BULGARIAN ACADEMY OF SCIENCES

PROBLEMS OF ENGINEERING CYBERNETICS AND ROBOTICS • 2020 • Vol. **72**, pp. 69-78
p-ISSN: 2738-7356; e-ISSN: 2738-7364
https://doi.org/10.7546/PECR.72.20.07

Methods of Composing Hierarchical Knowledge Graphs of Telecommunication Networks

Radoslav Yoshinov[1], *Igor Kulikov*[2], *Nataly Zhukova*[3]

[1] Bulgarian Academy of Sciences (BAS), Sofia, Republic of Bulgaria
[2] Saint Petersburg Electrotechnical University "LETI", St. Petersburg, Russia
[3] St. Petersburg Institute for Informatics and Automation of the Russian Academy of Sciences, St. Petersburg, Russia
Emails: yoshinov@cc.bas.bg, i.a.kulikov@gmail.com, nazhukova@mail.ru

Abstract: The article is devoted to the issue of developing and analyzing methods of composing hierarchical knowledge graphs of telecommunication networks with a view to representing network data. The article substantiates the timeliness of discussing this issue and describes the classes of problems that can be solved using hierarchical graph models. It discusses the structure of particular hierarchical graph models, their layers, and how it is possible to connect these models layer by layer and create a single hierarchical graph model. It also discusses data formats and approaches to adding dynamic data that are applicable to the resulting hierarchical knowledge graph. It studies the advantages of the hierarchical knowledge graph over one-level analogs. To explain the advantages of using a hierarchical knowledge graph, an example of modeling such a knowledge graph for a cable TV operator's network is given and compared with a one-level knowledge graph using a specific problem encountered by the operator. It is shown that it is much less difficult to execute queries of the same kind using a hierarchical graph. At the end of the article, conclusions are made and directions for future research are discussed.

Keywords: knowledge graph, monitoring system, ontology, semantic web telecommunication network.

1. Introduction

1.1. Graph models of telecommunication networks

Graph models are widely used for describing different applications [1] including telecommunication networks. In particular, graph models are used to describe the structure of networks and their functionality. Below are presented the ones that are most frequently used. The list was compiled based on the authors' personal experience in designing telecommunication networks and on the conclusions made from studying academic literature on the topic.

1. Graph models describing the network structure for various network layers:
    - network devices and data transmission channels;
    - network applications and the links between them;
    - network hierarchy;
    - functions and the links between them;
    - network restrictions;
    - graph data model [2].
2. Graph security models (attack graphs). They are used to simulate an intruder's sequence of actions resulting in network vulnerabilities. [3]
3. Access entitlements models (access rights graphs). They are used for subject-object modeling of access rights allocation. [4]
4. User interface models (for example, voluminous catalogs).
5. Network device status models.
6. Particular graph models. They are designed based on data available and used to solve specific problems.

All the graph models mentioned above are homogeneous in terms of both graph nodes and edges, which makes these models applicable in solving problems of the same kind: message routing, network hierarchy optimization, data structure optimization, and security or access rights issues. Each of these models can be described using a one-level or hierarchical graph.

1.2. Problem definition

In order to solve a large number of problems, it is necessary to combine a lot of graph models into a single model that will make it possible to solve such classes of problems which cannot be solved (or are difficult to solve) using graph models that are traditionally kept separate. These problems include:

- Problems in analyzing data from telecommunication network monitoring systems;
- Search problems;
- Forming personal recommendations;
- Problems in the analytical processing of semantically linked data from traditional IT systems.

A generalized knowledge graph can be plotted in two different ways:

- by making direct links between data at the bottom level of the hierarchy using RDF technology [5], [6], which produces a one-level model of the knowledge graph;
- by adding links between models at all possible levels of the hierarchy of the original graph models.

Both approaches make it possible to produce a working model. However, SPARQL queries are processed in different ways in one-level and hierarchical models, and these models differ in terms of their size. The search for the optimal approach to designing the knowledge graph of a telecommunication network is an important issue since the choice of methodology directly affects the size of the model and the performance of IT systems which are designed based on this model.

