INFOTEH-JAHORINA Vol. 16, March 2017.

The SecGENE ontologies framework

Valentina Nejkovic, Filip Jelenkovic, Milorad Tosic
Intelligent Information Systems Laboratory

Nenad Milosevic, Zorica Nikolic
Laboratory for Telecommunications Networks and Signal Processing

Faculty of Electronic Engineering, University of Nis
Nis, Serbia
(valentina, fillip.jelenkovic, milorad.tosic, nenad.milosevic, zorica.nikolic)@elfak.ni.ac.rs

Abstract — **The SecGENE experimentation framework aims to provide a solid practical evidence that automatic code generation based on semantic descriptions of experiments on testbeds and other infrastructure provided by EU H2020 SoftFIRE project is capable of supporting advanced experimentation with proposed use of new-coming 5G technologies. The flexibility of such experimental environment is addressed by adoption of domain and system ontologies for formal representation of semantics of the problem. This paper describes the ontologies framework that is needed for the SecGENE implementation. (Abstract)**

Keywords - ontology; automatic code generator; networking testbed (key words)

I. INTRODUCTION

Demand for testbed infrastructures that support 5G technologies and mechanisms for experimenting by research community and industry is getting higher. At the same time, testbed operators do not deploy required infrastructure because of the constraints posed by vendors that restrict the configuration capabilities to a certain extent. The EU H2020 project titled Software Defined Networks and Network Function Virtualization Testbed within FIRE+ (SoftFIRE) deals with integrated infrastructure that has the goal to enable, nurture and support an ecosystem of different companies that are interested in developing services and applications. Also, the project supports companies to create a state of the art control middleware based on network function virtualization (NFV) and software defined networks (SDN) technologies in 5G perspective [1]. SEmantics driven Code GENEration for 5G networking experimentation (SecGENE) builds upon the SoftFIRE platform to assist experimentators by generating automatically software code for experiments from a high-level specification. This paper presents an ontology experimentation framework developed within the scope of the SoftFIRE project with a goal to provide a solid practical evidence that SoftFIRE testbeds, together with complementing experimentation infrastructure, are capable to support advanced experimentation using automatic experiment code generator.

SecGENE experimentation framework includes a set of the domain and system ontologies, which gives semantic descriptions of network sensing and experiment execution flow. SecGENE ontologies are: Coordination by Spectrum Sensing for LTE-U Ontology (CoordSS Ontology - CO), Open-Multinet Ontology (OMN), Command Line Ontology (CLOnt), Semantic OEDL ontology (SOEDL), Network Capability ontology (NWC), Experiment Flow Ontology (EFO), Network Sensor ontology (NS) and SecGENE Client ontology (SCO).

We used OMN ontology to define experiment topology [2]. Using the ontology driven user interface, user can design experiment execution flow. Then, the code generator generates experiment code in the form of The OMF (Control, Management and Measurement Framework) Experiment Description Language (OEDL) [3] and Shell Script, which can be executed on nodes of testbed. OMF consists of software components that are used by operators for testbed management and by users for experiment orchestration. Today, OMF operates on several testbed deployments worldwide with many different types of resources and technologies. OMF provides a set of tools for describing and executing experiments and collecting experiment's results; and set of services for managing and operating testbed resources. Generated code could be executed on testbed nodes. For the communication among nodes and SecGENE server, SecGENE client application is used, which is semantically described by SCO.

iPerf software tool is used for implementation of network sensing. Thus, we need semantic descriptions of iPerf. NWC provides definitions of concepts of iPerf parameter, component and experiment execution. The NWC is complemented with Command Line ontology (CLOnt) and Semantic OEDL ontology (SOEDL). CLOnt is reused from the CoordSS ontologies framework [4]. Execution experiment flow provides semantic definitions for sequencing of execution steps in an experiment flow.

Section II of this paper gives background on federated testbeds because the aim of SecGENE project is seamless experimenting on networking testbeds in order to facilitate different experiments. Then we briefly introduce ontologies as the crucial constituent part of the framework. The main part of this paper is Section III where we introduce and discuss the SecGENE framework ontologies hierarchy. The last section is Section IV, which concludes paper.

