



Московский государственный технический университет им. Н.Э. Баумана

ГИБРИДНЫЕ ИНТЕЛЛЕКТУАЛЬНЫЕ ИНФОРМАЦИОННЫЕ СИСТЕМЫ НА ОСНОВЕ МЕТАГРАФОВОГО ПОДХОДА

Гапанюк Ю.Е., к.т.н., доцент кафедры ИУ-5

План

1. УРОВНИ РАЗВИТИЯ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ
2. ГИБРИДНЫЕ ИНТЕЛЛЕКТУАЛЬНЫЕ ИНФОРМАЦИОННЫЕ СИСТЕМЫ (ГИИС)
3. РЕАЛИЗАЦИЯ ГИИС НА ОСНОВЕ ХОЛОНИЧЕСКОЙ МАС
4. СЛОЖНЫЕ СЕТИ. АНСАМБЛИ СЛОЖНЫХ СЕТЕЙ. СЛОЖНЫЕ СЕТИ С ЭМЕРДЖЕНТНОСТЬЮ.
5. ГИПЕРГРАФЫ.
6. ГИПЕРСЕТИ.
7. МНОГОУРОВНЕВЫЕ СЕТИ.
8. МЕТАГРАФОВАЯ МОДЕЛЬ А. БАЗУ И Р. БЛЭННИНГА
9. МЕТАГРАФОВАЯ МОДЕЛЬ С МЕТАВЕРШИНАМИ
10. ИЕРАРХИЧЕСКАЯ МЕТАГРАФОВАЯ МОДЕЛЬ С МЕТАВЕРШИНАМИ И МЕТАРЕБРАМИ
11. АННОТИРУЕМАЯ МЕТАГРАФОВАЯ МОДЕЛЬ
12. СРАВНЕНИЕ МЕТАГРАФОВОГО ПОДХОДА И RDF
13. ОСНОВНЫЕ ПОДХОДЫ К ХРАНЕНИЮ МЕТАГРАФОВОЙ МОДЕЛИ
14. МУЛЬТИАГЕНТНЫЙ ПОДХОД ДЛЯ ОБРАБОТКИ МЕТАГРАФОВОЙ МОДЕЛИ
15. ФОРМАЛИЗАЦИЯ МНОГОМЕРНОЙ МОДЕЛИ. МНОГОМЕРНО-МЕТАГРАФОВАЯ МОДЕЛЬ
16. ОСНОВНЫЕ ПОЛОЖЕНИЯ И ОПРЕДЕЛЕНИЯ ГРАНУЛЯЦИИ ИНФОРМАЦИИ
17. МЕТАВЕРШИНА КАК БАЗОВАЯ СТРУКТУРА МЕТАГРАФОВОЙ МОДЕЛИ. МЕТАГРАФОВОЕ ИСЧИСЛЕНИЕ
18. ГРАНУЛЯРНОСТЬ МЕТАГРАФОВОЙ МОДЕЛИ
19. ПРОТОГРАФЫ И АРХИГРАФЫ. МЕТАГРАФОВАЯ МОДЕЛЬ НА ОСНОВЕ ТОЧЕК СОЕДИНЕНИЯ
20. ПРИМЕРЫ РАБОТ, В КОТОРЫХ ИСПОЛЬЗУЮТСЯ ГИИС И ПОДХОД НА ОСНОВЕ МЕТАГРАФОВ
21. НАПРАВЛЕНИЯ ИССЛЕДОВАНИЙ И РАЗРАБОТОК

Лекция посвящается светлой памяти дорогих учителей и соавторов

Эдуарда Николаевича Самохвалова



14.04.1936 - 17.06.2017

Валерия Борисовича Тарасова

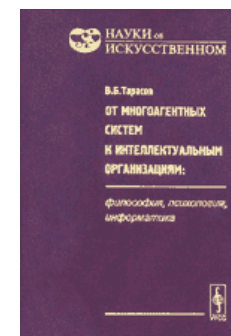


16.02.1955 - 22.07.2021

Валерия Михайловича Чёрненко



13.05.1941 - 07.06.2022



Уровни развития интеллектуальных систем

1. Отдельные модели машинного обучения, отдельные нейросетевые архитектуры.
2. Сложные нейросетевые ансамбли.
3. Когнитивные архитектуры (биологически инспирированные и искусственные, в перспективе сильный ИИ).
 - ГИИС рассматривается как разновидность относительно простой когнитивной архитектуры.



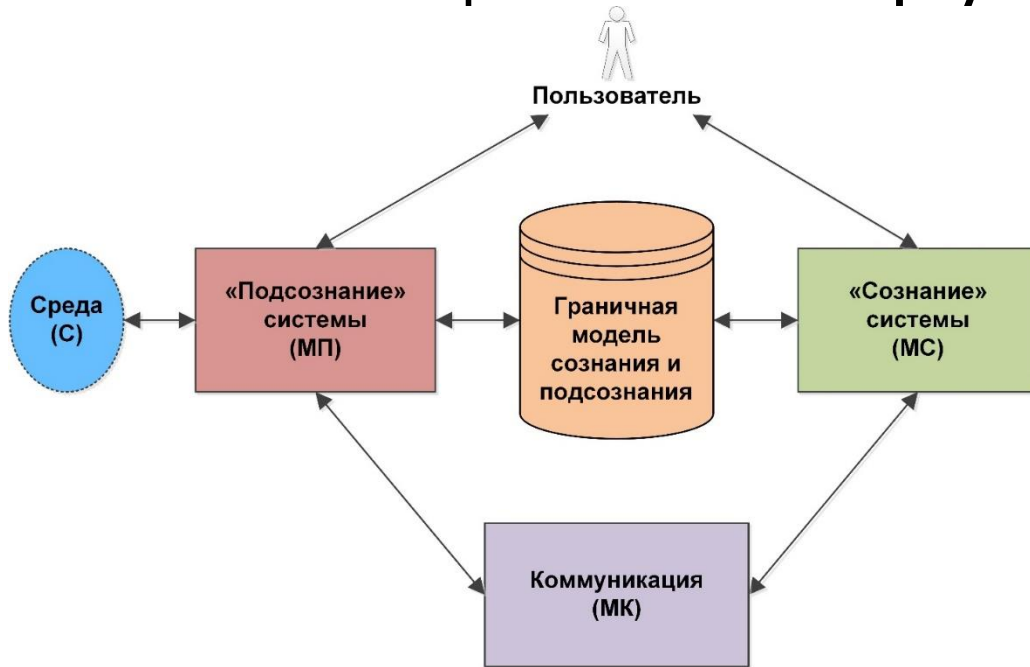
ГИС и ГИИС

- В настоящее время можно отметить явную тенденцию к совместному использованию различных интеллектуальных методов для решения различных классов задач. Это привело к появлению такого направления как **«гибридные интеллектуальные системы» (ГИС)**. Основополагающими работами в области ГИС можно считать работы Александра Васильевича Колесникова.
- В настоящее время интеллектуальные системы, как правило, не разрабатываются отдельно, но встраиваются в виде модулей в традиционные информационные системы для решения задач, связанных с интеллектуальной обработкой данных и знаний. Такую комбинированную систему назовем **гибридной интеллектуальной информационной системой (ГИИС)**.
- ГИИС обладает следующими особенностями:
 - сочетает различные методы, используемые для построения интеллектуальных систем, и в этом смысле является ГИС;
 - сочетает интеллектуальные методы с традиционными методами, используемыми для разработки данных в информационных системах, и в этом смысле является комбинацией ГИС и информационной системы, предназначенной для обработки данных.

Принцип гибридности

- Ключевым вопросом является вопрос о реализации принципа гибридности.
- В работах Надежды Глебовны Ярушкиной сформулирован следующий принцип гибридности: «В литературе встречаются схемы гибридизации нейроинформатики и ИИ, построенные по следующему принципу: **правое полушарие – нейрокомпьютер; левое полушарие – основанная на знаниях система**, а вопрос лишь в их взаимодействии или балансе право- и лево-полушарности. **В реальном поведении человека невозможно разделить восприятие и логическую обработку, поэтому более успешной представляется схема глубинной интеграции**».
- Таким образом, ГИИС должна сочетать элементы системы, построенной на основе мягких вычислений, и системы построенной на обработке данных и знаний.
- Метафора право- и лево-полушарности возможно не совсем точна, скорее стоит говорить о «подсознании» и «сознании» гибридной ИС. «Подсознание» строится на основе мягких вычислений, а «сознание» на основе логической обработки данных и знаний.

Обобщенная структура ГИИС



Основой системы являются «подсознание» системы (модуль подсознания, МП) и «сознание» системы (модуль сознания, МС). «Подсознание» связано со средой, в которой функционирует ГИИС.

Основной задачей МП является обеспечение взаимодействия ГИИС со «средой», или «выживание» ГИИС в среде.

- Поскольку среда может быть представлена в виде набора непрерывных сигналов, то в качестве методов обработки данных «подсознания» хорошо подходят методы, основанные на нейронных сетях и нечеткой логике, в том числе и комбинированные нейронечеткие методы.
- Модель данных «подсознания» максимально приближена к «понятийной системе» среды, представляет собой набор данных, который позволяет максимально эффективно взаимодействовать со средой. Часть этих данных может не иметь «физического смысла» с точки зрения МС, однако позволяет МП взаимодействовать со средой с нужной производительностью.

«Сознание» ГИИС - обработка

- «Сознание» ГИИС строится на принципах обработки данных и знаний. Обработка данных в МС может вестись на основе традиционных языков программирования или технологии workflow. Однако, в последнее время, все большую популярность приобретает подход на основе продукционных правил (rule-based programming).
- В настоящее время появляются гибридные продукты, в частности система Drools, которая позволяет проводить обработку как с использованием workflow-подхода, так и с использованием rule-based-подхода.
- Отметим, что в зависимости от особенностей предметной области правила могут быть нечеткими или вероятностными, что вносит в МС элементы МП. Это одно из проявлений принципа холоничности.
- К достоинствам подхода на основе правил можно отнести гибкость, так как в этом случае программа не кодируется жестко, а «выводится» из правил на основе данных. К недостаткам можно отнести возможность зацикливания правил, а также сложность обработки большого объема правил. В настоящее время для обработки большого объема правил используется алгоритм RETE (разработанный Ч. Форджи) и его модификации.

«Сознание» ГИИС – модель данных

- В качестве модели данных МС используются модели «онтологического» класса. Это могут быть классические онтологии, разработанные в рамках технологии Semantic Web (стандарты RDF, RDFa, OWL, OWL2).
- Также к моделям этого класса можно отнести (возможно, с некоторыми ограничениями) и классическую объектно-ориентированную модель. Классическая модель ООП обладает рядом ограничений по сравнению с онтологиями Semantic Web, но на практике именно она используется для моделирования предметных областей в большинстве современных информационных систем. С использованием средств объектно-реляционного отображения (Object-Relational Mapping, ORM) обеспечивается хранение элементов этой модели в реляционных СУБД.
- Как правило, большинство моделей «онтологического» класса обладает следующими свойствами:
 - явное выделение «абстрактных» понятий (классов) и «конкретных» понятий (объектов, экземпляров);
 - возможность работы как с абстрактными понятиями (например, наследование классов) так и с конкретными понятиями (например, создание объекта класса);
 - возможность работы как с непрерывными типами данных (целые, действительные числа), так и возможность перечисления объектов, относящихся к классу (перечисляемый тип в ООП).

«Сознание» ГИИС - функции

МС, базируясь на моделях «онтологического» класса, выполняет следующие функции:

- обработка данных и знаний на основе модели данных «онтологического» типа;
- логический контроль и проверка непротиворечивости данных, поступающих от МП;
- реализация функций ввода и вывода для среды (посредством МП), для модуля коммуникации и для взаимодействия с пользователем.
- реализация функции поддержки принятия решений (в этом случае МС выполняет функцию СППР);
- реализация функции планирование действий системы (автоматизированное планирование).

Граничная модель сознания и подсознания

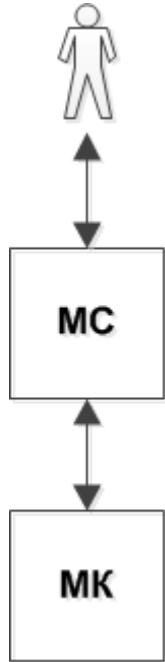
- Граничная модель сознания и подсознания предназначена для глубинной интеграции модулей сознания и подсознания и представляет собой интерфейс между этими модулями с функцией хранения данных.
- В качестве данных выступает комплексная онтология, которая используется как сознанием, так и подсознанием.
- Основной задачей подсознания является распознавание из среды элементов онтологии. Модуль подсознания воспринимает понятийную систему в виде отдельных (возможно несвязанных) признаков. Требования к «осознанию» целостности модели не предъявляется. Основным критерием является эффективность взаимодействия системы со средой.
- Если рассматривать сознание как разновидность экспертной системы, то распознанные элементы онтологии могут рассматриваться в качестве элементов операционной памяти экспертной системы, которые приводят к срабатыванию соответствующих правил. В зависимости от целей системы, правила могут формировать выходную информацию для пользователя или сигналы для модуля подсознания, которые оказывает требуемое воздействие на среду.

ГИИС - коммуникация

С точки зрения коммуникации в ГИИС возможны следующие варианты или их комбинации:

- Коммуникация осуществляется через среду. МП читает данные из среды, преобразует и передает в МС. МС осуществляет логическую обработку и возвращает результаты обработки в МП. МП записывает результирующие данные в среду, откуда они могут быть прочитаны другими ГИИС.
- Для коммуникации с другими ГИИС используется модуль коммуникации (МК). В зависимости от решаемых задач с МК может взаимодействовать МС (что характерно для традиционных информационных систем) или МП (что более характерно для систем на основе мягких вычислений).
- Взаимодействие с пользователем также может осуществляться через МС (что характерно для традиционных информационных систем) или через МП (что может быть использовано, например, в автоматизированных тренажерах).

Частные случаи структуры ГИИС - 1



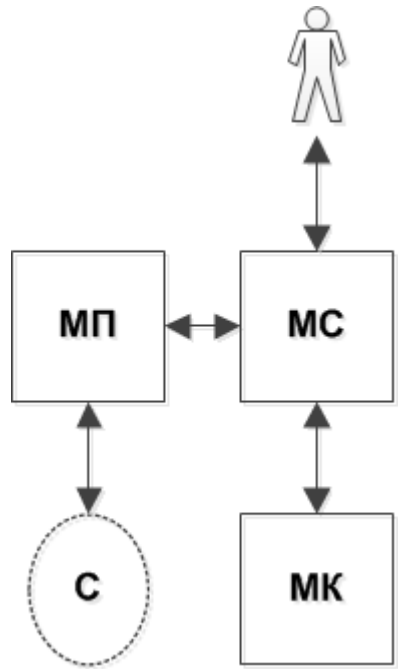
Классическая информационная система, в которой осуществляется только обработка данных и знаний (которую выполняет МС), реализуется коммуникация с другими системами (которую выполняет МК) и взаимодействие с пользователем.

Частные случаи структуры ГИИС - 2



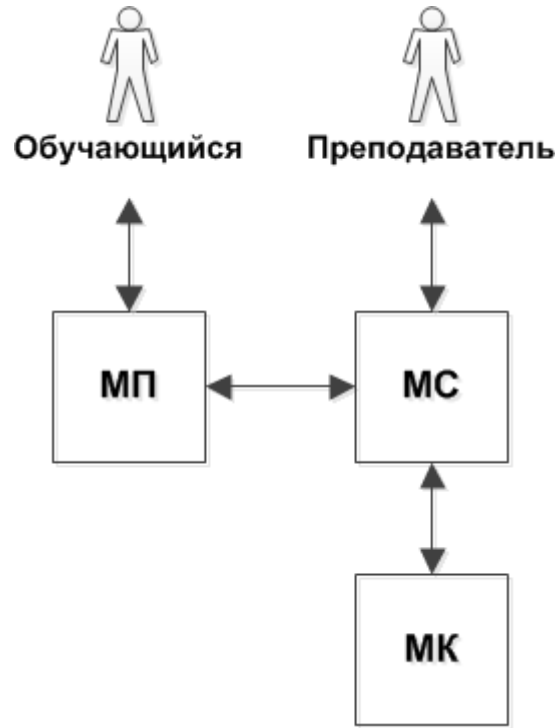
Простейшая система распознавания сигналов, поступающих из среды, с помощью МП. Сигналы могут иметь различную природу. Это может быть система распознавания музыкальной партитуры по звуковому сигналу, система распознавания элементов в видеопотоке и др. Данная система является простейшей, так как в ней отсутствует МС, который должен осуществлять коррекцию логических ошибок. Здесь эта задача возлагается на МП, что может приводить к сложным правилам при распознавании и обработке сигналов.

Частные случаи структуры ГИИС - 3



- Усовершенствованная система распознавания сигналов. Сигналы выделяются из среды с помощью МП и преобразуются в элементы онтологии, которые обрабатывает МС. МС осуществляет дополнительный логический контроль. Например, для системы распознавания музыкальной партитуры, МП выделяет ноты из входного сигнала (здесь ноты являются элементами онтологии), а МС может скорректировать неверно распознанную ноту на основе правил музыкальной гармонии. В этом случае модуль коммуникации может не использоваться.
- Медицинская система функциональной диагностики. В этом случае роль среды выполняют сигналы от медицинских приборов. МП преобразует сигналы в элементы онтологии, МС на основе продукционных правил может осуществлять поддержку принятия решений. Пользователем является врач, модуль коммуникации может осуществлять коммуникацию с другими информационными системами.
- АСУТП (автоматизированная система управления технологическими процессами), использующая методы мягких вычислений. В этом случае роль среды выполняют наблюдаемые параметры технологического процесса. МП преобразует сигналы в элементы онтологии, МС на основе продукционных правил может осуществлять логический контроль поступающей информации и поддержку принятия решений. Пользователем является оператор АСУТП, модуль коммуникации может осуществлять коммуникацию с другими информационными системами.

Частные случаи структуры ГИИС - 4



Автоматизированная система виртуального тренажера. В этом случае действия обучающегося по управлению тренажером поступают на МП. МП преобразует сигналы в элементы онтологии, МС на основе продукционных правил может осуществлять логический контроль поступающей информации, поддержку принятия решений и выдачу информации преподавателю. Модуль коммуникации может быть использован в случае группы тренажеров.

Реализация на основе холонической МАС - 1

- Под программным агентом будем понимать программный модуль, который выполняется в виде автономной задачи (не зависит от других агентов), способен обмениваться информацией со средой и другими агентами. Под МАС будем понимать систему однородных или разнородных агентов, функционирующих в среде.
- Для реализации ГИИС наиболее интересным представляется подход на основе холонической многоагентной системы (холонической МАС). Такой класс систем рассмотрен в работах В.Б.Тарасова. В соответствии с определением холон – это «целое, рассматриваемое в то же время как часть целого».
- С точки зрения данного подхода, рассмотренные компоненты, такие как МП, МС, МК являются агентами. В тоже время они являются частями системы, которая в свою очередь является агентом.
- При этом МП является сложной структурой, которая включает агенты нижнего уровня, каждый из которых может в свою очередь включать МП, МС, МК, предназначенные для решения конкретных задач данного агента. Не смотря на то, что агент нижнего уровня находится в составе МП, он может включать в свою структуру МС, предназначенный для решения задач МП более высокого уровня. Поэтому с точки зрения данного подхода нет ничего удивительного в том, что в МС могут использоваться нечеткие продукционные правила, а в МП входят «классические» модули обработки данных.