2. Methods of composing hierarchical graph models of telecommunication networks

2.1. Source data and models

Let us consider an abstract network in order to analyze telecommunication network models. This abstract network provides its customers with a number of services and applications. Clients use both stationary and mobile devices and pay for the services provided by the operator. In addition to providing access to services and applications, the operator also provides access to data stored inside the telecommunication network. A list of models and data relevant in this case is given in Table 1.

Table 1. Models and data of a telecommunication network

Model	Purpose	Model layers (top to bottom)	Layer description
Network topology	To describe the structure of a network	Datacenter layer	The top level of the hierarchy
		Regional network layer	Model breakdown into regions
		Local network layer	Model breakdown within each region
		Device types layer	Data breakdown by device types (stationary, mobile, etc.)
		Device model layer	Data breakdown by device models
		Device layer	The bottom level – separate devices
Billing model	To describe the system of client accounts, tariffs, households, and payments	Regional layer	Client account breakdown by regions
		Account type layer	Types of client accounts (natural / legal person, educational organization, etc.)
		Household layer	Data breakdown by households
		Tariff layer	Data breakdown by tariffs
		User account layer	The bottom level – data breakdown by users
Access rights model	To describe lists of services, applications, and data that are available to different layers of the billing model	Regional layer	Access rights breakdown by regions
		Tariff layer	Access rights breakdown by tariffs
		User account layer	Access rights breakdown by users
		Household layer	Access rights breakdown by households
		Entitlement lists layer	The bottom level – entitlement lists
Service provision model	To describe the structure of the services provided by the network	Service platform layer	Data breakdown by service platforms (for example, services for mobile or stationary devices)
		Service category layer	Data breakdown by service categories (data transmission, voice services, video services, etc.)

Model	Purpose	Model layers (top to bottom)	Layer description
		Service groups layer	Service groups within categories
		End-user services layer	The bottom level – separate end-user services
User application hierarchy model	To describe the structure of user applications provided by the network	Application platform layer	Data breakdown by platforms (for example, services for mobile or stationary devices)
		Application category layer	Data breakdown by application types (games, business applications, information applications, etc.)
		Application groups layer	Application groups within categories
		End-user applications layer	The bottom level – separate end-user applications
A model of the data provided by the network	To describe the structure of the data provided by the network	Data type layer	Data breakdown by data types (for example, video, audio, books, pictures, academic papers, etc.)
		Data category layer	Data breakdown by data categories (for example, film genres, such as comedy, action, melodrama, etc.)
		Data groups layer	Data groups within categories
		Data assets layer	The bottom level – separate data assets

Graph models can be decomposed into layers in different ways depending on the simulation objectives. Table 1 shows only one of the many possible decomposition options.

2.2. Linking graph models layer by layer

The nodes in the above graph models can be linked both at the level of instance objects (in a one-level model) and at higher levels (in a hierarchical model). Fig. 1 shows an example of combining two graphs into a one-level model, whereas Fig. 2 shows how a hierarchical model can be used. The billing model and the data model were taken as initial models, and access rights granted to Tariff 1 users to access Category 2 data and those granted to Tariff 2 users to access Category 1 data were taken as linking criteria.

It can be seen from the example given above that when initial models are linked layer by layer, it causes a significant decrease in the number of edges in the resulting graph, which affects the size of the resulting model and the performance of IT systems which are designed based on this model.

