Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ

КАФЕДРА СИСТЕМЫ ОБРАБОТКИ ИНФОРМАЦИИ И УПРАВЛЕНИЯ

РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

К НАУЧНО-ИССЛЕДОВАТЕЛЬСКОЙ РАБОТЕ

НА ТЕМУ:

Публикация статьи “The Development of the Metagraph Data and Knowledge Model” на Википедии

Студент ИУ5-33М ________________ **Д.Ю. Уткин**
(Группа) (Подпись, дата) (И.О.Фамилия)

Руководитель ________________ **Ю.Е. Гапанюк**
(Подпись, дата) (И.О.Фамилия)

2023 г.

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

УТВЕРЖДАЮ
Заведующий кафедрой ___ИУ5____
(Индекс)
______________ ___В.И. Терехов__
(И.О.Фамилия)
«_04_» ____сентября_______ 2023 г.

З А Д А Н И Е
на выполнение научно-исследовательской работы

по теме Публикация статьи “The Development of the Metagraph Data and Knowledge Model” на Википедии

Студент группы _ИУ5-33М__

Уткин Дмитрий Юрьевич
(Фамилия, имя, отчество)

Направленность НИР (учебная, исследовательская, практическая, производственная, др.)
____________________учебная____________________
Источник тематики (кафедра, предприятие, НИР) _____КАФЕДРА______________________________

График выполнения НИР: 25% к ___ нед., 50% к ___ нед., 75% к ___ нед., 100% к ___ нед.

Техническое задание в ходе выполнения НИР необходимо провести редактирование статьи на Википедии “Портовые графы”. Необходимо улучшить структуру и содержание статьи.
Оформление научно-исследовательской работы:

Расчетно-пояснительная записка на _____ листах формата А4.
Перечень графического (иллюстративного) материала (чертежи, плакаты, слайды и т.п.)

Дата выдачи задания «_04_» ____сентября_______ 2023 г.

Руководитель НИР ________________ ______Ю.Е. Гапанюк______
(Подпись, дата) (И.О.Фамилия)

Студент ________________ ______Д.Ю. Уткин_____
(Подпись, дата) (И.О.Фамилия)

Примечание: Задание оформляется в двух экземплярах: один выдается студенту, второй хранится на кафедре.

Оглавление

Введение

В современном мире, где телекоммуникационные системы играют ключевую роль в обеспечении связности и эффективного обмена информацией, особенно важным становится адекватное моделирование и анализ их структуры.

Настоящее исследование направлено на публикацию английской версии статьи на Википедии, посвященную метаграфу. Путем детального рассмотрения мы стремимся обогатить информацию, предоставляемую Википедией, и создать более полную и доступную базу знаний о телекоммуникационных системах.

1. Исходный код

```
'''Metagraphs''' in mathematics are a generalization of the concepts of graph structures. There are elements of both graphs and hypergraphs in the metagraph, the process metagraph itself is built on the basis of a hierarchical graph. Formally, the metagraph is defined by the set <V,MV, E, ME>, where V is the set of vertices, MV is the set of metapersons, E is the set of edges, ME is the set of metaperms.

== The Original Metagraph Model of A. Basu and R. Blanning ==
In the monograph by A. Bazu and R. Blanning<ref>Basu A., Robert W. Blanning. Metagraphs and their applications. — New York: Springer, 2007. — C. 14.</ref> , the following definitions are given that describe the metagraph model.

The'''''generating set''''' of a metagraph is the set of elements: <math>X = \{x_1,x_2,...,x_n\}</math> , which represent variables of interest, and which occur in the edges of the metagraph.

An '''''edge''''' <math>e = \langle V_e,W_e\rangle \in E</math> (where <math>E</math> is the set of edges) consisting of an '''''invertex'''' ' <math>V_e\subset X</math> and an '''''outvertex'''''  <math>W_e\subset X</math> . each of which may contain any number of elements. The different elements in the invertex are coinputs of each other '''''outvertex''''' ('''''cooutputs''''').

A '''''metagraph''''' <math>S = \langle X,E\rangle</math> is then a graphical construct specified by its generating set <math>X</math> and a set of edges <math>E</math> defined on the generating set.

A '''''simple path''''' <math>h(x, y)</math> from an element x to an element y is a sequence of edges <math>\langle e_1,e_2,...,e_n\rangle</math> such that:
*: <math>x \in invertex(e_1)</math> ;
*: <math>y \in outvertex(e_n)</math> ;

*: for all <math>e_i,i=1,...,n-1</math>, <math>outvertex(e_i) \cap invertex(e_{i+1}) \ne \varnothing</math>.
[[File:Метфграф.png|305x305px|thumb|right|Figure 1: An example of a description of a metagraph]]

The monograph <ref name="автоссылка1">Basu A., Robert W. Blanning. Metagraphs and their applications. — New York: Springer, 2007.</ref> gives an example of a metagraph shown in Figure 1, for which the following set-theoretic interpretation is given:
*: <math>S = <X, E></math>, where
```

