Architecture of a Telecommunications Network Monitoring System Based on a Knowledge Graph

Kirill Krinkin, Igor Kulikov, Alexander Vodyaho
Saint-Petersburg Electrotechnical University "LETI"
Saint-Petersburg, Russia
kirill@krinkin.com, i.a.kulikov@gmail.com,
aivodyaho@mail.ru

Nataly Zhukova
St. Petersburg Institute for Informatics and Automation of the Russian Academy of Sciences
St. Petersburg, Russia
nazhukova@mail.ru

***Abstract*—The article focuses on developing a new architecture of telecommunications network monitoring systems based on knowledge graphs that allow comfortably integrate static network models and the dynamic data obtained by monitoring within a single architecture. The article analyses the tasks accomplished by traditional monitoring systems used today, and determines the challenges that are not met by the systems. It offers a new architecture for a system based on a knowledge graph, which allows solve the newly defined challenges by integrating various network models and the dynamic data obtained by monitoring. The structure of the system is defined, and the ontological models it is based on are described. The article shows the capabilities of the system and defines the limits for its applicability. It describes the implementation of the system, including the requirements for its integration with external systems. The options for production implementation of the system components are listed. In order to practically evaluate the architecture proposed, a fragment of a monitoring system is drawn up and the results of its application are presented. Conclusions are formulated, and the areas of further research are identified.**

I. INTRODUCTION

A. *Tasks accomplished by monitoring systems*

The common tasks successfully accomplished by the monitoring systems existing today are summed up in the following list [1]:

- Monitoring of network devices and data transmission channels;
- Monitoring of network performance;
- Monitoring of key performance indicators of a network;
- Monitoring of application operation;
- Generation of reports and event notifications.

In order to analyze today's requirements to telecommunication systems that users impose on monitoring systems the authors carried out data analysis for one of the active telecommunications networks of a major North American cable television operator. Table I shows the results of the requirements analysis in the context of the parties concerned and the options for accomplishing the tasks set by means of traditional monitoring systems (the table includes only the tasks that either cannot be accomplished by traditional systems, or tackling them is not the key feature of such systems).

TABLE I. ANALYSIS OF NEEDS OF MONITORING SYSTEM USERS

Interested user	**Monitoring task**	**Solubility of tasks through traditional monitoring systems**
User (customer layer)	Monitoring relevant data on constraints for customers	The monitoring model must be supported with the data on access rights distribution.
	Receiving personal recommendations (offers of services and data for purchasing)	Impossible task. Substantial data model extension is required.
Network owners (business layer)	Monitoring of information about customer interests (for the purpose of personal ads and producing personal recommendations)	Impossible task. Substantial data model extension is required.
	Determining target customer groups for advertising purposes	Impossible task. Substantial data model extension is required.
Network operations (operations layer)	Quick identification of causes of incidents users may face	Finding the cause of a problem necessitates obtaining comprehensive data from several network models.

As we see from the above list of common and new tasks, today's monitoring systems provide relevant information on a variety of network performance facets, but without any interrelation (or the interrelation is not obvious and needs additional means to be shown), and without any reference to other telecommunications network models, such as:

- Billing model (personal accounts, user devices, tariffs, geoinformation, payments);
- Access rights model (access to services, applications and data);
- Models of services, applications and data accessible to the user;
- Behaviour statistics (queried services, results of request processing).

Extending the monitoring system base model with the data listed above allows not only to integrate within a monitoring system the analysis of technical features of the network performance but also to link them with business data, which in its turn increases the number of interested monitoring system

users by including business units of network operators and facilitates incident investigation for operation services through obtaining the monitoring data associated with the user. Besides that it will be possible to integrate statistical data related to the network with dynamic data of monitoring within one system. Extending the base model will reduce the costs of solving the following types of tasks:

- Grouping users by various cross-sections considering the data both of traditional monitoring and related systems (billing, geographical location, access rights distribution, statistics on the use of services, applications and data);
- Searching for information associated with the model components (from the perspective of users, services, applications, or data);
- Analysis of user interest trends;
- Facilitating identification of the key causes of incidents;
- Dynamic management of telecommunications network parameters based on monitoring metrics, including metrics of user interests and user action statistics.

The list may be extended as a result of analyzing the demand of telecommunications network operators for monitoring data.

B. Parametres of telecommunications network monitoring systems

In comparative study of telecommunications network monitoring systems existing today the authors of the reviews assess the systems by the following key parametres [1], [33]:

1. Generating reports on basic indicators of network quality according to SLA;
2. Identifying the tendencies of change in basic indicators of network quality;
3. Forecasting the trends of basic indicators of network quality;
4. Analysis of network topology;
5. Maintenance of SNMP protocol;
6. Using the agent model of monitoring;
7. Event logging;
8. Maintaining a variety of modes of delivering messages to users.