Реализация на основе холонической МАС - 2

- Хотя для решения задач МС могут быть использованы методы обработки правил, а для решения задач МП нейронечеткие методы, все эти методы являются статическими. То есть предполагается, что логические правила, структура нейросети и т.д. задаются на этапе проектирования ГИИС и не изменяются в процессе работы.
- Однако, подобный статический подход является недостаточным по следующим причинам:
 - нет возможности использования эволюционных методов (генетические алгоритмы, генетическое программирование и др.);
 - в настоящее время для разработки ГИС используются самоорганизующиеся нейронные сети (в частности такие топологии как SOINN, hyperNEAT), их использование предполагает динамическое изменение топологии нейронной сети во время работы;
 - нет возможности использования других подходов, связанных с изменением порядка действий, таких как динамические workflow, алгоритмы автоматизированного планирования.

Реализация на основе холонической МАС - 3

- Сформулируем основные требования к холонической МАС, предназначенной для реализации ГИИС:
- **Требование 1.** Агент должен реализовывать правила работы для МП или для МС.
- Агент может быть аналогом программной процедуры, которая вычисляет функцию активации нейрона. Может быть реактивным агентом, который реализует поведение на основе заданных правил. Может быть проактивным агентом, который реализует интеллектуальные алгоритмы планирования действий и взаимодействия с другими агентами.
- **Требование 2.** Агенты должны поддерживать принцип холонической организации. То есть агент может быть построен как структура из агентов нижнего уровня, которые агент считает «элементарными», но которые в свою очередь могут состоять из агентов более низкого уровня.
- **Требование 3.** Для реализации свойства динамичности должна существовать возможность перестройки как структуры связей между агентами, так и внутренней структуры самого агента.
- Для реализации требований используется подход на основе сложных сетей.

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 [3, 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

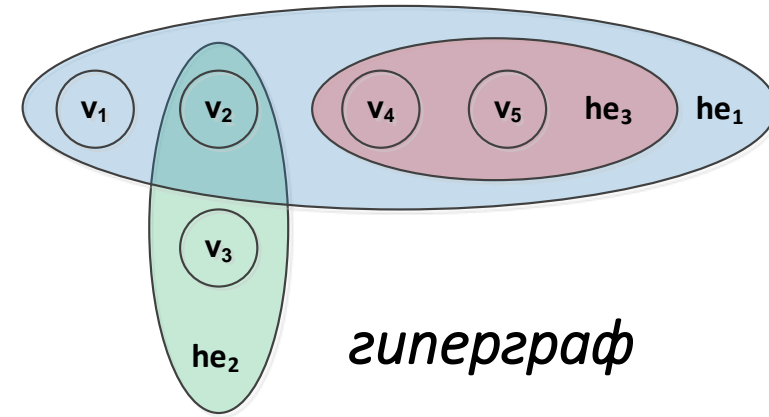
- Плоские графы (сети) очень большой размерности.
- Такие сети могут включать миллионы и более вершин.
- Ребра, соединяющие вершины, могут быть ненаправленными или направленными.
- Иногда используется модель мультиграфа, в этом случае две вершины могут соединяться не одним, а несколькими ребрами.
- Такие 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».



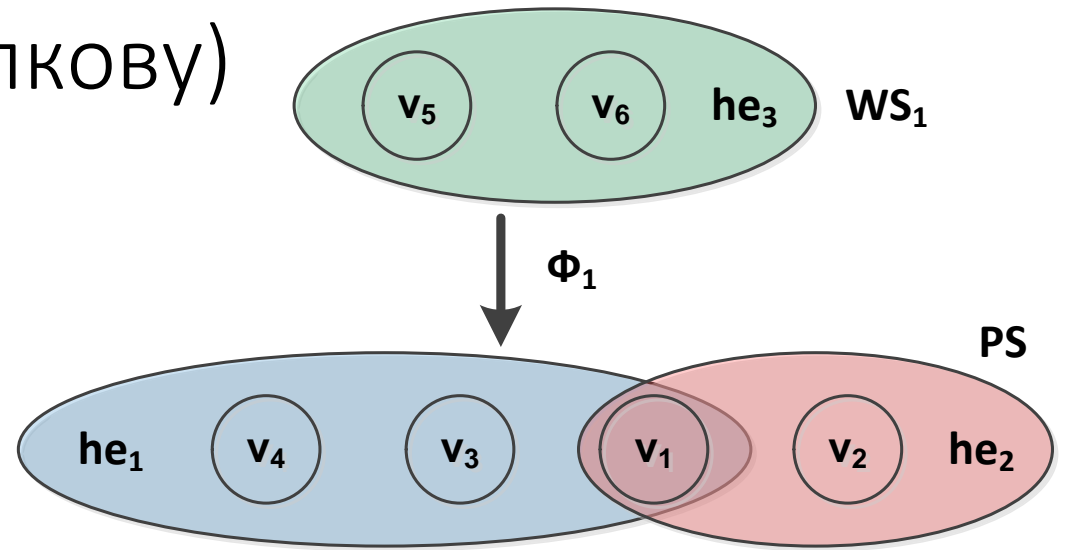
Гиперсетевая модель

- Достаточно типичной является ситуация, когда для семейства схожих моделей сложных сетей используется одинаковое название. Примером является гиперсетевая модель.
- Модель, с одинаковым названием «гиперсетевая», была независимо предложена профессором В.К. Попковым [6] и профессором Дж. Джонсоном [5].
- С одной стороны, концепции предлагаемых вариантов гиперсетевой модели во многом схожи.
- Но, с другой стороны, разница между двумя вариантами моделей, использование различающихся математических аппаратов, и само изложение материала авторами моделей достаточно убедительно свидетельствует об отсутствии возможных заимствований.

Гиперсетевая модель (по В.К. Попкову)

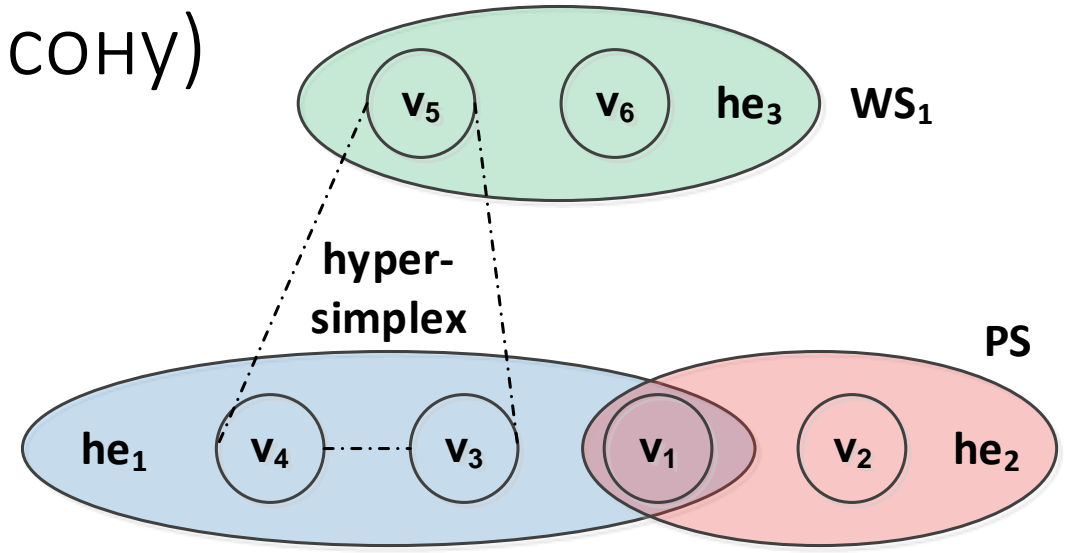
- В первый раз гиперсетевая модель предложена д.ф.м.н. профессором Владимиром Константиновичем Попковым в 1980-х годах.
- Фактически это первая модель «сети с эмерджентностью».
- Пусть даны гиперграфы $PS \equiv WS_0, WS_1, WS_2, \dots, WS_K$
- Гиперграф PS или WS_0 называется первичной сетью. Гиперграф WS_i называется вторичной сетью i -го порядка.
- Также задана последовательность отображений между сетями различных уровней:

$$\{\Phi_i\}: WS_K \xrightarrow{\Phi_K} WS_{K-1} \xrightarrow{\Phi_{K-1}} \dots WS_1 \xrightarrow{\Phi_1} PS$$
- Тогда иерархическая абстрактная гиперсеть порядка K : $AS^K = \langle PS, WS_1, \dots, WS_K; \Phi_1, \dots, \Phi_K \rangle$
- Эмерджентность в гиперсети возникает при переходе между уровнями за счет использования отображений между «слоями» гиперребер.



Гиперсетевая модель (по Дж. Джонсону)

- Во второй раз гиперсетевая модель была предложена профессором Джеффри Джонсоном в его монографии 2013 года.
- Эмерджентность в такой гиперсети возникает при переходе между уровнями за счет возникновения гиперсимплексов. Основание гиперсимплекса содержит множество элементов одного уровня, а его вершина образуется описанием их отношений и приобретает интегральные свойства, делающие ее элементом сети более высокого уровня.
- Профессор Константин Владимирович Анохин считает гиперсетевую модель (в интерпретации Дж. Джонсона) основой своей модели когнитума [7].



Series on Complexity Science - Vol. 3

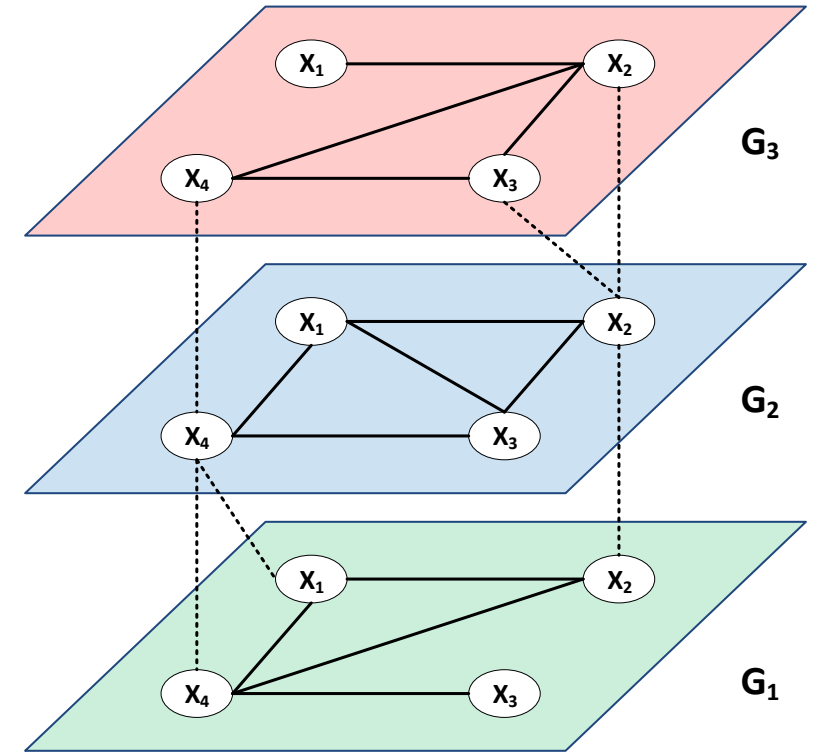
Hypernetworks in the Science
of Complex Systems



Imperial College Press

Многоуровневая сеть

- Пример многоуровневой сети представлен на рисунке.
- Сеть содержит три уровня G_1 , G_2 , G_3 . В данном примере каждый уровень является плоским ненаправленным графом. Ребра графов E_k показаны сплошными линиями. Примеры связей E_{ij} между уровнями показаны пунктирными линиями.
- Сравним модель многоуровневой сети и гиперсетевую модель. Как и гиперсетевая модель, модель многоуровневой сети является послойной.
- Если в гиперсетевой модели на каждом уровне применяются гиперграфы, то в многоуровневой сети уровнем является более простая модель – обычный плоский граф. Связи между уровнями E_{ij} можно рассматривать как частный случай отображения Φ в гиперсети на основе модели В.К. Попкова.
- Таким образом, модель многоуровневой сети можно считать частным упрощенным случаем гиперсетевой модели в интерпретации В.К. Попкова.

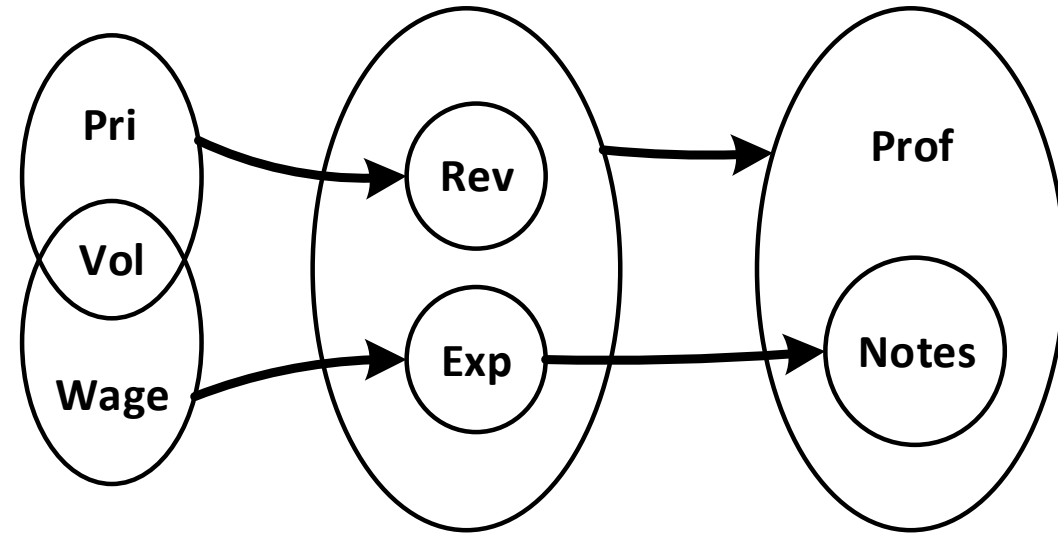


Метаграфовая модель А. Базу и Р. Блэннинга

- Исторически монография А. Базу и Р. Блэннинга [1] была первым источником, в котором появился термин «метаграф». В монографии даются следующие определения, характеризующие метаграфовую модель.
- **Порождающее множество метаграфа** – это множество переменных, встречающихся в ребрах метаграфа: $X = \{x_1, x_2, \dots, x_n\}$.
- **Ребро метаграфа** $e = \langle V_e, W_e \rangle \in E$ (где E – множество ребер) содержит **входную вершину (invertex)** $V_e \subset X$ и **выходную вершину (outvertex)** $W_e \subset X$. Входная и выходная вершины могут содержать произвольное количество элементов. Различные элементы, принадлежащие входной (выходной) вершине, называются соответственно **совходами (совыходами)**.
- Тогда **метаграф** $S = \langle X, E \rangle$ – это графовая конструкция, определяемая порождающим множеством X и множеством ребер E , при этом множество ребер определено на том же порождающем множестве.
- **Простым путем** $h(x, y)$ из элемента x в элемент y это последовательность ребер $\langle e_1, e_2, \dots, e_n \rangle$, такая что:
 - x является входной вершиной e_1 , $x \in invertex(e_1)$;
 - y является выходной вершиной e_n , $y \in outvertex(e_n)$;
 - для всех $e_i, i = 1, \dots, n - 1$ выполняется условие: $outvertex(e_i) \cap invertex(e_{i+1}) \neq \emptyset$, то есть путь из начальной вершины в конечную не прерывается.

Метаграфовая модель А. Базу и Р. Блэннинга (пример)

- На рисунке представлен пример метаграфа, для которого приводится следующая теоретико-множественная интерпретация:



- $S = \langle X, E \rangle$;
 - $X = \{Exp, Notes, Prof, Rev, Pri, Vol, Wage\}$;
 - $E = \langle \{Pri, Vol\}, \{Rev\} \rangle, \langle \{Vol, Wage\}, \{Exp\} \rangle,$
 - $\langle \{Rev, Exp\}, \{Prof, Notes\} \rangle, \langle \{Exp\}, \{Notes\} \rangle$.
- Эмерджентность в модели А. Базу и Р. Блэннинга достигается за счет использования ребер. Понятие метавершины в данной модели отсутствует.
 - Можно отметить, что данный вариант метаграфовой модели более подходит для описания направленных процессов, чем для описания сложных графовых структур данных.
 - В дальнейшем модель получила ряд расширений, которые независимо предлагались различными группами исследователей.

Метаграфовая модель с метавершинами

- Отсутствие естественного механизма для описания сложных графовых структур данных привело к появлению расширений исходной модели А. Базу и Р. Блэннинга. В моделях появились новые элементы – метавершины и метаребра.
- В работе [11] (Л.С. Глоба, М.Ю. Терновой, Е.С. Штогрина) появляется понятие метавершины. В этой работе даются следующие определения метаграфовой модели.
- **Метаграф** – это тройка множеств вершин, метавершин и ребер соответственно: $S = \langle V, M, E \rangle$, где $V = \{v_r\}$ – множество вершин метаграфа (порождающее множество); $M = \{m_q\}$ – множество метавершин метаграфа; $E = \{e_h\}$ – множество ребер метаграфа.
- **Метавершина метаграфа** $m_q = \{v_r | v_r \in V, r = 1, \dots, N_{m_q}\}$ определяется как множество вершин v_r , входящих в метавершину m_q , где N_{m_q} – мощность множества.
- Интересным следует считать следующее замечание авторов модели [11, стр. 238]: «... если две или больше метавершин соответствуют одному и тому же множеству вершин, то такие вершины считаются одинаковыми и рассматривается только одна из таких метавершин». Назовем данное свойство модели [11] **свойством анти-аннотируемости**.
- Интересно, что для задания ребер, авторы модели [11] вводят понятие узла метаграфа $mv \in (V \cup M)$, принадлежащего объединенному множеству вершин и метавершин. Ребро определяется как $e_h = \langle mv_{out}, mv_{in} \rangle$, то есть характеризуется исходящим и входящим узлами метаграфа. Но использование понятия узла для создания иерархических метавершин авторами модели не предлагается.

Иерархическая метаграфовая модель с метавершинами и метаребрами

- В работе [12] (С.В. Астанин, Н.В. Драгныш, Н.К. Жуковская) появляется не только понятие метавершины, но также понятия метаребра и иерархии вершин.
- **Метаграф** в модели [12] определяется как $S = \langle X, X_M, E, E_M \rangle$, где X – множество вершин метаграфа (порождающее множество); X_M – множество метавершин метаграфа; E – множество ребер метаграфа; E_M – множество метаребер метаграфа, заданных на множестве $X_M \cup X$.
- Таким образом, под метаребром в данной модели понимается ребро, которое может соединять вершину и метавершину или две метавершины.
- Важной особенностью данной модели является то, что авторы вводят понятие **вложенного метаграфа**, который является «обобщением обычных графов, гиперграфов и метаграфов» [12].
- В данной модели множество вершин X рассматривается как иерархическое, вводится индекс i , определяющий уровень вложенности вершины.
- Свойство анти-аннотируемости авторами модели не утверждается и не опровергается. При этом, приводимые в статье примеры неявно используют свойство анти-аннотируемости.
- Необходимо отметить, что относительно небольшая по объему работа [12] цитируется в большинстве более поздних статей по тематике метаграфов, что говорит о важности центрального вопроса данной статьи – описания иерархий в метаграфовой модели.

Пример 1 из статьи [12]

Если ребро n -мерного графа является направленным, то граф называется ориентированным n -мерным графом. Вложенные метаграфы являются обобщением обычных графов, гиперграфов и метаграфов. В общем случае, вершины x_2^p являются гиперребрами графов $g_1^p(x_1^p, e_1^p)$, вершины x_2^r являются гиперребрами графов $g_2^p(x_2^p, e_2^p)$, и т.д. Ребра могут связывать вершины любого уровня представления, т.е. как отдельные вершины, так и гиперребра, что характерно для метаграфов. Подобное описание позволяет представлять вложенные структуры, каждая вершина которой может быть устроена по типу «револьверной матрешки» (рис.3).

