



Московский государственный технический университет им. Н.Э. Баумана

ПРОГРАММНАЯ ПЛАТФОРМА ОБРАБОТКИ МЕТАГРАФОВ

Гапанюк Ю.Е., Нардид А.Н., кафедра ИУ-5

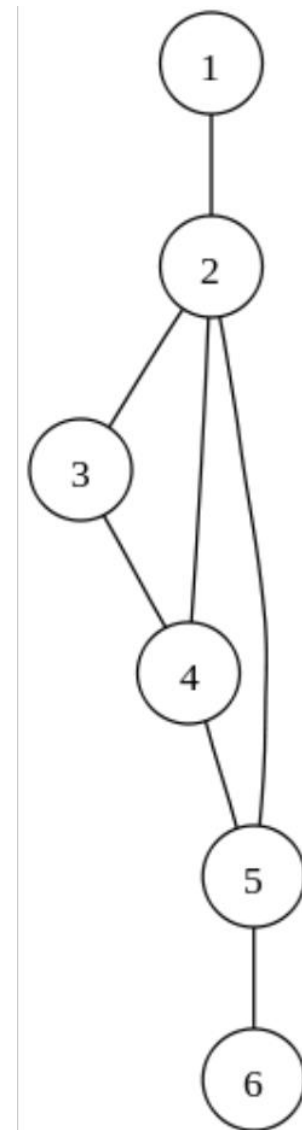
Простые графы

Простой граф

Определение. *Простой граф* $G(V, E)$ есть совокупность двух множеств – непустого множества V и множества E неупорядоченных пар различных элементов множества V . Множество V называется *множеством вершин*, множество E называется *множеством рёбер*

$$G(V, E) = \langle V, E \rangle, \quad V \neq \emptyset, \quad E \subseteq V \times V, \quad \{v, v\} \notin E, \quad v \in V,$$

то есть множество E состоит из 2-элементных подмножеств множества V .



Неориентированный граф с шестью вершинами и семью рёбрами

Мультиграфы

Мультиграф

Мультиграф $G(V, \mathbf{E})$ есть совокупность двух множеств – непустого множества V и мультимножества \mathbf{E} неупорядоченных пар различных элементов множества V .

$$G(V, \mathbf{E}) = \langle V, \mathbf{E} \rangle, \quad V \neq \emptyset, \quad \mathbf{E} \subseteq V \times V \quad \{v, v\} \notin \mathbf{E}, \quad v \in V$$

- *Кратными рёбрами* называются одинаковые элементы мультимножества $\{e, e, \dots, e\} \in \mathbf{E}$, то есть ребра, чьи концевые вершины совпадают.

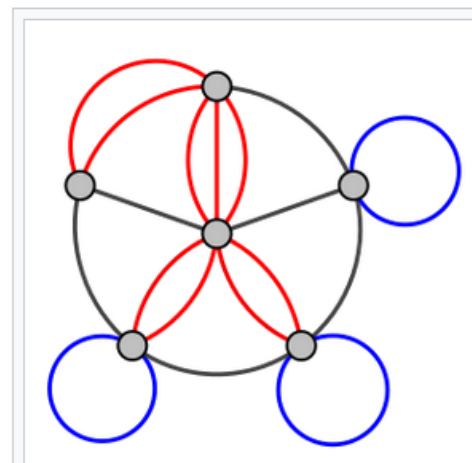
Другими словами Если E не множество, а семейство, то есть если \mathbf{E} содержит одинаковые элементы, то такие элементы называются кратными рёбрами, а граф называется мультиграфом

Псевдомультиграф

Псевдомультиграф $G(V, \mathbf{E})$ есть совокупность двух множеств – непустого множества V и мультимножества \mathbf{E} неупорядоченных пар элементов множества V .

$$G(V, \mathbf{E}) = \langle V, \mathbf{E} \rangle, \quad V \neq \emptyset, \quad \mathbf{E} \subseteq V \times V$$

Другими словами Если \mathbf{E} семейство содержащее одинаковые элементы (кратные ребра) и \mathbf{E} может содержать петли, то граф называется псевдомультиграфом



Псевдомультиграф с кратными рёбрами (красные) и петлями (синие).

Сложные сети

- В настоящее время термины «сложная сеть» или «комплексная сеть» (являются различными переводами англоязычного термина «complex network») и термин «сложный граф» (англ. «complex graph») часто употребляются как синонимы.
- В работе [1, стр. 14] отмечается, что термин «сложная сеть», как правило, употребляется для обозначения реальной исследуемой системы, в то время как термин «сложный граф» обычно используют для обозначения математической модели такой системы.
- Наибольшие разночтения вызывает термин «сложный» применительно к графовым моделям. Как правило, термин «сложный» трактуется в двух вариантах:

Сложные сети, вариант I

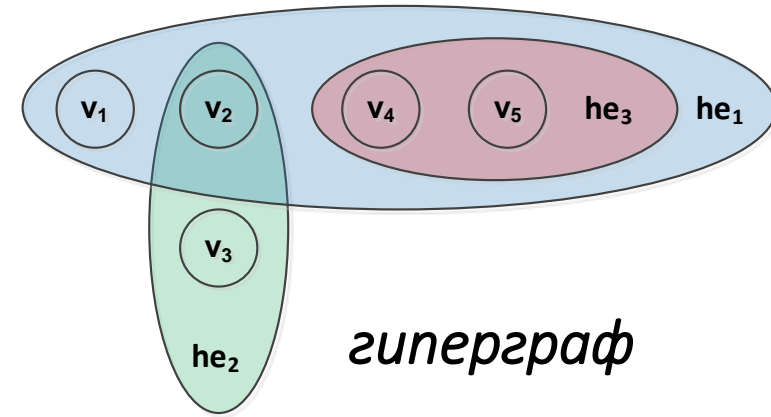
- Плоские графы (сети) очень большой размерности.
- Такие сети могут включать миллионы и более вершин.
- Ребра, соединяющие вершины, могут быть ненаправленными или направленными.
- Иногда используется модель мультиграфа, в этом случае две вершины могут соединяться не одним, а несколькими ребрами.
- Такие модели представляют интерес при изучении социальных сетей, глобальных компьютерных сетей, различных социологических и биологических моделей. Но они не очень хорошо помогают при описании сложных моделей данных и знаний.

Сложные сети, вариант II

- Сложные графы, в которых используется сложное (комплексное) описание вершин, ребер и/или их расположения.
- Часто в таких моделях отказываются от плоского расположения вершин и ребер.
- Именно подобные модели могут быть наиболее полезны при описании сложных моделей данных.
- На сегодняшний день известны четыре подобных модели: **гиперграф, гиперсеть, метаграф и многоуровневая сеть** (которая является упрощенным вариантом гиперсети).
- В настоящее время в литературе еще не появился единый «собирательный термин» для моделей такого класса. Авторы моделей, как правило, используют собственные названия для каждой модели, не всегда даже указывая на родство предлагаемой модели со сложными графами (сетями).
- Для подобного класса моделей можно предложить такой «собирательный термин» как **«ансамбли сложных сетей (графов)»**.
- Для гиперсетевой и метаграфовой моделей может быть использован термин **«сложные сети (графы) с эмерджентностью»**, так как данные модели реализуют принцип эмерджентности, хорошо известный в общей теории систем.

Гиперграф

- Гиперграф $HG = \langle V, HE \rangle$, $v_i \in V$, $he_j \in H$, V – множество вершин гиперграфа; HE – множество непустых подмножеств V , называемых гиперребрами; v_i – вершина гиперграфа; he_j – гиперребро гиперграфа. Гиперребро ненаправленного гиперграфа включает множество вершин, а ребро направленного гиперграфа задает последовательность обхода вершин.



- Гиперребро he_1 включает вершины v_1 , v_2 , v_4 , v_5 ; гиперребро he_2 включает вершины v_2 и v_3 ; гиперребро he_3 включает вершины v_4 и v_5 . Гиперребра he_1 и he_2 имеют общую вершину v_2 . Все вершины гиперребра he_3 также являются вершинами гиперребра he_1 . Но «вложенность» гиперребра he_3 в гиперребро he_1 является скорее «визуальным эффектом», потому что операция вложенности для гиперребер формально не определена.
- Поэтому, хотя гиперграф и содержит гиперребра, но не позволяет моделировать сложные иерархические зависимости и не является полноценной «сетью с эмерджентностью».

Аннотируемая метаграфовая модель

Определим аннотируемый метаграф следующим образом:

$$MG = \langle V, MV, E, ME \rangle,$$

где MG – метаграф; V – множество вершин метаграфа; MV – множество метавершин метаграфа; E – множество ребер метаграфа, ME – множество метаребер метаграфа.

