








# An Analytical Computing Infrastructure for Monitoring Dynamic Networks Based on Knowledge Graphs

Igor Kulikov<sup>1</sup> , Gerhard Wohlgenannt<sup>2</sup> , Yulia Shichkina<sup>1</sup>  ,  
and Nataly Zhukova<sup>3</sup> 

<sup>1</sup> Saint-Petersburg Electrotechnical University “LETI”, Saint-Petersburg, Russia  
i. a. kulikov@gmail.com, strange.y@mail.ru

<sup>2</sup> ITMO University, St. Petersburg, Russia  
gwohlg@itmo.ru

<sup>3</sup> St. Petersburg Institute for Informatics and Automation of the Russian  
Academy of Sciences, St. Petersburg, Russia  
nazhukova@mail.ru

**Abstract.** Dynamic network monitoring systems are typically designed to solve a predefined number of tasks, new requirements lead to expensive development efforts and sometimes even require changes in the system architecture. Knowledge graphs are powerful and flexible tools for information integration and supported by a set of standardized vocabularies and languages (the “Sematic Web” toolset). In this work, we discuss the application of knowledge graphs to develop and analyze an analytical computing infrastructure for a dynamic network monitoring system. As a typical dynamic network, a multiservice telecommunication network is considered. The presented system combines static models of a telecommunication network and dynamic monitoring data and makes it possible to obtain complex analytical reports using SPARQL queries over the knowledge graph. Those reports are of crucial importance to network stakeholders for improving the network services and performance. First, we analyze problems solved by traditional monitoring systems, and identify the classes of problems such systems cannot solve. Then we propose an analytical monitoring system architecture based on knowledge graphs to address these classes of problems. We present the system structure and detailed descriptions of the ontological and mathematical models of the resulting knowledge graph. In order to test the architecture discussed, we create an example task of the analytical monitoring system and analyze system performance depending on the size of the knowledge graph. The results of the analysis are presented using a number of SPARQL queries.

**Keywords:** Knowledge graph · Dynamic network · Monitoring system · Ontology · Domain ontology · Semantic web

## 1 Introduction

Dynamic network monitoring systems are designed for solving a known set of problems. New tasks are expensive to implement and often cannot be solved without architecture changes. We offer a more flexible system and data architecture.

As a typical dynamic network, a multiservice telecommunication network (TN) is considered. Telecom operators are in need of an analytical computing infrastructure to provide solutions for complex monitoring tasks, and changes in requirements and tasks. In this paper, we aim to

- a) investigate the set of features of current monitoring solutions,
- b) define a set of requirements for an analytical computing monitoring infrastructure,
- c) propose an architecture and data model for solving the discussed problems,
- d) create an example for a new task in this solution that provides dynamic modelling capabilities,
- e) define the benefits for TN stakeholders.

### 1.1 Traditional Telecommunication Network Monitoring Systems

We first discuss the goals, features and components of traditional Telecommunication Network Monitoring Systems (TN-MS) in order to establish which functionality is lacking in traditional systems. TN-MS are described as data providers for network management systems [1]. There are three main goals for traditional telecommunication network monitoring systems [2]:

- network performance monitoring;
- emergency monitoring;
- user account monitoring.

These goals overlap with the functional areas of network management systems [3]. As mentioned in the goals of the paper, we first discuss the problems that traditional monitoring systems can solve (different monitoring systems can be compared regarding their ability to solve these problems) [3–5]:

1. Report generation based on the main indicators of network quality according to the Service-Level-Agreement (SLA);
2. Trend identification concerning the main network performance indicators;
3. Trend forecasting for the main network performance indicators;
4. Network topology analysis;
5. SNMP support;
6. Application of an agent-based monitoring model;
7. Event logging;
8. Message delivery support for different delivery methods.

All of these tasks are elements in the structure of goals for monitoring systems. A typical monitoring system contains the following components [6]:

- The main server, including the server software core, the DBMS, the subsystem for interacting with agents, the user notification subsystem, the graphical user interface, the report generation subsystem, and the event logging subsystem.
- Agents, including the agent software core, the server interaction subsystem, the configuration subsystem, the monitoring subsystem (including the monitoring of physical parameters, the operating system status, the network host status, and the application status).

Data models of traditional monitoring systems are designed based on network performance indicators according to the SLA. As a rule, systems existing today store their data in SQL databases. It should be noted that traditional systems do not solve the problem of analyzing the relationship between monitoring parameters, the network structure, the structures of available data, the distribution of access rights provided by services and applications, and user behavior. Such problems can be solved by the analytical computing infrastructure of the monitoring system.