*: $X = \{Exp, Notes, Prof, Rev, Pri, Vol, Wage\}$, and

*: <math>E = <\{Pri,Vol\},\{Rev\}>, <\{Vol,Wage\},\{Exp\}>, <\{Rev,Exp\},\{Prof , Notes\}>, <\{Exp\}, \{Notes\}></math>.

The emergence in the model of A. Bazu and R. Blanning is achieved through the use of edges. The concept of a metavertex is not present in this model.

It can be noted that this version of the metagraph model is more suitable for describing directed processes than for describing complex graph data structures.

== The Metagraph Model with Metavertices ==
The lack of a natural mechanism for describing complex graph data structures led to the appearance of extensions of the original model by A. Basu and R. Blanning. New elements appeared in the model – metavertices and metaedges.

In<ref name="автоссылка2">Globa L.S. Metagraph based representation and processing of fuzzy knowledgebases / L.S. Globa, M.Y. Ternovoy, O.S. Shtogrina</ref>, the concept of a metavertex appears, and the following definitions of a metagraph model are given.

'''''The metagraph''''' is a triple of sets of vertices, metaverses, and edges, respectively: $S = \langle V,W,E \rangle$, where $V = \{\nu_r\}$ – the set of vertices of the metagraph (generating set); $M = \{m_q\}$ — the set of the metagraph metavertices; $E = \{e_h\}$ — the set of edges of the metagraph.

'''''The metagraph metavertex''''' $m_q = \{\nu_r \vert \nu_r \in V, r=1,...,N_{mq}\}$ is defined as a set of vertices included in a metavertex m_q , where N_{mq} is a cardinality of the set.

The following remark by the authors of the model<ref>Globa L.S. Metagraph based representation and processing of fuzzy knowledgebases / L.S. Globa, M.Y. Ternovoy, O.S. Shtogrina -pp. 238.</ref>should be considered very interesting: “... if two or more metavertices correspond to the same set of vertices, then such vertices are considered identical, and only one of these metavertices is considered.” We call this property of the model<ref name="автоссылка2" /> an '''''anti-annotability property''''', the features of which will be considered later.

To define the edges, the authors of the model<ref name="автоссылка2" /> introduce the concept of a metagraph node $m\nu \in(V \cup M)$ belonging to the combined set of vertices and metavertices. An edge is defined as $e_h = \langle m\nu_{out}, m\nu_{in} \rangle$, that is, characterized by outgoing and incoming nodes of a metagraph. But the use of the concept of a metagraph node for creating hierarchical metavertices is not proposed in this model.

== The Hierarchical Metagraph Model with Metavertices and Metaedges ==
In <ref name="автоссылка3">Astanin S.V., Zhukovskaya N.K. Business processes control via modeling by fuzzy situational networks. Autom. Remote Control 75, (2014): 570–579.</ref>, not only the concept of a metavertex appears but also the concept of a metaedge and the idea of the metavertices hierarchy.

'''''The metagraph''''' in the model<ref name="автоссылка3" /> is defined as $S = \langle X,X_M,E,E_M \rangle$, where X is the set of vertices of the metagraph (generating set); X_M is the set of metavertices; E is the set of edges; E_M is the set of metaedges defined on the union of metavertices and vertices $X_M \cup X$.

Thus, the metaedge in this model means an edge that can connect a vertex and a metavertex or two metavertices.

An essential feature of this model is that the authors introduce the concept of a '''''nested metagraph''''', which, according to the authors, is a “generalization of flat graphs, hypergraphs, and metagraphs”<ref name="автоссылка3" />.

In this model, the set of vertices X is considered hierarchical, and the index i is entered, which determines the nesting level of the vertex.

The property of anti-annotability is neither confirmed nor refuted by the authors of the model. At the same time, examples implicitly use a property of anti-annotability.