The research leading to these results has received funding from the European Union's Horizon2020 Programme under grant agreement no 687860, named SoftFIRE.

II. BACKGROUND

A. Federated testbeds

In order to conduct testing for computational tools and new technologies, testbeds are widely used today. Networking testbeds are related to networking and computing in general. Networking testbeds are widely used in different application domains, platforms and environments in order to support experimenting necessary for development of new emerging communication technologies.

Usually testbeds are built for a specific project or to study a specific technology. Testbeds can be heterogeneous and can provide different environmental settings. Testbeds can be joint and federated. In the federation, each testbed maintains its own administration scope, architecture and functionality but delegates its resources into a common shared pool. There are a number networking testbeds, such as: 5GIC [5], FUSECO Playground from FOKUS Fraunhofer/TUB [6], Orbit [7], Nitos [8], Fibre [9], PlanetLab [10], Multi Radio Access Technology (MultiRAT)-Testbed [11] etc..

5G research activities, which include experimenting over testbeds, are very important and a number of projects have just been finished or are still running in this area, under EU FP7 and HORIZON 2020 funding schemes. SoftFIRE [1] includes federated testbeds infrastructure comprising experimental networks and programmable resources designed according to NFV/SDN principles. Such testbed infrastructure supports the orchestration of resources according to the needs of different applications. FLEX (FIRE LTE testbeds for open experimentation) [12] aims at the improvement of FIRE's infrastructure, i.e. cellular access technologies and LTE. FLEX's testbeds feature both open source platforms and configurable commercial equipment for different cellular setups. FLEX was built on the basis of current FIRE testbed management and experiment control tools and developed new monitoring tools. WiSHFUL (Wireless Software and Hardware platforms for Flexible and Unified radio and network controL) [13] deals with faster wireless solution development and its shorter testing cycles. It defines software modules with unified interfaces that permit wireless developers to quickly implement and validate advanced wireless network solutions. WiSHFUL also develops portable testbeds, which may be delivered anywhere where needed.

B. Ontologies

Different scientific domains used ontologies in order to formalize knowledge and organize information [14]. They are used for different tasks, such as improving communication between agents and reusing knowledge about data model schema. Value of data and information is improved by using ontologies due to support for proven highly effective meta-data management. Ontology represents a formalized way of representation of human knowledge. It is scalable, distributed, agile, code-independent, understandable by machines, open, supported by communities and enterprises, standardized and manageable. By using ontologies knowledge can be organized as: 1) a set of concepts and properties for these concepts; 2) a set of facts associated with the concepts.

Ontologies were considered for adoption to different problems within networking testbed community [15][16][17][18]. The main focus so far was on semantic resource description [19] and provisioning [20]. Also, ontologies are successfully adopted to IoT infrastructure [21].

Regarding the platforms federation and interoperability support, some research initiatives have been focusing either on middleware frameworks or on the modeling of related knowledge. In the scope of the Fed4FIRE project [22][22] efforts are made to define a homogeneous way of describing heterogeneous resources within federated testbeds. Starting under the umbrella of Open-Multinet (OMN) [23], they have been working towards a set of upper ontologies to describe federated infrastructures and their underlying resources. These ontologies support a number of use cases to semantically manage the whole life-cycle of a resource: discovery, selection, reservation, provisioning, monitoring, control, termination, authentication, authorization, and trustworthiness.

The ontology driven experimentations framework provides a flexible support to testbeds and other experimentation infrastructure as proven in the EU FP7 FLEX project by the Coordss framework [4][24]. Such a support is needed for the advanced experimentation required for the design and analysis of the components of 5G systems, for instance in the spectrum usage analysis for the coordination in the unlicensed frequency bands.