While developing the new architecture of telecommunications network monitoring system it is to be expected that the new system must be competitive with regard to the parametres listed and provide ample opportunities of extending the list of soluble tasks through joint use of a multitude of statistical models and dynamic data.

C. Architecture of standard monitoring systems

A standard monitoring system contains the following components [2]:

- Main server, including server software core, DBMS, agent interaction subsystem, user notification subsystem, graphic user interface, reporting subsystem, event logging subsystem.
- Agents, including agent software core, server interaction subsystem, configuraing subsystem, monitoring subsystem (monitoring of physical parametres, operating system status, network host status, application status).

The models of these traditional monitoring systems are built based on indicators of network quality according to SLA. In the existing systems data is usually stored in SQL database. The new architecture proposes integration of telecommunications network static models with dynamic data within a shared data model, which allows to extend the range of soluble monitoring tasks.

II. MONITORING SYSTEM DATA MODEL

The authors propose the knowledge graph method [3], [4] as a base for developing an integrated model of telecommunications system monitoring. This approach allows reach the following advantages:

- integrate all required models, thus providing semantic connections among all the elements;
- low cost of a new monitoring events adding (defined by knowledge graph architecture);
- ability of integrating with 3rd party systems (open ontology model);
- ability to find new classes of tasks that can be solved by analyzing semantic connected monitoring data.

This approach can be used for different multiservice telecommunication networks which provide services, applications and data access for different types of end-user devices.

A knowledge graph of a telecommunications network monitoring system contains both a static component that determines the network structure in its various perspectives and dynamic monitoring data. The structure of static data for a telecommunications network knowledge graph is presented in “Fig. 1. Model of knowledge graph static data”.

The knowledge graph is developed based on static and dynamic data given below.

1. Static data.
- Billing model (fragment):
  - User – a network customer;
  - Account – user account identifier;
- Access rights model (lists of access rights in the perspective of network users):
  - Entitlements –node element of access rights control for a group of users;
  - User entitlements list – list of resources available to the user;
- Network topology model (fragment):
  - User –a network customer;
  - Device –a client device;
- Network applications hierarchy model (fragment):
  - Applications – node element of application hierarchy;

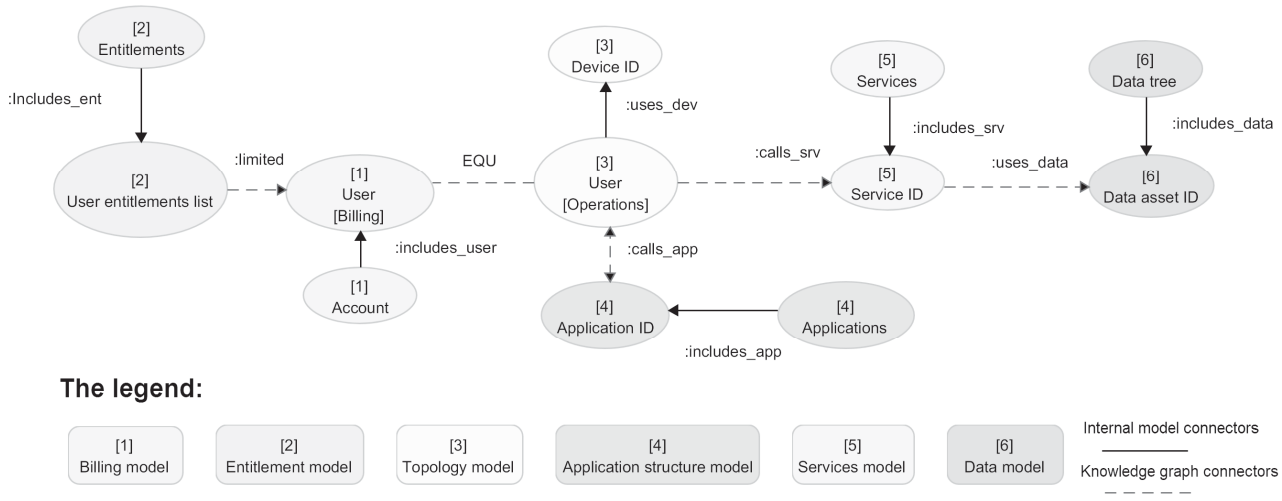

Fig. 1. Model of knowledge graph static data

- Application ID – network application;
- Network service hierarchy model (fragment):
  - Services – node element of service hierarchy;
  - Service ID – network service;
- Data model (fragment):
  - Data tree – node element of data hierarchy;
  - Data asset ID – data target accessible for users, services and applications.

2. Dynamic data:

- Data from traditional monitoring systems;
- Data from operations logs;
- Data on user actions.