Вершина метаграфа характеризуется множеством атрибутов:

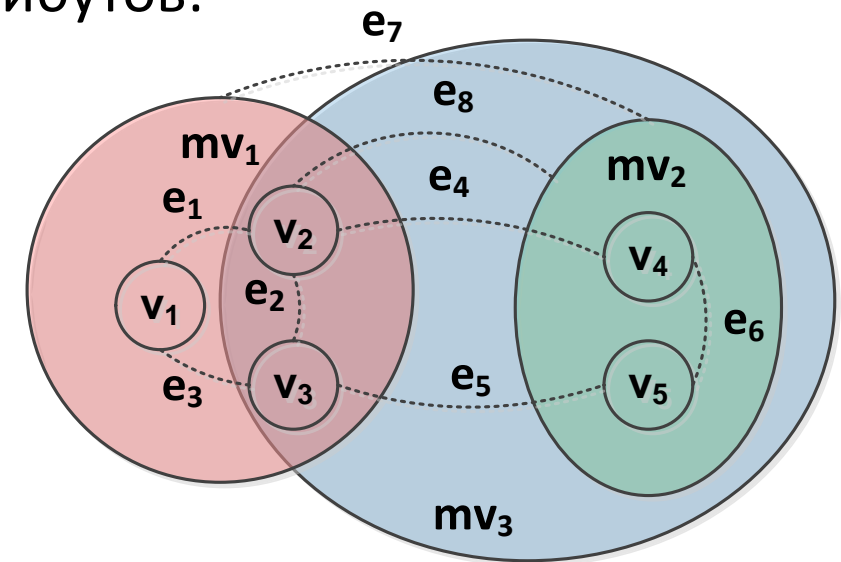
$$v_i = \{atr_k\}, v_i \in V,$$

где v_i – вершина метаграфа; atr_k – атрибут.

Ребро метаграфа характеризуется множеством атрибутов, исходной и конечной вершиной:

$$e_i = \langle v_S, v_E, \{atr_k\} \rangle, e_i \in E,$$

где e_i – ребро метаграфа; v_S – исходная вершина (метавершина) ребра; v_E – конечная вершина (метавершина) ребра; atr_k – атрибут.



Аннотируемая метаграфовая модель - 2

Фрагмент метаграфа:

$$MG_i = \{ev_j\}, ev_j \in (V \cup E \cup MV \cup ME),$$

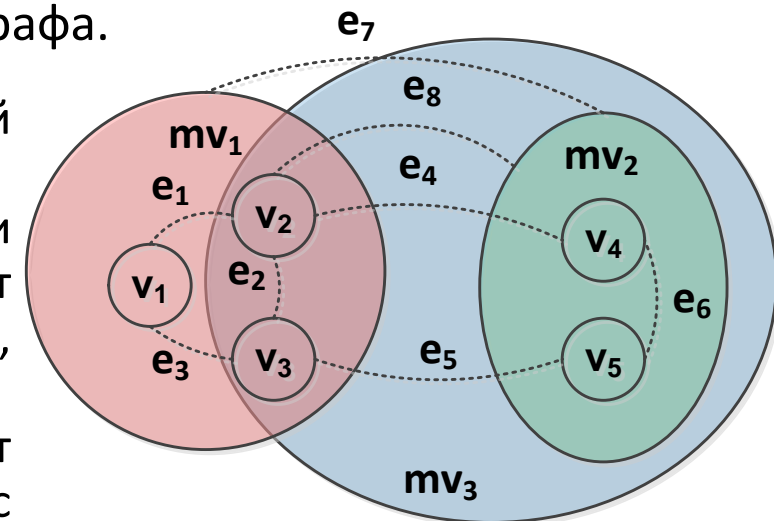
где MG_i – фрагмент метаграфа; ev_j – элемент, принадлежащий объединению множеств вершин, метавершин, ребер и метаребер метаграфа.

Фрагмент метаграфа в общем виде может содержать произвольные вершины (метавершины) и ребра.

Метавершина метаграфа: $mv_i = \langle \{atr_k\}, MG_i \rangle, mv_i \in MV,$

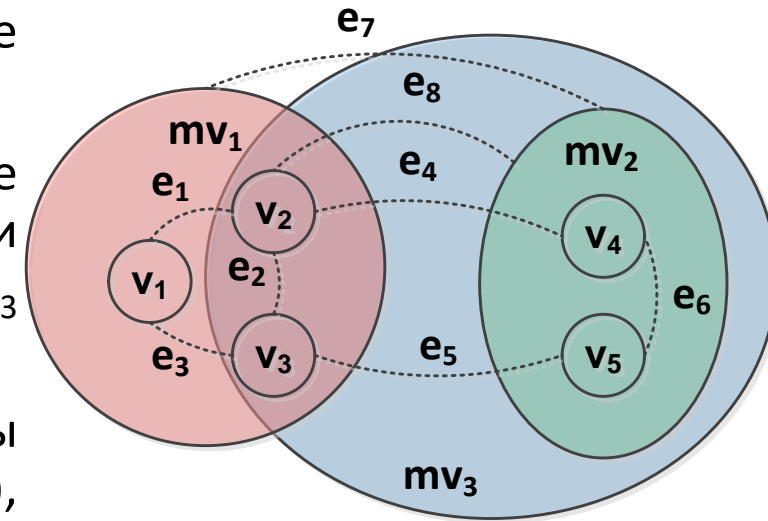
где mv_i – метавершина метаграфа; atr_k – атрибут, MG_i – фрагмент метаграфа.

- Метавершина в дополнение к свойствам вершины включает вложенный фрагмент метаграфа.
- Наличие у метавершин собственных атрибутов и связей с другими вершинами является важной особенностью метаграфов. Это соответствует принципу эмерджентности, то есть приданию понятию нового качества, несводимости понятия к сумме его составных частей.
- Как только вводится новое понятие в виде метавершины, оно «получает право» на собственные свойства, связи и т.д., так как в соответствии с принципом эмерджентности новое понятие обладает новым качеством и не может быть сведено к подграфу базовых понятий.



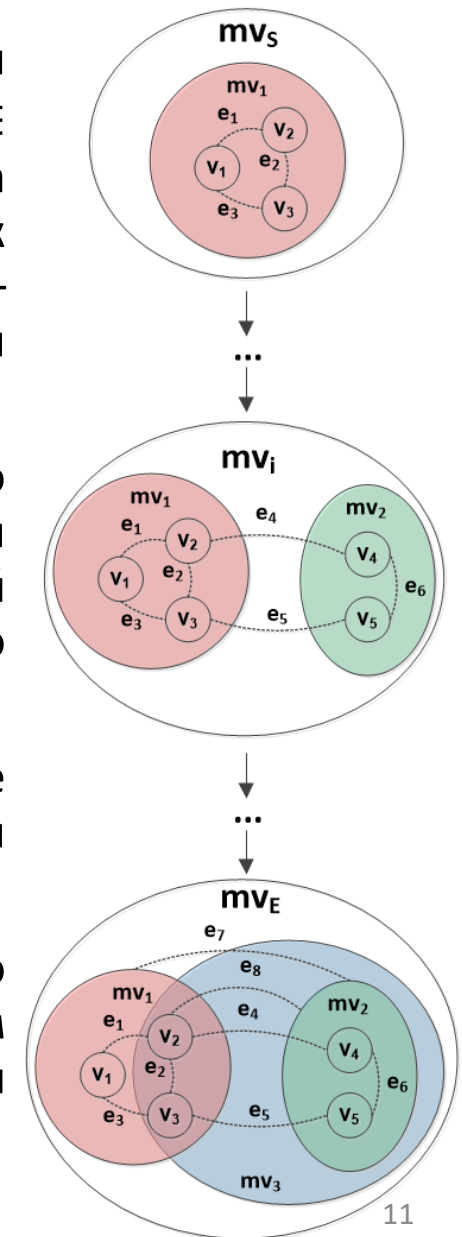
Аннотируемая метаграфовая модель – (пример)

- Метаграф позволяет естественным образом моделировать сложные иерархические зависимости и является «сетью с эмерджентностью».
- Метаграф содержит вершины, метавершины и ребра. На рисунке показаны три метавершины: mv_1 (которая включает вершины v_1, v_2, v_3 и ребра e_1, e_2, e_3), mv_2 (которая включает вершины v_4, v_5 и ребро e_6) и mv_3 (которая включает метавершину mv_2 , вершины v_1 и v_2 и ряд ребер).
- Ребро метаграфа может соединять вершины внутри одной метавершины (e_1, e_2, e_3, e_6), вершины между различными метавершинами (e_4, e_5), метавершины (e_7), вершины и метавершины (e_8).
- Метавершина позволяет выделять фрагмент графа (метаграфа), аннотировать его дополнительными свойствами, проводить к нему (как к целому) ребра.
- Отметим, что в отличие от [5], в данной модели не выполняется свойство анти-аннотируемости. Одинаковый набор вершин и ребер может быть включен в несколько различных метавершин, которые могут представлять различные ситуации и быть аннотированы различными атрибутами.
- Также, в предлагаемой модели, метавершина может включать как вершины, так и ребра.