III. SecGENE Ontologies Framework

The proposed SecGENE framework shown in Figure 1. includes a set of the domain and system ontologies for semantic descriptions of the SecGene client, network sensing, and experiment execution flow. Using Ontology driven user interface user defines experiment topology, where OMN ontology is used for defining experiment topology. Using the ontology driven user interface, user can further design experiment execution flow. Then, the code generator generates OEDL experiment code and Shell Script code. The code generator uses SecGENE ontologies framework. iPerf software tool is used for implementation of network sensing **Error! Reference source not found.**. Thus, we need semantic descriptions of iPerf. Network Capability ontology (NWC) provides definitions of concepts of iperf parameters, component and experiment execution. The NWC is complemented with Command Line ontology (CLOnt) and Semantic OEDL ontology (SOEDL). CLOnt is reused from the CoordSS ontologies framework [4]. Execution experiment flow provides semantic definitions for sequencing of execution steps in an experiment flow.

A. OMN ontology

Everything in the SecGENE environment is considered as a accessible resource. With a goal of knowledge reuse and interoperability, we adopted the OMN Resource ontology which provides the concepts and methods to describe resources present in Future Internet platforms. In particular, OMN imports NOVI ontology with Resource ontology as a constituent part developed in the NOVI EU-funded project [2][26]. The OMN defines a formal information model for federated infrastructure management. Corresponding data models are developed in order to enable the communication among the various components in software architecture. The NOVI information model describes resources at a conceptual level, including all the components required to support the operation of the NOVI software. This ontology also holds information on how resources are connected together in a federated infrastructure.

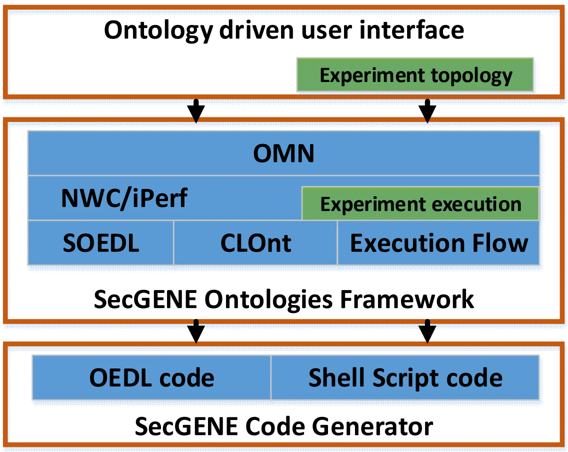

Figure 1. SecGENE framework ontologies

The main class in Resource Ontology is omn:Resource, and subclasses are:

- omn:Node, which is used to represent physical nodes.
- omn:NetworkElement, is an abstract class with following subclasses: omn:Interface, omn:Link and omn:Path.
- omn:NodeComponent is an abstract class, which give description of components of nodes (for example, CPU, Memory, Storage, SwitchingMatrix and LoginComponent).
- omn:Service allows the user to express the service level desired and to decouple the desired service from the actual physical implementation.

Classes which describe properties of resources are:

- omn:Location, which provides a way to describe location of resources.
- omn:Lifetim, which describes the time dimension of other objects, such as reservations, or availability of nodes.
- omn:Group describe groups of resources, which provides support for main concepts (omn:Platform and omn:Topology subclasses) within the NOVI federation. A omn:Platform describes a testbed within testbeds federation, while omn:Topology define a group of resources that user requests [26].

OMN Monitoring ontologies are identified as useful for SecGENE purposes. Those set of ontologies covers various monitoring services provided for federation administrators, experimenters, and testbeds federation services. For the purpose of the SecGENE, the most relevant are OMN Monitoring Data Ontology and OMN Monitoring Tool Ontology. From OMN Monitoring Data Ontology we reused classes: omn:Metric, omn:MeasurementData, omn:Data and omn:DataFormat. Also, we used subclasses omn:SimpleMeasurement and omn:MeasurementParameter, from the main class omn:MeasurementData. From OMN Monitoring Tool ontology we used omn:MeasurementTool and omn:MonitoringTool.

B. *CoordSS ontology*

The CoordSS ontologies framework is evaluated and used for experimenting over testbeds federation and coordination in heterogeneous communication networks [4]. One problem considered through the CoordSS ontologies framework is the coordination in heterogeneous communication networks with no constraints in the band to operate whether licensed or unlicensed. The CoordSS framework represents a base for coordination protocols based on semantic technologies and ontologies.