Dynamic data must contain the event time mark, so to describe the dynamic data we propose the structure of Statement about statement type presented in “Fig. 2. Model of knowledge graph dynamic data”.

The described structure of dynamic data related to a monitoring event contains the following information:

- Calling node – the component of knowledge graph that causes the event recorded in monitoring;
- Request ID – monitoring event identifier;
- Monitoring event time mark;
- :is_requested_with – type of association between Request ID node and Statement data; the monitoring event predicat (more than one type is possible depending on the ontological model);
- Object node – static model node; monitoring event object;
- Subject node – static model node; monitoring event subject.

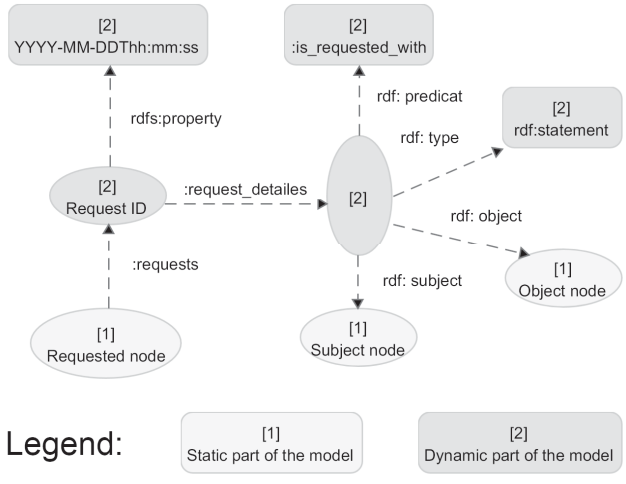

Fig. 2. Model of knowledge graph dynamic data

The dynamic monitoring data is integrated into the generalized knowledge graph model in the following way:

- The Request ID object is connected by a respective predicat to a subject, the knowledge graph static node that caused the monitoring event (User, Service, Application, etc.)
- The Statement dynamic data model node is connected to the Subject and Object, the static model nodes, depending on the nature of the monitoring event (Services, Applications, Data assets, etc.)

The presented model of dynamic data is universal and agrees with the general concept of knowledge graph. Both static and dynamic knowledge graph models use a single ontological model, which allows to make queries to data as to a shared graph data base (using SPARQL).

III. Monitoring SYSTEM ARCHITECTURE

A. Structure diagram

The scheme of the proposed monitoring system based on knowledge graph is presented in “Fig. 3. Structural scheme of a monitoring system based on knowledge graph”.

The proposed system consists of the following components:

A. The monitoring system core. The core includes:

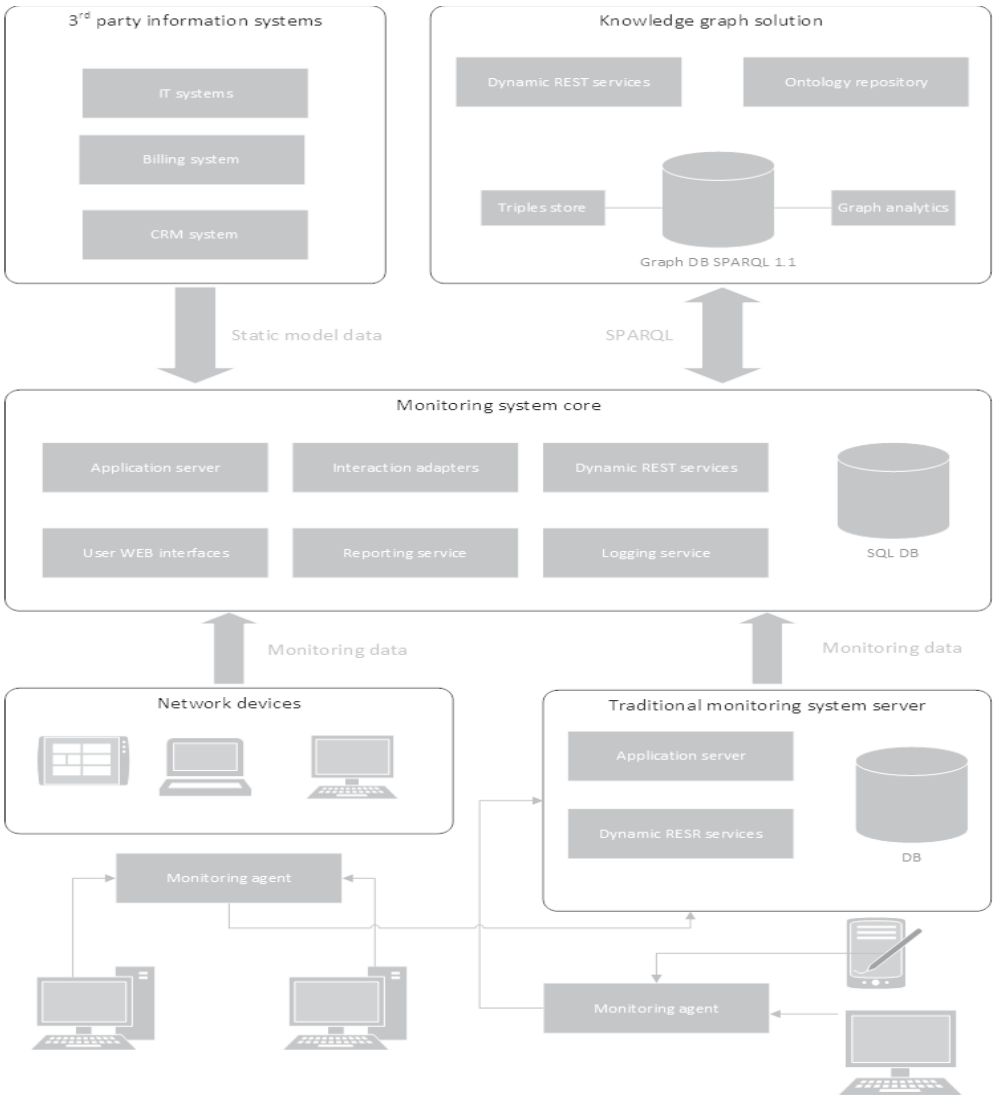

Fig. 3. Structural scheme of a monitoring system based on knowledge graph

- 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.)
- WEB interface for the system users and administrators.
- Reporting service with the possibility of representing reports in WEB interface or sending 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.

B. 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 data analytics module. It stores both static and dynamic graph data connected by 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 [34], RDFS [35], OWL [36].
- A dynamic REST service supporting API for interaction with external systems, in particular, with the monitoring system core.

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

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

D. Traditional monitoring systems (can be used as sources for aggregated monitoring data). From the perspective of interoperability traditional monitoring systems include:

- Application server accommodating the business-logic of the system.
- Dynamic REST service supporting API for interaction with external systems, in particular, with the core of the monitoring system.
- Own SQL storage for network monitoring data.

E. Own system agents supplying aggregated monitoring data to the monitoring system core.

B. Data processing levels

In the proposed architecture data is processed at a variety of logical levels (Table 1).

TABLE II. ANALYSIS OF NEEDS OF MONITORING SYSTEM USERS

Logical level of data	Data scope	Data storage	Data processing methods
Primary monitoring data	Device and application logs Device parametres Network events data User activity history (Data Collection primary data)	File system DBMS for primary monitoring systems	Log parsing Queries to database
Level of representing data in monitoring systems	System performance indicators Event logs	Monitoring system data storages	API of monitoring systems Monitoring systems database search
Level of representing data within the knowledge graph	Static graph network models Dynamic data obtained from monitoring systems and own monitoring agents User activity logs	Knowledge graph	Semantic search queries to knowledge graph (integration of static and dynamic data)

C. Telecommunication networks ontologies

In order to achieve the tasks of monitoring an individual model is developed for every network based on a knowledge graph proceeding from the types of tasks to be performed. As a base ontology to build a telecommunications network knowledge graph it would be appropriate to use the Telecommunications Service Domain Ontology (TSDO) [5] developed industry process ontological model with the ontology described in OWL (Web Ontology Language) [36]. The model involves several levels of ontological model construction aimed at addressing practical necessities. The specific features of operator telecommunications networks are to be considered in the ontology of levels of applications extending the definitions of TSDO. The hierarchical model for developing the ontology for a telecommunications network monitoring system knowledge graph is presented in “Fig. 4. Ontological model of telecommunications network monitoring system knowledge graph”.

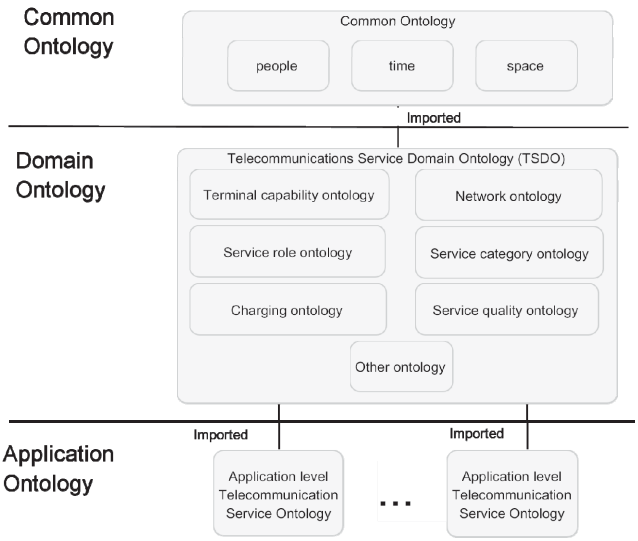

Fig. 4. Ontological model of telecommunications network monitoring system knowledge graph

The application level ontology is introduced with the aim to describe the static model of a knowledge graph. The dynamic data is described within the same ontology (with no further development required).