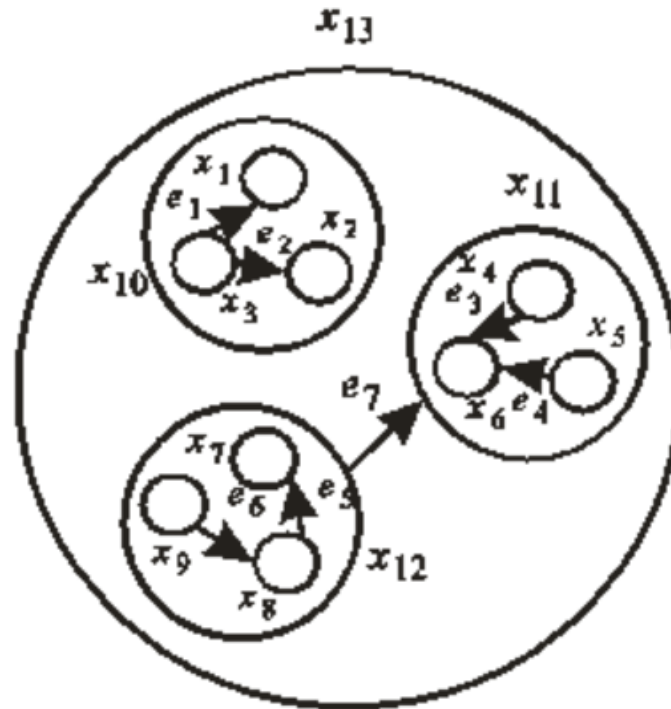


Рис.3.- Пример вложенного метаграфа трехмерной размерности

Пример 2 из статьи [12]

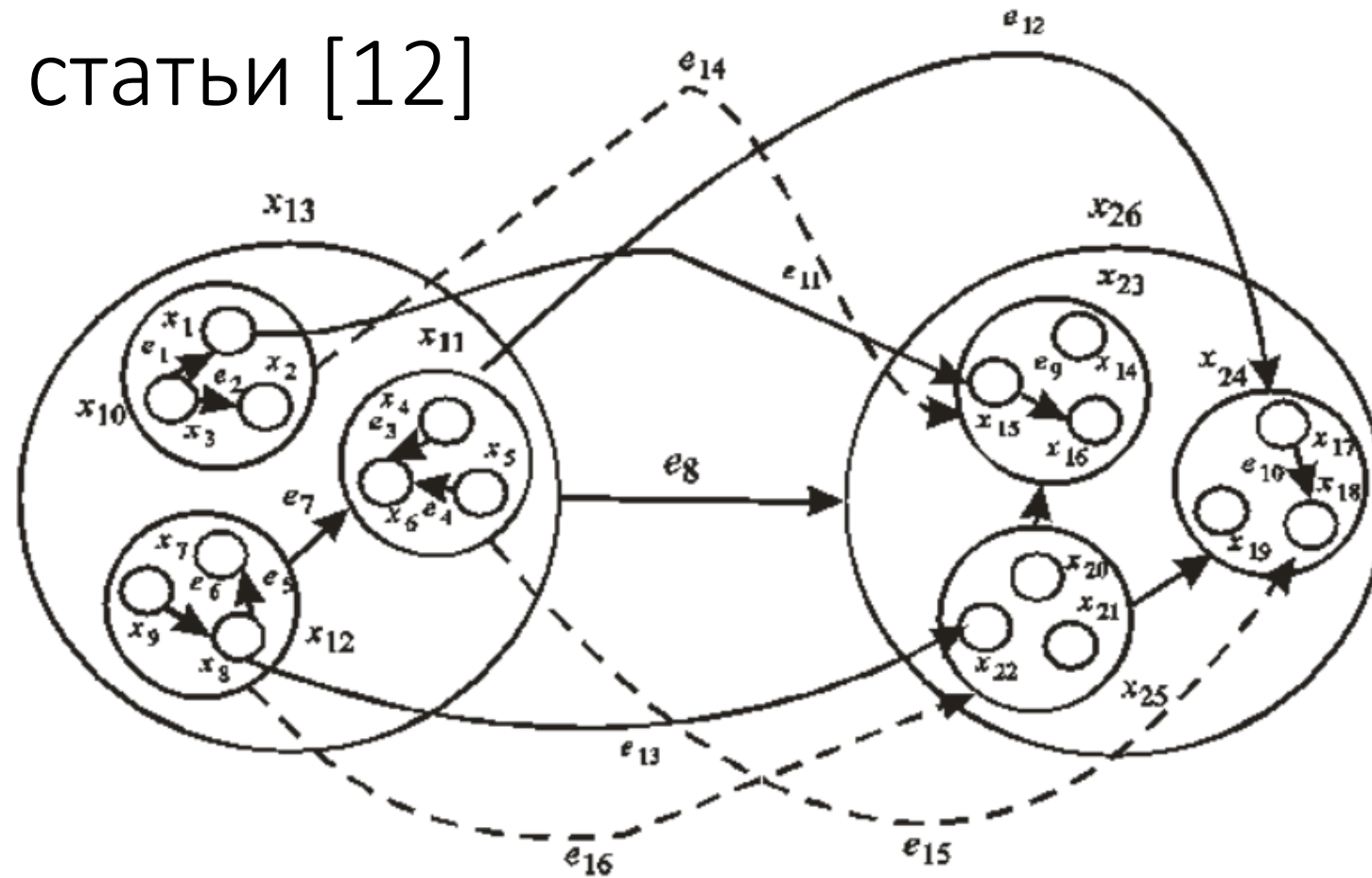


Рис.4.- Фрагмент ситуационной сети бизнес-процесса

Состояния сети x_{13} и x_{26} представлены метаграфами, причем состояние x_{13} является обобщением ситуаций x_{10} , x_{11} и x_{12} , а состояние x_{26} – обобщением ситуаций x_{23} , x_{24} и x_{25} . Каждое метаребро является управляющим воздействием, позволяющим перевести бизнес-процесс из одного состояния в другое состояние. При этом возможен анализ различных уровней в зависимости от текущих обстоятельств. Например, если в момент времени t ожидаемым состоянием является x_{26} , а фиксируется состояние x_{13} при управляющих воздействиях e_{11} , e_{12} , e_{13} , то анализируются причины на других уровнях управления, не позволившие процессу перейти в состояние x_{26} .

Аннотируемая метаграфовая модель

Определим 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:

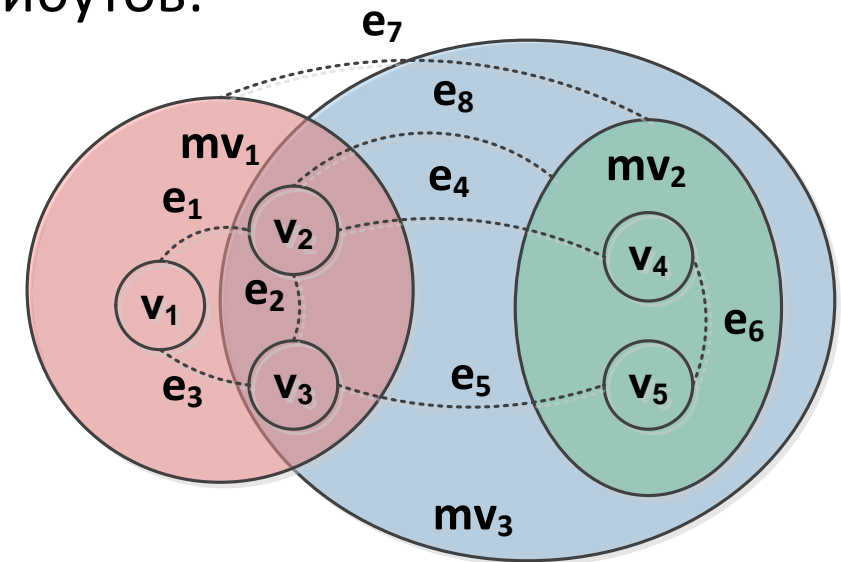
$$v_i = \{atr_k\}, v_i \in V,$$

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:

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

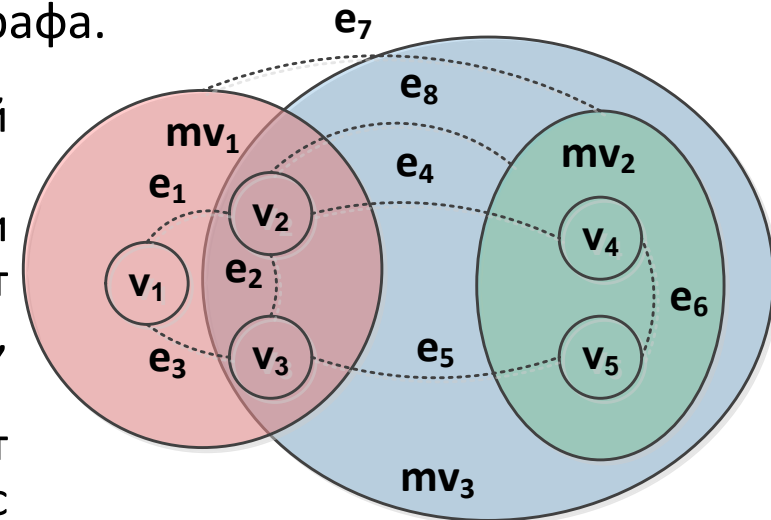
где MG_i – фрагмент метаграфа; ev_j – элемент, принадлежащий объединению множеств вершин, метавершин, ребер и метаребер метаграфа.

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

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

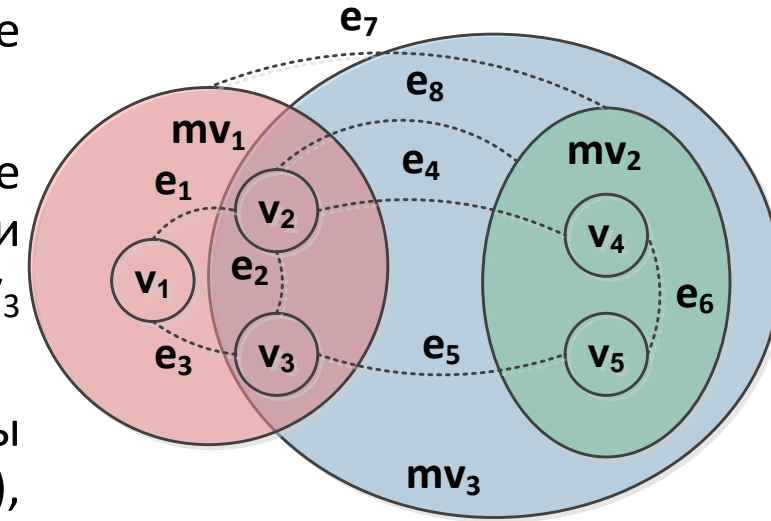
где mv_i – метавершина метаграфа; atr_k – атрибут, MG_i – фрагмент метаграфа.

- 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.



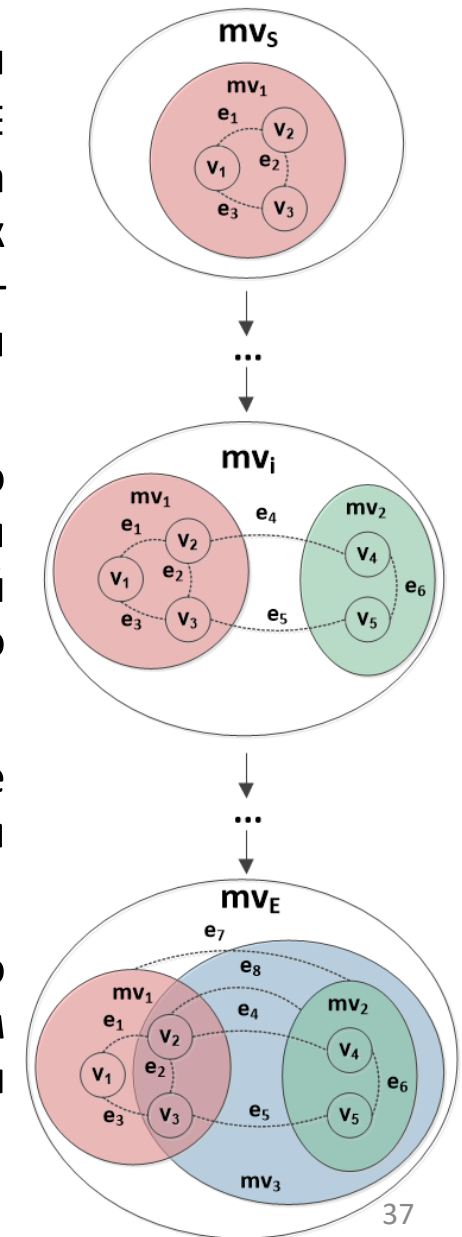
Аннотируемая метаграфовая модель – (пример)

- 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 [11], 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.



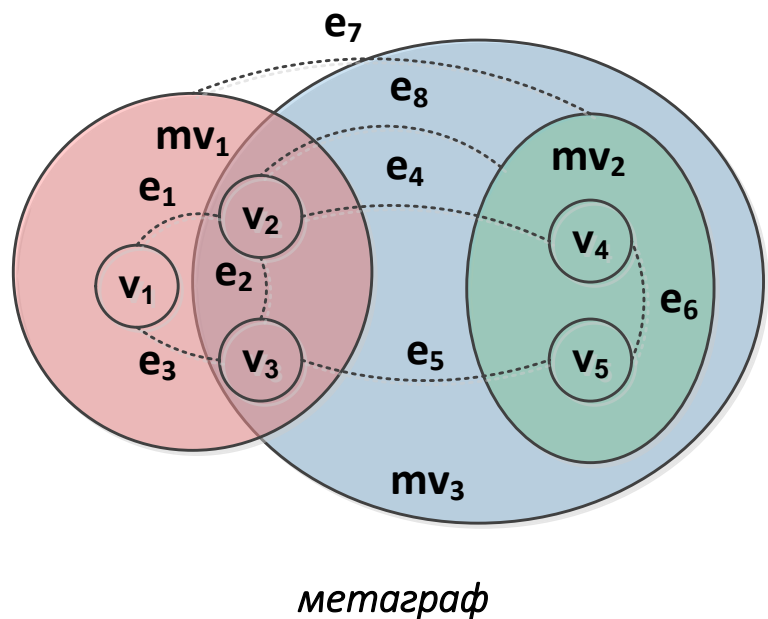
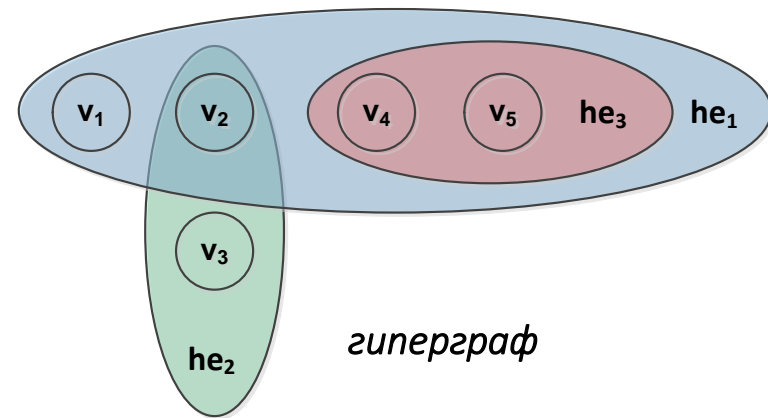
Аннотируемая метаграфовая модель – метаребро

- **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 [12] 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».



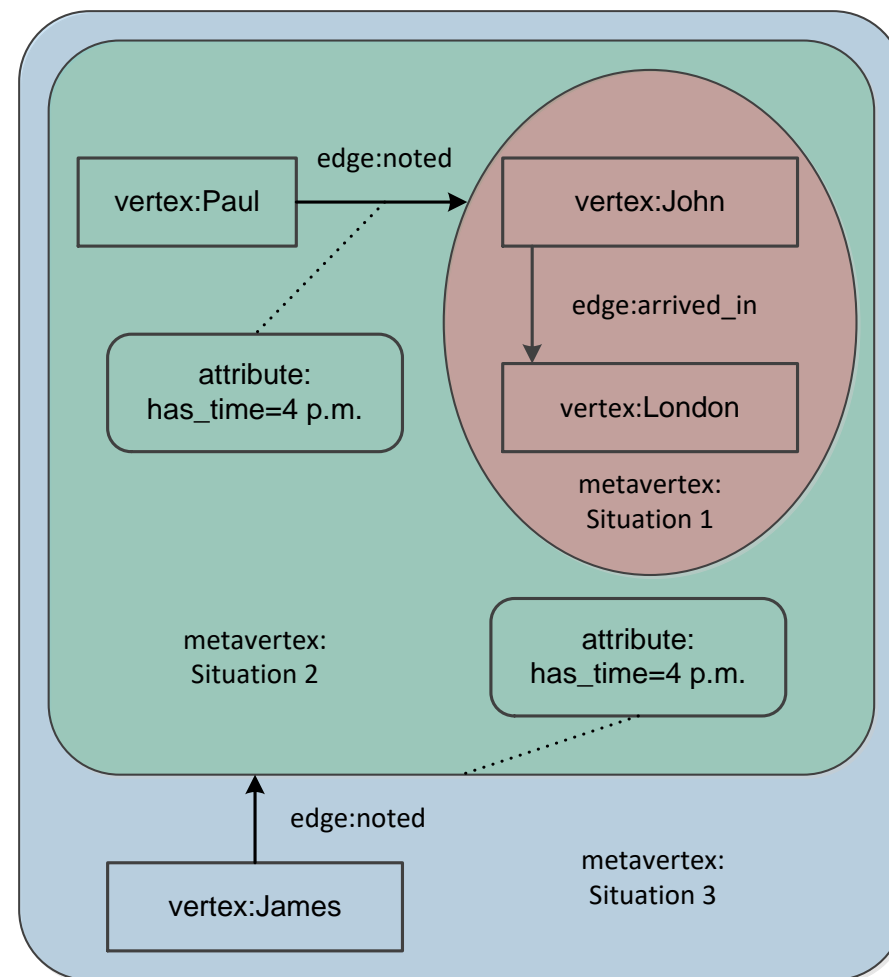
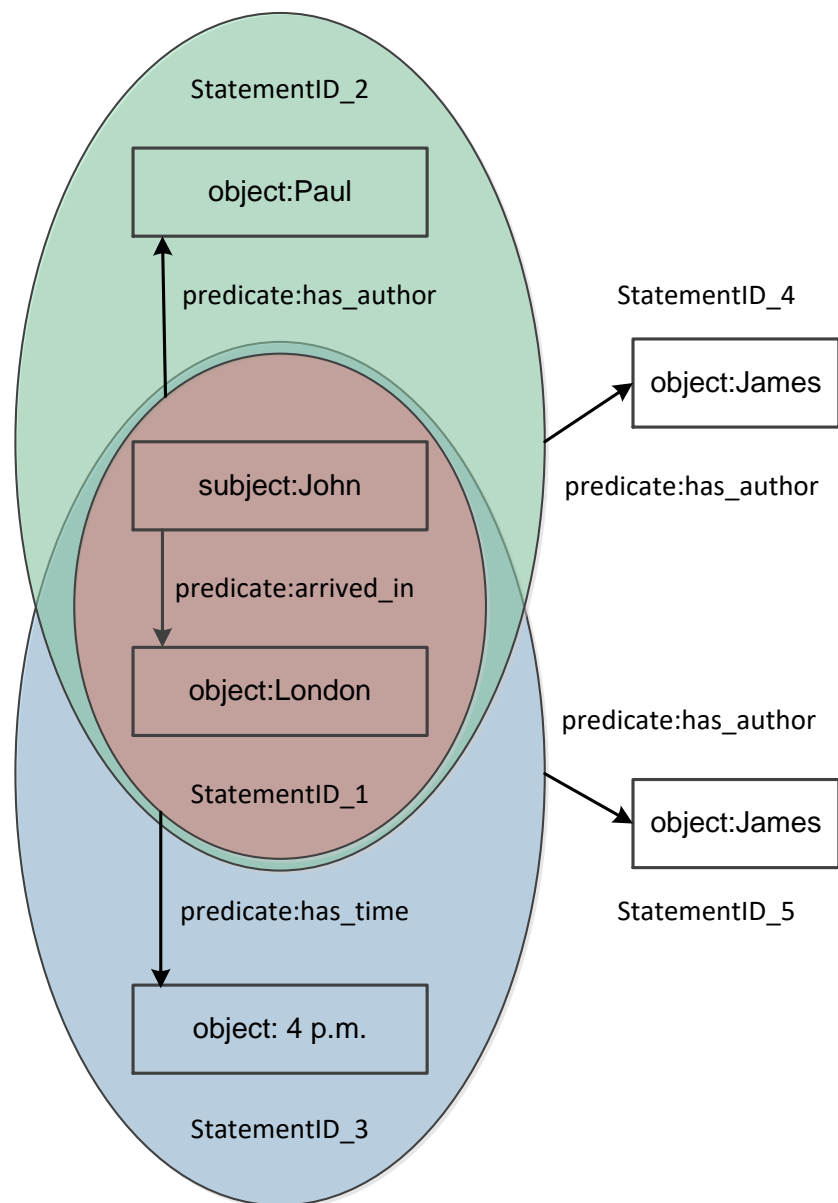
Метаграфы и гиперсети

