Council

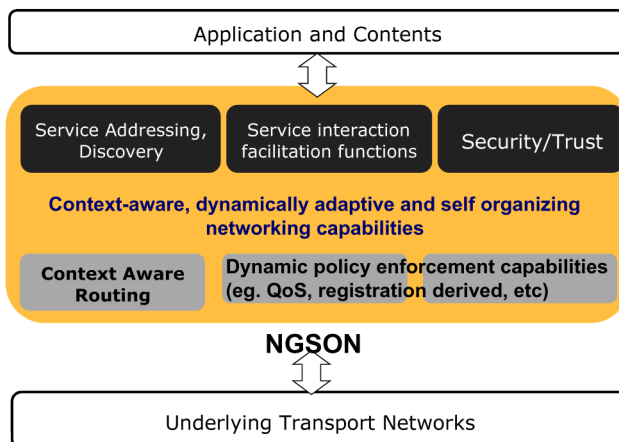
2 Service Composition Ecosystem

2.1 Similarities between Service Composition, Workflows and Business Processes

Most networking services may be considered as self-contained, self-describing, modular applications that can be published, located, and invoked across a network. However, the ability to efficiently and effectively select and integrate inter-organizational and heterogeneous services at runtime is an important step towards the development of service applications. In particular, if no single service can satisfy the functionality required by the user, there should be a possibility to combine existing services together in order to fulfill the request.

Despite previous efforts, the service composition still is a highly complex task, and it is often already beyond the human capability to deal with the whole process manually. The complexity, in general, comes from the following sources. First, the number of services available increased significantly during the recent years, and one can expect to have a huge service repository to be searched. Second, services can be created and updated on the fly, thus the

NGSON Framework Diagram



Geneva Switzerland, 17 October 2011

Figure 2 NGSON Framework

composition system needs to detect the updating at runtime and the decision should be made based on the up to date information. Third, services can be developed by different organizations, which use different concept models to describe the services, however, there does not exist a unique language to define and evaluate the services in an identical means. Therefore, building composite services with an automated or semiautomated tool is critical. To that end, several methods for this purpose have been proposed. In particular, most researches conducted fall in the realm of workflow composition or AI planning. For the former, one can argue that, in many ways, a composite service is similar to a workflow [2]. The definition of a composite service includes a set of atomic services together with the control and data flow among the services. Similarly, a workflow has to specify the flow of work items. The current achievements on flexible workflow, automatic process adaptation and cross-enterprise integration provide the means for automated services composition as well. In addition, the dynamic workflow methods provide the means to bind the abstract nodes with the concrete resources or services automatically.

A similar scenario can be observed in the modeling of business processes, where the Object Management Group has released a Business Process Model and Notation standard that uses workflow notational patterns to model business processes. There are also a few competing business process modeling languages available [3].

2.2 Using Deontic Logic to address drawbacks in current modeling schemes

A drawback of many current modeling schemes is that modalities are only implicitly expressed through the structure of the process flow.

All activities (service executions) are implicitly obligatory, i.e. mandatory, and whenever something should be permissible, i.e. optional, a branching node is used to split the modeling flow to offer the possibility to execute the activity or perhaps wait for an event or do nothing. This implies that the decision whether to execute one or more activities is described within that branching node, therefore the separation of decision and execution requires additional modeling elements and a comprehensive understanding of the entire process to identify obligatory and permissible activities.

Applying deontic logic can help here. While in classical logic, statements have values of either true or false, in deontic logic statements can be qualified by modalities. Modal logic began with Aristotle's analysis of statements containing the words "necessary" and "possible". These are but two of a wide range of modal connectives, or modalities that are abundant in natural and technical languages. Briefly, a modality is any word or phrase that can be applied to a given statement *S* to create a new statement that makes an assertion about the mode of truth of *S*: about when, where or how *S* is true, or about the circumstances under which *S* may be true.

Von Wright divided modal concepts into three main groups:

1. Alethic modes or modes of truth (necessary, possible and contingent)
2. Epistemic modes or modes of knowing (verified, undecided, and falsified)
3. Deontic modes or modes of obligation (obligatory, permitted, and forbidden)