The CoordSS framework comprises of ontologies as shown in Figure 2: command line ontology (CLOnt), wireless ontology (WDSO), spectrum sensing capabilities ontology (SSCO), Meta System Architecture ontology (MSAO), Resource Description Framework ontology (RDFO). They specify semantic descriptions of radio spectrum, coordination, frequency selection, dynamic spectrum access, command line, wireless, and spectrum sensing capabilities of supporting software.

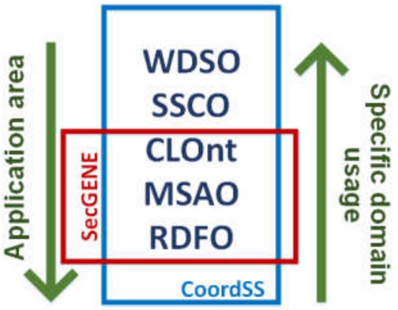

Figure 2. The CoordSS framework ontologies

We reuse CoordSS framework ontologies as needed for the SecGENE framework. As depicted in Figure 2, for the purposes of SecGENE we used CLOnt, MSAO and RDFO. RDFO represents a very general ontology, and can be used to describe any concept. MSAO represents ontology for describing in computational systems. MSAO gives descriptions of entity, element and attribute concepts, with main class Entity and subclasses Element and Attribute. RDFO is the most generic ontology in the framework, and describes value and property concepts.

C. *Network Sensing Ontology*

The fundamental concepts for network sensing domain are specified in the Network Sensing Ontology (NS). Network sensing measurements of the bandwidth in IP networks are practically considered in the context of one possible implementation represented by the iPerf **Error! Reference source not found.**. The Network Sensor has implementation in a form of a software tool called iPerf. IPerf has command line arguments, receives data from nodes and writes results to the database. The NS ontology semantically describes iPerf software module that implements throughput measuring. IPerf supports different parameters related to timing, buffers and protocols, such as TCP, UDP, SCTP with IPv4 and IPv6. For each test iPerf reports the bandwidth. Since iPerf receives arguments via command line, NS uses CLOnt for semantic description of concepts used in command line. CLOnt has Command and Argument classes.

IPerf implements Network Sensor. Further, iPerf has iPerf:iPerfCommand that executes on nwc:iPerf module, and which has iPerf:iperfArgument. Arguments that we semantically described are: port, format, and bind, which hold its shortnames, comments and names, and they are semantically described. nwc:iPerfParameter represents feature of iPerf. nwc:iPerfParameter is in connection with nwc:Parameter (iPerfParameter is subclass of nwc:Parameter). iPerf:iPerfParameters are interface, port, period and protocol. iPerf:iPerfParameter is implemented by iPerf:iPerfArgument.

D. SecGENE Client Ontology

SecGENE client application is responsible for communication among server and testbed nodes in SecGENE environment. SecGENE Client Ontology (SCO) contains statements describing SecGENE client application with its parameters and command line arguments is called SecGeneClient ontology (with prefix sgc). Parameters are represented as individuals of the sgc:SecGENEparameter class. Main parameters for the client are:

- server represents the server that is responsible for sending and receiving client messages.
- agent is the user of the client. It can be human user or another software component.
- channel represents communication channel on which agents sends and receives messages.
- message contains data that are send from one agent to another via given channel.

SecGENE client represents a console application. Arguments of this application includes: -a (URI of the agent), -ch (URI of the channel), -s (URL of the server), -sub -pub -reg -rmsgs (subscribe, publish, register and read messages, corresponding operations). SecGENEargument is the class of which all arguments are type of. SecGENEcommand class is connected with arguments, via hasArg property, and with SecGENEClient class via executesOnSecGENEclient property.