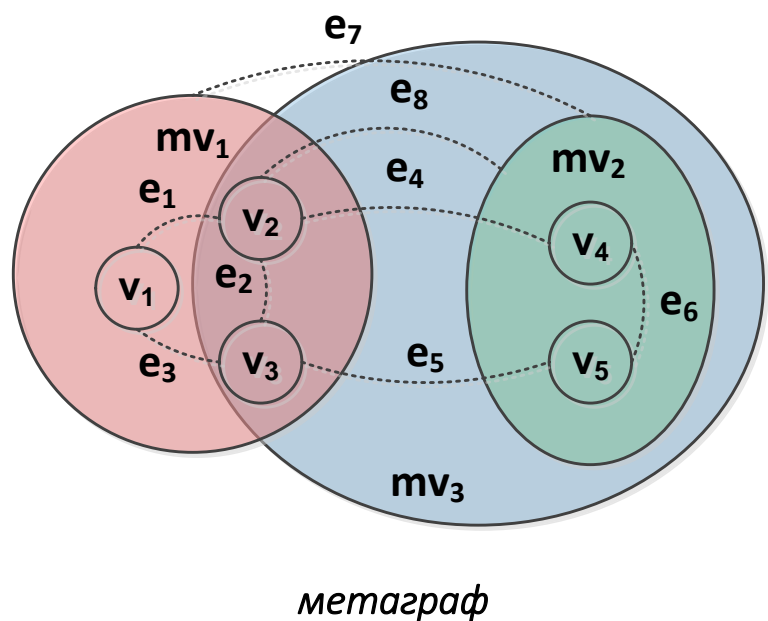
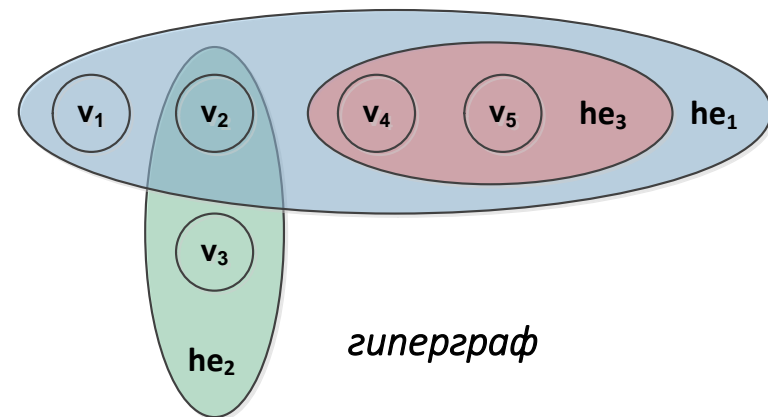
Аннотируемая метаграфовая модель – метаребро

- **Метаребро метаграфа** в дополнение к свойствам ребра включает вложенный фрагмент метаграфа: $me_i = \langle v_S, v_E, eo, \{atr_k\}, \{ev_j\} \rangle, e_i \in E, eo = true|false, ev_j \in (V \cup E \cup MV \cup ME)$, где me_i – метаребро метаграфа; v_S – исходная вершина (метавершина) ребра; v_E – конечная вершина (метавершина) ребра; eo – признак направленности метаребра ($eo=true$ – направленное метаребро, $eo=false$ – ненаправленное метаребро); atr_k – атрибут; ev_j – элемент, принадлежащий объединению множеств вершин (метавершин) и ребер (метаребер) метаграфа.
- Пример описания метаребра метаграфа представлен на рисунке. Метаребро содержит метавершины $v_S, \dots, v_i, \dots, v_E$ и связывающие их ребра. Исходная метавершина содержит фрагмент метаграфа. В процессе преобразования исходной метавершины v_S в конечную метавершину v_E происходит дополнение содержимого метавершины, добавляются новые вершины, связи, вложенные метавершины.
- Таким образом, иерархическому метаребру из модели [6] соответствует обычное ребро в предлагаемой нами модели. А под метаребром понимается последовательность изменения метавершин метаграфа.
- Если метавершины предназначены прежде всего для описания данных и знаний, то метаребра предназначены в большей степени для описания процессов. Таким образом, аннотируемая метаграфовая модель позволяет в рамках единой модели описывать данные, знания и процессы.



Метаграфы и гиперграфы

- Гиперграф $HG = \langle V, HE \rangle$, $v_i \in V$, $he_j \in H$, V – множество вершин гиперграфа; HE – множество непустых подмножеств V , называемых гиперребрами; v_i – вершина гиперграфа; he_j – гиперребро гиперграфа. Гиперребро ненаправленного гиперграфа включает множество вершин, а ребро направленного гиперграфа задает последовательность обхода вершин.
- Гиперребро he_1 включает вершины v_1, v_2, v_4, v_5 ; гиперребро he_2 включает вершины v_2 и v_3 ; гиперребро he_3 включает вершины v_4 и v_5 . Гиперребра he_1 и he_2 имеют общую вершину v_2 . Все вершины гиперребра he_3 также являются вершинами гиперребра he_1 . Но «вложенность» гиперребра he_3 в гиперребро he_1 является скорее «визуальным эффектом», потому что операция вложенности для гиперребер формально не определена. Поэтому, хотя гиперграф и содержит гиперребра, но не позволяет моделировать сложные иерархические зависимости и не является полноценной «сетью с эмерджентностью».
- Если гиперребро гиперграфа может включать только вершины, то метавершина метаграфа может включать как вершины (или метавершины), так и ребра.
- В отличие от гиперграфа, метаграф позволяет естественным образом моделировать сложные иерархические зависимости и является «сетью с эмерджентностью».



Разновидности агентов для обработки метаграфовой модели

- Метаграф является пассивной структурой данных. Как реализовать активность при моделировании сложной сети с эмерджентностью? Используется идея преобразования метаграфов с помощью агентов.
- Идея преобразования плоских графов с помощью агентов принадлежит профессору Владимиру Васильевичу Голенкову (графодинамическая парадигма).
- Для обработки метаграфов предлагается использовать два вида агентов: агент-функцию и метаграфовый агент. Рассмотрим данные виды агентов более подробно.

Агент-функция

- Агент-функция:

$$ag^F = \langle MG_{IN}, MG_{OUT}, AST \rangle,$$

- где ag^F – агент-функция; MG_{IN} – метаграф, который выполняет роль входного параметра агента-функции; MG_{OUT} – метаграф, который выполняет роль выходного параметра агента-функции; AST – абстрактное синтаксическое дерево агента-функции, которое может быть представлено в виде метаграфа.

Метаграфовый агент (определение)

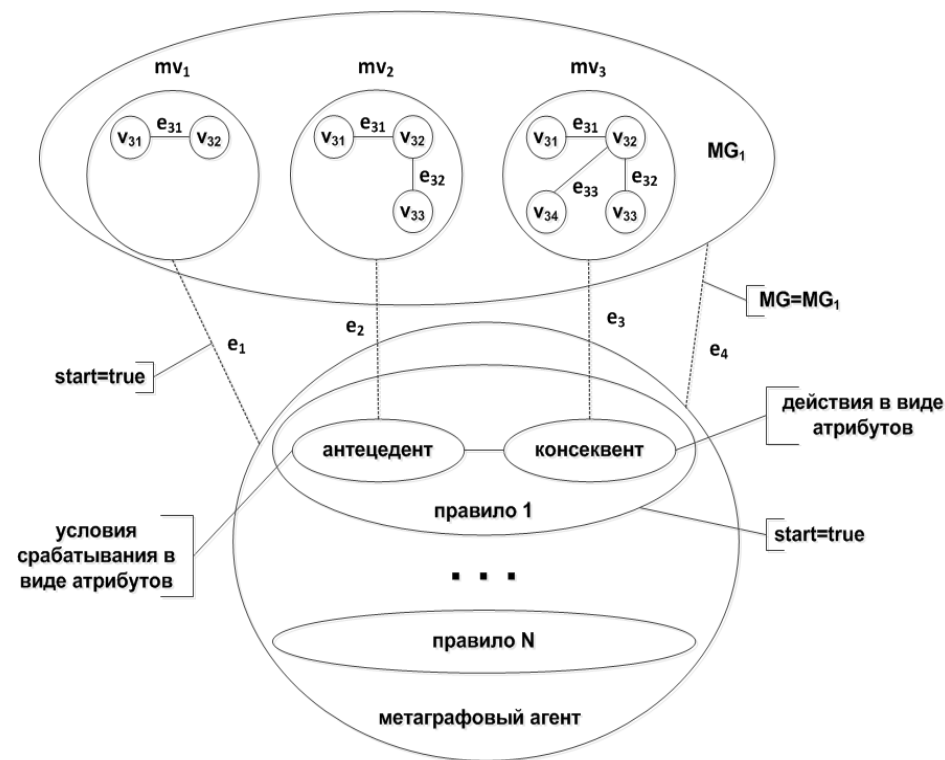
- Метаграфовый агент:

$$ag^M = \langle MG, R, AG^{ST}, \{ag_i^M\} \rangle, R = \{r_j\},$$

- где ag^M – метаграфовый агент; MG – метаграф, на основе которого выполняются правила агента; R – набор правил (множество правил r_j); AG^{ST} – стартовое условие выполнения агента (фрагмент метаграфа, который используется для стартовой проверки правил, или стартовое правило).
- При этом агент ag^M содержит множество вложенных агентов ag_i^M что соответствует принципам организации холонической многоагентной системы. Агент верхнего уровня может активизировать агентов нижнего уровня для решения подзадач.
- Структура правила метаграфового агента:

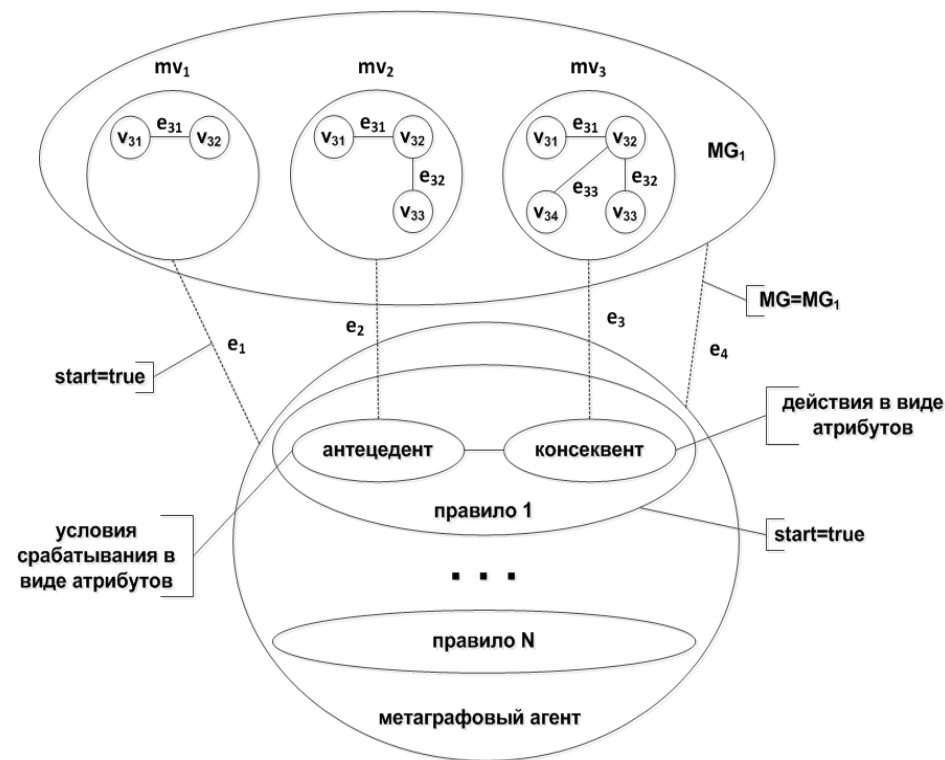
$$r_i : MG_j \rightarrow OP^{MG},$$

- где r_i – правило; MG_j – фрагмент метаграфа, на основе которого выполняется правило; OP^{MG} – множество действий, выполняемых над метаграфом.
- Антецедементом правила является фрагмент метаграфа, консеквентом правила является множество действий, выполняемых над метаграфом.



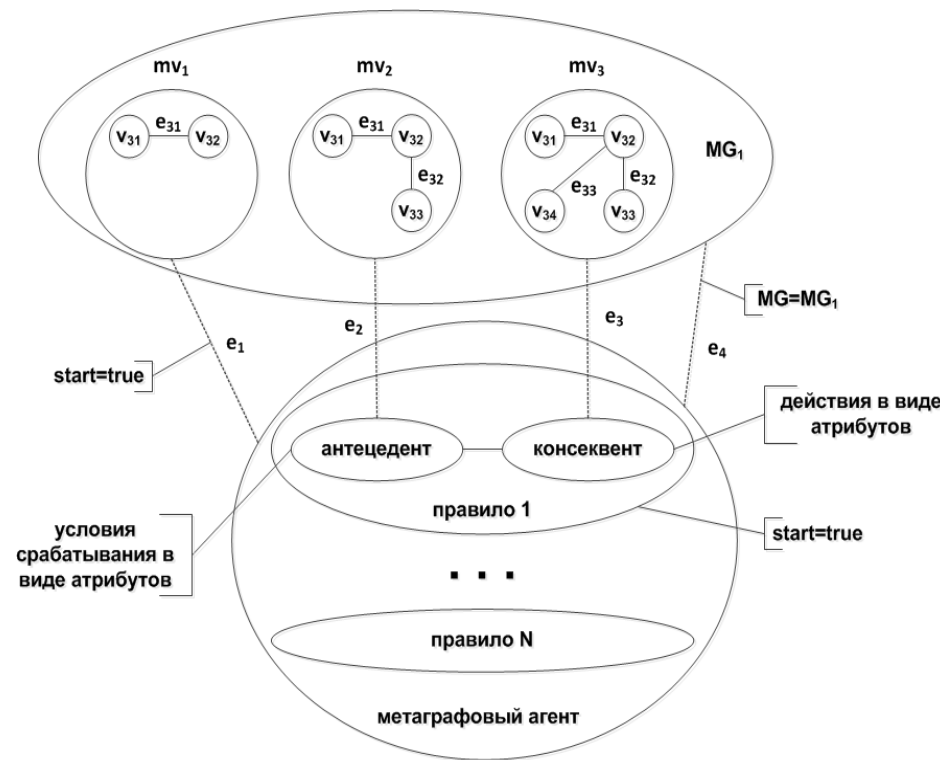
Метаграфовый агент (правила)

- Правила метаграфового агента можно разделить на замкнутые и разомкнутые.
- Разомкнутые правила не меняют в правой части правила фрагмент метаграфа, относящийся к левой части правила. Можно разделить входной и выходной фрагменты метаграфа. Данные правила являются аналогом шаблона, который порождает выходной метаграф на основе входного.
- Замкнутые правила меняют в правой части правила фрагмент метаграфа, относящийся к левой части правила. Изменение метаграфа в правой части правил заставляет срабатывать левые части других правил. Но при этом некорректно разработанные замкнутые правила могут привести к заикливанию метаграфового агента.
- Таким образом, метаграфовый агент позволяет генерировать один метаграф на основе другого (с использованием разомкнутых правил) или модифицировать метаграф (с использованием замкнутых правил).



