Ideas of Metagraph-Based Types

Yuriy E. Gapanyuk, Anatoly N. Nardid, and Dmitry A. Zuev

Bauman Moscow State Technical University, Moscow, Russia
gapyu@bmstu.ru, nazgul109@gmail.com, zuynew@yandex.ru

Abstract

The metagraph model is a kind of "complex networks with emergence" model. To process and transform metagraph data, the metagraph agents are used. The combination of the metagraph data model and the metagraph agent model makes it possible to represent various type systems in the form of a metagraph model.

According to the HOTT book [5]: "the basic concept of type theory, that the term a is of type A, which is written: $a : A$. This expression is traditionally thought of as akin to: 'a is an element of the set A'. However, in homotopy type theory we think of it instead as: 'a is a point of the space A' ."

We propose the basic ideas of an approach in which a is a subgraph in a complex graph A. According to [1]: "a complex network is a graph (network) with non-trivial topological features – features that do not occur in simple networks such as lattices or random graphs but often occur in graphs modeling of real systems." The terms "complex network" and "complex graph" are often used synonymously. According to [2]: "the term 'complex network,' or simply 'network,' often refers to real systems while the term 'graph' is generally considered as the mathematical representation of a network." In this paper, we also consider these terms synonymous.

One of the essential kinds of such complex network models is "complex networks with emergence." The term "emergence" is used in general system theory. The emergent element means a whole that cannot be separated into its component parts. As far as the authors know, currently, there are two "complex networks with emergence" models that exist: hypernetworks and metagraphs.

The hypernetwork model [4] is mature, and it helps to understand many aspects of complex networks with an emergence. However, from the authors' point of view, the metagraph model is more flexible and convenient than a hypernetwork model for use in information systems [3].

According to paper [3], the metagraph approach may be considered as a higher-level structural framework for the representation of dynamical complex graph structures.

The metagraph is described as follows: $MG = \langle V, MV, E \rangle$, where MG – metagraph; V – set of metagraph vertices; MV – set of metagraph metavertices; E – set of metagraph edges.

Metagraph vertex is described by set of attributes: $v_i = \{atr_k\}, v_i \in V$, where atr_k – attribute.

Metagraph edge is described by set of attributes, the source and destination vertices (or metavertices): $e_i = \langle v_S, v_E, \{atr_k\} \rangle, e_i \in E$, where e_i – metagraph edge; v_S – source vertex (metavertex) of the edge; v_E – destination vertex (metavertex) of the edge; atr_k – attribute.

The metagraph fragment is defined as $MG_i = \{ev_j\}, ev_j \in (V \cup E \cup MV)$, where MG_i – metagraph fragment; ev_j – an element that belongs to union of vertices, edges and metavertices.

The metagraph metavertex: $mv_i = \langle \{atr_k\}, MG_f \rangle, mv_i \in MV$, where mv_i – metagraph metavertex; atr_k – attribute, MG_f – metagraph fragment.

From the general system theory point of view, metavertex is a particular case of manifestation of emergence principle, which means that metavertex with its private attributes and connections became whole that cannot be separated into its component parts. The example of metagraph representation is represented in Fig. 1.

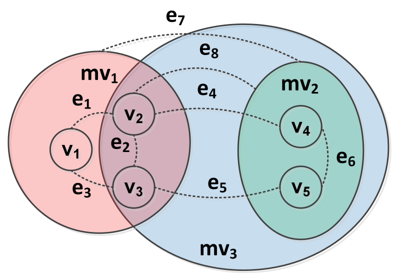

Figure 1: The example of metagraph representation.