E. Experiment Flow Ontology

Flow of the experiment consists of the sequence of steps, which are represented by the FlowElement class. Experiment instructions are represented by the ExecutionStep class, and the sequence of steps is created by using the TransferStep class and its sub classes. TransferStep represents the concept that is used as an abstraction of sequence. Properties passesTo and receviesFrom are used to annotate which ExecutionStep follows and which one precedes. We differentiate four types of TransferSteps, with the corresponding definition:

- execStart represents the first step in experiment execution, e.g. start of the experiment.
- execEnd represents final step in experiment execution.
- execNext executes one ExecutionStep that is connected via passesTo property.
- execAll executes all ExecutionSteps that are connected vai passesTo property. All ExecutionsSteps are executed in parallel.

Classes FlowElement and ExecutionStep have seeAlso property that reference them to concpets in OMN ontology, Dependency and ExecutionService classes.

F. Semantic OEDL Ontology

SOEDL semantically describes basic building blocks of the OEDL language. Main SOEDL concepts are:

- soeld:OEDLElement is the base class for OEDL specific commands used to describe experiments, which is further specialized to Application, Measurement, Group, Event and Experiment.
- soeld:OEDLProperty represents possible arguments for commands. For connection among command and its argument property hasProperty is used. By subclassing this property we have created new properties that described possible arguments for each command.

For concrete values representation for each OEDL command and command argument two semantic properties are used:

- soeld:elAttribute is attribute associated to an OEDL element, and represents value for that attribute.
- soeld:propAttribute contains more detailed description for arguments of property, such as description, type and name. Values for this attributes are set by defining soeld:propAttrValue.

Since our experiment requires iPerf software module we first created definition of that module in SOEDL. Attributes for this application are set according to OMF requirements. Every argument that iPerf application have is defined as soedl:ApplicationProperty. For every argument there are attributes that are required by OMF. Definitions of arguments are connected to knowledge base via soedl:implementsArgument relation. Using this relation we have connected experimenter knowledge with experiment that can be executed in OMF based testbed.

After created definition of iPerf software module in SOEDL, experiment should be described in SOEDL, which will be done automatically based on users given descriptions of experiment topology and execution flow over Ontology driven user interface. For example, the network sensing experiment has two properties, one resource group (that uses previously defined iperf software module) and one event. Resource group gives parameters of testbeds nodes involved in experiment, while event describe experiment flow, which contains plain Ruby code that is executed by OMF framework.

After experiment network expressed over SOEDL ontology, this semantic description of experiment could be transformed to the OMF executable code using SecGENE experimentation framework.

SOEDL ontology contains all OEDL concepts as described in [28] and by populating them with concrete instances and values we can create semantic description for an arbitrary experiment. After that we create experiment code that can be executed on any OMF capable testbed.

IV. Conclusion

The SecGENE framework ontologies overview is given in this paper. The main impact of the SecGENE ontologies framework is to facilitate experimentation based on flexibility of semantic automatic code generation. That magnifies the importance of heterogeneous testbed infrastructures for the experimentation in diversity of environments.

The presented SecGENE ontologies framework provides the semantic foundation for automatic code generation for experiments conducted on testbeds. The OMF framework and OEDL language as well as different shell scripts could be used as the target platform for practical implementation of the automatic code generator. The ontological foundation will enable easy integration into the wider context of the FIRE initiative, particularly having in mind the SecGENE natural interoperability with the FIRE foundational information model.

References

[1] Software Defined Networks and Network Function Virtualization Testbed within FIRE+ (SoftFIRE), https://www.softfire.eu/, [accessed 15.01.17].

[2] Y. Al-Hazmi, Open-Multinet Ontologies – Their Hierarchy and Tooling, http://groups.geni.net/geni/raw-attachment/wiki/.../Session4/OMN-ontology.pdf, [accessed 15.01.17].

[3] T. Rakotoarivelo, M. Ott, G. Jourjon, I. Seskar, "OMF: a control and management framework for networking testbeds", ACM SIGOPS Operating Systems Review, Vol 43-4, 54-59, 2010

[4] M. Tosic, V. Nejkovic, F. Jelenkovic, N. Milosevic, Z. Nikolic, I. Seskar, "The CoordSS experimentation framework ontologies", 23nd Telecommunication Forum Telfor 2015, Belgrade, Serbia, November 25-27th, 2015.