Метаграфовый агент (самоотображаемость)

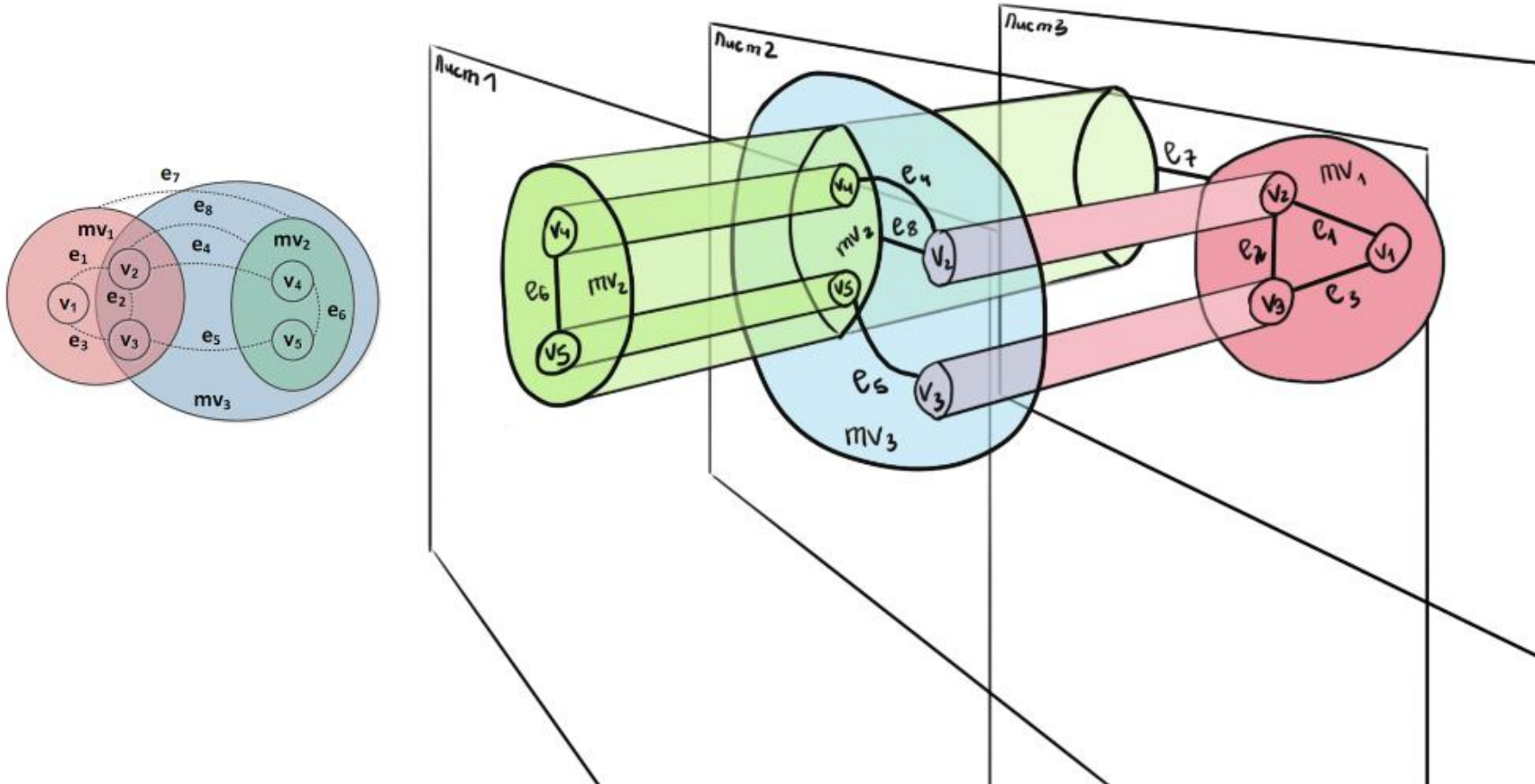
- Особенностью метаграфового агента является то, что его структура может быть представлена в виде фрагмента метаграфа. Это соответствует принципу самоотображаемости (англ. homoiconicity) в языках программирования. Самоотображаемость – это способность языка программирования анализировать программу на этом языке как структуру данных этого языка.
- Структура агента может быть изменена как данные с помощью правил агентов верхнего уровня.
- Метаграфовый агент представлен в виде метавершины метаграфа. В соответствии с определением он связан с метаграфом MG_1 , на основе которого выполняются правила агента. Данная связь показана с помощью ребра e_4 .
- Метаграфовый агент содержит множество вложенных метавершин, соответствующих правилам (правило 1 – правило N). В данном примере с антецедентом правила связана метавершина данных mv_2 , что показано ребром e_2 , а с консеквентом правила связана метавершина данных mv_3 , что показано ребром e_3 . Условия срабатывания задаются в виде атрибутов соответствующих вершин.
- Стартовое условие выполнения агента задается с помощью атрибута «start=true». Если стартовое условие задается в виде стартового правила, то данным атрибутом помечается метавершина соответствующего правила, в данном примере это правило 1. Если стартовое условие задается в виде стартового фрагмента метаграфа, который используется для стартовой проверки правил, то атрибутом «start=true» помечается ребро, которое связывает стартовый фрагмент метаграфа с метавершиной агента, в данном примере это ребро e_1 .



Программная платформа обработки метаграфов

- Технологии:
 - Backend – Rust
 - Frontend – JavaScript, TypeScript, React, Svelte
- Задачи:
 - Модуль хранения метаграфов
 - Модуль работы с метаграфовыми агентами
 - Метаграфовые агенты [на основе правил](#)
 - Метаграфовые агенты [на основе нечеткой логики](#)
 - Модуль пользовательского интерфейса
 - Реализация библиотеки визуализации метаграфов для использования в веб-приложениях.
 - Реализация мультивидового редактора для метаграфовых данных (редактирование данных в виде объектов, списков, таблиц).
 - Реализация генератора отчетов для метаграфовых данных (отчеты, [дашборды](#)).
 - Прикладные системы на основе метаграфового подхода
 - Реализация семантической файловой системы на основе метаграфового подхода

Визуализация метаграфов – 1 – 1



Визуализация метаграфов – 1 – 2

Примеры визуализаций

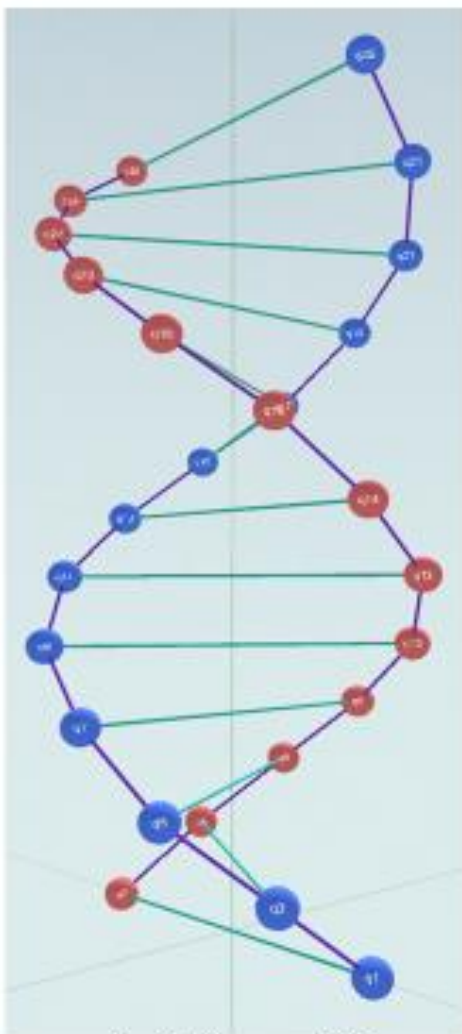


Рис.12. Цепочка ДНК

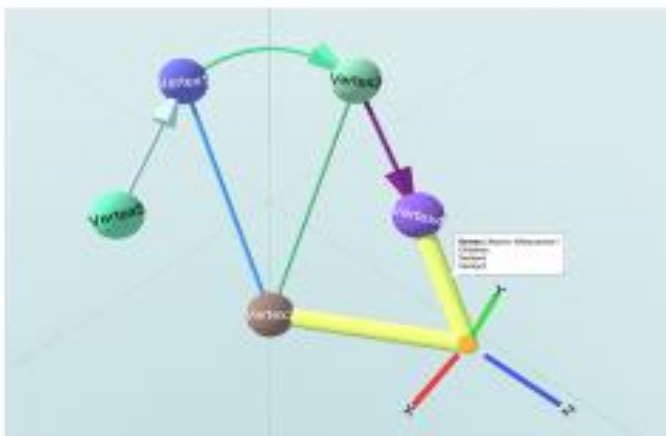


Рис.14. Пример расстановки метаграфа

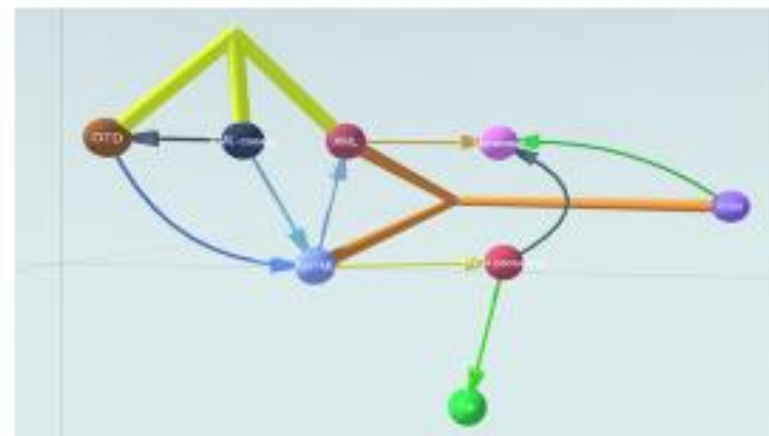


Рис.15. Метаграф «Языки структуризации»