D. Operating scenarios

A generalized system operating scenario is presented in “Fig. 5. Operating scenario of a monitoring system based on knowledge graph”.

Description of system operating scenario:

- External IT systems, either on schedule or on some event, transfer the updates of the network static model to the monitoring system core; the data is considered with the updating of the static model. This is the main scenario for updating the knowledge graph static model.

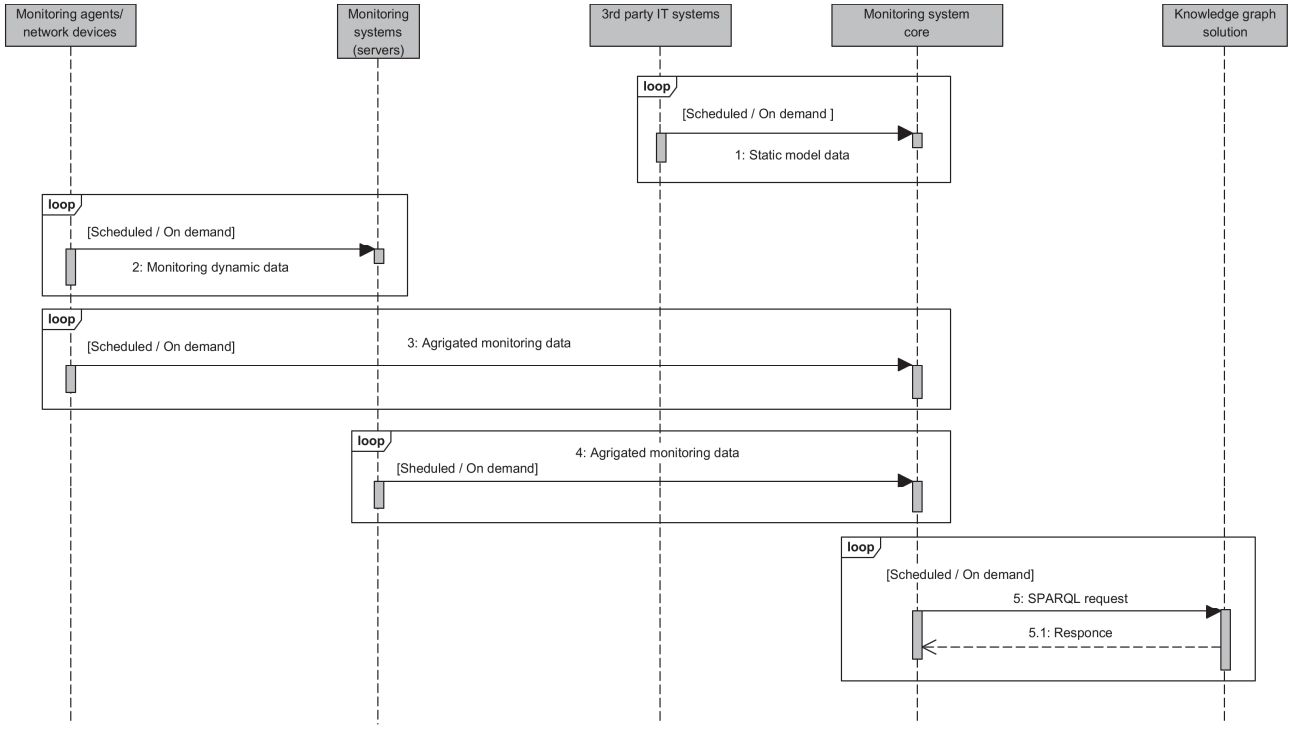

Fig. 5. Operating scenario of a monitoring system based on knowledge graph

- Traditional monitoring system agents and network devices, either on schedule or on some event, transfer to traditional monitoring systems the primary data, which after processing is placed in their local data storages.
- Agents of monitoring system core, either on schedule or on some event, transfer the aggregated monitoring data to the monitoring system core; after processing the data is transferred to the knowledge graph.
- Traditional monitoring systems, either on schedule or on some event, transfer the aggregated monitoring data to the monitoring system core; after processing the data is transferred to the knowledge graph.
- The monitoring system core, either on schedule or on some event, transfers to the knowledge graph the aggregated monitoring data (the dynamic component of knowledge graph). Dynamic data present sets of triples which are built on the operator network ontological model and do not require a separate ontological model. Also the system core generates queries to knowledge graph in order to receive the necessary network operation parametres (SPARQL queries). The SPARQL queries operate with full set of knowledge graph data (both static and dynamic, integrated into a single ontological model).