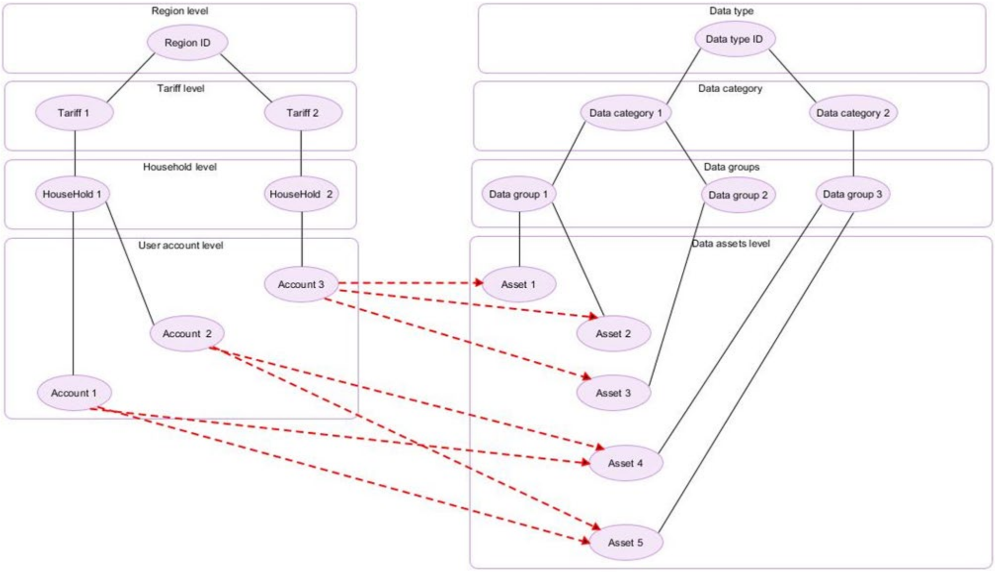

Fig. 1. Linking graph models at the bottom level (a one-level model)

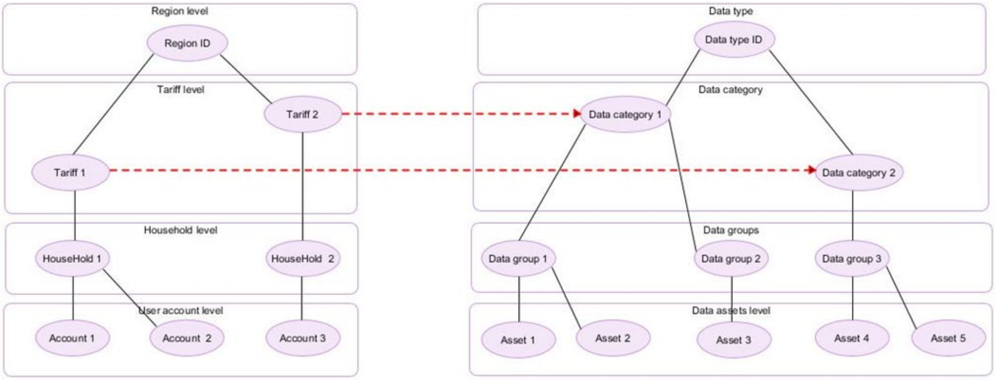

Fig. 2. Graph models being linked layer by layer (a hierarchical model)

2.3. A description of the resulting hierarchical model in the form of a hierarchical knowledge graph

We chose an RDF (Resource Description Framework) graph comprising "subject – predicate – object" RDF triples, a technology used to represent knowledge graphs, as a tool for representing the generalized hierarchical model. In this configuration, a multitude of RDF statements form a directed graph with subjects and objects as apices and links between them as edges [5], [6].

The initial hierarchical models are transformed into hierarchical RDF graphs, with nodes of different layers of the graphs that are being linked acting as subjects and objects of the combined RDF graph and links of different types acting as predicates. This approach is illustrated in Fig. 3.