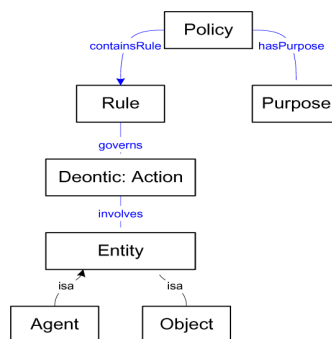


Figure 3 Top Level of Service Composition Policy Ontology

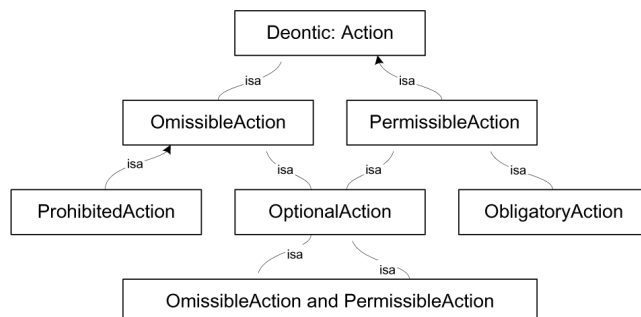


Figure 4 Deontic actions

Deontic Logic, therefore, is sometimes characterized as a form of modal logic describing the logic of prohibitions, permissions and obligations or the logic of ideal (according to certain norms) versus actual behavior.

Using W3C OWL2 Service Composition policies can be expressed in an ontology like:

‘Modalizing’ the ontology classes the reasoner will be able to deduct further service composition rules by applying Deontic Logic methods:

3 Interfacing with NFV and SDN

OWL2 ontologies may be augmented by forms of Rule Interchange Format, a further W3C standard. Rules in RIF provide the necessary functionality to specify procedural attachments in an OWL2 ontology, i.e. programs written in a procedural language that can be invoked by rules. These procedural programs may provide the necessary interfacing algorithms and code for communication with NFV and SDN functions.

4 Outlook

After compiling a current overview on available knowledge and semantic representation techniques as part of a Workshop on Software Defined, Context Aware and Dynamically Adaptive Services Overlay Networks (EcoSDSON) an attempt to provide a first, yet incomplete, OWL2 ontology in XML/RDF syntax should be undertaken.

Future research should also attempt to include temporal aspects into service composition design to accommodate for time relevant and dependent changes from a service as well as network related view.

References

- [1] Jinghai Rao and Xiaomeng Su, “A Survey of Automated Web Service Composition Methods”, Norwegian University of Science and Technology, Department of Computer and Information Science, N-7491, Trondheim, Norway, {jinghai,xiaomeng}@idi.ntnu.no
- [2] F. Casati, M. Sayal, and M.-C. Shan. “Developing e-services for composing eservices.” In Proceedings of 13th International Conference on Advanced Information Systems Engineering(CAiSE), Interlaken, Switzerland, June 2001. Springer Verlag.
- [3] C Natschläger-Carpella, “Extending BPMN with Deontic Logic”, Dissertation at Johannes Kepler Universität Linz, Austria, 2012
- [4] “OWL 2 Web Ontology Language Structural Specification and Functional-Style Syntax (Second Edition).” W3C Recommendation, 11 December 2012. Available at: <http://www.w3.org/TR/owl2-syntax/>
- [5] “OWL 2 Web Ontology Language Direct Semantics (Second Edition).” W3C Recommendation, 11 December 2012. Available at: <http://www.w3.org/TR/owl2-direct-semantics/>.
- [6] “OWL 2 Web Ontology Language Profiles (Second Edition).” W3C Recommendation, 11 December 2012. Available at: http://www.w3.org/TR/owl2-profiles/#Feature_Overview_3
- [7] “RIF Framework for Logic Dialects (Second Edition).” W3C Recommendation, 5 February 2013. Available at: http://www.w3.org/TR/rif-fld/#Semantic_Framework.
- [8] “RIF Core Dialect (Second Edition).” W3C Recommendation 5 February, 2013. Available at: <http://www.w3.org/TR/rif-core/>.
- [9] C. Masolo, S. Borgo, A. Gangemi, “DOLCE : a Descriptive Ontology for Linguistic and Cognitive Engineering”. Technical report, Institute of Cognitive Science and Technology, Italian National Research Council, 2003
- [10] Handbook of Philosophical Logic, Vol II: Extensions of Classical Logic, Kluwer Academic Publishers Group, Dordrecht, Holland, 1984

Biography



Reinhard Schrage is an independent Telecommunications Consultant. He received his MSc degree in Mathematics and Computer Science at the University of Hannover in 1980. He has held several engineering and managerial posts at cutting edge technology providing enterprises and telecommunication incumbents. His assignments ranged from designing and providing customer tailored software for Government departments to responsible network planning for a globally operating financial services network. Now based in Germany he has studied in the USA, lived in New Zealand, the UK, Canada, Spain and Switzerland. He is a contributor to the IEEE P1903 'Next Generation Service Overlay Networks' Workgroup.