E. Industrial solutions for the system components

The feasible industrial solutions for the system components are presented in Table III [6], [27].

TABLE III. FEASIBLE INDUSTRIAL SOLUTIONS FOR THE SYSTEM COMPONENTS

System component	Variant solutions used
Knowledge graph	Virtuoso 8.3 [8], GraphDB [9], Stardog [10], Oracle 19c [11], Apache Jena-Fuseki [12], Metafactory (Blazegraph) [13], CumulusRDF [14], Strabon [15], 4store [16], h2rdf+ [17]
Monitoring systems	Datadog [18], LogicMonitor [19], SolarWinds Network Performance Monitor [20], Microsoft System Center [21], NinjaRMM [22], SolarWinds NetFlow Traffic Analyzer [23], Wireshark [24], AteraPRT [25], GNagios XI [26]
SQL DBMS	Oracle 19c [28], MySQL [29], Microsoft SQL Server [30], PostgreSQL [31], IBM DB2 [32]

F. General requirements to system performance

Developing a monitoring system based on a knowledge graph necessitates considering performance parametres of the existing solutions. For a monitoring system, the major types of query to a knowledge graph are as follows:

- Establishing a triple;
- Search in the knowledge graph.

According to the outcomes of investigating the productivity of systems based on knowledge graph [6], the average speed of executing a search query to a knowledge graph with the graph size of 1M triplets searched for the best solution for RDF storage (Virtuoso 7.2.4) is no more than 1 second, and can increase up to 4 seconds. With simultaneous inputting new data in the knowledge graph, which is an acceptable figure for report generation in a monitoring system. In this case using a combination solution, when static data is kept in RDF storage, with dynamic data in SQL DBMS, and SPARQL queries transforming into SQL for the search of

dynamic data, does not provide any advantage. This said, while developing a monitoring system based on knowledge graph the main criteria of its productivity will be as follows:

- Maximum size of a knowledge graph;
- Maximum allowable time for query processing;
- Frequence of recording new triplets containing monitoring data.

IV. Example solution

Using the Metaphactory environment [37], a fragment of static knowledge graph for a cable television operator network monitoring system was built, with the following parametres:

- Number of users: 1 000
- Number of personal accounts: 1 000
- Number of user devices: 1 000
- Number of network services: 4
- Number of data assets: 4

A. Example scenario

Task to be accomplished:
To produce lists of users who viewed a specific show on any channel on a certain date. In addition, the focus is on the users who have a specified tariff plan and a certain model of a user device.
Scenario description:

- Users watch TV shows; in this context, if a user has viewed a show for over 50% of its time, the system considers it as viewed to the end;
- The events related to watching TV shows are communicated to the monitoring system server and recorded in the knowledge graph;
- The data on tariff plan, model of the user device used, list of accessible network services and program schedule are communicated to the monitoring system from the respective operator IT systems.