Thus, the central issue<ref name="автоссылка3" /> of this article is the description of hierarchies in the metagraph model.

== The Annotating Metagraph Model ==
The annotating metagraph model proposed by us extends the ideas of the original model by A. Bazu and R. Blanning<ref name="автоссылка1" /> and the ideas of the work<ref name="автоссылка3" />. The contents of the work<ref name="автоссылка2" /> at the time of the appearance of the first version of the annotating metagraph model<ref name="автоссылка4">Yu.E. Gapanyuk. Metagraph Approach to the Information-Analytical Systems Development. In: Proceedings of the 6th International Conference Actual Problems of System and Software Engineering, Moscow, Russia, 2019, pp. 428-439.</ref> were unknown to us.

The'''''metagraph''''' model may be described as follows: $S = \langle V, MV, E, ME\rangle$, where MG – metagraph; V – set of metagraph vertices; MV – set of metagraph metavertices; E – set of metagraph edges; ME – set of metagraph metaedges.

'''''Metagraph vertex''''' is described by '''''a set of attributes''''': $\nu_i = \{atr_k\}, \nu_i \in V$, where$\nu_i$ – metagraph vertex; atr_k – attribute.

'''''Metagraph edge''''' is described by a set of attributes, the source, and destination vertices and edge direction flag: $e_i = \langle \nu_S, \nu_E, eo, \{atr_k\}\rangle, e_i \in E, eo = true \vert false,$ where e_i – metagraph edge; ν_S – source vertex (metavertex) of the edge; ν_E – destination vertex (metavertex) of the edge; eo – edge direction flag ($eo=true$ – directed edge, $eo=false$ – undirected edge); atr_k – attribute.

The '''''metagraph fragment''''': $MG_i=\{e\nu_j\}, e\nu_j \in (V \cup E \cup MV \cup ME)$, where MG_i – metagraph fragment; $e\nu_j$ – an element that belongs to the union of vertices, edges, metavertices, and metaedges.

The'''''metagraph metavertex''''': $m\nu_i=\langle \{atr_k\},MG_j\rangle, m\nu_i \in MV$, where $m\nu_i$ – metagraph metavertex belongs to a set of metagraph metavertices MV; atr_k – attribute, MG_j – metagraph fragment.

Thus, metavertex, in addition to the attributes, includes a fragment of the metagraph. The presence of private attributes and connections for metavertex is a distinguishing feature of the metagraph. It makes the definition of metagraph holonic – metavertex may include a number of lower-level elements and, in turn, may be included in a number of higher-level elements.

[[File:Метаграф в аннотируемой метаграфовой модели.png|thumb|Figure 2: The example of data metagraph]]

The example of data metagraph (shown in Figure 2) contains three metavertices: '''mv1, mv2''' and '''mv3'''. Metavertex '''mv1''' contains vertices '''v1, v2, v3''' and connecting them edges '''e1, e2, e3'''. Metavertex '''mv2'''contains vertices '''v4, v5''' and connecting them edge '''e6'''. Edges '''e4, e5''' are examples of edges connecting vertices '''v2-v4''' and '''v3-v5'''are contained in different metavertices '''mv1''' and '''mv2'''. Edge '''e7''' is an example of the edge connecting metavertices '''mv1''' and '''mv2'''. Edge '''e8''' is an example of the edge connecting vertex '''v2''' and metavertex '''mv2'''. Metavertex '''mv3''' contains metavertex '''mv2''', vertices '''v2, v3''' and edge '''e2''' from metavertex '''mv1''' and also edges '''e4, e5, e8,''', showing the holonic nature of the metagraph structure.

Note that, unlike <ref name="автоссылка2" />, the anti-annotability property is not fulfilled in this model. The same set of vertices and edges can be included in several different metavertices, which can represent different situations and can be annotated with different attributes. This is why we called our model an '''annotating metagraph model'''.