[5] 5G Innovation Centre, http://www.surrey.ac.uk/5gic; [accessed 15.01.17].

[6] FUSECO Playground, https://www.fokus.fraunhofer.de/go/de/fokus_testbeds/fuseco_playground; [accessed 15.01.17].

[7] OrbitLab, http://www.orbit-lab.org; [accessed 15.01.17].

[8] NITlab, NITOS, http://nitos.inf.uth.gr/; [accessed 15.01.17].

[9] T. Salmito et al, FIBRE - An International Testbed for Future Internet Experimentation, 2014, http://sbrc2014.ufsc.br/anais/files/ salao/SF-ST1-1.pdf; [accessed 15.01.17].

[10] PlanetLab Europe, http://www.planet-lab.eu/; [accessed 15.01.17].

[11] https://www.softfire.eu/dttestbed/; [accessed 15.01.17].

[12] FIRE LTE testbeds for open Experimentation (FLEX), http://www.flex-project.eu/ [accessed 15.01.17].

[13] http://www.wishful.eu/; [accessed 15.01.17].

[14] T.R. Gruber, "A translation approach to portable ontology specifications," Knowledge Acquisition, vol. 5, no. 2, p. 199–220, 1993.

[15] C. Fortuna, M. Tosic, M. Chwalisz, P. deValck, I. Moerman, I. Seskar, "Representation of spectrum sensing experimentation functionality for federated management and control", In IFIP/IEEE International Symposium on Integrated Network Management (IM), May, 2015, pp. 1226-1229.

[16] Fourth GENI & FIRE Collaboration Workshop, Washington DC, September 17-18, 2015, http://groups.geni.net/geni/wiki/GENIFire CollaborationWorkshopSeptember2015, [accessed 15.01.17].

[17] M. Giatili, C. Papagianni, and S. Papavassiliou, "Semantic aware resource mapping for future internet experimental facilities." 19th IEEE International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD), 2014.

[18] V. Maglaris, et al., "Toward a holistic federated future internet experimentation environment: the experience of NOVI research and experimentation", IEEE Communications Magazine, vol. 53, no. 7, 2015, pp. 136-144.

[19] F. Jelenkovic, M. Tosic, and I. Seskar, "A semantic approach to wireless networking testbed infrastructure." 14th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt)., 2016.

[20] W. Vandenberghe, et al., "Architecture for the heterogeneous federation of future internet experimentation facilities", In Proceedings of the Future Network and Mobile Summit (FutureNetworkSummit), Lisbon, Portugal, 3–5 July 2013; pp. 1–11.

[21] J. Lanza, et al. "A Proof-of-Concept for Semantically Interoperable Federation of IoT Experimentation Facilities", Sensors, vol. 16, no. 7, 2016, pp. 1006.

[22] A. Willner, D. Nehls, "Semantic-based management of federated infrastructures for future internet experimentation.", 2016, https://www.netsys2015.com/wp-content/uploads/NetSys2015_PhD-Forum_Willner.pdf, [accessed 15.01.17].

[23] Y. Al-Hazmi, Y. Magedanz, "Towards Semantic Monitoring Data Collection and Representation in Federated Infrastructures", In Proceedings of the 3rd International Conference on Future Internet of Things and Cloud (FiCloud), Rome, Italy, 24–26 August 2015; pp. 17–24.

[24] M. Tošić, V. Nejković, F. Jelenković, N. Milošević, Z. Nikolić, N. Makris, T. Korakis, "Semantic coordination protocol for LTE and Wi-Fi coexistence," Proceedings of papers European Conference on Networks and Communications (EuCNC) 2016, Athens, Greece, June 2016, pp. 69-73.

[25] iPerf, https://iperf.fr/; [accessed 15.01.17].

[26] W. Adianto, C. de Laat and P. Grosso, Future Internet Ontologies: The NOVI Experience, www.semantic-web-journal.net/system/files/ swj580.pdf, [accessed 15.01.17].

[27] The OMF Experiment Description Language (OEDL), https://omf.mytestbed.net/projects/omf6/wiki/OEDLOMF6, [accessed 15.01.17].