The example contains three metavertices: mv_1, mv_2, and mv_3. Metavertex mv_1 contains vertices v_1, v_2, v_3 and connecting them edges e_1, e_2, e_3. Metavertex mv_2 contains vertices v_4, v_5, and connecting them edge e_6. Edges e_4, e_5 are examples of edges connecting vertices v_2–v_4 and v_3-v_5 are contained in different metavertices mv_1 and mv_2. Edge e_7 is an example of the edge connecting metavertices mv_1 and mv_2. Edge e_8 is an example of the edge connecting vertex v_2 and metavertex mv_2. Metavertex mv_3 contains metavertex mv_2, vertices v_2, v_3 and edge e_2 from metavertex mv_1 and also edges e_4, e_5, e_8 showing emergent nature of metagraph structure.

Consider the basics of the object-oriented data structures representation using the metagraph approach. We review only data structures containing data fields in form $name : type : value$ where $type$ may be atomic type, complex type or list (collection) type.

The data structure formally may be defined as follows: $DS = \langle ds_T, DS_F \rangle, ds_T \in TP, DS_F = \{fld^i\}$, where DS – data structure; ds_T – data structure type belongs set of types TP; DS_F – set of data structure fields fld^i.

The field is defined as follows: $fld^i = \langle fld_N, fld_T, fld_V \rangle, fld_T \in TP$, where fld_N – field name; fld_T – field type belongs set of types TP, fld_V – field value of type fld_T.

Every type tp belonging to set of types TP must be either atomic type TP_A or complex type TP_C or list (collection) type TP_L. The atomic type TP_A corresponds to the only value. The complex type TP_C contains a set of corresponding field types fld_T. The list type TPL is a collection of elements of any type: $(\forall tp \in TP) tp = TP_A | TP_C = \{fld_T\} | TP_L = [TP]$.

The example showing one of the possible cases of metagraph representation of object–oriented data structure is represented in Fig. 2.

Data structure DS and its corresponding type are represented as a metavertices bound with edge ds_T. The set of data structure fields DS_F (also represented as a metavertex) consists of three fields fld^1, fld^2 and fld^3.

Field fld^1 with the name "field1" corresponds to the atomic type "int" with value "1". Field fld^1 is represented as a metavertex, field name fld^1_N, and value fld^1_V are represented as inner vertices. The field type is represented as edge fld^1_T bound field metavertex with atomic type TP_A vertex.

Field fld^2 with the name "field2" corresponds to the complex type consists of fields "field2_1" of type "int" with value "2" and "field2_2" of type "string" with value "string2". Field fld^2 is represented as a metavertex, field name fld^2_N is represented as inner vertex, and value fld^2_V is represented as inner metavertex containing metavertices fld^{2_1} and fld^{2_2} correspondings to subfields "field2_1" and "field2_2" with their values. Field fld^2 type is represented as edge fld^2_T bound field metavertex with complex type TP_C metavertex. The TP_C metavertex contains