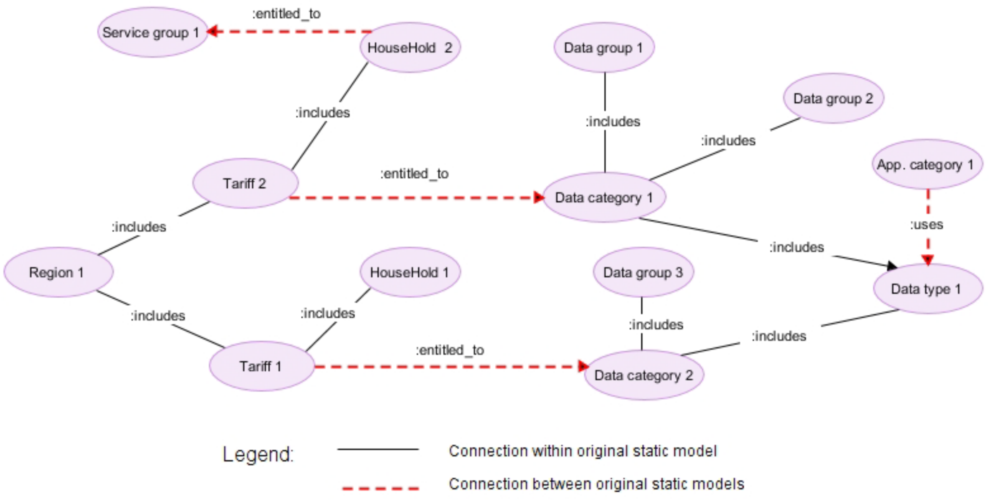

Fig. 3. The structure of the hierarchical knowledge graph

2.4. A performance analysis of SPARQL queries to the knowledge graph

We suggest that a bipartite graph should be used to give a formal description of the knowledge graph [7]. The set of nodes and edges that form the knowledge graph is represented as follows:

- The whole set objects in the network forms a set of n nodes V_i;
- The whole set of links between the objects forms a set of m nodes E_j.

The knowledge graph is transformed into a bipartite graph $B(V_L, V_R, E_B)$, for which the following conditions are met:

1. For any pair of nodes $V_{i_1} \times V_{i_2} = \{(v_{i_1}, v_{i_2}) | 1 \le i_1 < i_2 \le n \wedge v_{i_1} \in V_{i_1} \wedge v_{i_2} \in V_{i_2}\}$.
2. $V_L = \cup_{1 \le i_1 < i_2 \le n}(V_{i_1} \times V_{i_2})$.
3. $V_R = \cup_{1 \le j \le m} E_j$.
4. The ***e*** set of links between the nodes $(v_{i_1}, v_{i_2}) \in V_L$ and the nodes $p_j \in V_R (1 \le i_1 < i_2 \le n, 1 \le j \le m)$ is defined as $e(\langle v_{i_1}, v_{i_2} \rangle, p_j)$. The node weight is $w(e) = w(\langle v_{i_1}, v_{i_2} \rangle, p_j)$, where $w(\langle v_{i_1}, v_{i_2} \rangle, p_j)$ is the cost of forming a triple.

A SPARQL query can be represented as a query graph $Q(V^Q, E^Q, FL)$, where:

- V^Q – is a set of graph nodes from V_L associated with the subjects and objects of the SPARQL query;
- E^Q – is a set of links associated with the properties of the SPARQL query;
- FL – are filter limits specified in the SPARQL query.

Execution of a SPARQL query over the knowledge graph can be considered as a problem of finding all the subgraphs corresponding to the query $Q(V^Q, E^Q, FL)$ in the original graph [8]. The maximum possible number of object pairs in the graph over which the subgraphs are searched is determined by the expression $\binom{n}{2}^{m}$. As it is shown in [9], the performance of SPARQL queries to the knowledge graph decreases more than tenfold with an increase in the number of objects.

3. Example solution

3.1. Use-case

Initial data: A telecommunication network provides services, applications, and access to content. The devices used are both stationary and mobile. There are certain rules concerning granting access rights to data, services, and applications at different levels of the hierarchy of network models.

Task: Build both one-level and hierarchical knowledge graphs based on the initial data and compare query execution time taken to compile a list of users who have access to a selected application. In the one-level model, customer accounts are linked with applications. In the hierarchical model, application categories are linked with tariff groups.

3.2. Knowledge graph models

The resulting one-level model of the knowledge graph is shown in Fig. 4 and the resulting hierarchical model of the knowledge graph is shown in Fig. 5.