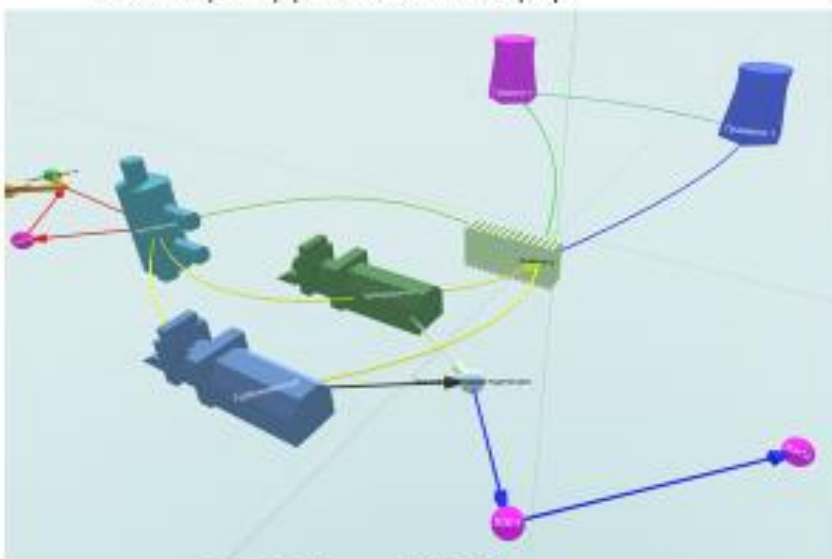


Рис.13. Модель АЭС с 3D моделями

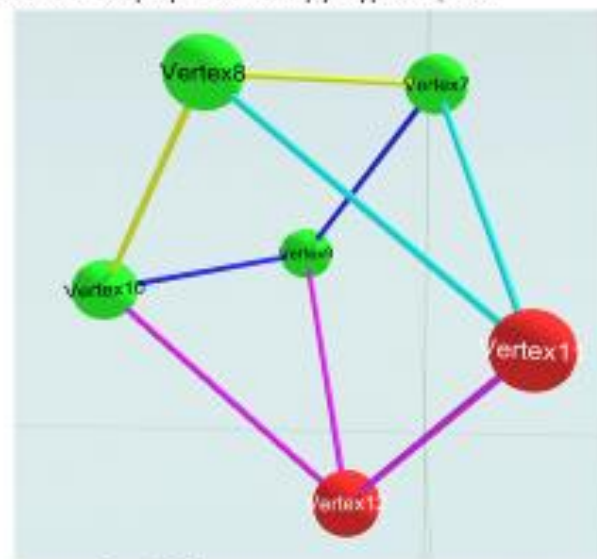
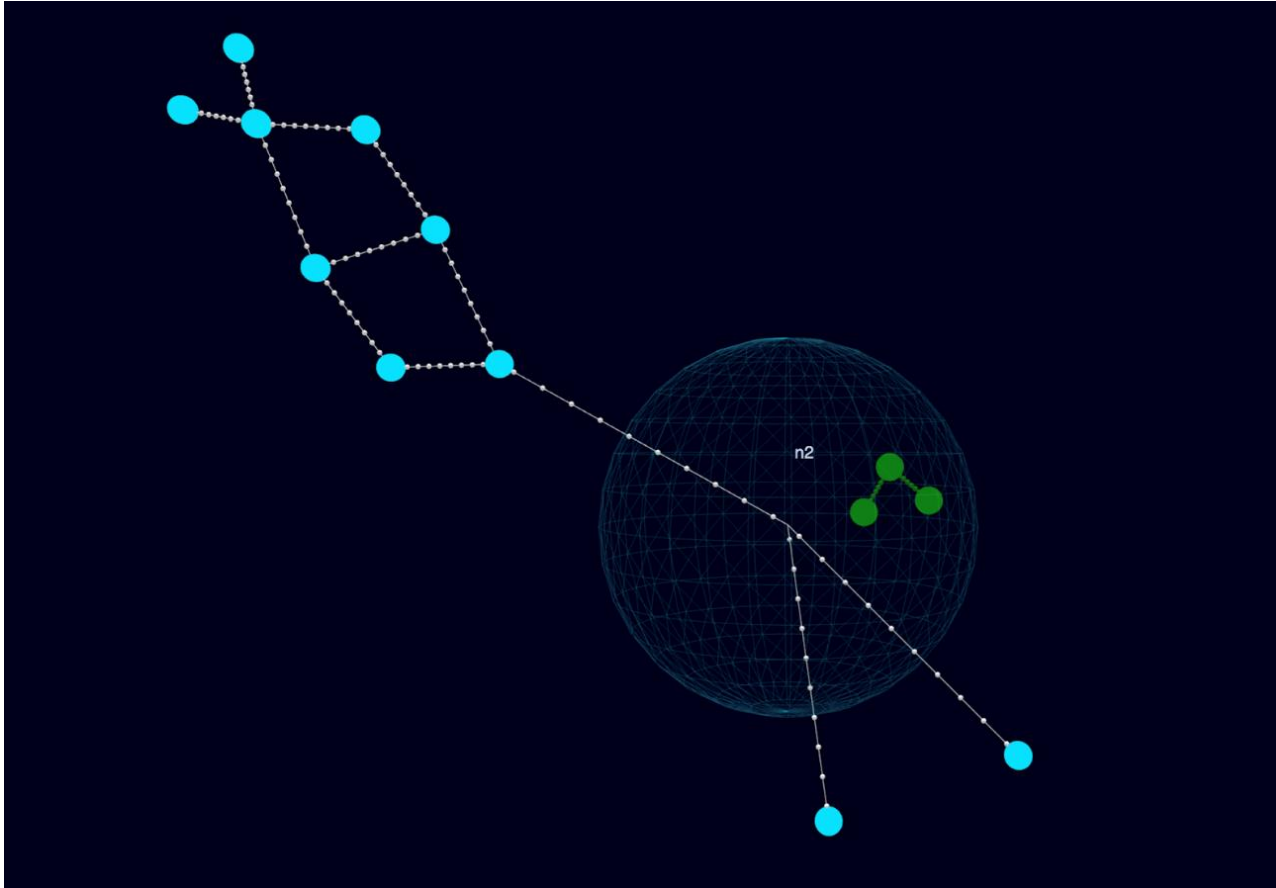


Рис.16. Пример стилизации цвета

Визуализация метаграфов – 2 – 1



- <https://github.com/vasturiano/react-force-graph>
- <https://aframe.io/>

Реализация семантической файловой системы (иерархия)