- В соответствии с определением гиперсеть является «послойным» описанием графов. Предполагается, что слои-гиперграфы идут последовательно и имеют регулярную структуру. Метаграф позволяет с помощью метавершин группировать произвольные элементы, наличие регулярных уровней не обязательно, что делает подход метаграфов более гибким. Фактически, каждый гиперсимплекс может быть представлен отдельной метавершиной.
- Гиперсеть состоит из разнородных элементов (гиперграфов, отображений, гиперсимплексов). Метаграф позволяет с помощью метавершин обеспечивать связь как между элементами одного уровня, так и между элементами различных уровней (при этом, не обязательно соседних). Это делает метаграфовый подход более унифицированным и удобным в описании, так как для описания используются не разнородные структуры (гиперграфы и отображения), а только метавершины (и связи как элементы метавершин). Метаграфовый подход позволяет рассматривать сеть не только в виде «горизонтальных» слоев, но и в виде «вертикальных» колонок.
- Эмерджентность в гиперсети обеспечивается за счет отображений или гиперсимплексов и фактически возникает только при переходе между соседними уровнями. Эмерджентность в метаграфах обеспечивается за счет использования метавершин и может применяться на одном уровне или между уровнями (не обязательно соседними), что делает реализацию эмерджентности в метаграфах более гибкой.
- Необходимо подчеркнуть, что метаграфы и гиперсети являются лишь различными формальными описаниями одних и тех же процессов, которые происходят в «сетях с эмерджентностью». Также необходимо отметить, что настоящее время теория гиперсетей является намного более зрелой по сравнению с теорией метаграфов и именно благодаря теории гиперсетей исследователям удалось понять многие аспекты «сетей с эмерджентностью».

Сравнение метаграфового подхода и [RDF](#)

- Модель RDF широко используется для хранения знаний.
- Но проблема в том, что она плохо подходит для описания сложных вложенных контекстов.
- Рассмотрим пример описания ситуации на естественном языке: «James noted that Paul noted at 4 p.m. that John arrived in London».
- Представление ситуации в виде RDF-триплетов:
 1. *StatementID_1 John arrived_in London*
 2. *StatementID_2 StatementID_1 has_author Paul*
 3. *StatementID_3 StatementID_1 has_time “4p.m.”*
 4. *StatementID_4 StatementID_2 has_author James*
 5. *StatementID_5 StatementID_3 has_author James*

Сравнение метаграфового подхода и RDF



Сравнение метаграфового подхода и RDF

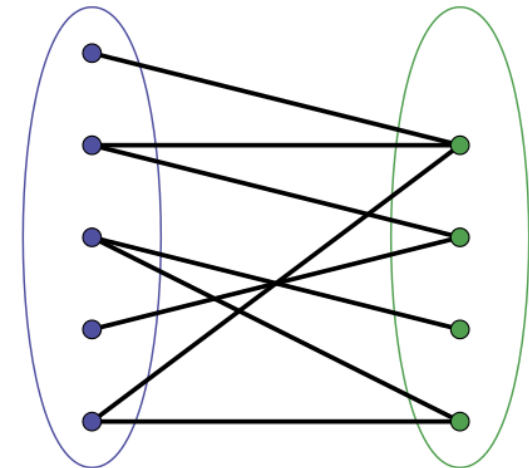
- RDF-триплет является слишком «мелким» элементом, который не позволяет описывать сложные контексты, требуется искусственно вводить вспомогательные триплеты.
- Метаграфовая модель позволяет описывать сложные контексты с использованием метавершин.

Основные подходы к хранению метаграфовой модели

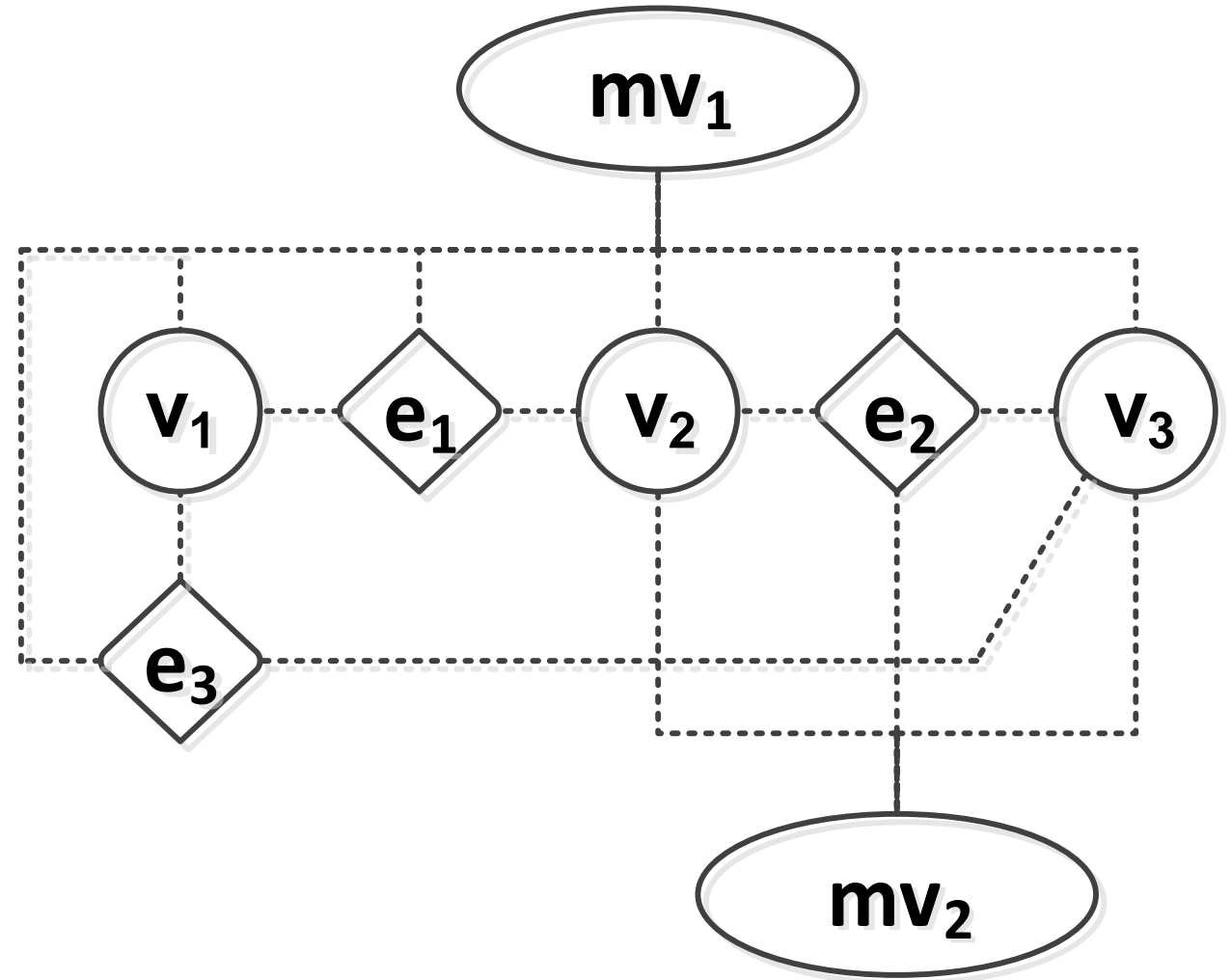
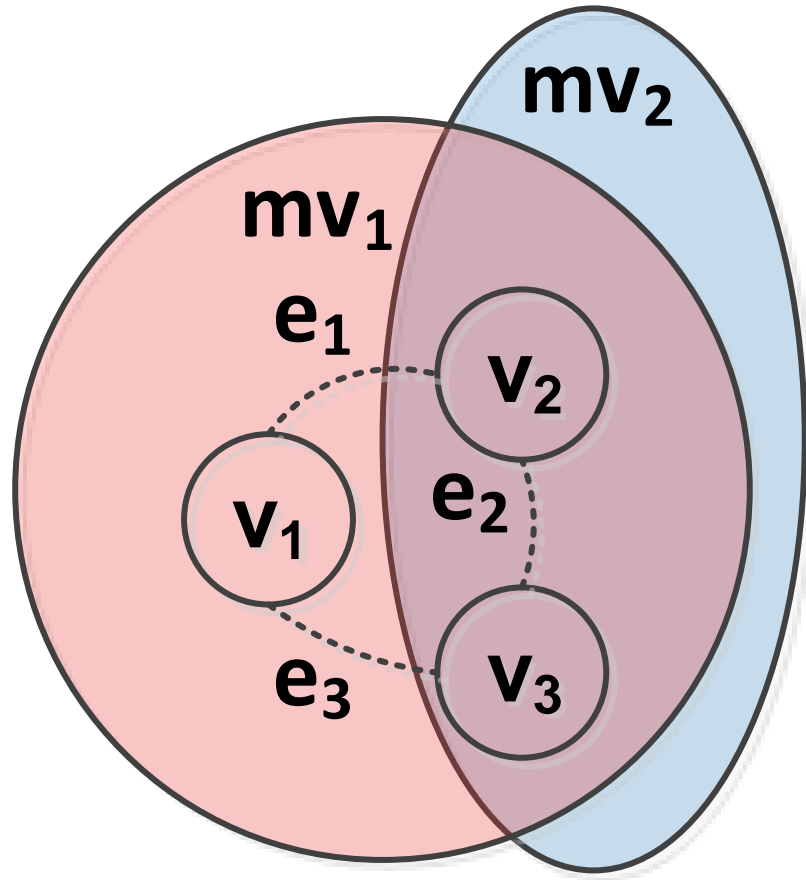
- Метаграфовая модель является аналогом «логической» модели СУБД и ей могут соответствовать различные «физические» модели.
- Мы рассмотрим три варианта «физической» модели:
 1. Модель на основе плоских графов.
 2. Документно-ориентированная модель.
 3. Реляционная модель.

Хранение – плоские графы

- Чтобы превратить холоническую (иерархическую) метаграфовую модель в плоскую графовую модель можно использовать подход на основе многодольных графов.
- Двудольный граф позволяет превратить вершины и ребра графа в подмножества вершин другого графа: $FG = \langle FG^V, FG^E \rangle$, $BFG = \langle BFG^{VERT}, BFG^{EDGE} \rangle$, $BFG^{VERT} = \langle FG^{BV}, FG^{BE} \rangle$, $FG^V \leftrightarrow FG^{BV}$, $FG^E \leftrightarrow FG^{BE}$
- Для метаграфа используется трехдольный граф:
- $TFG = \langle TFG^{VERT}, TFG^{EDGE} \rangle$, $TFG^{VERT} = \langle TFG^V, TFG^E, TFG^{MV} \rangle$,
 $TFG^V \leftrightarrow MG^V$, $TFG^E \leftrightarrow MG^E$, $TFG^{MV} \leftrightarrow MG^{MV}$



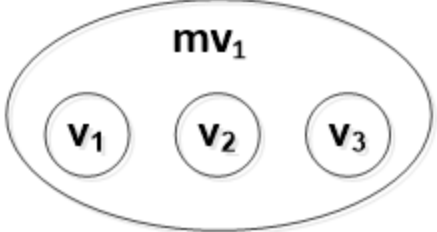


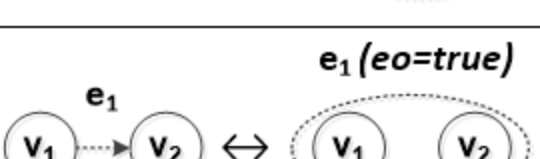
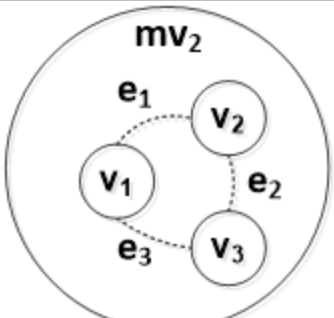
Хранение – плоские графы (пример)

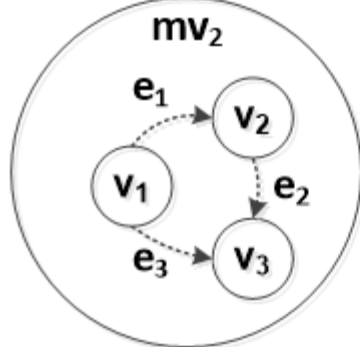
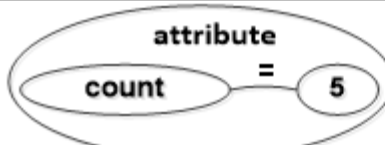
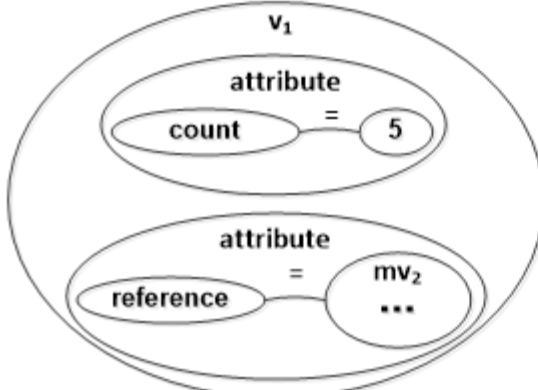


Хранение – документы

- В качестве документоориентированного представления используется Prolog-подобное предикатное описание: *predicate(atom, \dots, key = value, \dots, predicate(\dots), \dots)*.
- Структуры, используемые в данном описании изоморфны структурам, применяемым в JSON-модели: иерархически организованные массивы и пары ключ-значение.
- Основные элементы метаграфовой модели могут быть отображены в предикатное описание:

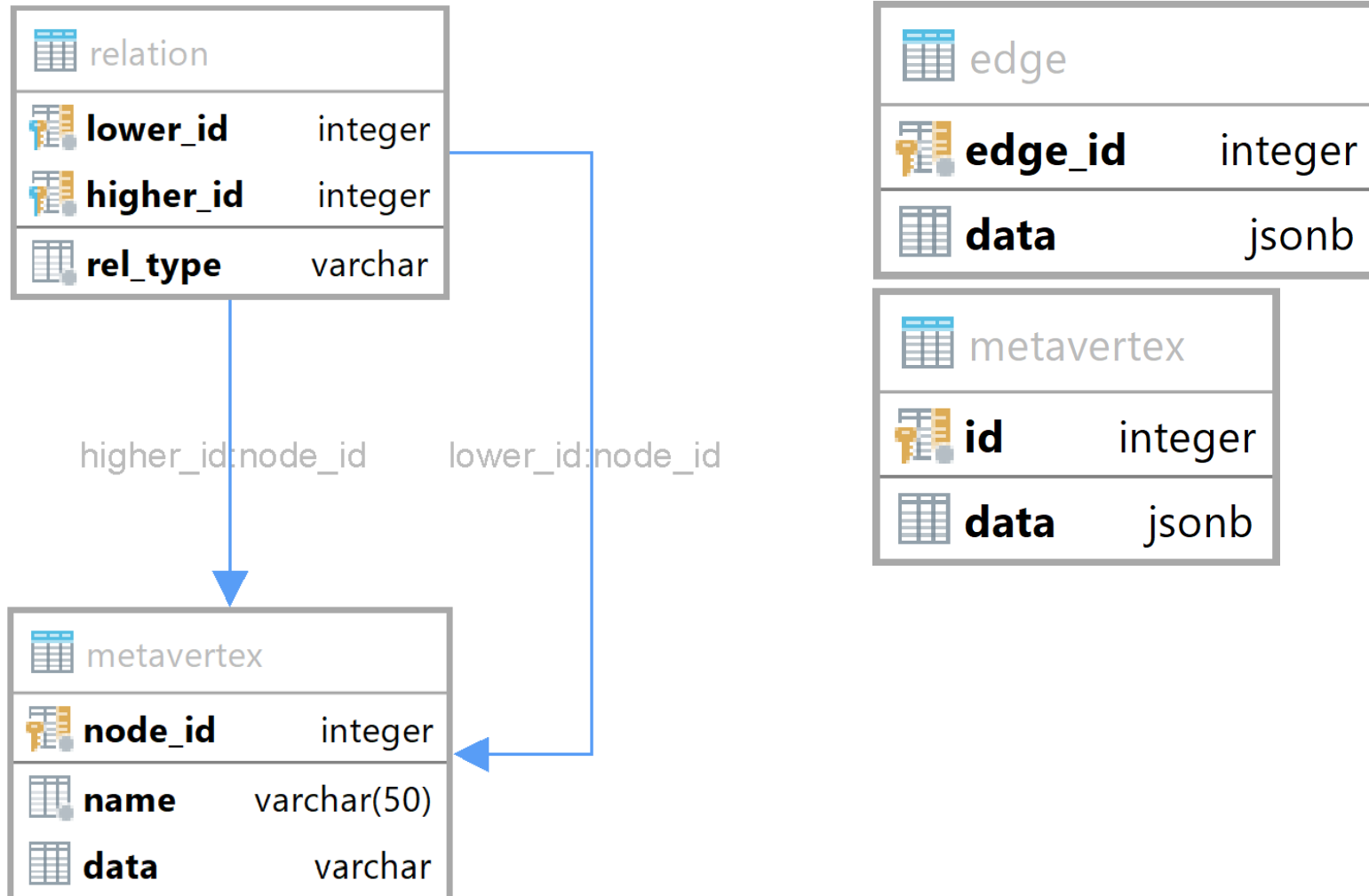
Хранение – документы

	Metavertex(Name=mv ₁ , v ₁ , v ₂ , v ₃)
	Edge(Name=e ₁ , v ₁ , v ₂)
	Edge(Name=e ₁ , v ₁ , v ₂ , eo=false)
	1. Edge(Name=e ₁ , v ₁ , v ₂ , eo=true) 2. Edge(Name=e ₁ , v _S =v ₁ , v _E =v ₂ , eo=true)
	Metavertex(Name=mv ₂ , v ₁ , v ₂ , v ₃ , Edge (Name=e ₁ , v ₁ , v ₂), Edge(Name=e ₂ , v ₂ , v ₃), Edge(Name=e ₃ , v ₁ , v ₃))

	Metavertex(Name=mv ₂ , v ₁ , v ₂ , v ₃ , Edge(Name=e ₁ , v _S =v ₁ , v _E =v ₂ , eo=true), Edge(Name=e ₂ , v _S =v ₂ , v _E =v ₃ , eo=true), Edge(Name=e ₃ , v _S =v ₁ , v _E =v ₃ , eo=true))
	Attribute(count, 5)
	Vertex(Name=v ₁ , Attribute(count, 5), Attribute(reference, mv ₂))

Хранение – реляционная модель

- Использование чистого реляционного подхода или документно-ориентированного хранилища, встроенного в реляционную СУБД.