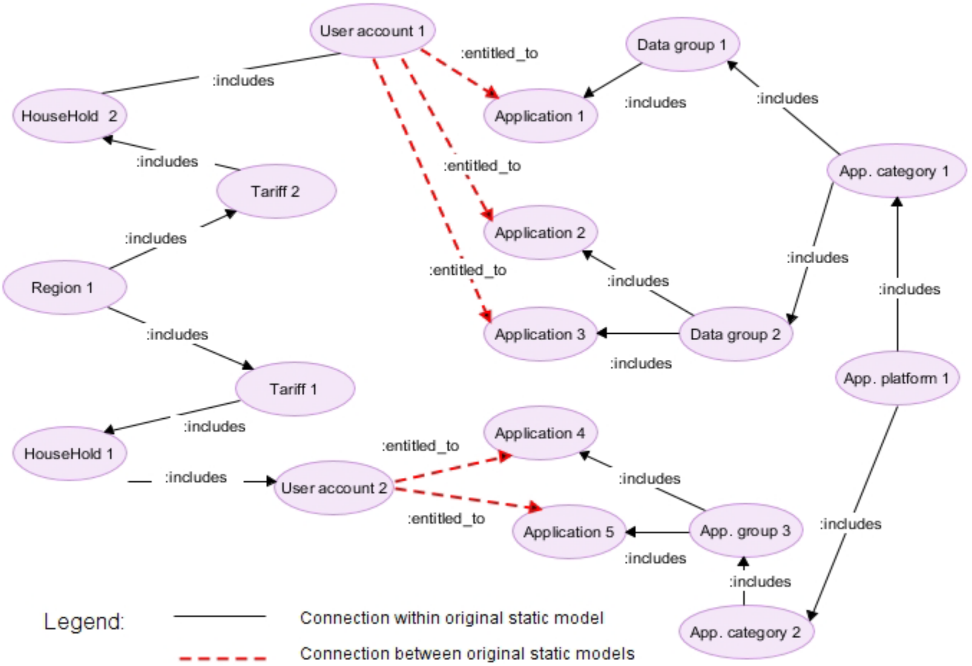

Fig. 4. A one-level model of the knowledge graph of a telecommunication network

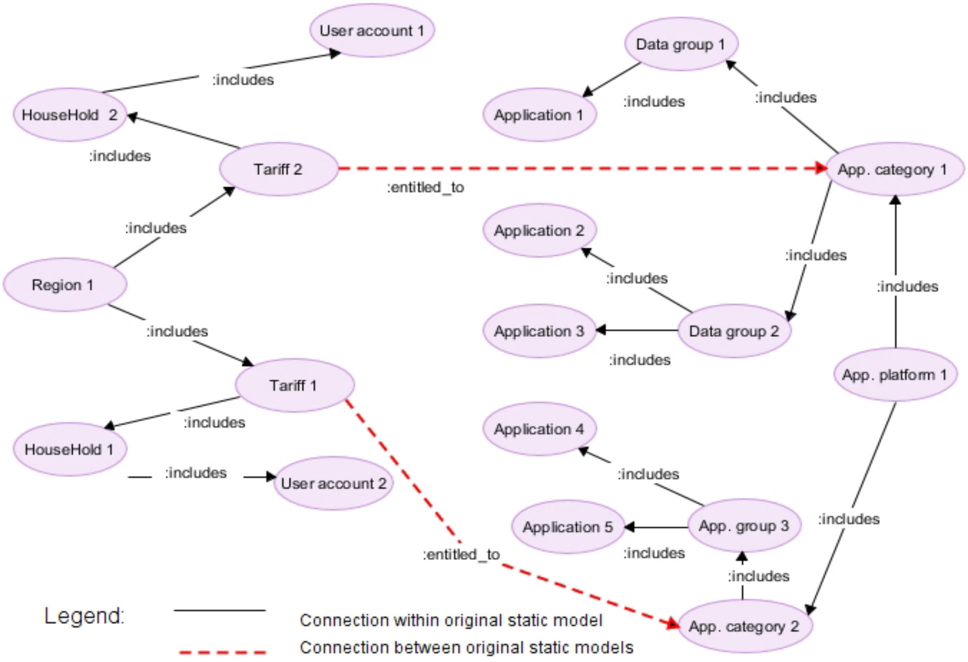

Fig. 5. A hierarchical model of the knowledge graph of a telecommunication network

3.3. SPARQL Requests

SPARQL REQUEST #1 – for the one-layer model:

```
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

{

 ?Account_ID my:has_id "A1" .

 ?Account_ID my:entitled_to ?Application_ID .

}
```

SPARQL REQUEST #1 – for the one-layer model:

```
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
        {
          ?Account_ID my:has_id "A1" .
          ?Account_ID my:entitled_to ?Application_ID .
        }
```

3.4. Dynamic system modeling

To compare the performance of the one-level and hierarchical models, the authors carried out tests aimed at measuring SPARQL query execution time depending on the size of the knowledge graph model. The Metaphactory platform was used for simulation [10]. The parameters of the models used in the tests and simulation results are presented in Table 2.