## 1.2 Problem Definition

By analyzing of incoming requests statistic from the stakeholders of the TN of a major cable TV operator in North America, we identify the following groups of features not available in traditional monitoring systems:

- User classification based on different criteria taking into account both traditional monitoring data and data from other systems (e.g. data on billing, location, distribution of access rights, statistics on the use of services, applications, and data);
- Search for information associated with network elements such as metadata associated with data assets, services schedules, previous behavior statistic etc. (with information broken down by users, services, applications, and data);
- Analysis of user interests (and their changes);
- Streamlining the search for key causes of incidents;
- Dynamic control of telecommunication network parameters based on monitoring metrics, including metrics on user interests and activity.

This is not an exhaustive list; it can be expanded after a more detailed analysis of the needs of telecommunication network operators in monitoring data. The problems mentioned above can be solved by creating an analytical computing infrastructure built on both a single traditional monitoring system and group of monitoring systems.

## 2 Requirements for Ther Analytical Computing Infrastructure for Monitoring a Telecommunication Network

### 2.1 Use-Cases for the Introduction of an Analytical Monitoring System

Here we discuss how monitoring system data can be analyzed along with various static models of the telecommunication network and data on user behavior regarding the use

of resources, services, and applications. Several use-cases divided into layers depending on the user groups are presented in Table 1.

**Table 1.** Analytical monitoring system scenarios

User role	Use-case
End-user (customer layer)	Monitoring current access restriction data
	Receiving personalized recommendations (offers of services and data for purchase)
	Improving the search service by taking into account the interests of the user when sorting search results
	Collecting geographic information and location data on user devices when providing services to the user
Network owners (business layer)	Monitoring information on user interests (for developing customized ads and giving personalized recommendations)
	Identification of target user groups for advertising purposes (analysis of statistics on user preferences)
	Generation of comprehensive analytical reports based on data from static network models and dynamic monitoring data
	Search for new semantic links between data from monitoring systems
	Two-way analysis of trends in various network performance indicators and their relationship with statistics on user behavior
Network operation (operations layer)	Instant search for the causes of problems encountered by users (expert search)

## 2.2 General Static and Dynamic Telecommunication Network Model Requirements

In order to solve analytical problems, it is necessary to combine a variety of static models of telecommunication networks that are available and add dynamic monitoring data. We suggest combining the following models and types of data:

### 1. Static models

- Billing model;
- Access permission model;
- Network topology model;
- Application hierarchy model;
- Service hierarchy model;
- Data model;

## 2. Dynamic data

- Data from traditional monitoring systems;
- Data from operational logs;
- Data on user activity.

The dynamic data needs to be connected with the static models.

### 2.3 Requirements for the Interaction Between a Static Model, Traditional Monitoring Data, and the Statistics on User Activity

In order for the analytical computing infrastructure to be able to solve the problems discussed, it is necessary to fulfill the following requirements for the structure of dynamic data and the interaction between these data and a static model:

- The data on the event being monitored should contain the following information:
  - event identifier;
  - time stamp;
  - event type identifier;
  - geographic information (if applicable);
  - a set of logical links between the event and the static network model.
- Events and network parameters that are fed into the analytical computing infrastructure of the monitoring system should be selected in such a way that they allow solving the problems at hand.
- Data flow parameters (data recording schedule, the number of monitoring parameters, methods and parameters for deleting obsolete data) need to be selected so that both the requirements for analytical reports are fulfilled and the desired performance is achieved (a system optimization issue connected with system design or configuration).

## 3 The Knowledge Graph as a Solution Core

### 3.1 The Knowledge Graph as a Core of the Solution

In general, using of knowledge graphs can support knowledge-driven applications and serve as a smart knowledge factory generating new knowledge. Knowledge graphs are used for both open-source projects (open knowledge graphs) and corporate ones (industrial knowledge graphs). Well-known open knowledge graphs are DBpedia [9], Google Knowledge Graph [10], YAGO [11], Wikidata [12]. Knowledge graphs provide an opportunity to expand our understanding of how knowledge can be managed on the Web and how that knowledge can be distinguished from more conventional Web-based data publication schemes such as Linked Data [24]. Standard problems solved by industrial knowledge graphs are for example [13]:

- Creating digital twins of real equipment.
- Risk management.
- Process monitoring.
- Operating services for sophisticated equipment.