Выводы (по результатам экспериментов с хранением метаграфовой модели)

- В случае добавления, обновления и удаления данных наиболее эффективным вариантом является использование PostgreSQL. Но в случае выборки иерархических данных графовые СУБД показывают значительно лучшие результаты.
- Для графовых СУБД (как для ArangoDB так и для Neo4j) время выполнения запросов на получение иерархических данных сопоставимо с временем выполнения запросов на добавление, обновление и удаление данных.
- В PostgreSQL добавление, обновление и удаление данных в целом производятся быстрее, чем в графовых СУБД или за сопоставимое время. Но при этом время выполнения запросов на получение иерархических данных во много раз больше времени добавления, обновления и удаления данных.
- В целом наиболее производительным вариантом является использование СУБД ArangoDB с плоской графовой моделью.

Мультиагентный подход 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 того, чтобы объединить метаграфовую модель данных с агентами для ее обработки, введем понятие **активного метаграфа**:

$$MG^{ACTIVE} = \langle MG^D, AG^{MG} \rangle, AG^{MG} = \{ag_i\},$$

- где MG^{ACTIVE} – активный метаграф; MG^D – метаграф данных; AG^{MG} – множество метаграфовых агентов ag_i , связанных с метаграфом данных.

Холоническая МАС для реализации активности - 1

- Как выполнить рассмотренные требования 1, 2, 3?
- **Требование 1.** Агент должен реализовывать правила работы для МП или для МС.
- Для реализации требования 1 предлагается использовать два вида агентов: агент-функцию и метаграфовый агент.

Холоническая МАС для реализации активности - 2

- **Требование 2.** Агенты должны поддерживать принцип холонической организации. То есть агент может быть построен как структура из агентов нижнего уровня, которые агент считает «элементарными», но которые в свою очередь могут состоять из агентов более низкого уровня.
- Для реализации требования 2 предлагается использовать контейнерный агент.
- **Требование 3.** Для реализации свойства динамичности должна существовать возможность перестройки как структуры связей между агентами, так и внутренней структуры самого агента.
- Для реализации требования 3 предлагается использовать динамический 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)

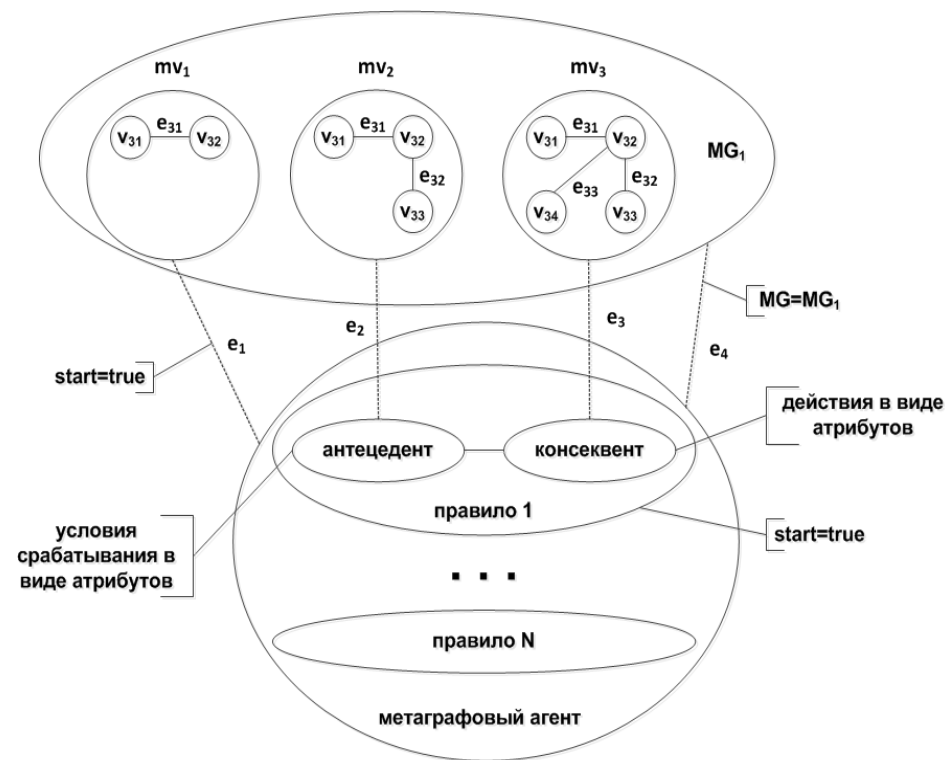
- Метаграфовый агент:

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

- 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:

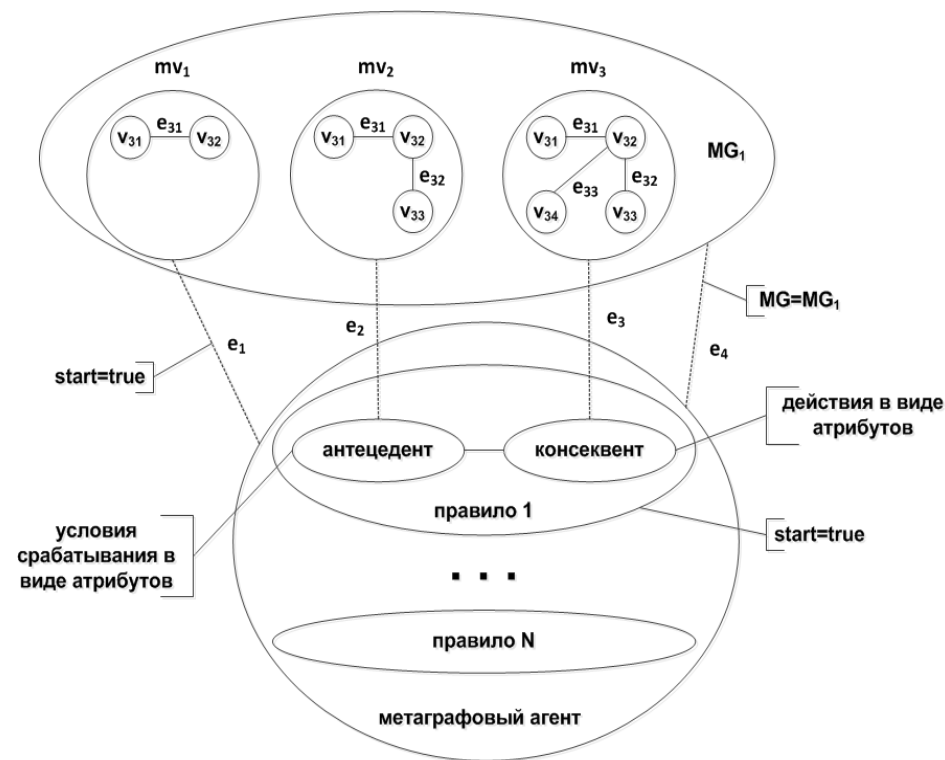
$$r_i : MG_j \rightarrow OP^{MG},$$

- 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.



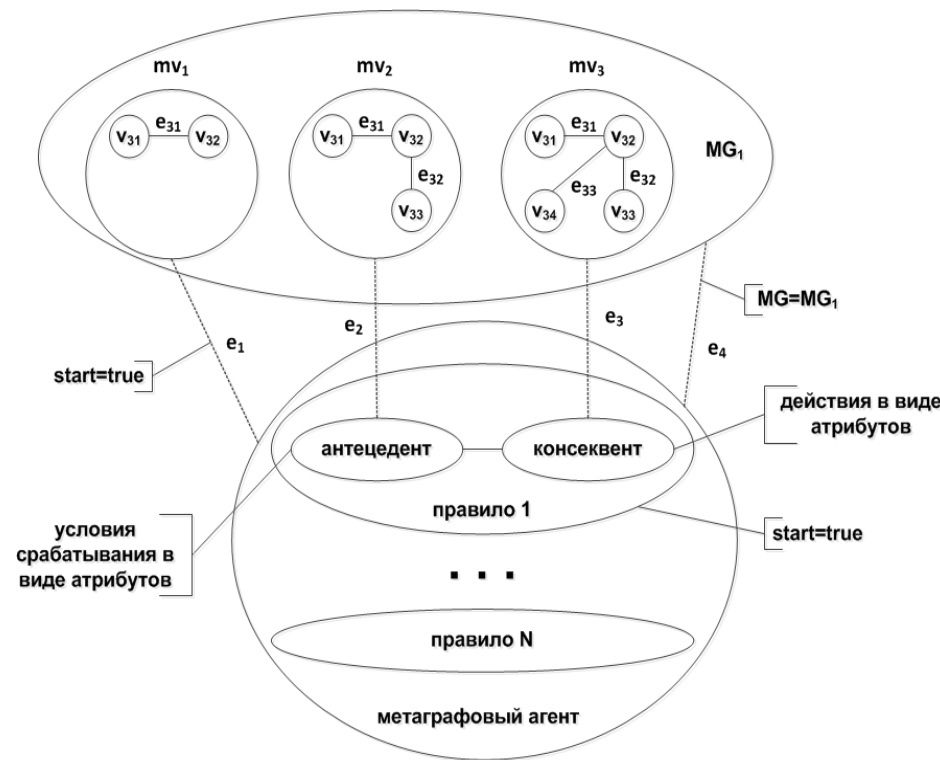
Метаграфовый агент (правила)

- 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 .



Контейнерный агент

- Контейнерный агент:

$$ag^C = MG, v_i \equiv ag_i, v_i \in V, mv_i \equiv ag_i, mv_i \in MV,$$

- где ag^C – контейнерный агент; MG – метаграф; v_i – вершина метаграфа; ag_i – агент; V – множество вершин метаграфа; mv_i – метавершина метаграфа; MV – множество метавершин метаграфа.
- Контейнерный агент, представляет собой метаграф, вершины и метавершины которого являются агентами.

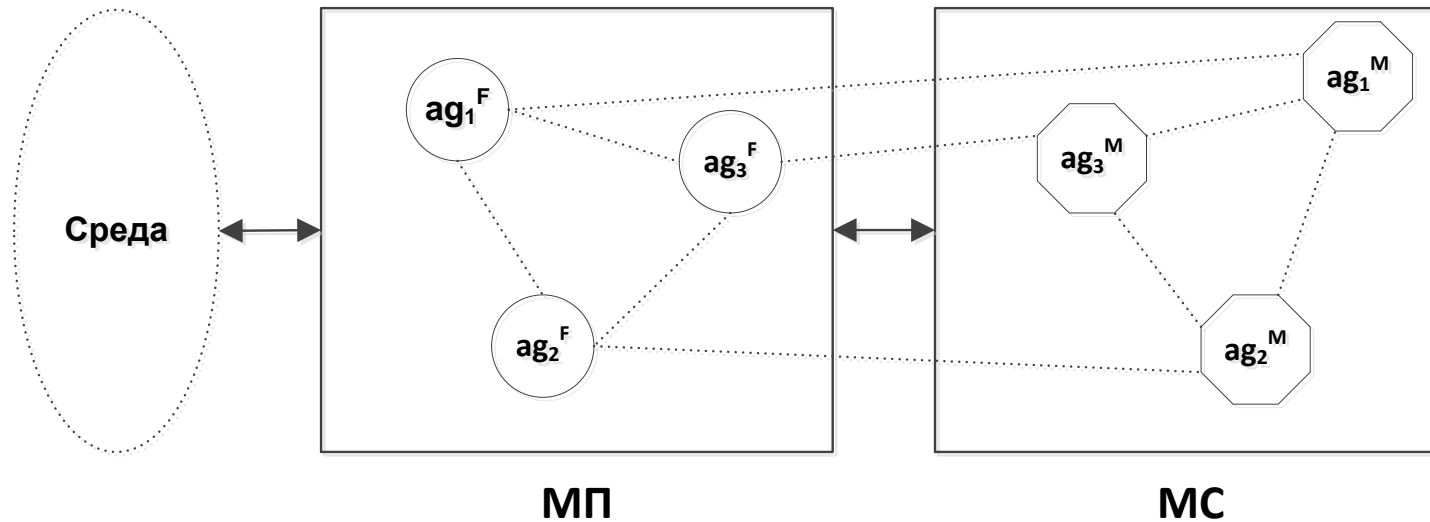
Динамический метаграфовый агент

- Динамический метаграфовый агент:

$$ag^{MD} = \langle (ag^C \cup ag^{MD}), R, AG^{ST} \rangle, R = \{r_j\},$$

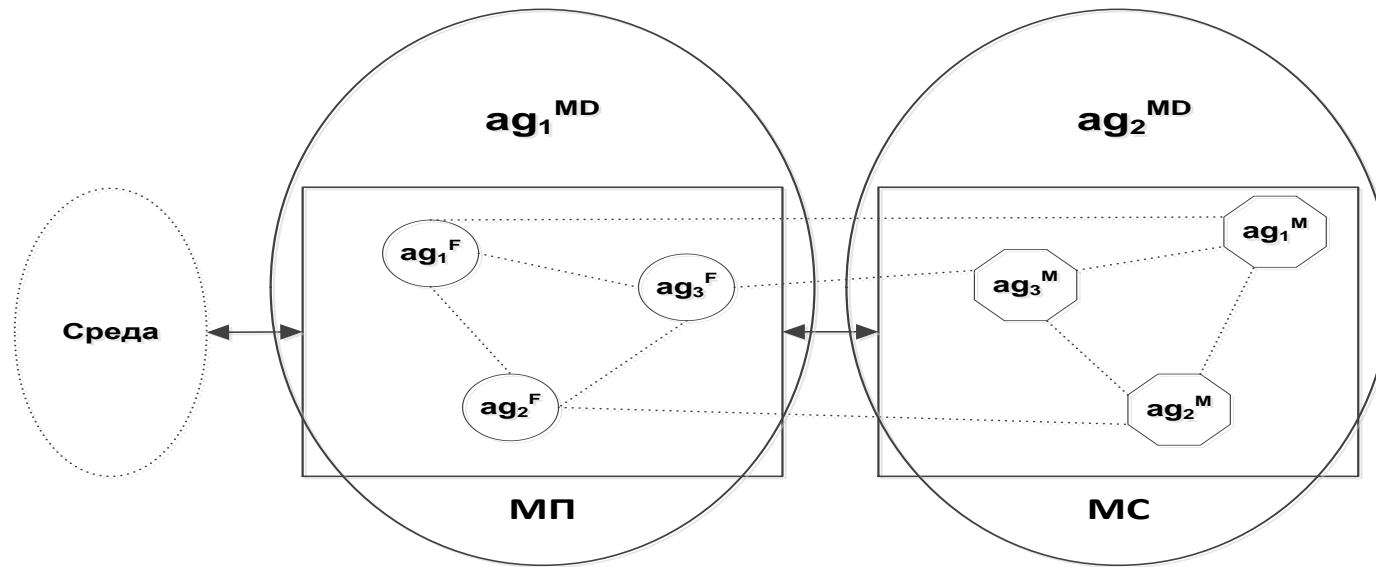
- где ag^{MD} – динамический метаграфовый агент; ag^C – контейнерный агент, на метаграфе которого выполняются правила агента; R – набор правил (множество правил r_j); AG^{ST} – стартовое условие выполнения агента (фрагмент метаграфа, который используется для стартовой проверки правил, или стартовое правило).
- Правила обработки динамического метаграфового агента выполняются не на метаграфе данных и знаний, а на метаграфе агентов для заданного контейнерного агента.
- По определению контейнерный агент включает все рассмотренные ранее виды агентов: агенты-функции и метаграфовые агенты. Поэтому динамический метаграфовый агент может изменять все виды агентов.
- Определение данного агента использует тот факт, что в предлагаемой модели все агенты являются метаграфами, поэтому любые элементы структуры агентов доступны для обработки агентами верхнего уровня. Эта особенность является аналогом свойства «самоотображаемости» в традиционных языках программирования.
- Отметим, что данное определение является рекурсивным. Динамические метаграфовые агенты первого уровня могут обрабатывать статические контейнерные агенты, метаграфовые агенты второго уровня могут обрабатывать метаграфовые агенты первого уровня и так далее. По мере необходимости систему можно надстраивать требуемыми уровнями динамики.
- В зависимости от условий динамический метаграфовый агент может решать следующие задачи:
 - первичное развертывание, создание, системы агентов более низкого уровня;
 - изменение системы нижнего уровня (изменение внутренней структуры агентов, изменение связей между агентами, удаление агентов).

Статическая структура ГИИС



- На рисунке представлена система, МП которой является нейронной сетью, а МС построен на основе обработки правил.
- Агенты-нейроны показаны в виде окружностей, а метаграфовые агенты обработки данных в виде восьмиугольников. МП и МС выполняют роль контейнерных агентов. В данном примере используются одноуровневые контейнеры, однако, возможно использование произвольной вложенности контейнеров.
- Отметим, что вся система холонических агентов представляет собой метаграф, при этом каждый агент также является метаграфом.

Динамическая структура ГИИС



- По сравнению с предыдущим рисунком добавились два динамических метаграфовых агента.
- Агент, отвечающий за МП, может изменять структуру самоорганизующейся нейронной сети. Или может применять эволюционные методы для оптимизации конфигурации нейронной сети.
- Агент, отвечающий за МС, может изменять связи между агентами для решения задачи автоматизированного планирования. Или может применять эволюционные методы для оптимизации конфигурации агентов.
- На рисунке показаны только динамические метаграфовые агенты первого уровня, однако, количество таких уровней не ограничено. Над показанными динамическими метаграфовыми агентами могут быть надстроены динамические метаграфовые агенты более высоких уровней.

Реализация ГИИС

- Классическое понимание

- В качестве методов обработки данных «подсознания» хорошо подходят методы, основанные на машинном обучении, нейронных сетях, нечеткой логике, в том числе и комбинированные нейронечеткие методы. Эти методы основаны на векторном (матричном, тензорном) представлении признаков и их обработке.
- В качестве методов обработки данных «сознания» используются онтологии и правила, экспертные системы.

- Современное понимание

- Методы, основанные на векторном представлении все активнее используются для обработки данных «сознания». Пример – Graph Neural Network (GNN).
- Использование векторного представления можно рассматривать как **«схему глубинной интеграции»** сознания и подсознания.
- Концепции «сознания» и «подсознания» ГИИС не исчезают, но могут быть реализованы унифицированным образом на основе векторного представления признаков.

Многомерная модель - 1

Страна	Товар	Год	Объем продаж
Аргентина	Бытовая электроника	1988	105
Аргентина	Бытовая электроника	1989	117
Аргентина	Бытовая электроника	1990	122
Аргентина	Резиновые изделия	1988	212
Аргентина	Резиновые изделия	1990	217
Бразилия	Бытовая электроника	1988	313
Бразилия	Бытовая электроника	1989	342
Бразилия	Бытовая электроника	1990	337
Бразилия	Резиновые изделия	1988	515
Бразилия	Резиновые изделия	1989	542
Бразилия	Резиновые изделия	1990	566
Венесуэла	Бытовая электроника	1988	94
Венесуэла	Бытовая электроника	1989	96
Венесуэла	Бытовая электроника	1990	102
Венесуэла	Резиновые изделия	1988	153
Венесуэла	Резиновые изделия	1989	147
Венесуэла	Резиновые изделия	1990	162



Многомерная модель - 2

- Общие вопросы многомерной организации метаграфовой информации рассматривались в работах [8, 9], в данной работе полученные ранее результаты систематизируются и объединяются с моделью метаграфовых процессов.
- Классическая многомерная модель данных, предложенная Эдгаром Ф. Коддом, позволяет работать с числовыми данными (мерами), привязывая их к иерархическим таксономиям (измерениям) [10].
- Многомерная модель данных является ядром информационных систем OLAP (online analytical processing). Было предложено множество вариантов формализации такой модели, но в данном разделе мы используем нашу собственную упрощенную версию формализации в соответствии с [8], которая поможет описать предлагаемую многомерно-метаграфовую модель.
- Многомерный гиперкуб (как классический, так и метаграфовый) может быть описан следующим образом: $HC = \langle MSR, HCD, HCF, HCR, HCP \rangle$, $MSR = \{msr_i\}$, $HCD = \{hcd_i\}$, $HCF = \{hcf_i\}$, $HCR = \{hcr_i\}$, $HCP = \{hcp_i\}$ где HC – гиперкуб; MSR – множество мер (msr_i – мера); HCD – множество измерений (hcd_i – измерение); HCF – множество фактов (hcf_i – факт); HCR – множество правил агрегации (hcr_i – правило); HCP – множество метаграфовых процессов (hcp_i – процесс), данный компонент присутствует только в многомерно-метаграфовой модели (и отсутствует в классической).