МетаПроцесс

Файл Редактирование Вид Окно

Классика

Связи

Категории

Инструменты

Режим

Страна

Германия

Австрия

композиторы

Бах

Бетховен

Век

17

18

19

жанр

барокко

классика

форма

симфония

багатель

инвенция

соната

токатта

сюита

тональность

C-moll

A-moll

F-moll

D-moll

B-min

инструмент

орган

оркестр

фортепиано

Бах

Бах-германия

Бах-17

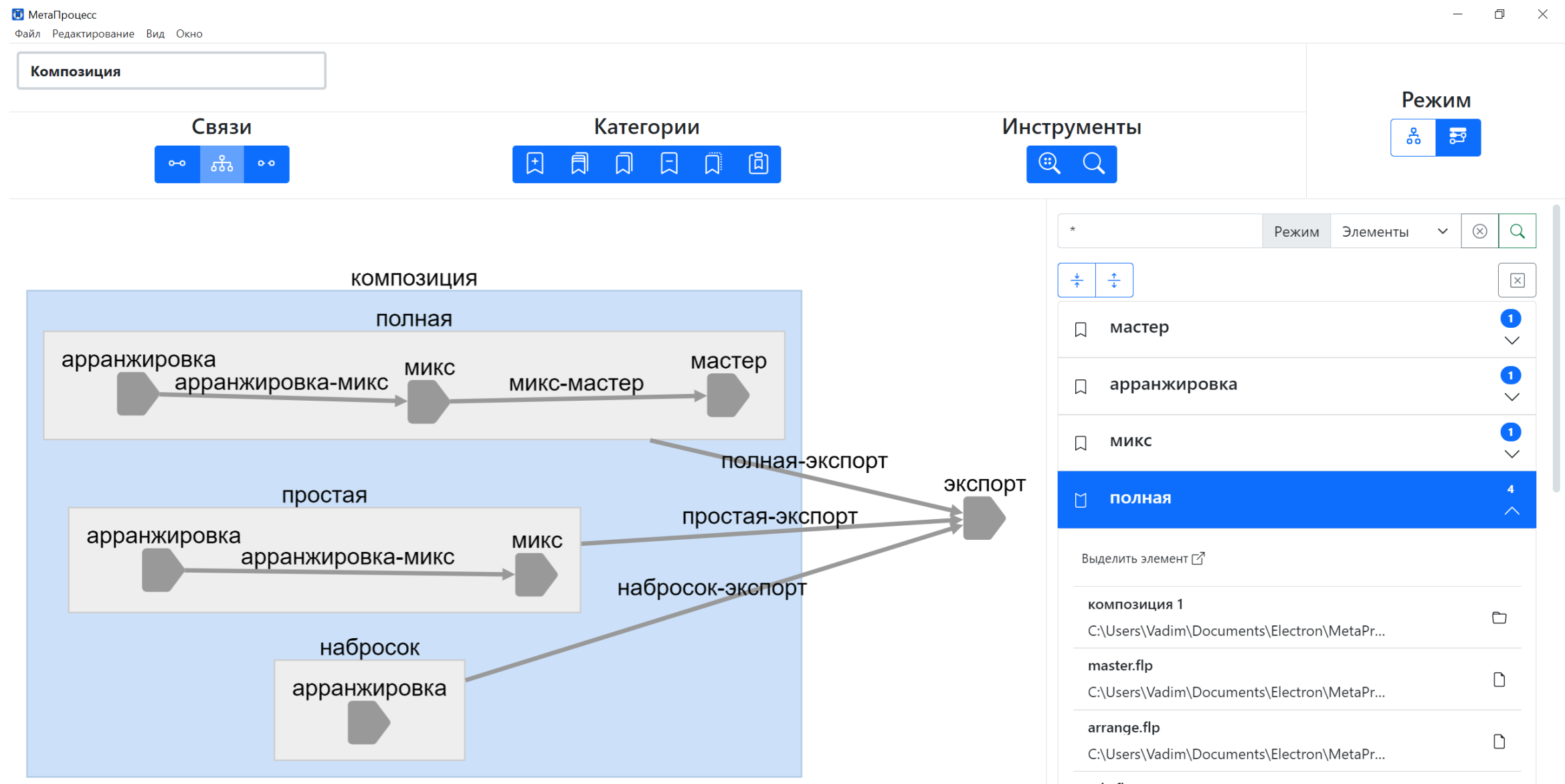
Бах-18

Бах-барокко

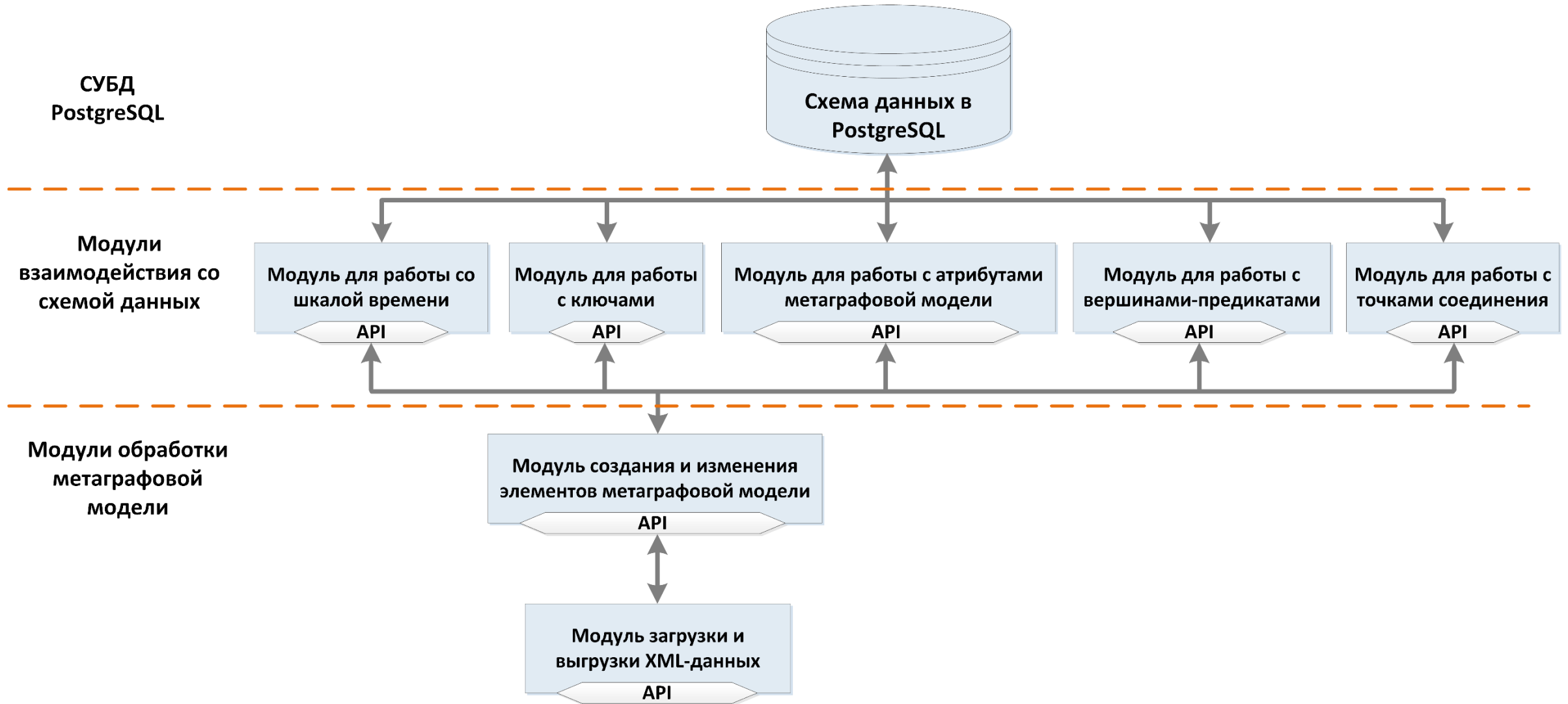
фортепиано

название	полный путь	тип	действия
Inveciya_do_mazh...	C:\Users\Vadim\Docu...	📁	📄
K_JElize.mp3	C:\Users\Vadim\Docu...	📁	📄
Piano_Sonata_No_2...	C:\Users\Vadim\Docu...	📁	📄

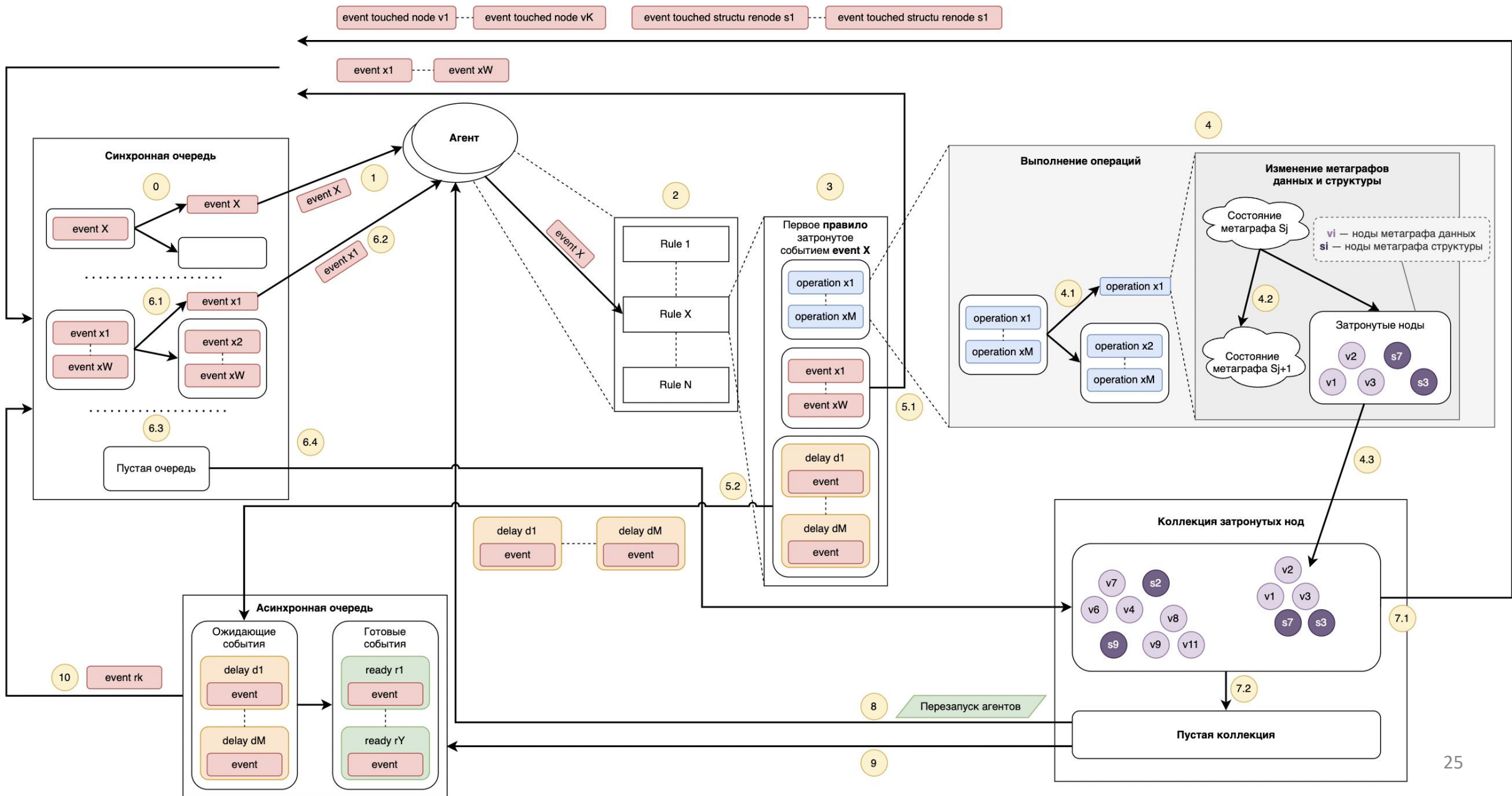
Реализация семантической файловой системы (процессы)