B. Knowledge graph model

To accomplish the task, the following knowledge graph

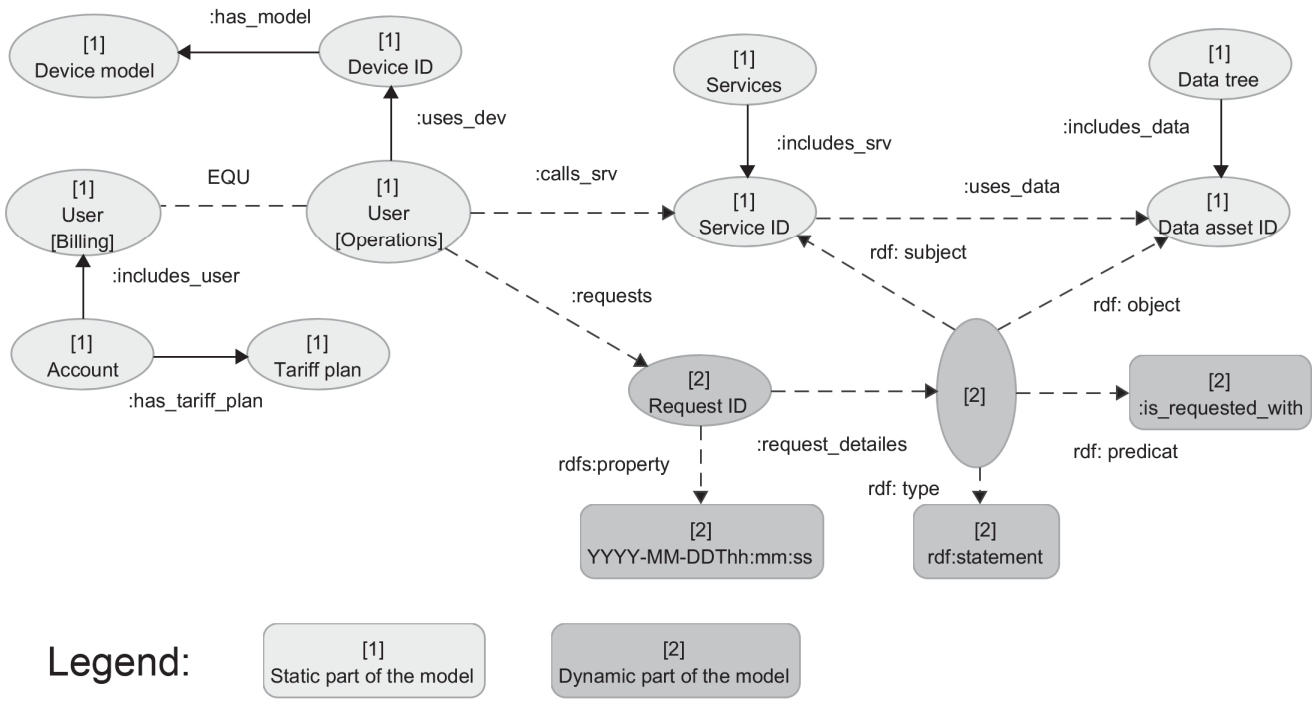

Fig. 6. Data model for a knowledge graph (example of system implementation)

- Number of device models: 3
- Number of tariff plans: 4
- Number of monitoring events in 24 hours: 50 000

The knowledge graph was developed in RDF/XML format and imported in Metaphactory environment using standard data import tools.

data model was developed ("Fig. 6. Data model for a knowledge graph (example of system implementation)").

C. SPARQL request / response – solution for use-case

For query, the following parameter values were selected:

- Identifier of hub to which devices are connected: H1;
- Tariff: Promo;

- Device model: Moto2k;
- Event date: 2020-02-01;
- Service used: WatchTV;
- ID of the programme viewed: Asset1.

Query and response are shown below:

```
SPARQL REQUEST:
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX my: <http://127.0.0.1/bg/ont/test1#>
SELECT *
WHERE
{
 ?Accounts my:includes_user ?User_account .
 ?Accounts my:has_tariff_plan "Promo" .
 ?User my:has_id ?User_account .
 ?Device my:is_connected_to_hub "H1" .
 ?Device my:has_the_device_model "Moto2k" .
 ?Device my:has_id ?Device_id .
 ?User my:uses_device ?Device_id .
 ?User my:requests ?Request_ID .
 ?Request_ID rdf:property ?Date
 FILTER contains(?Date, "2020-02-01") .
 ?Request_ID my:request_detailes ?Detailes_ID .
 ?Detailes_ID rdf:subject "WatchTV" .
 ?Detailes_ID rdf:object ?Asset_ID .
 ?Asset_ID my:has_id "Asset1"
}
```

```
RESPONCE:
    {
     "Asset_ID" : {
       "type" : "uri",
       "value" : "http://127.0.0.1/Asset_1/"
     },
     "Detailes_ID" : {
       "type" : "bnode",
       "value" : "t272118"
     },
     "Request_ID" : {
       "type" : "uri",
       "value" : "http://127.0.0.1/Request_10834/"
     },
     "User" : {
       "type" : "uri",
       "value" : "http://127.0.0.1/User_431/"
     },
     "Device_id" : {
       "type" : "literal",
       "value" : "D431"
     },
     "User_account" : {
       "type" : "literal",
       "value" : "U431"
     },
     "Accounts" : {
       "type" : "uri",
       "value" : "http://127.0.0.1/Account_431/"
     },
     "Device" : {
       "type" : "uri",
       "value" : "http://127.0.0.1/Device_431/"
     },
     "Date" : {
       "datatype" : "http://www.w3.org/2001/XMLSchema#datetime",
       "type" : "literal",
       "value" : "2020-02-01 22:40:12"
     }
    }
```

Application for generation of RDF/XML model of knowledge graph, the RDF/XML model itself and the SPARQL queries are available in an open repository at GitHub [38].