Многомерная модель - 3

- Для классической многомерной модели определения могут быть раскрыты следующим образом:
- Определение измерения гиперкуба: $hcd_i = \langle \{hcd_i^k\}, < \rangle$, где hcd_i^k – элемент измерения гиперкуба; $<$ – отношение частичного порядка на множестве элементов измерения гиперкуба.
- В большинстве случаев элементы измерения гиперкуба организованы в виде иерархической структуры, в случае измерения времени, например, *год* \rightarrow *месяц* \rightarrow *неделя* \rightarrow *день*. Но организация с частичным порядком более правильна, чем иерархическая организация, потому что организация с частичным порядком позволяет описывать неоднородные иерархии, в случае измерения времени, например, иерархии *месяц* \rightarrow *неделя* \rightarrow *день* и *месяц* \rightarrow *декада* \rightarrow *день* могут существовать одновременно в одном измерении.

Многомерная модель - 4

- Определение факта: $hcf_i = \langle \{hcd_i^{ref}\}, \{msr_n\} \rangle$, где hcd_i^{ref} – ссылка на элемент измерения гиперкуба; msr_n – значение меры.
- Определение правила агрегации:
- $hcr_k: \{hcf_{OUT}\} = agf(\{hcf_{IN}\}, HCD^{ag}), HCD^{ag} \subset HCD$,
- где hcf_{OUT} – выходные (агрегированные) факты; agf – функция агрегации; hcf_{IN} – входные (неагрегированные) факты; HCD^{ag} – подмножество измерений гиперкуба, принимающее участие в агрегации.
- Типичными функциями агрегации являются определение суммы, среднего, минимального и максимального значений меры.

Многомерно-метаграфовая модель - 1

- На основе формализации для классической многомерной модели предложено описание многомерно-метаграфовой модели.
- Мерой в предлагаемой модели является фрагмент метаграфа: $msr_i \equiv MG_j$ где msr_i – мера; MG_j – фрагмент метаграфа.
- Определение измерения : $hcd_i = \langle \{hcd_i^k\}, < \rangle, hcd_i \in MV, hcd_i^k \in (V \cup MV)$, измерение может быть представлено в виде метавершины, а элемент измерения гиперкуба в виде вложенной вершины или метавершины. Таким образом, измерения гиперкуба могут быть представлены в виде иерархически организованных метавершин. Элементы измерения гиперкуба, соответствующие листьям дерева, могут быть представлены в виде вершин, а элементы более высоких уровней – в виде метавершин.

Многомерно-метаграфовая модель - 2

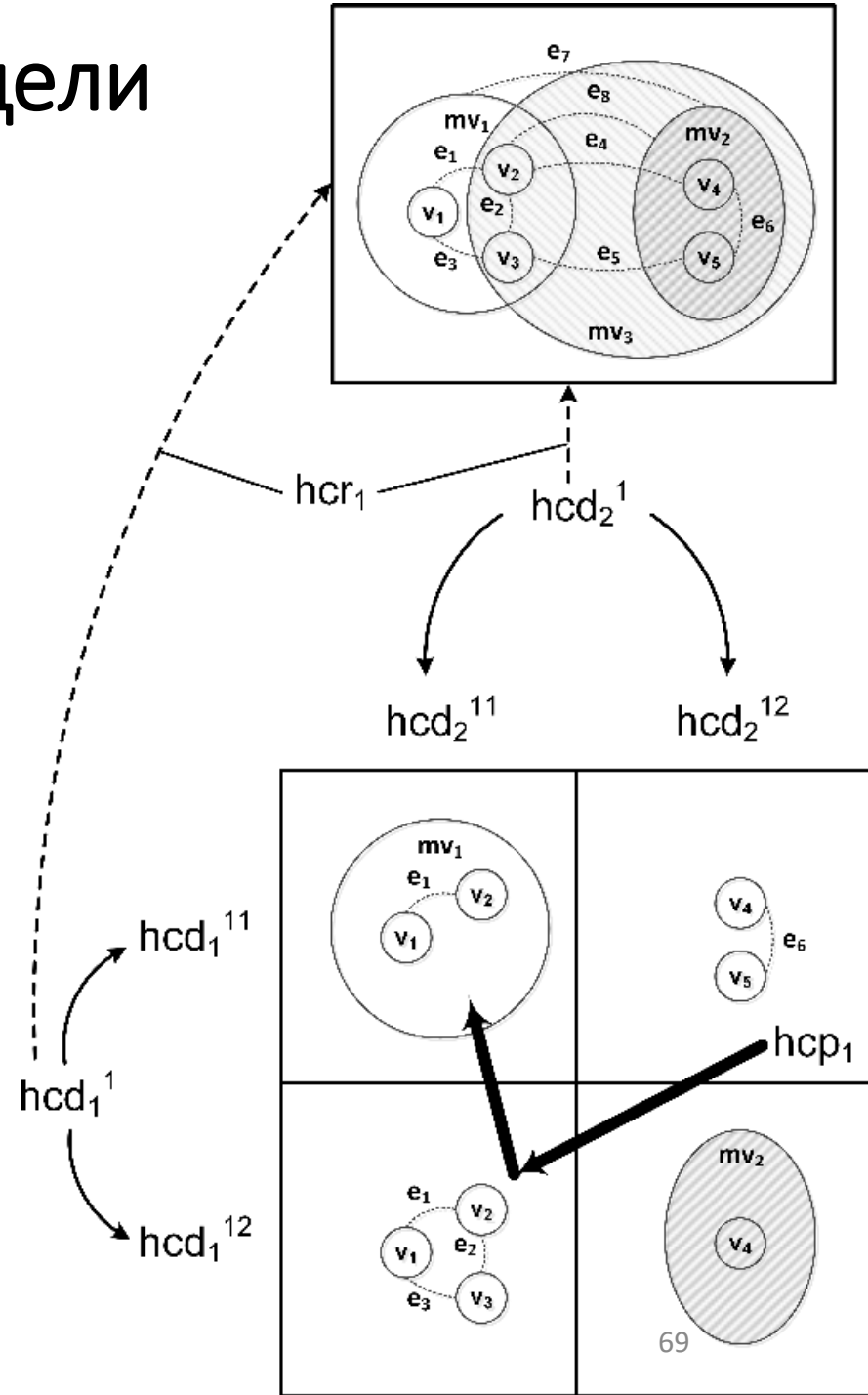
- Определение факта: $hcf_i = \langle \{hcd_i^{ref}\}, \{msr_n\} \rangle$, $hcd_i^{ref} \in (V \cup MV)$, $msr_n \equiv MG_j$. В соответствии с рассмотренными ранее определениями, ссылка на элемент измерения гиперкуба hcd_i^{ref} представляет собой связь с вершиной или метавершиной, которая принадлежит измерению, а мерой msr_n может быть произвольный фрагмент метаграфа.
- Определение правила агрегации:
- $hcr_k: \{hcf_{OUT}\} = ag^{MG}(\{hcf_{IN}\}, HCD^{ag}), HCD^{ag} \subset HCD$,
- вместо функции агрегации agf используется агрегирующий метаграфовый агент ag^{MG} , который выполняет задачу агрегации для произвольного фрагмента метаграфа MG_j , который является мерой.

Многомерно-метаграфовая модель - 3

- Определение метаграфового процесса:
- $hcr_i = \langle v_S, v_E, eo = true, \{atr_k\}, MG_j[mv_{node} \equiv MG_{node}^{ACTIVE}] \rangle, mv_{node} \in MV,$
- метаграфовый процесс является метаребром, в котором узлы mv_{node} вложенного фрагмента метаграфа MG_j являются активными метаграфами MG_{node}^{ACTIVE} .
- Агенты активного метаграфа обрабатывают данные на соответствующем шаге метаребра, соответствующего метаграфовому процессу.
- Метаграфовые процессы могут быть использованы для агрегации метаграфовых данных (в этом случае они являются аналогами метаграфовых правил агрегации), для обработки данных на одном уровне многомерно-метаграфовой модели, для детализации данных с переходом на нижние уровни.

Пример многомерно-метаграфовой модели

- На рисунке изображен простой гиперкуб с двумя измерениями hcd_1 и hcd_2 .
- Факты гиперкуба соответствуют элементам измерения гиперкуба. Комбинации элементов hcd_1^{11} - hcd_2^{11} , hcd_1^{11} - hcd_2^{12} , hcd_1^{12} - hcd_2^{11} , hcd_1^{12} - hcd_2^{12} соответствуют фактам гиперкуба нижнего уровня. Комбинация hcd_1^1 - hcd_2^1 соответствует агрегированному факту гиперкуба с правилом агрегации hcr_1 .
- В процессе агрегации изменяются не только количественные характеристики, но и структура метаграфа для ячеек, соответствующих фактам гиперкуба.
- Работа правила агрегации hcr_1 показана пунктирными стрелками.
- Шаги метаграфового процесса hcr_1 (который выполняется на нижнем уровне многомерно-метаграфовой модели,) показаны утолщенными стрелками.
- Таким образом, многомерно-метаграфовая модель может обрабатывать произвольные данные, знания и процессы, представленные в форме метаграфа, с использованием агрегации между уровнями модели или с применением метаграфовых процессов.



Основные положения и определения грануляции информации - 1

- Этот раздел основан на работе [14]. Мы используем модель грануляции, рассматриваемую в [15].
- **Гранулу** можно определить как «маленькую частицу, в особенности как одну из многочисленных частиц, образующих большую единицу», существенным свойством гранулы является **атомарность**.
- Атомарность информационного элемента (гранулы) предполагает, что данный элемент может быть явно выделен из окружающих его элементов внешней системы, добавлен удален или перемещен в другое место внешней системы.
- Гранулы могут быть дополнительно разложены на более мелкие гранулы, называемые субгранулами. В зависимости от сложности, уровня абстракции и размера, субгранулы могут быть расположены на разных уровнях. «Универсум» определяется на уровне наиболее общей (грубой) гранулы. Гранулы на самом низком уровне состоят из элементов конкретной используемой модели. Таким образом, существенным свойством гранулярности является **иерархическая организация гранул**.
- **Иерархическая организация гранул** на самом деле является частным случаем **холонической организации**. Холоническая организация часто используется в мультиагентном подходе. Холон – целое, рассматриваемое в то же время как часть целого.
- Важным следствием холонической организации является ее **мультивариантность**. Находясь в середине иерархии холонов, можно одновременно как представить текущий холон в виде различных комбинаций холонов нижнего уровня, так и включить текущий холон в различные холоны верхнего уровня, влияя на их организацию.

Основные положения и определения грануляции информации - 2

- Согласно Л. Заде, существует две основные операции по формированию и разделению гранул: **грануляция** включает в себя разложение целого на части. И наоборот, **организация** предполагает интеграцию частей в целое. Таким образом, существенным свойством грануляции являются **операции конструирования и деконструирования (разделения) гранул**. Рассматривают также такое понятие как **«информационные гранулы высшего порядка»**, которые включают гранулы более низких порядков.
- Термин грануляция можно рассматривать более широко. Вместо того, чтобы вводить два понятия, т. е. **грануляцию и организацию**, грануляцию можно определить как процесс в обоих направлениях: **организация и декомпозиция**.
- Также можно выделить два вида гранулярных отношений: **внутренние и внешние связи**. Таким образом, важно разделять внутренние связи (внутри гранул) и внешние связи (между гранулами).

Операции и отношения между гранулами - 1

Утончение, огрубление и частичный порядок.

- Гранула g_1 – это утончение другой гранулы g_2 , или, что эквивалентно, g_2 – это огрубление g_1 , которое обозначается $g_1 \leq g_2$ или $g_2 \geq g_1$, если каждая подгранула или объект g_1 содержится в некоторых подгранулах g_2 . \leq – это отношение утончения, а \geq – отношение огрубления.
- Отношения утончения/огрубления могут иметь место не для всех гранул, то есть имеет место частичный порядок. Тогда отношения с учетом частичного порядка называются р-уточнением и р-огрублением. Гранула o_1 – это р-уточнение гранулы o_2 , или, что эквивалентно, o_2 – это р-огрубление гранулы o_1 . Эти отношения обозначаются как $o_1 \sqsubseteq o_2$ или $o_2 \sqsupseteq o_1$, если некоторые подгранулы o_1 содержатся в некоторых подгранулах o_2 . \sqsubseteq – это отношение р-уточнения, а \sqsupseteq – это отношение р-огрубления.
- Таким образом, отношения \sqsupseteq и \sqsubseteq реализуют холоническую организацию гранул с учетом частичного порядка.

Операции и отношения между гранулами - 2

Отношение сходства.

Сходство является ключом к формированию взаимосвязей между гранулами. Для определения сходства предлагается использовать подходы, принятые в кластерном анализе, и расстояния в метрических пространствах, например, евклидово или манхэттенское расстояния.

Грануляции с точки зрения разбиений и покрытий.

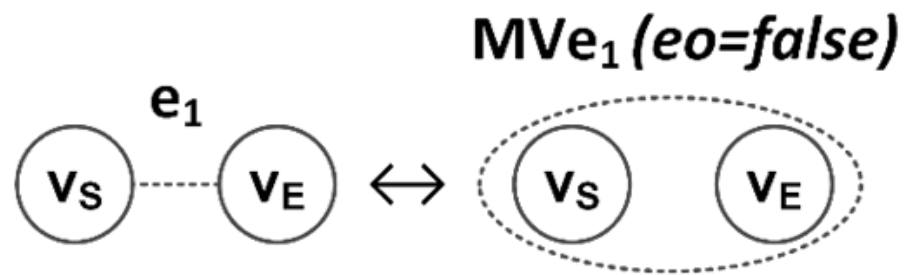
Разбиение состоит из непересекающихся подмножеств, а покрытие состоит из, возможно, пересекающихся подмножеств. Отношения между этими подмножествами называются отношениями разбиения и отношениями покрытия соответственно.

Меры гранулярности.

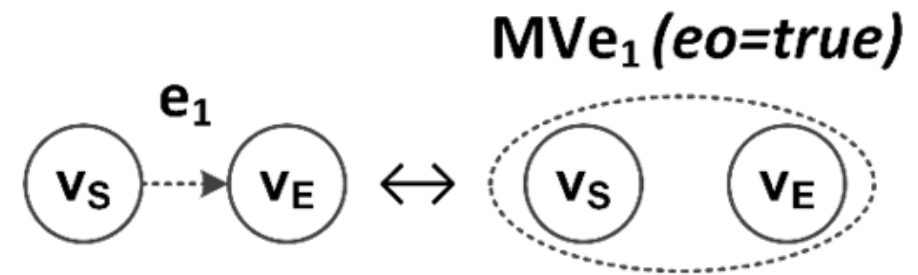
Изучение мер гранулярности привлекло внимание исследователей, чтобы лучше понять взаимосвязь, тонкость и грубость гранул. Многие меры гранулярности, основаны на нечетком подходе.

Метавершина как базовая структура метаграфовой модели - 1

- Рассмотрим, как элементы метаграфовой модели могут быть представлены в виде различных вариантов метавершины метаграфа.
- По определению, **вершина** является частным случаем метавершины. Вершина содержит вложенные атрибуты, но не содержит вложенного фрагмента метаграфа.
- **Ребро метаграфа** может быть представлено как частный случай метавершины, содержащей исходную и конечную вершины в виде вложенных элементов. В этом случае направленность ребра может быть представлена как атрибут метавершины. На рисунках показаны примеры представления ненаправленного ребра (слева) и направленного ребра (справа).



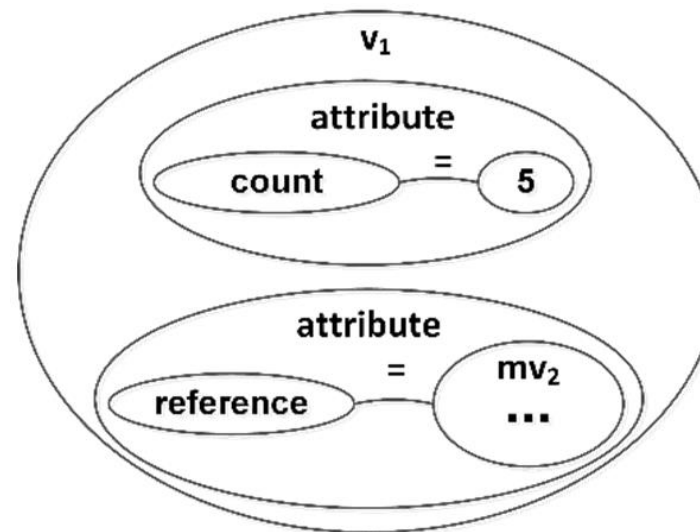
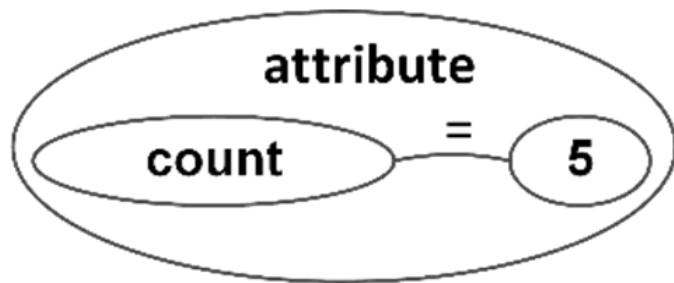
ненаправленное ребро



направленное ребро

Метавершина как базовая структура метаграфовой модели - 2

- **Фрагмент метаграфа** может содержать вершины, метавершины и ребра. Каждый из этих элементов индивидуально представим в виде метавершины. Представления отдельных элементов могут быть искусственно объединены в общую метавершину.
- **Атрибут** может быть представлен как частный случай метавершины, содержащей вложенные вершины для имени и значения атрибута. Ребро между вершинами имени и значения может быть интерпретировано как отношение “=”.
- На левом рисунке показано представление простого числового атрибута “count=5”. На правом рисунке показан пример вершины v_1 , содержащей числовой атрибут, и ссылочный атрибут, который ссылается на метавершину mv_2 .
- Назовем служебную метавершину **вершиной-предикатом** или **предикатом**.



Метаграфовое исчисление - 1

- Метаграфовое исчисление – это формальное исчисление для работы с метаграфами. Оно используется в качестве основы для преобразования метаграфов. Все конструкции исчисления представляют собой строки, разделенные на левую и правую части оператором присваивания, в качестве которого используется знак равенства «=».
- Слева от присваивания стоит имя конструируемого предиката, а справа предикаты и операторы исчисления.
- В отличие от функциональных языков программирования на переменные, соответствующие вершинам-предикатам, не накладывается ограничение неизменяемости. Переменная может фигурировать и в правой, и в левой части одного выражения, то есть может быть перезаписана.
- Для обозначения комментария используется общепринятая в C++-подобных языках конструкция «//».

Метаграфовое исчисление - 2

- **Оператор конструирования.**
- Оператор конструирования предназначен для создания новых вершин-предикатов на основе существующих. Применение этого оператора создает новую эмерджентность более высокого уровня. Оператор соответствует **операции организации гранул**, предложенной Л. Заде.
- Для удобства записи все рассматриваемые операторы могут использовать традиционную и префиксную формы. В префиксной форме используется общепринятый синтаксис группировки данных с помощью круглых скобок и запятой.