Also, unlike the previously considered models, in the model we propose, a metavertex can include both vertices and edges.
The '''''metagraph metaedge''''':
$me_i = \langle \nu_S, \nu_E, eo, \{atr_k\},MG_j\rangle, e_i \in E, eo = true \vert false, e\nu_j \in (V \cup E \cup MV \cup ME)$ where me_i – metagraph metaedge belongs to set of metagraph metaedges ME; ν_S – source vertex (metavertex) of the metaedge; ν_E – destination vertex (metavertex) of the metaedge; eo – metaedge direction flag ($eo=true$ – directed metaedge, $eo=false$ – undirected metaedge); atr_k – attribute; MG_j – metagraph fragment.
[[File:Метаребра_метаграфа_в_аннотируемой_метаграфовой_модели.png|thumb|386x386пкс|Figure 3: The example of the directed metaedge]]
The example of a directed metaedge is shown in Figure 3. The directed metaedge contains metavertices $m\nu_S,\dots,m\nu_i,\dots,m\nu_E$ and connecting them edges. The source metavertex contains a nested metagraph fragment. During the transition to the destination metavertex, the nested metagraph fragment became more complex, new vertices, edges, and inner metavertices are added. Thus, metaedge allows binding the stages of nested metagraph fragment development to the steps of the process described with metaedge.
Thus, the hierarchical metaedge from the model<ref name="автоссылка3" /> corresponds to the usual edge in our model. And a metaedge in our model means a sequence of changes of metagraph vertices.

== Using Agents to Process Annotated Metagraph Model ==
The metagraph model is aimed for complex data descriptions. However, it is not aimed for data transformation. To solve this issue, the metagraph agent aimed for data transformation is proposed. There are two kinds of metagraph agents: '''''the metagraph function agent''''' and '''''the metagraph rule agent'''''.

The metagraph function agent serves as a function with input and output parameters in the form of metagraph: $ag^F=\langle MG_{IN}, MG_{OUT}, AST \rangle,$ where ag^F – metagraph function agent; MG_{IN} – input parameter metagraph; MG_{OUT} – output parameter metagraph; AST – abstract syntax tree of metagraph function agent in the form of metagraph.

The metagraph rule agent is rule-based: $ag^M=\langle MG_D, R, AG^{ST} \rangle, R=\{r_j\}$ where ag^M – metagraph rule agent; MG_D – working metagraph, a metagraph on the basis of which the rules of the agent are performed; R – set of rules r_j; AG^{ST} – start condition (metagraph fragment for start rule check or start rule); MG_j - a metagraph fragment on the basis of which the rule is performed; OP^{MG} - set of actions performed on metagraph.

$r_i\colon MG_j \to OP^{MG}$ where r_i – the rule; MG_j – a metagraph fragment on the basis of which the rule is performed; OP^{MG} – set of actions performed on metagraph.

The antecedent of the rule is a condition over the metagraph fragment, the consequent of the rule is a set of actions performed on the metagraph. Rules can be divided into open and closed.
Note that the rules of a metagraphic agent can be divided into '''''closed and open'''''.

The '''''open rules''''' do not change the metagraph fragment on the right side of the rule that belongs to the left side of the rule. You can separate the input and output fragments of the metagraph. These rules are analogous to a template that generates an output metagraph based on an input one.

'''''Closed rules''''' change the metagraph fragment on the right side of the rule that belongs to the left side of the rule. Changing the metagraph on the right side of the rules causes the left parts of the other rules to be triggered. But at the same time, incorrectly developed closed rules can lead to cycling of the metagraphic agent. Thus, a metagraph agent allows you to generate one metagraph based on another (using open rules) or modify a metagraph (using closed rules).

== References ==
*: Chapela V., Regino Criado, Santiago Moral, Miguel Romance. Intentional risk management through complex networks analysis. – Springer, 2015: SpringerBriefs in optimization.

*: B.S. Manoj, A. Chakraborty, R. Singh. Complex Networks: A Networking and Signal Processing Perspective. Pearson, New York, 2018.

*: A.T. Akhmediyarova, J.R. Kuandykova, B.S. Kubekov, I.T. Utepbergenov, V.K. Popkov. Objective of Modeling and Computation of City Electric Transportation Networks Properties. In: International Conference on Information Science and Management Engineering (Icisme 2015), Destech Publications, Phuket, 2015, pp. 106–111.

*: Johnson J. Hypernetworks in the science of complex systems. – London, Hackensack NJ: Imperial College Press, 2013.

*: Basu A., Robert W. Blanning. Metagraphs and their applications. – New York: Springer, 2007.

*: Globa L.S. Metagraph based representation and processing of fuzzy knowledgebases / L.S. Globa, M.Y. Ternovoy, O.S. Shtogrina // Proceedings of 9th Open Semantic Technologies for Intelligent Systems Conference (OSTIS-2015), pp. 237-240. BGUIR Publishing, Minsk 2015.