In order to build the analytical computing infrastructure for monitoring a telecommunication network, we propose combining structural graph models of networks with dynamic data on network parameters and statistics on user activity in a single knowledge graph. This will allow making connections between monitoring data and the data from static network models as well as between different types of monitoring data (through semantic links within a single model). As a result, it will be possible to generate complex analytical reports that include data on both the network status and the links between different network processes.

We propose to represent the telecommunication network knowledge graph as an RDF (Resource Description Framework) graph, i.e. in “subject – predicate – object” triple format. In this configuration, a multitude of RDF statements form a directed graph with subjects and objects as nodes and links between them as edges [14, 15, 20, 21].

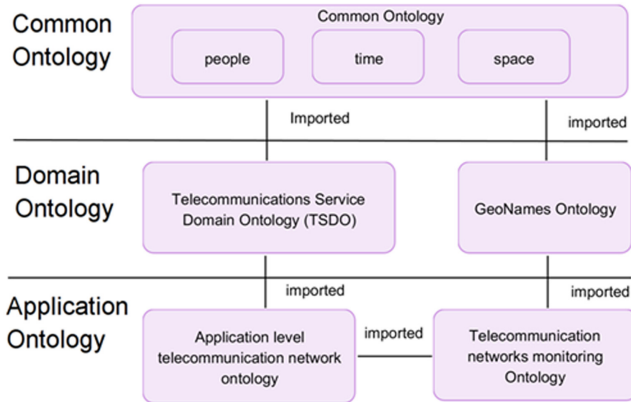
A telecommunication network provides users with services, which may include data transmission services (voice transmission or data transmission) or access to applications and/or data. Telecommunication networks are used by end users, business units of network operators, and owners. Each end user has access entitlements regarding services and data, and there can also be financial arrangements (billing). Communication channels can be different in both their physical properties (wired communication/optical communication/radio relay transmission) and bandwidth. A generalized model for monitoring a telecommunication network combines all of the components mentioned within a knowledge graph. The knowledge graph consists of mostly static structural models of the network, and of dynamic monitoring data that reflect user activity, services invocation, service performance statistics, errors, emergencies, and other events.

In contrast to traditional approaches, KGs make it possible to easily add new entity types to the model (static and dynamic), to use common domain ontologies to integrate external data, and to apply graph query languages for powerful search functionality.

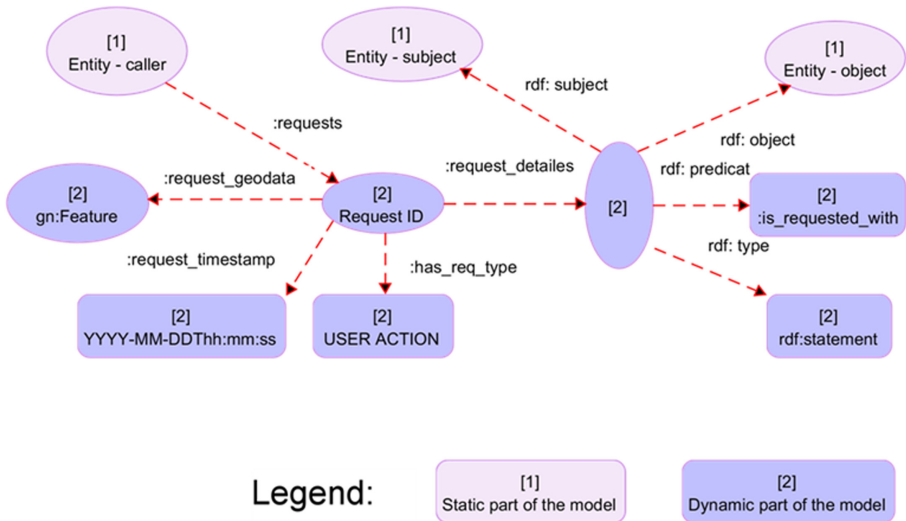
The static component of the knowledge graph of a telecommunication network is based on the Telecommunications Service Domain Ontology (TSDO) [16]. Based on the architecture of the semantic services of telecommunication networks [17], the analytical computing infrastructure has the following layers:

- Semantic web-service based on Unified Service Architecture;
- Common Service Facilities and Value-added service Layer;
- Personalized Application.

The use of a generally accepted ontological model is critical for the subsequent integration of the analytical computing infrastructure for monitoring the telecommunication network with external applications and systems that deal with semantically linked data. The structure of the ontological model is shown in Fig. 1.