// Пример конструирования метавершины mv1

// Традиционная форма

$mv1 = v1 + v2 + v3 + e1 + e2 + e3$

// Префиксная форма

$mv1 = +(v1, v2, v3, e1, e2, e3)$

Метаграфовое исчисление - 3

- **Оператор деконструирования.**
- Оператор деконструирования предназначен для разделения элементов верхнего уровня на элементы нижнего уровня. Оператор соответствует **операции грануляции**, предложенной Л. Заде.
- Данный оператор напоминает оператор получения элементов кортежей, который применяется в функциональном программировании. В качестве синтаксиса используется стрелка влево «<=».

// Пример деконструирования метавершины mv1

v1, v2, v3, e1, e2, e3 <= mv1

Метаграфовое исчисление - 4

- **Оператор удаления.**
- Оператор удаления предназначен для удаления вершин-предикатов нижнего уровня из вершин-предикатов верхнего уровня. В этом случае используется синтаксис «-».

// Пример удаления ребра e1 из метавершины mv1

// Традиционная форма

$mv1 = mv1 - e1$

// Префиксная форма

$mv1 = -(mv1, e1)$

- На оператор удаления в метаграфовой модели накладывается несколько ограничений. Эти ограничения связаны с обеспечением логической целостности элементов метаграфовой модели.
- **Ограничение D1.** Поскольку ребро содержит ровно две вершины, то ни одну из этих двух вершин невозможно удалить. В этом случае нарушается логическая целостность ребра.
- **Ограничение D2.** Невозможно удалить предикат A, из предиката B, если предикат B содержит вложенный предикат C, ссылающийся на предикат A. В этом случае нарушается логическая целостность предиката C.
- Пример нарушения целостности:

$mv1 = v1 + v2$

$mv2 = v1 + v2 + mv1$

$mv2 = mv2 - v1$ // Ошибка - ссылка на v1 присутствует в mv1

Метаграфовое исчисление - 5

- **Оператор транзитивного удаления.**
- Для преодоления ограничения D2 в исчисление введен оператор транзитивного удаления с синтаксисом «-*». В этом случае также транзитивно удаляются все элементы, теряющие логическую целостность в результате удаления. Пример транзитивного удаления:

$mv1 = v1 + v2$

$mv2 = v1 + v2 + mv1$

$mv2 = mv2 - *v1$

//После удаления $mv2 = v2$

- До удаления элемент $mv2$ содержит элементы $v1$, $v2$, $mv1$. Элемент $v1$ удаляется из $mv2$ по запросу. Поскольку элемент $mv1$, вложенный в $mv2$, содержит ссылку на удаляемый элемент $v1$, то $mv1$ должен быть транзитивно удален из $mv2$. Поэтому в результате выполнения оператора транзитивного удаления $mv2$ содержит только элемент $v2$.

Метаграфовое исчисление с точки зрения атомарности - 1

Атомарность информационного элемента (гранулы) предполагает, что данный элемент может быть явно выделен из окружающих его элементов внешней системы, добавлен удален или перемещен в другое место внешней системы:

- Все элементы метаграфа можно четко отличить от окружающих элементов.
- Оператор конструирования используется для добавления элементов.
- Оператор удаления используется для удаления элементов.
- Перемещение элемента можно рассматривать как удаление его из некоторой части метаграфа, а затем добавление его в другую часть метаграфа.

Метаграфовое исчисление с точки зрения атомарности - 2

Если в результате операций добавления, удаления или перемещения элемента во внешней системе возникают нарушения структуры, то количество таких нарушений должно быть конечным, а местоположение во внешней системе для каждого такого нарушения должно быть однозначно локализуемо относительно других элементов:

- В предлагаемом исчислении операция добавления не может привести к нарушениям.
- Поскольку перемещение является последовательностью удаления и добавления, то только нарушения при удалении могут привести к нарушениям при перемещении.
- В предлагаемом исчислении только операция удаления может привести к нарушениям.
- Число нарушений может быть бесконечным тогда и только тогда, когда число элементов метаграфа бесконечно. Но метаграфы с бесконечным числом элементов в предлагаемом подходе не рассматриваются.
- При попытке удалить элементы, нарушающие ограничения D1 или D2, элементы метаграфа, нарушающие эти условия, автоматически локализуются.
- Таким образом, метаграфовое исчисление полностью соответствует определению атомарности, и элементы модели метаграфа, преобразованные с использованием исчисления, можно считать атомарными.

Гранулярность метаграфовой модели - 1

Свойство атомарности.

- Метаграфовое исчисление полностью соответствует определению атомарности, и элементы модели метаграфа, преобразованные с использованием исчисления, можно считать атомарными.

Холоническая организация.

- Все элементы метаграфовой модели могут быть представлены в виде метавершин специального вида. Согласно определению, метавершины по умолчанию организованы холонически.
- Для метаграфовой модели также выполняется мультивариантность. Находясь в середине иерархии метавершин, можно одновременно как представить текущую метавершину в виде различных комбинаций метавершин нижнего уровня, так и включить текущую метавершину в различные метавершины верхнего уровня, влияя на их организацию.

Операции конструирования и деконструирования для гранул.

- Операции конструирования и деконструирования для гранул реализуются с использованием метаграфового исчисления для элементов метаграфовой модели.

Информационные гранулы высшего порядка.

- В метаграфовой модели «информационные гранулы высшего порядка» реализуются как организация результатов предыдущей организации. Используя оператор конструирования (организации), предложенный Л. Заде и реализованный в метаграфовом исчислении, можно рекурсивно включать метавершины более низкого уровня в метавершины более высокого уровня.

Гранулярность метаграфовой модели - 2

Внутренние и внешние связи между гранулами.

- Оба вида связей могут быть реализованы в метаграфовой модели с использованием ребер. Если мы рассматриваем метавершину как гранулу, то ребра внутри метавершины могут рассматриваться как внутренние связи, а ребра между метавершиной и другими элементами могут рассматриваться как внешние связи.

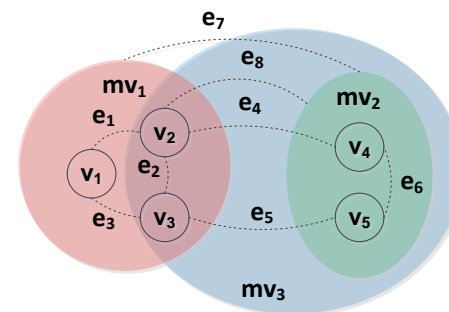
Утончение, огрубление и частичный порядок.

- Если мы определим отношение \sqsubseteq для вложенных метавершин, то мы можем рассматривать организацию метавершины как отношение частичного порядка. Тогда на рисунке mv_2 – это р-уточнение для mv_3 ($mv_2 \sqsubseteq mv_3$), а mv_3 – это р-огрубление mv_2 ($mv_3 \sqsupseteq mv_2$).

Отношение сходства.

- Сходство двух метавершин может быть определено как доля общих вложенных элементов. Если mv_1^{SET} это множество вложенных метавершин для метавершины mv_1 , а mv_2^{SET} это множество вложенных метавершин для метавершины mv_2 , то отношение сходства можно определить:

$$Similarity(mv_1, mv_2) = \frac{mv_1^{SET} \cap mv_2^{SET}}{mv_1^{SET} \cup mv_2^{SET}}.$$



Гранулярность метаграфовой модели - 3

Грануляция с точки зрения разбиений и покрытий.

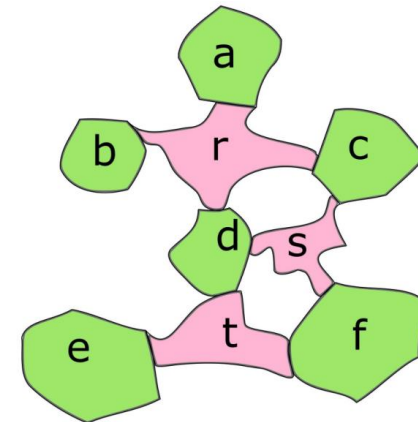
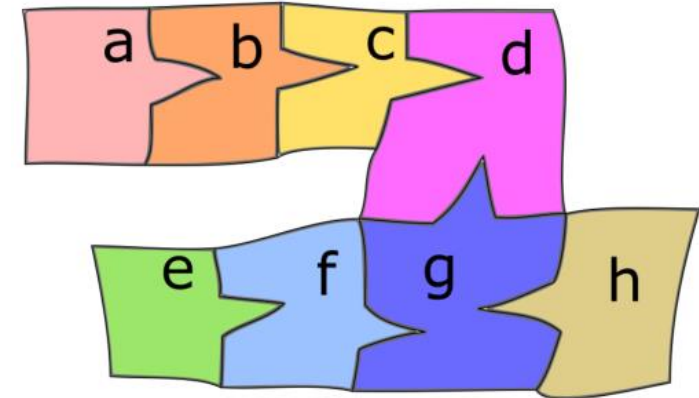
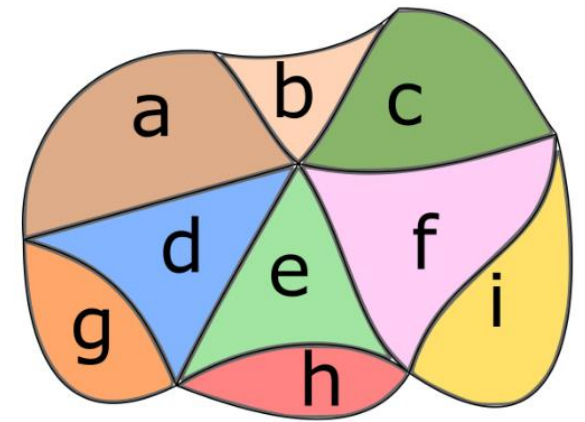
- Мы можем рассматривать метавершину как «универсум», полный набор гранул. Тогда разбиения и покрытия можно рассматривать как внутренние метавершины, содержащие вершины и метавершины более низкого уровня.

Меры гранулярности.

- Все элементы модели метаграфа могут быть сведены к метавершинам специального вида. Если mv_1^{SET} это множество вложенных метавершин для метавершины mv_1 , то мерой гранулярности метавершины mv_1 является количество вложенных метавершин: $Granularity(mv_1) = |mv_1^{SET}|$. Для вершины, не содержащей вложенных элементов, $mv_1^{SET} = \emptyset$, и $Granularity(mv_1) = 0$.

Протографы и архиграфы

- Протограф P задается множеством элементов $\{p_i\}$, $i = 1, n$ и матрицей соседства (смежности) $M_{n \times n}$, состоящей из 0 и 1, где 1 означает соседство (смежность) элемента a элементу b .
- Протограф можно рассматривать как граф, не имеющий ребер; роль ребер выполняет прилегание вершин друг к другу.
- Поскольку определение протографа основано на матрице смежности, то речь идет именно о попарном прилегании элементов протографа друг к другу. Также в [16] отмечается, что прилегание может быть направленным (ориентированным).
- В работе [16] также вводится понятие архиграфа: «Архиграф с числом n – это протограф с разбиением элементов на n классов и соответствующим правилом разбиения. Семейство архиграфов с числом n – семейство отличных разбиений протографов на n классов». Каждому классу можно сопоставить цвет, тогда архиграф является протографом, раскрашенным в n цветов, соответствующих классам.
- В работе [16] рассматривается следующий пример: «Простой неориентированный граф является протографом с числом разбиения 2, где элементы 1 класса не могут соседствовать с элементами 1 класса, но каждый элемент 1 класса может иметь сколько угодно отношений соседства с элементами 2 класса».
- В работе [17] отмечается, что «аннотируемые метаграфы это архиграфы с пятью классами элементов: атрибуты, вершины, метавершины, ребра и метаребра».



Операции прилегания

Для неориентированного протографа операция « $\overset{*}{*}$ » прилегания элементов протографа может быть определена как двухместная операция: $p_1 \overset{*}{*} p_2$.

Исходя из определения неориентированного протографа, операция прилегания является:

1. Коммутативной: $p_1 \overset{*}{*} p_2 = p_2 \overset{*}{*} p_1$. Порядок объединения двух соседних элементов протографа неважен.
2. Ассоциативной: $p_1 \overset{*}{*} (p_2 \overset{*}{*} p_3) = (p_1 \overset{*}{*} p_2) \overset{*}{*} p_3$. Порядок объединения нескольких соседних элементов протографа также неважен.

Для ориентированного протографа операция « $\overset{*}{*}$ » направленного прилегания элементов протографа также может быть определена как двухместная операция (предполагается, что прилегание направлено от левого операнда к правому): $p_1 \overset{*}{*} p_2$.

Исходя из определения ориентированного протографа, операция направленного прилегания является:

1. Некоммутативной: $p_1 \overset{*}{*} p_2 \neq p_2 \overset{*}{*} p_1$. В данном случае выражения $p_1 \overset{*}{*} p_2$ и $p_2 \overset{*}{*} p_1$ обладают различной направленностью.
2. Ассоциативной: $p_1 \overset{*}{*} (p_2 \overset{*}{*} p_3) = (p_1 \overset{*}{*} p_2) \overset{*}{*} p_3$. Порядок объединения нескольких соседних элементов протографа в случае сохранения направленности неважен.

На основе введенных операций может быть предложено другое определение протографа:

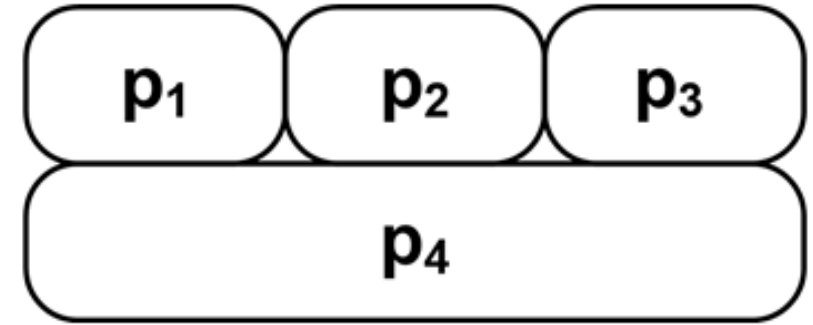
$$P = \langle PE, PL^U, PL^D \rangle, PE = \{p_i\}, PL^U = \{pl_j^U\}, PL^D = \{pl_k^D\}$$

где P – протограф; PE – множество элементов протографа $\{p_i\}$; PL^U – множество ненаправленных прилеганий элементов протографа $\{pl_j^U\}$; PL^D – множество направленных прилеганий элементов протографа $\{pl_k^D\}$.

Примеры протографов

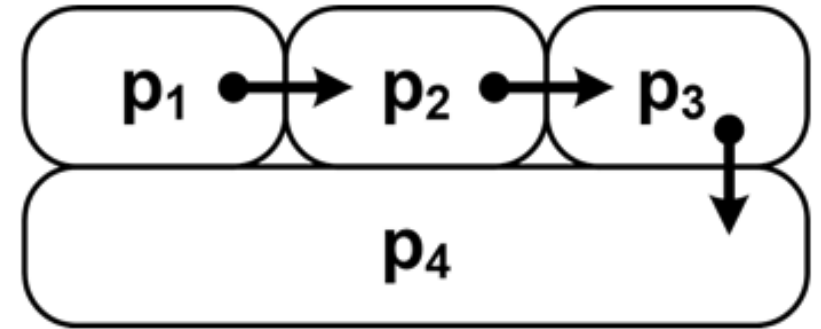
Пример неориентированного протографа:

- $PE = \{ p_1, p_2, p_3, p_4 \}$,
- $PLU = \{ p_1 \ast p_2, p_2 \ast p_3, p_1 \ast p_4, p_2 \ast p_4, p_3 \ast p_4 \}$,
- $PLD = \emptyset$.



Пример ориентированного протографа:

- $PE = \{ p_1, p_2, p_3, p_4 \}$,
- $PLU = \{ p_1 \ast p_4, p_2 \ast p_4 \}$,
- $PLD = \{ p_1 \ast\ast p_2, p_2 \ast\ast p_3, p_3 \ast\ast p_4 \}$.

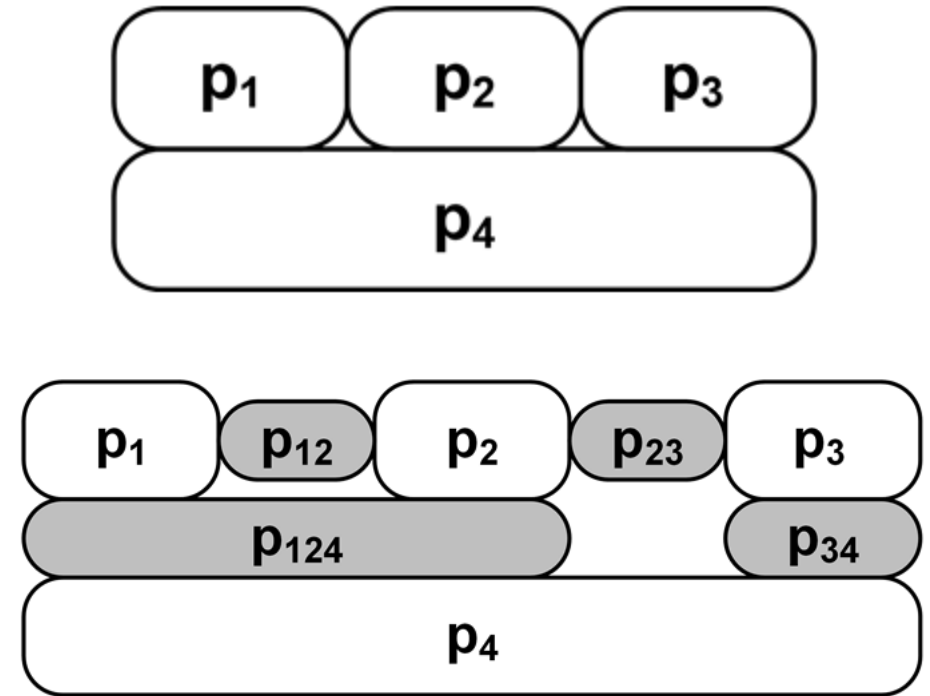


Таким образом, протографовая модель позволяет отказаться от традиционного понимания графа как совокупности вершин и ребер и рассматривать граф как совокупность прилегающих элементов.

Расширение протографа

В [16] вводится операция расширения протографа (расширения архиграфа): «операция расширения заключается в том, что все элементы протографа помечаются одним цветом (то есть относятся к одному классу), но между соседними элементами добавляются элементы второго класса».

На рисунке представлен пример протографа и его расширения до двух классов. Элементы второго класса показаны с помощью темного фона, их индексы соответствуют индексам прилегающих элементов первого класса.



Определение одноклассового протографа на основе формулы можно расширить до определения архиграфа (многоклассового протографа):

$$AR = \langle P, CLS^n, P.PE \xrightarrow{\varphi} CLS^n \rangle$$

где AR – архиграф; P – базовый протограф, на основе которого строится архиграф; CLS^n – множество классов вершин архиграфа мощностью n ; $P.PE \xrightarrow{\varphi} CLS^n$ – отображение множества вершин PE базового протографа P на множество классов CLS^n (раскраска элементов протографа соответствующими классами из множества CLS^n).

Отметим, что операция расширения протографа фактически позволяет добавлять соединительные элементы (второго класса) между основными элементами протографа (первого класса). Этот подход может быть использован для расширения аннотируемой метаграфовой модели.