*: Astanin S.V., Zhukovskaya N.K. Business processes control via modeling by fuzzy situational networks. Autom. Remote Control 75, (2014): 570–579.

*: V.M. Chernenkiy, Yu.E. Gapanyuk, A.N. Nardid, A.V. Gushcha, Yu.S. Fedorenko. The Hybrid Multidimensional-Ontological Data Model Based on Metagraph Approach. In: A.K. Petrenko, A. Voronkov (eds.) Perspectives of systems informatics. 11th International Andrei P. Ershov Informatics Conference, PSI 2017, Moscow, Russia, June 27-29, 2017, Revised Selected Papers / edited by Alexander K. Petrenko, Andrei Voronkov, vol. 10742. Lecture notes in computer science, 0302-9743, vol. 10742, pp. 72–87. Springer (2018). doi: 10.1007/978-3-319-74313-4_6

*: Yu.E. Gapanyuk. Metagraph Approach to the Information-Analytical Systems Development. In: Proceedings of the 6th International Conference Actual Problems of System and Software Engineering, Moscow, Russia, 2019, pp. 428-439.

== Notes ==
{{notes}}
[[Category:Mathematics]]

2. Вид статьи на Википедии

Metagraphs in mathematics are a generalization of the concepts of graph structures. There are elements of both graphs and hypergraphs in the metagraph, the process metagraph itself is built on the basis of a hierarchical graph. Formally, the metagraph is defined by the set <V,MV, E, ME>, where V is the set of vertices, MV is the set of metapersons, E is the set of edges, ME is the set of metaperms.

The Original Metagraph Model of A. Basu and R. Blanning [edit source]

In the monograph by A. Bazu and R. Blanning[1] , the following definitions are given that describe the metagraph model.

The***generating set*** of a metagraph is the set of elements: $X = \{x_1, x_2, \ldots, x_n\}$, which represent variables of interest, and which occur in the edges of the metagraph.

An ***edge*** $e = \langle V_e, W_e \rangle \in E$ (where E is the set of edges) consisting of an ***invertex'*** ' $V_e \subset X$ *and an* **outvertex** $W_e \subset X$ *. each of which may contain any number of elements. The different elements in the invertex are coinputs of each other* **outvertex** *(***cooutputs***).*

A ***metagraph*** $S = \langle X, E \rangle$ is then a graphical construct specified by its generating set X and a set of edges E defined on the generating set.

A ***simple path*** $h(x, y)$ from an element x to an element y is a sequence of edges $\langle e_1, e_2, \ldots, e_n \rangle$ such that:

- $x \in invertex(e_1)$;
- $y \in outvertex(e_n)$;
- for all $e_i, i = 1, \ldots, n-1$, $outvertex(e_i) \cap invertex(e_{i+1}) \neq \varnothing$.

The monograph [2] gives an example of a metagraph shown in Figure 1, for which the following set-theoretic interpretation is given:

- $S =< X, E >$, where
- $X = \{Exp, Notes, Prof, Rev, Pri, Vol, Wage\}$, and

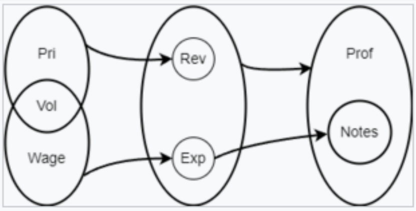

Figure 1: An example of a description of a metagraph

- $E =< \{Pri, Vol\}, \{Rev\} >, < \{Vol, Wage\}, \{Exp\} >, < \{Rev, Exp\}, \{Prof, Notes\} >, < \{Exp\}, \{Notes\} >$.

The emergence in the model of A. Bazu and R. Blanning is achieved through the use of edges. The concept of a metavertex is not present in this model.

It can be noted that this version of the metagraph model is more suitable for describing directed processes than for describing complex graph data structures.

The Metagraph Model with Metavertices [edit source]

The lack of a natural mechanism for describing complex graph data structures led to the appearance of extensions of the original model by A. Basu and R. Blanning. New elements appeared in the model – metavertices and metaedges.

In[3], the concept of a metavertex appears, and the following definitions of a metagraph model are given.