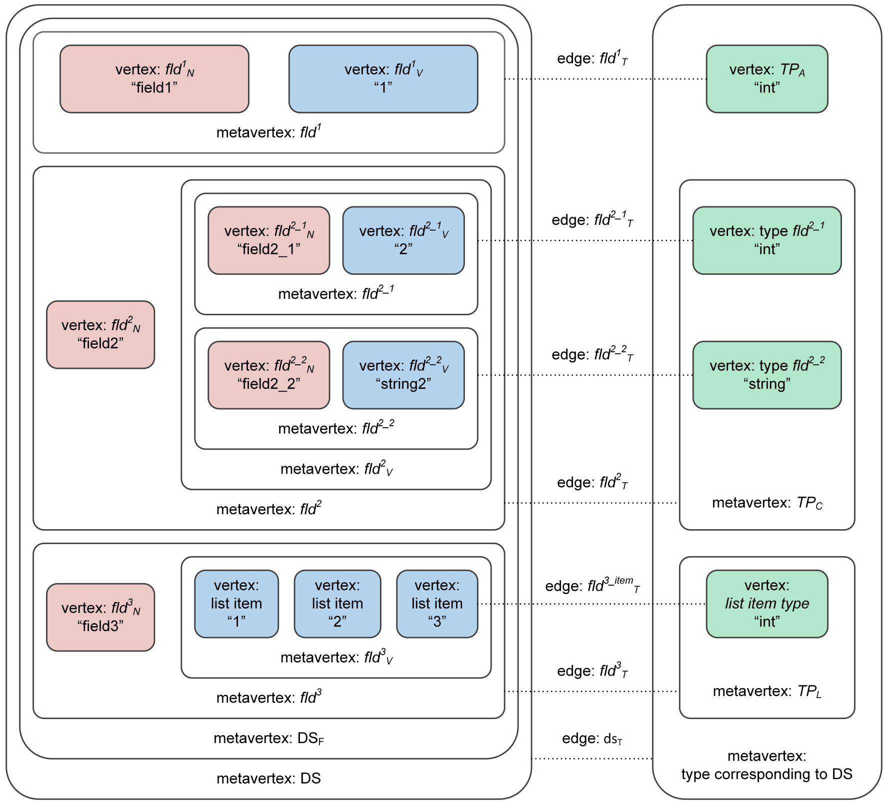

Figure 2: The metagraph representation of object-oriented data structure.

inner vertices corresponding to subfields $fld^{2\text{-}1}$ and $fld^{2\text{-}2}$ types. The edges $fld_T^{2\text{-}1}$ and $fld_T^{2\text{-}2}$ bound subfields $fld^{2\text{-}1}$ and $fld^{2\text{-}2}$ metavertices with corresponding subtypes vertices.

Field fld^3 with the name "field3" corresponds to the list (collection) type "list of int" with value "1, 2, 3". Field fld^3 is represented as a metavertex, field name fld_N^3 is represented as inner vertex and value fld_V^3 is represented as inner metavertex corresponding to the list containing vertices corresponding to the list items. The field type is represented as edge fld_T^3 bound field metavertex with list (collection) type TP_L metavertex. The TP_L metavertex contains inner vertex corresponds to the list item type. List items bound with list item type with $fld_T^{3\text{-}item}$ edge (shown only for list item "3" in order not to clutter the figure).

The example shows that the object–oriented data structure may be represented using the metagraph approach without losing detailed information. In conclusion, we note other features of the metagraph model related to the description of types:

- Types are considered as fragments of a complex graph, in which not only the values of the vertices are important, but also the relationships between them. This makes the metagraph model related to the ontological knowledge model.

- To process and transform metagraph data, the metagraph agents are used. The metagraph agent may be represented as a set of metagraph fragments. The distinguishing feature of the metagraph agent is its homoiconicity, which means that it can be data structure for itself.

- The combination of the metagraph data model and the metagraph agent model makes it possible to represent various type systems in the form of a metagraph model. In this case, the relationships between the elements of the model are represented explicitly.

References

[1] Manoj B.S., Abhishek Chakraborty, and Rahul Singh. *Complex Networks: A Networking and Signal Processing Perspective*. Prentice Hall, 2018.

[2] Victor Chapela, Regino Criado, Santiago Moral, and Miguel Romance. *Intentional Risk Management through Complex Networks Analysis*. SpringerBriefs in Optimization. Springer International Publishing, first edition, 2015.

[3] Valeriy M. Chernenkiy, Yuriy E. Gapanyuk, Anatoly N. Nardid, Anton V. Gushcha, and Yuriy S. Fedorenko. The hybrid multidimensional-ontological data model based on metagraph approach. In Alexander K. Petrenko and Andrei Voronkov, editors, *Perspectives of System Informatics*, volume 10742 of *LNCS*, pages 72–87, Cham, 2018. Springer International Publishing.

[4] Jeffrey Johnson. *Hypernetworks in the Science of Complex Systems*. Series on Complexity Science. Imperial College Press, 2013.

[5] The Univalent Foundatiosn Program. *Homotopy Type Theory: Univalent Foundations of Mathematics*. http://homotopytypetheory.org/book/, 2013.