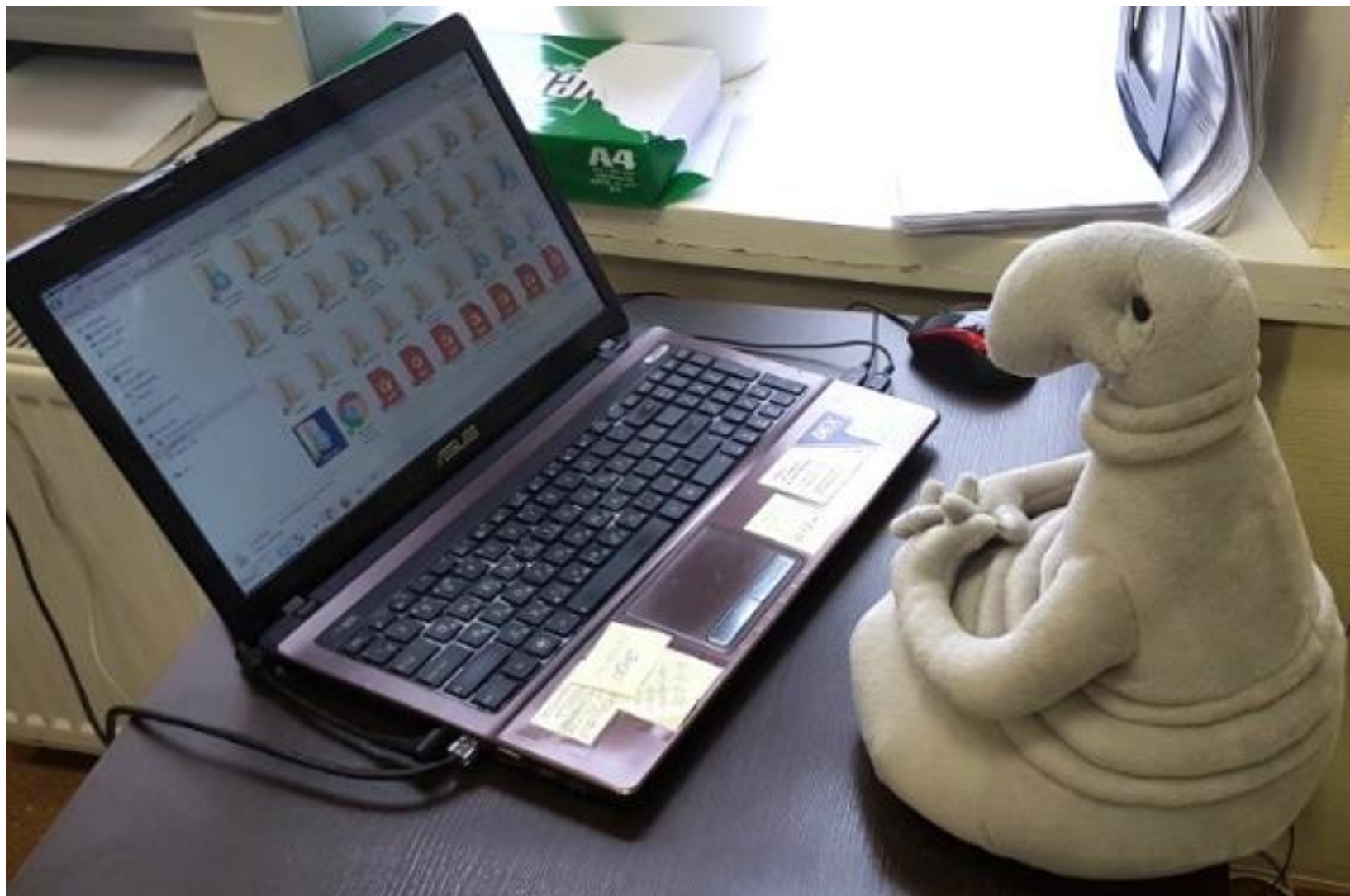
Фрагмент архитектуры системы



Метаграфовый агент (схема работы)



Ждем Вас в проекте



Список литературы

1. Chapela V., Regino Criado, Santiago Moral, Miguel Romance. *Intentional risk management through complex networks analysis*. – Springer, 2015: SpringerBriefs in optimization.
2. Попков В.К. *Математические модели связности*. Новосибирск: ИВ-МуМГ СО РАН, 2006.
3. Johnson J. *Hypernetworks in the science of complex systems*. – London, Hackensack NJ: Imperial College Press, 2013.
4. Basu A., Robert W. Blanning. *Metagraphs and their applications*. – New York: Springer, 2007.
5. Глоба, Л. С. Метаграфы как основа для представления и использования баз нечетких знаний / Л. С. Глоба, М. Ю. Терновой, Е. С. Штогрин // Открытые семантические технологии проектирования интеллектуальных систем = Open Semantic Technologies for Intelligent Systems (OSTIS-2015) : материалы V междунар. науч.-техн. конф. (Минск, 19-21 февраля 2015 года) / редкол. : В. В. Голенков (отв. ред.) [и др.]. – Минск : БГУИР, 2015. – С. 237-240.
6. Астанин С.В., Драгныш Н.В., Жуковская Н.К. Вложенные метаграфы как модели сложных объектов // Инженерный вестник Дона, 2012, №4. URL: ivdon.ru/ru/magazine/archive/n4p2y2012/1434
7. Самохвалов Э.Н., Ревунков Г.И., Гапанюк Ю.Е. Использование метаграфов для описания семантики и прагматики информационных систем. *Вестник МГТУ им. Н.Э. Баумана. Сер. «Приборостроение»*. 2015. Выпуск №1. С. 83-99.
8. Черненький В.М., Терехов В.И., Гапанюк Ю.Е. Представление сложных сетей на основе метаграфов // Нейроинформатика-2016. XVIII Всероссийская научно-техническая конференция. Сборник научных трудов. Ч. 1. М.: НИЯУ МИФИ, 2016. С. 173-178.
9. Черненький В.М., Гапанюк Ю.Е., Ревунков Г.И., Терехов В.И., Каганов Ю.Т. Метаграфовый подход для описания гибридных интеллектуальных информационных систем. *Прикладная информатика*. 2017. № 3 (69). Том 12. С. 57–79.
10. Анохин К.В. Когнитом: гиперсетевая модель мозга // Нейроинформатика-2015.
11. Aleta A., Moreno Y. *Multilayer Networks in a Nutshell*. <https://arxiv.org/pdf/1804.03488.pdf>
12. Boccaletti S., Bianconi G., Criado R., del Genio C. I., Gómez-Gardeñes J., Romance M., Sendiña-Nadal I., Wang Z., Zanin M. The structure and dynamics of multilayer networks // *Physics Reports*. – 2014. – Т. 544. № 1. С. 1–122.