The metagraph is a triple of sets of vertices, metaverses, and edges, respectively: $S = \langle V, W, E \rangle$, where $V = \{\nu_r\}$ – the set of vertices of the metagraph (generating set); $M = \{m_q\}$ — the set of the metagraph metavertices; $E = \{e_h\}$ — the set of edges of the metagraph.

The metagraph metavertex $m_q = \{\nu_r | \nu_r \in V, r = 1, \ldots, N_{mq}\}$ is defined as a set of vertices included in a metavertex m_q , where N_{mq} is a cardinality of the set.

The following remark by the authors of the model[4]should be considered very interesting: "... if two or more metavertices correspond to the same set of vertices, then such vertices are considered identical, and only one of these metavertices is considered." We call this property of the model[3] an ***anti-annotability property***, the features of which will be considered later.

To define the edges, the authors of the model[3] introduce the concept of a metagraph node $m\nu \in (V \cup M)$ belonging to the combined set of vertices and metavertices. An edge is defined as $e_h = \langle m\nu_{out}, m\nu_{in} \rangle$, that is, characterized by outgoing and incoming nodes of a metagraph. But the use of the concept of a metagraph node for creating hierarchical metavertices is not proposed in this model.

The Hierarchical Metagraph Model with Metavertices and Metaedges [edit source]

In [5], not only the concept of a metavertex appears but also the concept of a metaedge and the idea of the metavertices hierarchy.

The metagraph in the model[5] is defined as $S = \langle X, X_M, E, E_M \rangle$, where X is the set of vertices of the metagraph (generating set); X_M is the set of metavertices; E is the set of edges; E_M is the set of metaedges defined on the union of metavertices and vertices $X_M \cup X$.

Thus, the metaedge in this model means an edge that can connect a vertex and a metavertex or two metavertices.

An essential feature of this model is that the authors introduce the concept of a ***nested metagraph***, which, according to the authors, is a "generalization of flat graphs, hypergraphs, and metagraphs"[5].

In this model, the set of vertices X is considered hierarchical, and the index i is entered, which determines the nesting level of the vertex.

The property of anti-annotability is neither confirmed nor refuted by the authors of the model. At the same time, examples implicitly use a property of anti-annotability.

Thus, the central issue[5] of this article is the description of hierarchies in the metagraph model.

The Annotating Metagraph Model [edit source]

The annotating metagraph model proposed by us extends the ideas of the original model by A. Bazu and R. Blanning[2] and the ideas of the work[5]. The contents of the work[3] at the time of the appearance of the first version of the annotating metagraph model[6] were unknown to us.

The ***metagraph*** model may be described as follows: $S = \langle V, MV, E, ME \rangle$, where MG — metagraph; V — set of metagraph vertices; MV — set of metagraph metavertices; E — set of metagraph edges; ME — set of metagraph metaedges.

Metagraph vertex is described by ***a set of attributes***: $\nu_i = \{atr_k\}, \nu_i \in V$, where ν_i — metagraph vertex; atr_k — attribute.

Metagraph edge is described by a set of attributes, the source, and destination vertices and edge direction flag: $e_i = \langle \nu_S, \nu_E, eo, \{atr_k\} \rangle, e_i \in E, eo = true|false$, where e_i — metagraph edge; ν_S — source vertex (metavertex) of the edge; ν_E — destination vertex (metavertex) of the edge; eo — edge direction flag ($eo = true$ — directed edge, $eo = false$ — undirected edge); atr_k — attribute.

The ***metagraph fragment***: $MG_i = \{e\nu_j\}, e\nu_j \in (V \cup E \cup MV \cup ME)$, where MG_i — metagraph fragment; $e\nu_j$ — an element that belongs to the union of vertices, edges, metavertices, and metaedges.

The ***metagraph metavertex***: $m\nu_i = \langle \{atr_k\}, MG_j \rangle, m\nu_i \in MV$, where $m\nu_i$ — metagraph metavertex belongs to a set of metagraph metavertices MV; atr_k — attribute, MG_j — metagraph fragment.

Thus, metavertex, in addition to the attributes, includes a fragment of the metagraph. The presence of private attributes and connections for metavertex is a distinguishing feature of the metagraph. It makes the definition of metagraph holonic – metavertex may include a number of lower-level elements and, in turn, may be included in a number of higher-level elements.