**Fig. 1.** The ontological model of the knowledge graph of the analytical computing infrastructure.



**Fig. 2.** The structure of the dynamic part of the knowledge graph.

The services and applications are described using the Web Ontology Language (OWL) model [22], which is compatible with the ontology presented [17]. In order to add geographic data to the model, the GeoNames ontology is imported at the level of domain ontologies [7].

### 3.2 Dynamic and Static Parts of the Model

The following KG sketches the design of the dynamic data model (Fig. 2):

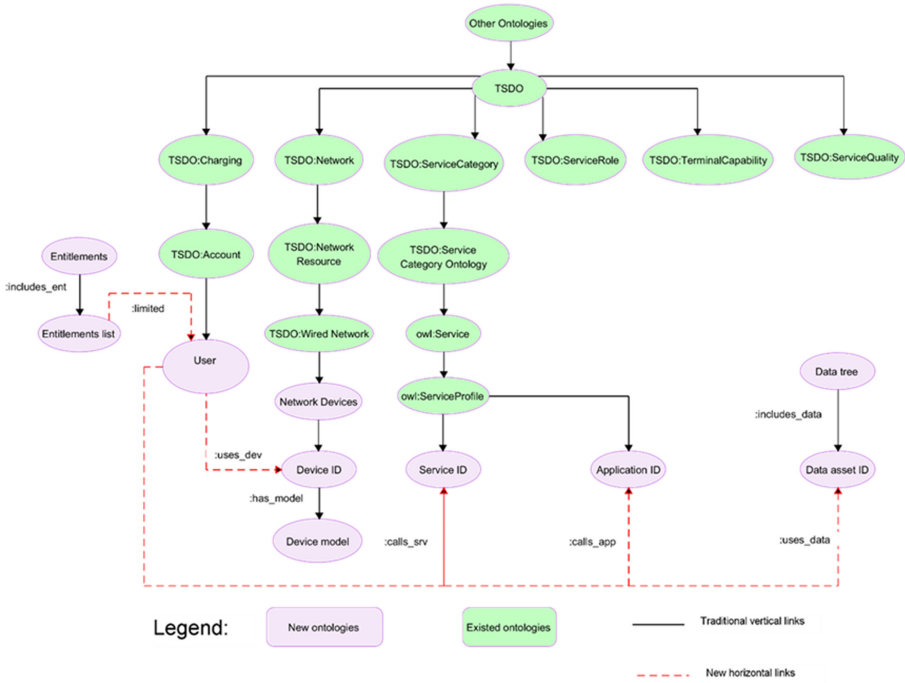


Fig. 3. The structure of the static part of the knowledge graph.

The static KG model is based on the ontological model shown in Fig. 1. The generalized hierarchy of the static model up to the application level (telecommunication network specialization) is shown in Fig. 3.

When designing the analytical computing infrastructure of a monitoring system, we start with the structural and ontological models of the static part of the knowledge graph. Next, the structure of dynamic data and the data arrival rate are defined.

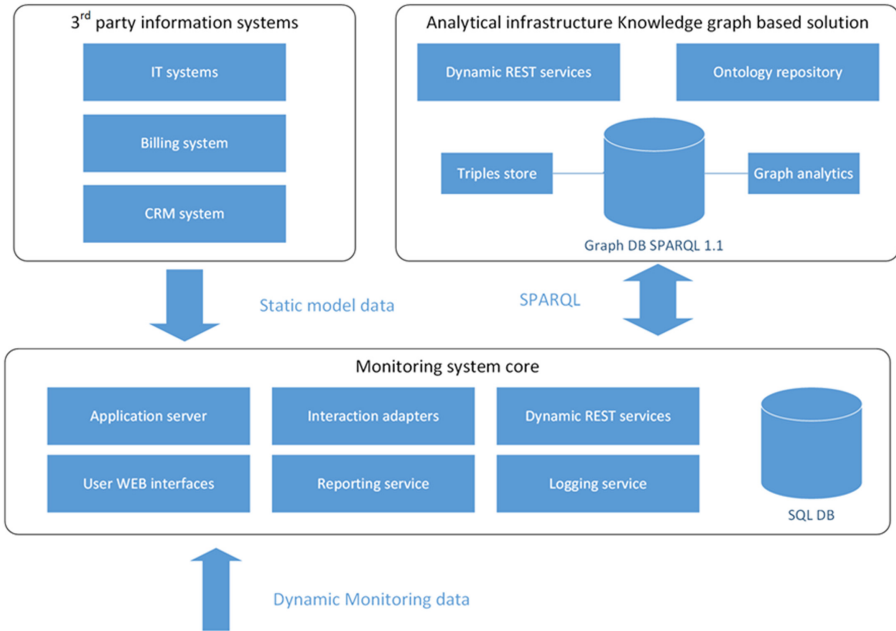
### 3.3 The Architecture of the Analytical Computer Infrastructure Based on the Knowledge Graph

The block chart of the proposed analytical computing infrastructure based on the knowledge graph is presented in Fig. 4.