V. Conclusion

In the paper the new approach of operating with monitoring data as semantic connected data is represented. The knowledge graph allows integrate a variety of telecommunications network static models into a single semantic model and to seamlessly add to it the dynamic monitoring data aggregated by traditional monitoring systems or respective agents. This joint model provides the opportunity for accomplishing new types of tasks insoluble in traditional monitoring systems. Also knowledge graph ability for integration with 3rd-party systems. In developing projects it is advisable to make maximum use of already developed domain ontological models to ensure further integration. In the course of further research dynamic simulation of the system should be carried out depending on the size of knowledge graph, and optimal parametres for the graph size and incoming data flow should be determined. The proposed architecture of telecommunications network monitoring system based on knowledge graph has the capacity to accomplish the tasks set when it is necessary to integrate a number of network models. The system load can be managed by determining the parametres of incoming flow of information placed in the knowledge graph. The already used common ontological models allow easy integration with other systems based on semantic data model. The example considered in the article discloses the advantages of analyzing dynamic monitoring data in an integrated network model within the framework of a shared knowledge graph. Future researches will be focused on methods of knowledge graph structure creation for both static and dynamic data and performance analysis.

Acknowledgment

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

References

[1] Stanford University, “Network Monitoring Tools” Stanford University. [Online]. Available: http://www.slac.stanford.edu/xorg/nmtf/nmtf-tools.html. [Accessed: Oct. 20, 2012]

[2] A. Natarov, A. Shirokii, “Next Generation Network Monitoring Systems — Critical Requirements and Design”. DOI: https://doi.org/10.15688/mpcm.jvolsu.2018.3.4

[3] https://www.w3.org/TR/rdf-primer/

[4] M. Farber, B. Ell, C. Menne, A. Rettinger, and F. Bartscherer. Linked Data Quality of DBpedia, Freebase, OpenCyc, Wikidata, and YAGO. Semantic Web Journal, 2016. http://www.scmantic-web-journal.net/contenv/linked-data-quality-dbpedia-freebase- opencyc-wikidata-and-yago [August, 2016] (revised version, under review)

[5] Xiuquan Qiao, Xiaofeng Li and Junliang Chen (March 30th 2012). Telecommunications Service Domain Ontology: Semantic Interoperation Foundation of Intelligent Integrated Services, Telecommunications Networks - Current Status and Future Trends, Jesus Hamilton Ortiz, IntechOpen, DOI: 10.5772/36794

[6] Pierfrancesco Bellini, Paolo Nesi, “Performance assessment of RDF graph databases for smart city services”. DOI: https://doi.org/10.1016/j.jvlc.2018.03.002

[7] Shuo Han, Lei Zou, Jeffery Xu Yu, Dongyan Zhao, “Keyword Search on RDF Graphs — A Query Graph Assembly Approach”. DOI: https://doi.org/10.1145/3132847.3132957
[8] Virtuoso: https://virtuoso.openlinksw.com/
[9] GraphDB: http://graphdb.ontotext.com/
[10] Stardog: https://www.stardog.com/platform/
[11] Oracle RDF Graph Features: https://www.oracle.com/database/technologies/spatialandgraph/rdf-graph-features.html
[12] Apache Jena-Fuseki: https://jena.apache.org/documentation/fuseki2/
[13] Metaphactory: https://metaphacts.com/product
[14] CumulusRDF: https://code.google.com/archive/p/cumulusrdf/
[15] Strabon: http://www.strabon.di.uoa.gr/
[16] 4store: http://4store.org/
[17] h2rdf+: https://github.com/zcourts/h2rdf/tree/master/H2RDF%2B
[18] Datadog: https://www.datadoghq.com/product/
[19] LogicMonitor: https://www.logicmonitor.com/
[20] SolarWinds Network Performance Monitor: https://www.solarwinds.com/network-performance-monitor
[21] Microsoft System Center: https://www.microsoft.com/ru-ru/cloud-platform/system-center
[22] NinjaRMM: https://www.ninjarmm.com/
[23] SolarWinds NetFlow Traffic Analyzer: https://www.solarwinds.com/netflow-traffic-analyzer
[24] Wireshark: https://www.wireshark.org/
[25] AteraPRT: https://www.atera.com/
[26] GNagios XI: https://www.nagios.com/
[27] DB-engines ranking: https://db-engines.com/en/ranking
[28] Oracle DB: https://www.oracle.com/ru/database/technologies/
[29] MySQL: https://www.mysql.com/
[30] Microsoft SQL Server: https://www.microsoft.com/ru-ru/sql-server/sql-server-2019
[31] PostgreSQL: https://www.postgresql.org/
[32] IBM DB2: https://www.ibm.com/support/knowledgecenter/SSEPGG_11.5.0/com.ibm.db2.luw.welcome.doc/doc/welcome.html
[33] Comparison of network monitoring systems: https://en.wikipedia.org/wiki/Comparison_of_network_monitoring_systems
[34] RDF: https://www.w3.org/RDF/
[35] RDFS: https://www.w3.org/TR/rdf-schema/
[36] OWL: https://www.w3.org/OWL/
[37] Metaphactory: https://metaphacts.com/
[38] Solution example GitHub repository: https://github.com/kulikovia/FRUCT-2020