Table 2. The results of dynamic system modeling

	The number of triples in the knowledge graph			
	100k	**1M**	**10M**	**25M**
RDF/XML data loading time, one-layer model, sec.	12,7	94,3	1097,8	3671,2
Query execution time, one-level model, ms	111	202	282	298
Number of triples, one-layer model	1,7M	14M	135M	339M
RDF/XML data loading time, hierarchical model, sec.	8,2	55,2	625,1	1471,2
Query execution time, hierarchical model, ms	118	122	129	129
Number of triples, hierarchical model	720k	4,3M	40M	100M

Based on the results of the dynamic tests, it can be concluded that in comparison with the one-level model, the hierarchical model significantly increases the performance of solutions that are designed based on such a model. The application for generating an RDF/XML model of the knowledge graph, the RDF/XML model itself, and the SPARQL queries are available in an open repository on GitHub [11].

4. Conclusion

The article proposes a hierarchical graph model of telecommunication networks. Within this model, separate graph models that already exist are combined. The unified model makes it possible to solve new classes of problems that cannot be solved using traditional systems. The article describes two approaches to designing such a unified model, namely creating links between network objects at the bottom level of the models being linked and making links between objects layer by layer.

Hierarchical models can significantly increase the performance of systems which use these models as their foundation. The article describes a mathematical model of the knowledge graph as a bipartite graph, which allows for using the already existing mathematical apparatus for data processing. The problem of processing a SPARQL query is reduced to transforming it into a corresponding graph and solving the problem of finding subgraphs in the original knowledge graph. An example of designing one-level and hierarchical models of a telecommunication network is given. The example analyzed in the article reflects the advantages of the hierarchical model in terms of reducing the size of the graph and increasing performance. The results of the tests that were carried out show the performance advantage hierarchical knowledge graphs have over one-level graphs. In the future, it is advisable to analyze different options in the dynamic plotting of hierarchical knowledge graphs.

Acknowledgment

This work is supported by National Science Program “Information and Communication technologies for unified Digital Market in Science, Education and Security”.

References

1. Moustakerov, I.: An improved network optimization model of transport processes control. Problems of Engineering Cybernetics and Robotics, 50, 47–53 (2000).
2. Spatial and Graph Topology Data Model and Network Data Model Graph Developer's Guide, https://docs.oracle.com/database/121/TOPOL/network-data-model-graph-overview.htm#TOPOL700, last accessed 2020/3/12.
3. Saha, S., Halappanavar M., Vullikanti A.: Identifying vulnerabilities and hardening attack graphs for networked systems, Virgina Tech, (2014) http://staff.vbi.vt.edu/ssaha/papers/attackgraph_dag.pdf.
4. Lipton, R. J., Snyder, L.: A linear time algorithm for deciding subject security. Journal of the ACM, 24 (3), 455–464 (1977).
5. RDF Primer, https://www.w3.org/TR/rdf-primer/, 2020/3/5.
6. Farber, M., Ell, B., Menne, C., Rettinger, A., Bartscherer, F.: Linked data quality of DBpedia, Freebase, OpenCyc, Wikidata, and YAGO. Semantic Web Journal 1, 1–45, (2016).
7. Han, S., Zou, L., Yu, J. X., Zhao, D.: Keyword search on RDF graphs – A query graph assembly approach, In Proc. of ACM on Conference on information and Knowledge Management, 227–236, (2017).
8. 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).
9. Bizer, C., Schultz, A.: The Berlin SPARQL benchmark. International journal on Semantic Web and information systems, 5(2), 1–24, (2009).
10. Haase, P., Herzig, D. M., Kozlov, A., Nikolov, A., Trame, J.: Metaphactory: A platform for knowledge graph management. Semantic Web, 10(6), 1109–1125 (2019).
11. GitHub repository, https://github.com/kulikovia/BAN-2020.