The proposed system consists of the following components:

1. The monitoring system core. The core includes:
  - Application server accommodating the business logic for the performance of the whole system: schedule of interaction with other components, data bus, message exchange, file storage.
  - Dynamic REST service supporting API for queries made by external systems.
  - Set of adapters for querying data from external systems (monitoring, operator IT systems, etc.)





**Fig. 4.** The block chart of the analytical computing structure of a monitoring system based on the knowledge graph.

- Web interface for the system users and administrators.
- Reporting service which can represent reports in Web interface or send them to external consumers.
- System event logging service.
- SQL database designed to store monitoring dynamic data appropriate for storing in the system but inappropriate for placing in the knowledge graph.

## 2. Knowledge graph which includes:

- SPARQL 1.1 compliant RDF data storage. This component is the key element to the solution holding knowledge graph triples (static and dynamic components) and supporting the functions of adding/removing triples and searching in the RDF storage. The storage also includes a data analytics module. It stores both static and dynamic graph data connected by the common ontology.
- Ontology repository storing replicas of all ontological models the knowledge graph is based on. The delivered standards for data and ontology description: RDF [20], RDFS [21], OWL [22].
- Dynamic REST service supporting API for interaction with external systems, in particular, with the monitoring system core.

## 3. Operator IT systems supplying static data for the model used. Within the proposed monitoring system, the following operator IT systems are considered:

- The IT system for network infrastructure management supplies data on network topology, network devices, network services, network applications, accessible data, and access rights.
- The billing system supplies data on users, their devices, personal accounts, tariffs, and payments.
- The CRM systems supply data on the history of operator-user interaction.

### 3.4 Dynamic System Modeling

In order to evaluate performance, – we carried out tests to measure the speed of executing SPARQL queries depending on the size of the static and dynamic models of the knowledge graph, using the Metaphactory platform [8]. The parameters of the models that were analyzed and test results are presented in Table 2.

**Table 2.** Dynamic system modeling results

	Static data: Number of nodes: 100,000		
	Number of events		
	100,000	1,000,000	10,000,000
Request#1* execution time	1sec, 76ms	7sec, 683ms	1min, 49sec, 426ms
RDF data upload time, sec	21,8	198	2 542
	Static data: Number of nodes: 1,000,000		
	Number of events		
	100,000	1,000,000	10,000,000
Request#1* execution time, ms	2sec, 493ms	14sec, 188ms	2min, 1sec, 726ms
RDF data upload time, sec	95,5	283	2 712

\*-Request#1 is defined below.

From the experiments, we can conclude that the analytical computing infrastructure for monitoring a telecommunication network based on a knowledge graph in the example presented has acceptable performance indicators if the knowledge graph has a size of 1 million nodes in the static network model and covers 10 million dynamic events.

Different approaches to optimizing the speed of query execution are described in [18] and [19].

## 4 Example Solution

### 4.1 Use-Case

The overall idea of the use case is to analyze service call frequency of end-users. This use-case has been chosen as example of analyzing dynamic data from different models and information systems.

**Initial Data:** A telecommunication network that provides services, applications, and sells access to content. The devices used are both stationary and mobile. When using services, data is generated about the period of use and the location of the device. In addition, emergencies happening to the operator’s equipment are monitored taking into account geographic information.

**Task:** In this use case, we want to break down data on service call frequency by for following criteria:

- hours
- device models
- city districts

We want to overlay data on emergencies happening to the operator’s network with data on service call frequency and break it down by the categories mentioned. This is not a regular task for the traditional TN-MS because the data to be analyzed is in different operator IT systems. Also, the available data in traditional TN-MS systems is aggregated and many initial data associations have been lost. This makes the use-case interesting and relevant, and it covers some of the discussed monitoring tasks.

### 4.2 The Knowledge Graph Model

To solve the problem, we propose the following KG model (Fig. 5).