The example of data metagraph (shown in Figure 2) contains three metavertices: **mv1, mv2** and **mv3**. Metavertex **mv1** contains vertices **v1, v2, v3** and connecting them edges **e1, e2, e3**. Metavertex **mv2**contains vertices **v4, v5** and connecting them edge **e6**. Edges **e4, e5** are examples of edges connecting vertices **v2-v4** and **v3-v5**are contained in different metavertices **mv1** and **mv2**. Edge **e7** is an example of the edge connecting metavertices **mv1** and **mv2**. Edge **e8** is an example of the edge connecting vertex **v2** and metavertex **mv2**. Metavertex **mv3** contains metavertex **mv2**, vertices **v2, v3** and edge **e2** from metavertex **mv1** and also edges **e4, e5, e8,**, showing the holonic nature of the metagraph structure.

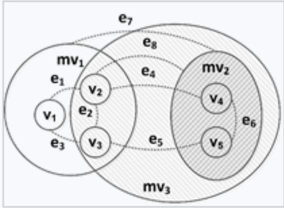

Figure 2: The example of data metagraph

Note that, unlike [3], the anti-annotability property is not fulfilled in this model. The same set of vertices and edges can be included in several different metavertices, which can represent different situations and can be annotated with different attributes. This is why we called our model an **annotating metagraph model**.

Also, unlike the previously considered models, in the model we propose, a metavertex can include both vertices and edges.

The ***metagraph metaedge***: $me_i = \langle \nu_S, \nu_E, eo, \{atr_k\}, MG_j \rangle, e_i \in E, eo = true|false, e\nu_j \in (V \cup E \cup MV \cup ME)$ where me_i — metagraph metaedge belongs to set of metagraph metaedges ME; ν_S — source vertex (metavertex) of the metaedge; ν_E — destination vertex (metavertex) of the metaedge; eo — metaedge direction flag ($eo = true$ — directed metaedge, $eo = false$ — undirected metaedge); atr_k — attribute; MG_j — metagraph fragment.

The example of a directed metaedge is shown in Figure 3. The directed metaedge contains metavertices $m\nu_S, \ldots, m\nu_i, \ldots, m\nu_E$ and connecting them edges. The source metavertex contains a nested metagraph fragment. During the transition to the destination metavertex, the nested metagraph fragment became more complex, new vertices, edges, and inner metavertices are added. Thus, metaedge allows binding the stages of nested metagraph fragment development to the steps of the process described with metaedge.

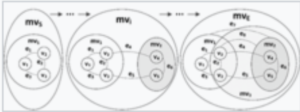

Figure 3: The example of the directed metaedge

Thus, the hierarchical metaedge from the model[5] corresponds to the usual edge in our model. And a metaedge in our model means a sequence of changes of metagraph vertices.

Using Agents to Process Annotated Metagraph Model [edit source]

The metagraph model is aimed for complex data descriptions. However, it is not aimed for data transformation. To solve this issue, the metagraph agent aimed for data transformation is proposed. There are two kinds of metagraph agents: ***the metagraph function agent*** and ***the metagraph rule agent***.

The metagraph function agent serves as a function with input and output parameters in the form of metagraph: $ag^F = \langle MG_{IN}, MG_{OUT}, AST \rangle$, where ag^F — metagraph function agent; MG_{IN} — input parameter metagraph; MG_{OUT} — output parameter metagraph; AST — abstract syntax tree of metagraph function agent in the form of metagraph.

The metagraph rule agent is rule-based: $ag^M = \langle MG_D, R, AG^{ST} \rangle, R = \{r_j\}$ where ag^M — metagraph rule agent; MG_D — working metagraph, a metagraph on the basis of which the rules of the agent are performed; R — set of rules r_j; AG^{ST} — start condition (metagraph fragment for start rule check or start rule); MG_j – a metagraph fragment on the basis of which the rule is performed; OP^{MG} – set of actions performed on metagraph.

$r_i : MG_j \to OP^{MG}$ where r_i — the rule; MG_j — a metagraph fragment on the basis of which the rule is performed; OP^{MG} — set of actions performed on metagraph.

The antecedent of the rule is a condition over the metagraph fragment, the consequent of the rule is a set of actions performed on the metagraph. Rules can be divided into open and closed.

Note that the rules of a metagraphic agent can be divided into ***closed and open***.

The ***open rules*** do not change the metagraph fragment on the right side of the rule that belongs to the left side of the rule. You can separate the input and output fragments of the metagraph. These rules are analogous to a template that generates an output metagraph based on an input one.