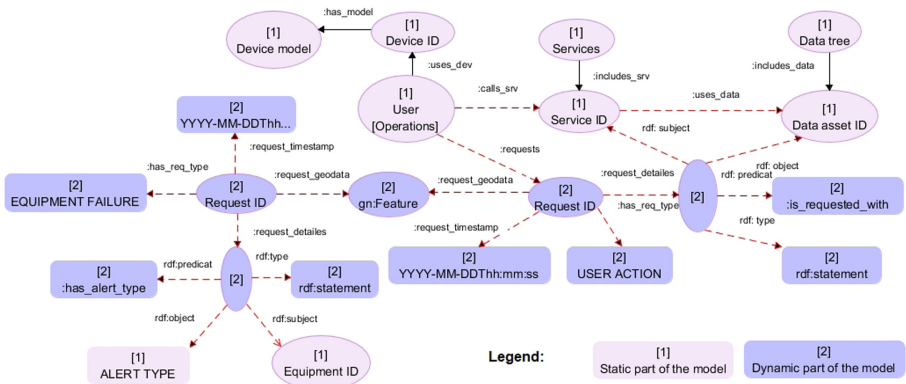


Fig. 5. The structure of the knowledge graph as an example of implementing the system.

### 4.3 SPARQL Requests/Responses

The application for generating an RDF/XML model of the knowledge graph, the RDF/XML model itself, and the SPARQL queries used in the paper are available on GitHub [23].

Below we provide a query and its response which limits the list of user’s events by the following criteria:

- Date: 2020-02-01
- Event type: USER\_ACTION
- Device model: Moto2k
- City district: <<https://sws.geonames.org/8504951/>>

```

SPARQL REQUEST #1:
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema>
PREFIX my: <http://127.0.0.1/bg/ont/test1#>
SELECT *
WHERE
{
  ?Device my:has_the_device_model "Moto2k" .
  ?Device my:has_id ?Device_id .
  ?User my:uses_device ?Device_id .
  ?Request_ID my:request_timestamp ?Date .
  FILTER contains(?Date, "2020-02-01") .
  ?Request_ID my:requests ?User .
  ?Request_ID my:has_req_type "USER_ACTION" .
  ?Request_ID my:request_geodata <https://sws.geonames.org/8504951/> .
  ?Request_ID my:request_details ?Details_ID .
  ?Details_ID rdf:subject ?Request_subject .
  ?Details_ID rdf:object ?Request_object .
}
    
```

The first rows of the response are shown in the Table 3.

**Table 3.** Response for the request #1

Device ID	User	Date	Request subject	Request object
D71	< <a href="http://127.0.0.1/User_71/">http://127.0.0.1/User_71/</a> >	2020-02-01T14:50:17	WatchTV	< <a href="http://127.0.0.1/Asset_7/">http://127.0.0.1/Asset_7/</a> >
D60	< <a href="http://127.0.0.1/User_60/">http://127.0.0.1/User_60/</a> >	2020-02-01T12:22:05	PPV	< <a href="http://127.0.0.1/Asset_4/">http://127.0.0.1/Asset_4/</a> >
D60	< <a href="http://127.0.0.1/User_60/">http://127.0.0.1/User_60/</a> >	2020-02-01T15:53:51	WatchTV	< <a href="http://127.0.0.1/Asset_10/">http://127.0.0.1/Asset_10/</a> >

The next SPARQL query (incl. result) retrieves equipment alerts for the following search criteria:

- Date: 2020-02-01
- Event type: EQUIPMENT\_FAILURE
- City district: <https://sws.geonames.org/8504951/>

**SPARQL REQUEST #2:**

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema>
PREFIX my: <http://127.0.0.1/bg/ont/test1#>
SELECT *
WHERE
{
  ?Request_ID my:request_timestamp ?Date .
  FILTER contains(?Date, "2020-02-01") .
  ?Request_ID my:has_req_type "EQUIPMENT_FAILURE" .
  ?Request_ID my:request_geodata <https://sws.geonames.org/8504951/> .
  ?Request_ID my:request_details ?Details_ID .
  ?Details_ID rdf:subject ?Request_subject .
  ?Details_ID rdf:object ?Request_object .
}
```

The first rows of the response are shown in the Table 4.

**Table 4.** Response for the request #2

Date	Request subject	Request object
2020-02-01T09:11:09	< <a href="http://127.0.0.1/Device_78/">http://127.0.0.1/Device_78/</a> >	STB_out_of_memory
2020-02-01T03:40:33	< <a href="http://127.0.0.1/Device_94/">http://127.0.0.1/Device_94/</a> >	STB_out_of_memory
2020-02-01T11:35:01	< <a href="http://127.0.0.1/Device_93/">http://127.0.0.1/Device_93/</a> >	STB_out_of_memory
2020-02-01T22:00:01	< <a href="http://127.0.0.1/Device_18/">http://127.0.0.1/Device_18/</a> >	STB_out_of_memory

Finally, we analyze the distribution of events for both user actions and equipment alerts. Request parameters:

- Date: 2020-02-01
- Event type: All event types
- Device model: All models
- City district: [https://sws.geonames.org/8504951](https://sws.geonames.org/8504951/)

```

SPARQL REQUEST #3:
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema>
PREFIX my: <http://127.0.0.1/bg/ont/test1#>
SELECT *
WHERE
{
  ?Request_ID my:request_timestamp ?Date .
  FILTER contains(?Date, "2020-02-01") .
  ?Request_ID my:request_geodata <https://sws.geonames.org/8504951/> .
  ?Request_ID my:request_details ?Details_ID .
  ?Details_ID rdf:subject ?Request_subject .
  ?Details_ID rdf:object ?Request_object .
}
    
```

The first few results are shown in the Table 5.

**Table 5.** Response for the request #3

Date	Request_subject	Request_object
2020-02-01T14:13:54	WatchTV	<http://127.0.0.1/Asset_2/>
2020-02-01T14:03:46	nPVR	<http://127.0.0.1/Asset_7/>
2020-02-01T14:50:17	WatchTV	<http://127.0.0.1/Asset_7/>
2020-02-01T09:11:09	<http://127.0.0.1/Device_78/>	STB_out_of_memory

## 5 Conclusion

With the proposed analytical computing infrastructure for monitoring a telecommunication network (as a typical most complex dynamic network) based on a knowledge graph it is possible to combine different static network models in a single semantic model and add dynamic monitoring data to the system. The KG model allows to address new classes of problems that could be tackled using traditional monitoring systems. Further, the KG (based on ontologies as backbone) can be easily integrated with other systems based on semantic data models. In addition to solving new classes of monitoring tasks, telecom operators can more easily realize complex analytical monitoring solutions and a more flexible architecture in general. From the end-user’s point of view, the operator can provide more personalized services. We discuss an example use case of an analytical problem of monitoring a telecommunication network. The example shows some of the benefits of analyzing dynamic monitoring data within a single knowledge graph. Test results show that such systems can process large amounts of data with acceptable performance. The suggested approach provides benefits when building an analytical monitoring infrastructure based on soft requirements and when the monitoring functionality needs to be extended in the future. KG

technologies allow to create powerful tools for system analysis. Also, this approach can be used in different subject areas for dynamic objects modelling, e.g. natural phenomena. The base of this model can be built using already existed models of machine learning [25–27]. In future work, we will study which kinds problems such models can solve in more detail, and how to optimize links in the KG, and then create a full prototype of the solution with the discussed benefits.

**Acknowledgment.** To Metaphacts GmbH, Daimlerstrasse 36, 69190, Walldorf, Germany for the license to model knowledge graphs on the Metaphactory platform.

**Funding.** The research was funded by Russian Foundation for Basic Research (RFBR) according to the research projects #18-57-34001 and #19-07-00784.

## References

1. Wong, E.: Network monitoring fundamentals and standards. Computer Science (2000). [https://www.cse.wustl.edu/~jain/cis788-97/ftp/net\\_monitoring/index.html](https://www.cse.wustl.edu/~jain/cis788-97/ftp/net_monitoring/index.html)
2. Stallings, W.: SNMP, SNMPv2, and RMON Practical Network Management, 2nd edn. Addison-Wesley Professional Computing and Engineering (1996). A good general reference in basics of RMON
3. Apostolopoulos, T.K., Daskalou, V.C.: On the implementation of a prototype for performance management services. In: IEEE Symposium on Computers and Communications, pp. 57–63 (1995). A research paper on a prototype for management services
4. Stanford University: Network monitoring tools. Stanford University. <http://www.slac.stanford.edu/xorg/nmtf/nmtf-tools.html>
5. Comparison of network monitoring systems. [https://en.wikipedia.org/wiki/Comparison\\_of\\_network\\_monitoring\\_systems](https://en.wikipedia.org/wiki/Comparison_of_network_monitoring_systems)