Closed rules change the metagraph fragment on the right side of the rule that belongs to the left side of the rule. Changing the metagraph on the right side of the rules causes the left parts of the other rules to be triggered. But at the same time, incorrectly developed closed rules can lead to cycling of the metagraphic agent. Thus, a metagraph agent allows you to generate one metagraph based on another (using open rules) or modify a metagraph (using closed rules).

References [edit source]

- Chapela V., Regino Criado, Santiago Moral, Miguel Romance. Intentional risk management through complex networks analysis. — Springer, 2015: SpringerBriefs in optimization.
- B.S. Manoj, A. Chakraborty, R. Singh. Complex Networks: A Networking and Signal Processing Perspective. Pearson, New York, 2018.
- A.T. Akhmediyarova, J.R. Kuandykova, B.S. Kubekov, I.T. Utepbergenov, V.K. Popkov. Objective of Modeling and Computation of City Electric Transportation Networks Properties. In: International Conference on Information Science and Management Engineering (Icisme 2015), Destech Publications, Phuket, 2015, pp. 106–111.
- Johnson J. Hypernetworks in the science of complex systems. — London, Hackensack NJ: Imperial College Press, 2013.
- Basu A., Robert W. Blanning. Metagraphs and their applications. — New York: Springer, 2007.
- [6] Globa L.S. Metagraph based representation and processing of fuzzy knowledgebases / [6] L.S. Globa, M.Y. Ternovoy, O.S. Shtogrina // Proceedings of 9th Open Semantic Technologies for Intelligent Systems Conference (OSTIS-2015), pp. 237-240. BGUIR Publishing, Minsk 2015.
- [7] Astanin S.V., Zhukovskaya N.K. Business processes control via modeling by fuzzy situational networks. Autom. Remote Control 75, (2014): 570–579.
- V.M. Chernenkiy, Yu.E. Gapanyuk, A.N. Nardid, A.V. Gushcha, Yu.S. Fedorenko. The Hybrid Multidimensional-Ontological Data Model Based on Metagraph Approach. In: A.K. Petrenko, A. Voronkov (eds.) Perspectives of systems informatics. 11th International Andrei P. Ershov Informatics Conference, PSI 2017, Moscow, Russia, June 27-29, 2017, Revised Selected Papers / edited by Alexander K. Petrenko, Andrei Voronkov, vol. 10742. Lecture notes in computer science, 0302-9743, vol. 10742, pp. 72–87. Springer (2018). doi: 10.1007/978-3-319-74313-4_6
- Yu.E. Gapanyuk. Metagraph Approach to the Information-Analytical Systems Development. In: Proceedings of the 6th International Conference Actual Problems of System and Software Engineering, Moscow, Russia, 2019, pp. 428-439.

Notes [edit source]

1. ^ Basu A., Robert W. Blanning. Metagraphs and their applications. — New York: Springer, 2007. — C. 14.
2. ^ a b Basu A., Robert W. Blanning. Metagraphs and their applications. — New York: Springer, 2007.
3. ^ a b c d e Globa L.S. Metagraph based representation and processing of fuzzy knowledgebases / L.S. Globa, M.Y. Ternovoy, O.S. Shtogrina
4. ^ Globa L.S. Metagraph based representation and processing of fuzzy knowledgebases / L.S. Globa, M.Y. Ternovoy, O.S. Shtogrina -pp. 238.
5. ^ a b c d e f Astanin S.V., Zhukovskaya N.K. Business processes control via modeling by fuzzy situational networks. Autom. Remote Control 75, (2014): 570–579.
6. ^ Yu.E. Gapanyuk. Metagraph Approach to the Information-Analytical Systems Development. In: Proceedings of the 6th International Conference Actual Problems of System and Software Engineering, Moscow, Russia, 2019, pp. 428-439.

Выводы

В проведенном исследовании метаграфов мы представили анализ этого инструмента моделирования и анализа телекоммуникационных систем.

Цель настоящего исследования также включала публикацию статьи на Википедии, посвященной метаграфам, с целью создания более информативной и понятной базы данных об этой теме для широкой аудитории.

В итоге, проделанные усилия направлены на обогащение знаний об этой важной области и поддержку цифрового общества, предоставляя точные и актуальные сведения о технологиях телекоммуникаций. Публикация английской версии статьи на Википедии о метаграфах способствует более глубокому пониманию современных тенденций в области телекоммуникаций и способствует развитию цифрового информационного пространства.