6. Natarov, A., Shirokii, A.: Next generation network monitoring systems—critical requirements and design. <https://doi.org/10.15688/mpcm.jvolsu.2018.3.4>
7. GeoNames ontology. <http://www.geonames.org/ontology/documentation.html>
8. Haase, P., Herzig, D.M., Kozlov, A., Nikolov, A., Trame, J.: metaphactory: a platform for knowledge graph management. *Semant. Web* **10**(6), 1109–1125 (2019)
9. DBpedia. <https://wiki.dbpedia.org/about>
10. Introducing the Knowledge Graph: things, not strings, 16 May 2012. <http://googleblog.blogspot.com/2012/05/introducing-knowledge-graph-things-not.html>
11. Suchanek, F.M., Kasneci, G., Weikum, G.: Yago: a core of semantic knowledge. In: WWW 2007: Proceedings of the 16th International Conference on World Wide Web, pp. 697–706, May 2007. <https://doi.org/10.1145/1242572.1242667>
12. Erxleben, F., Günther, M., Krötzsch, M., Mendez, J., Vrandečić, D.: Introducing wikidata to the linked data web. In: Mika, P., et al. (eds.) ISWC 2014. LNCS, vol. 8796, pp. 50–65. Springer, Cham (2014). [https://doi.org/10.1007/978-3-319-11964-9\\_4](https://doi.org/10.1007/978-3-319-11964-9_4)
13. Hubauer, T., et al.: Use cases of the industrial knowledge graph at siemens. In: International Semantic Web Conference (P&D/Industry/BlueSky) (2018)
14. RDF primer. <https://www.w3.org/TR/rdf-primer/>
15. Farber, M., Ell, B., Menne, C., Rettinger, A., Bartscherer, F.: Linked data quality of DBpedia, Freebase, OpenCyc, Wikidata, and YAGO. *Semantic Web J.* (2016). <http://www.scsemantic-web-journal.net/contentv/linked-data-quality-dbpeda-freebase-opencyc-wikidata-and-yago>. Accessed August 2016

16. Qiao, X., Li, X., Chen, J.: Telecommunications service domain ontology: semantic interoperability foundation of intelligent integrated services. In: Ortiz, J.H. (ed.) *Telecommunications Networks - Current Status Future Trends*, 30th March 2012. IntechOpen (2012). <https://doi.org/10.5772/36794>
17. Qiao, X., Li, X., You, T., Sun, L.: Semantic telecommunications network capability services. In: Domingue, J., Anutariya, C. (eds.) *ASWC 2008*. LNCS, vol. 5367, pp. 508–523. Springer, Heidelberg (2008). [https://doi.org/10.1007/978-3-540-89704-0\\_35](https://doi.org/10.1007/978-3-540-89704-0_35)
18. Han, S., Zou, L., Yu, J.X., Zhao, D.: Keyword search on RDF graphs - a query graph assembly approach. In: *CIKM 2017: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pp. 227–236, November 2017. <https://doi.org/10.1145/3132847.3132957>
19. Zou, L., Özsu, M.T., Chen, L., Shen, X., Huang, R., Zhao, D.: gStore: a graph-based SPARQL query engine. *VLDB J.* **23**(4), 565–590 (2014). <https://doi.org/10.1007/s00778-013-0337-7>
20. RDF. <https://www.w3.org/RDF/>
21. RDFS. <https://www.w3.org/TR/rdf-schema/>
22. OWL. <https://www.w3.org/OWL/>
23. GitHub repository link. <https://github.com/kulikovia/ICSSA-2020>
24. McCusker, J.: What is a knowledge graph? <http://www.semantic-web-journal.net/content/what-knowledge-graph>
25. Stankova, E.N., Balakshiy, A.V., Petrov, D.A., Korkhov, V.V.: OLAP technology and machine learning as the tools for validation of the numerical models of convective clouds. *Int. J. Bus. Intell. Data Min.* **14**(1/2), 254–266 (2019). <https://doi.org/10.1504/IJBIDM.2019.096793>. ISSN online 1743-8195, ISSN print 1743-8187
26. Stankova, E.N., Khvatkov, E.V.: Using boosted k-nearest neighbour algorithm for numerical forecasting of dangerous convective phenomena. In: Misra, S., et al. (eds.) *ICCSA 2019*. LNCS, vol. 11622, pp. 802–811. Springer, Cham (2019). [https://doi.org/10.1007/978-3-030-24305-0\\_61](https://doi.org/10.1007/978-3-030-24305-0_61)
27. Stankova, E.N., Ismailova, E.T., Grechko, I.A.: Algorithm for processing the results of cloud convection simulation using the methods of machine learning. In: Gervasi, O., et al. (eds.) *ICCSA 2018*. LNCS, vol. 10963, pp. 149–159. Springer, Cham (2018). [https://doi.org/10.1007/978-3-319-95171-3\\_13](https://doi.org/10.1007/978-3-319-95171-3_13)