Информационный элемент метаграфа

Информационный элемент метаграфа (ИЭМ) является элементарной строительной единицей для конструирования метаграфов:

$$ИЭМ = \left\langle id, NM, VAL, RL, \{lnk_i\}, \{atr_j\} \right\rangle, RL \in \{RL_V, RL_{MV}, RL_R\}$$

где id – уникальный идентификатор элемента; NM – наименование элемента; VAL – значение элемента; RL – роль элемента; lnk_i – ссылка на другой ИЭМ; atr_j – атрибут; RL_V – роль элемента «вершина»; RL_{MV} – роль элемента «метавершина»; RL_R – роль элемента «ребро».

Таким образом, ИЭМ может использоваться для хранения вершины, метавершины и ребра метаграфа. С использованием ссылок lnk_i реализуются нетипизированные связи между вершиной и ребром, а также связи между метавершиной и входящими в нее элементами.

Фактически, в данном случае ИЭМ изоморфен элементу протографа. Множество ссылок lnk_i задает нетипизированные связи между соседними ИЭМ аналогично операции ненаправленного прилегания « $\overset{*}{\sim}$ » в протографах. Роль элемента RL аналогична раскраске протографа, таким образом, формула задается архиграф из трех классов.

Но ИЭМ соответствует элементу метаграфовой модели (вершине, метавершине, ребру), в то время как связям между ИЭМ не уделяется достаточного внимания, связи задаются с помощью ссылок lnk_i .

С учетом рассмотренных положений протографового подхода (в особенности операции расширения протографа) и ИЭМ предложим метаграфовую модель, в которой достаточное внимание уделяется как элементам метаграфовой модели, так и связям между ними.

Метаграфовая модель на основе точек соединения - 1

Введем в метаграфовую модель новый элемент – точку соединения:

$$lp = \langle ИЭМ_1, ИЭМ_2, \{atr_k\} \rangle$$

где lp – точка соединения; $ИЭМ_1$ – первый информационный элемент метаграфа, соответствующий выходу точки соединения; $ИЭМ_2$ – второй информационный элемент метаграфа, соответствующий выходу точки соединения; atr_k – атрибут.

Точка соединения объединяет два информационных элемента метаграфа ненаправленным образом, аналогично операции ненаправленного прилегания « $\overset{*}{\times}$ » в протографах.

Дадим новое определение метаграфа, учитывающее точки соединения:

$$MG = \langle \{ИЭМ_i\}, \{lp_j\} \rangle$$

Таким образом, метаграф содержит множество информационных элементов $ИЭМ_i$ и множество точек соединения lp_j , связывающих информационные элементы.

Метаграфовая модель на основе точек соединения - 2

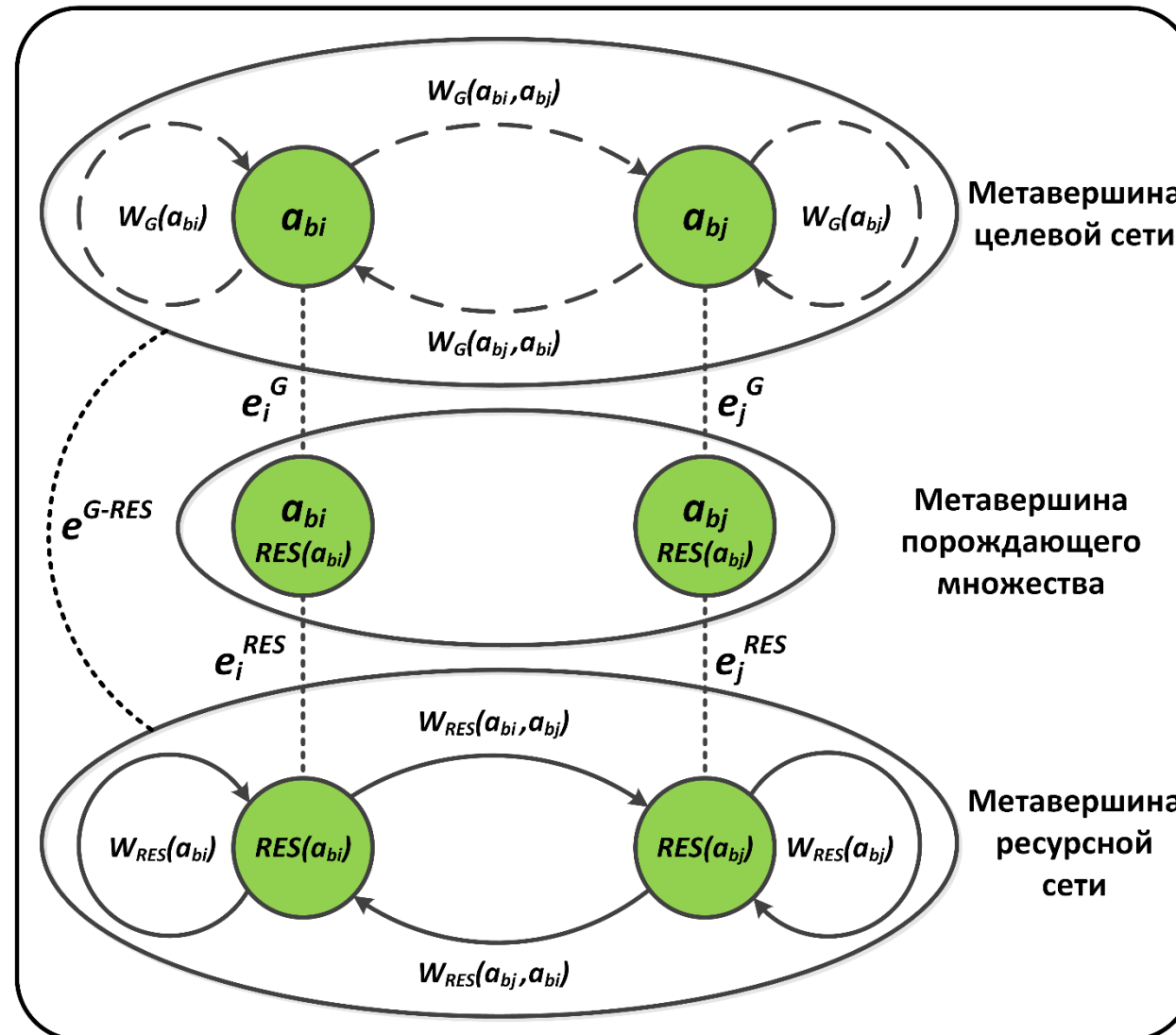
Предложенная модель метаграфа на основе точек соединения обладает следующими преимуществами:

1. Подход на основе ИЭМ позволяет унифицировать метаграфовую модель, представляя элементы метаграфовой модели (вершины, метавершины, ребра) в виде единой структуры ИЭМ с различными параметрами.
2. По сравнению с исходной моделью метаграфа и подходом на основе ИЭМ, предложенная модель позволяет гибко соединять элементы метаграфовой модели (вершины, метавершины, ребра) за счет использования точек соединения.
3. Использование точек соединения с набором атрибутов позволяет расширять метаграфовую модель, в частности, строить нечеткие или вероятностные метаграфы, не изменяя при этом основные элементы модели (вершины, метавершины, ребра). Например, для точки соединения между вершиной и метавершиной, можно задать атрибут, который будет интерпретироваться как мера нечеткой принадлежности вершины к метавершине.
4. Предложенный подход на основе точек соединения позволит унифицировать способы хранения метаграфовых данных в реляционных, документо-ориентированных и графовых СУБД.

Примеры работ, в которых используются ГИИС и подход на основе метаграфов

Пример 1. Метаграфовое представление ресурсно-целевой сети

Простая РЦС для представления взаимодействия между двумя благонамеренными агентами



Пример 2. Моделирование работы нейронной сети с использованием метаграфового подхода

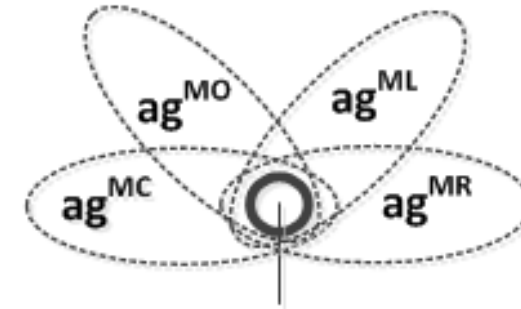
- На следующих 4 слайдах показан пример моделирования нейронной сети с использованием метаграфового подхода.
- Слайд А – Общая схема функционирования нейронной сети на основе метаграфовых агентов. Данная схема не рассматривает детально метаграфовое представление структуры отдельных нейронов, что рассматривается на слайдах Б, В, Г.
- Слайд Б – Описание функционирования персептрона на основе метаграфового подхода. Персептрон является базовым элементом для построения простых нейронных сетей.
- Слайды В,Г – Варианты описания глубокой нейронной сети с использованием различных стратегий регуляризации.
- Исследования в области метаграфового описания нейронных сетей продолжаются.

А) Описание нейронной сети с помощью метаграфовых агентов

- Представление нейронной сети в виде метаграфа может быть реализовано с помощью метаграфовых агентов.
- С использованием метаграфовых агентов может быть смоделирована работа нейросети в различных режимах.
- В примере используются следующие динамические метаграфовые агенты:

1. ag^{MC} – агент создания нейросети; 2. ag^{MO} – агент изменения нейросети;
3. ag^{ML} – агент обучения нейросети; 4. ag^{MR} – агент запуска нейросети.

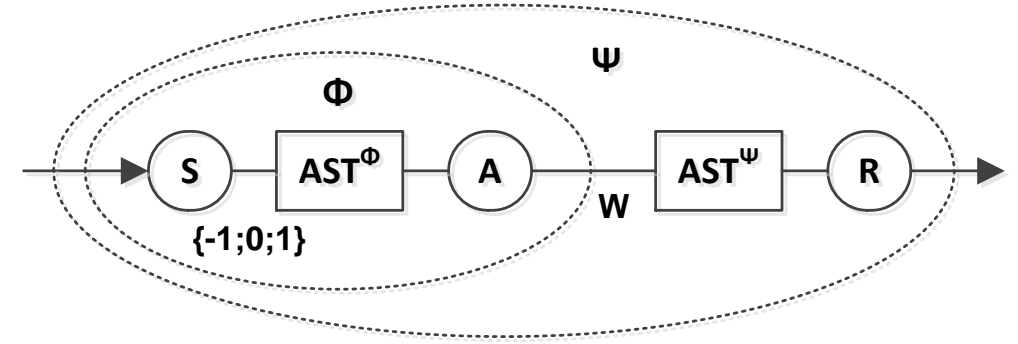
- Агент создания нейросети (ag^{MC}) реализует правила создания начальной топологии нейросети. Данный агент содержит как правила создания отдельных нейронов, так и правила соединения нейронов в нейросеть, в частности создает AST агентов-функций, моделирующих отдельные нейроны.
- Агент изменения нейросети (ag^{MO}) содержит правила изменения топологии сети в процессе работы. Это особенно важно для сетей с изменяемой топологией, таких как SOINN.
- Агент обучения нейросети (ag^{ML}) реализует один из алгоритмов обучения. При этом в результате обучения измененные значения весов записываются в метаграфовое представление нейросети. Возможна реализация нескольких алгоритмов обучения с использованием различных наборов правил для агента ag^{ML} .
- Агент запуска нейросети (ag^{MR}) реализует запуск и работу обученной нейросети в штатном режиме.
- Отметим, что агенты могут работать как независимо, так и совместно. Например, при обучении сети SOINN агент ag^{ML} может вызывать правила агента ag^{MO} для изменения топологии в процессе обучения.
- Каждый агент на основе заложенных в него правил фактически реализует специфическую программную «машину». Использование метаграфового подхода позволяет реализовать принцип «мультимашинности», когда несколько агентов с различными целями реализуют различные действия на одной и той же структуре данных.



**Представление нейронной
сети в виде метаграфа**

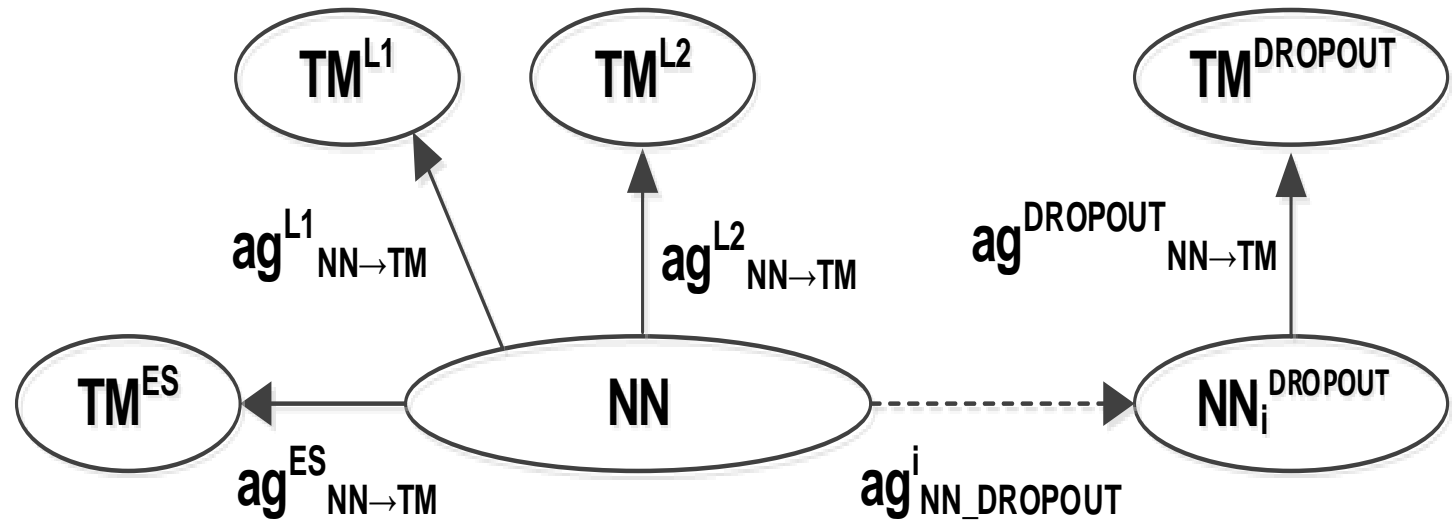
Б) Метаграфовое представление персептрона

- В соответствии с моделью Розенблатта, персептрон состоит из трех уровней: S, A и R.
- Слой сенсоров (S) представляет собой набор входных сигналов. Ассоциативный слой (A) включает набор промежуточных элементов, которые активизируются, если одновременно активизируется некоторый набор (образ) входных сигналов. Сумматор (R) активизируется, если одновременно активизируется некоторый набор A-элементов.
- В соответствии с обозначениями, принятыми в работе М. Минского и С. Пайперта, значение сигнала на A-элементе может быть представлено в виде предиката $\phi(S)$ а значение сигнала на сумматоре в виде предиката $\psi(A, W)$, где W – вектор весов. Под предикатом здесь понимается функция, принимающая только два значения «0» и «1».
- В зависимости от конкретного вида персептрона вид предикатов $\phi(S)$ и $\psi(A, W)$ может быть различным. Как правило, с помощью предиката $\phi(S)$ проверяется, что суммарный входной сигнал от сенсоров не превышает некоторый порог. Также с помощью предиката $\psi(A, W)$ проверяется, что взвешенная сумма от A-элементов не превышает некоторого порога.
- В нашем случае конкретный вид предикатов не важен, важно то, что предикаты являются обычными функциональными зависимостями, которые на программном уровне могут быть представлены в виде абстрактного синтаксического дерева и следовательно могут быть смоделированы агентами-функциями $\varphi^F = \langle S, A, AST^\phi \rangle$, $\psi^F = \langle \langle \{\varphi^F\}, W \rangle, R, AST^\psi \rangle$. Структура агентов-функций представлена на рисунке.
- Описание функций может содержать различные параметры, например пороги, но мы предполагаем, что эти параметры входят в описание абстрактного синтаксического дерева и могут быть переписаны с использованием метаграфовых агентов верхнего уровня.



В) Описание регуляризации с помощью метаграфов - 1

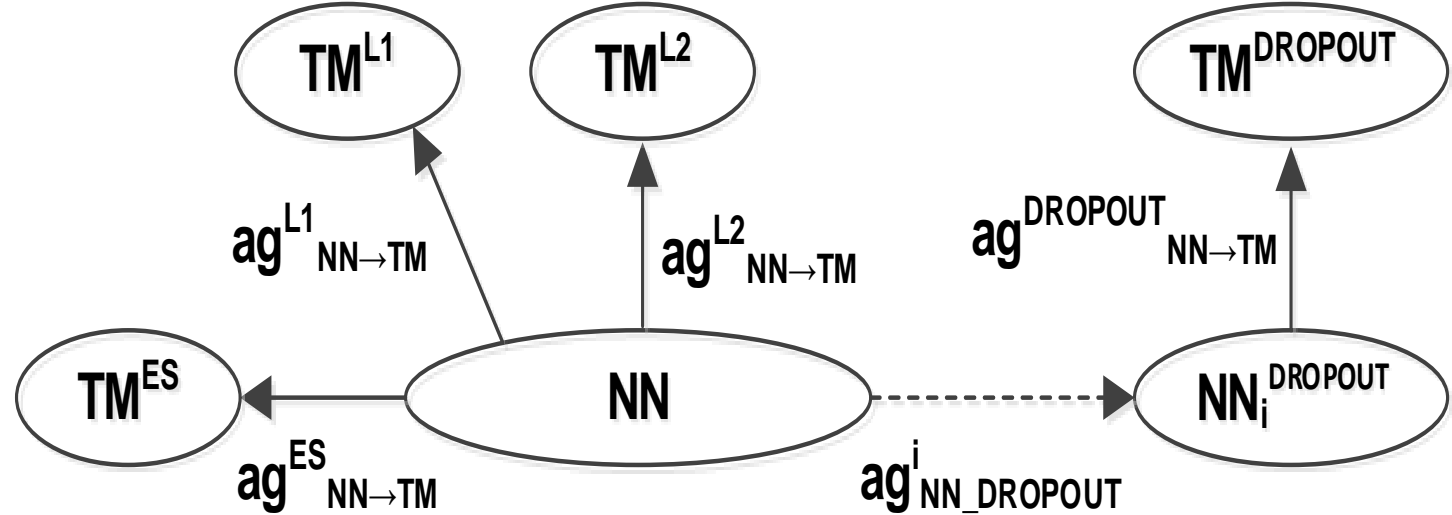
- Как и в случае отдельного персептрона, глубокая нейронная сеть может быть создана с использованием метаграфовых агентов. В зависимости от целей моделирования, нейрон может быть представлен в виде простой вершины или сложной метавершины, содержащей детализированное описание нейрона.
- На шаге создания нейросети создается структура “Neural Network” (NN), которая является плоским графом из нейронов, соединенных связями. Нейрон может быть описан как персептрон или аналогичным образом. Но при этом внутреннее представление нейрона может быть раскрыто в форме метавершины. Также в форме метавершины может быть представлен каждый уровень нейросети.



- В режиме обучения создается структура “Training Metagraph” (TM), изоморфная NN. Для каждой вершины-нейрона в NN создается метавершина в TM, связи сохраняются. Для создания TM на основе NN используется агент-функция $ag_{NN \rightarrow TM}$.
- TM является активным метаграфом, в котором с метаграфом данных связан метаграфовый агент ag_{TM} , отвечающий за обучение сети. Результаты обучения сохраняются в TM.
- Поскольку агенты могут быть представлены в виде метаграфов, то агент ag_{TM} создается с помощью агента $ag_{NN \rightarrow TM}$ ⁹⁸.

Г) Описание регуляризации с помощью метаграфов - 2

- Агент ag_{TM} может использовать различные стратегии регуляризации.
- Для NN может быть создано несколько структур TM, использующих различные стратегии регуляризации: L1, L2, ES (Early Stopping), dropout (обозначены верхними индексами).
- В случае использования стратегий L1, L2, ES не требуется изменения структуры графа нейросети в процессе обучения. Соответствующие агенты являются агентами-функциями.



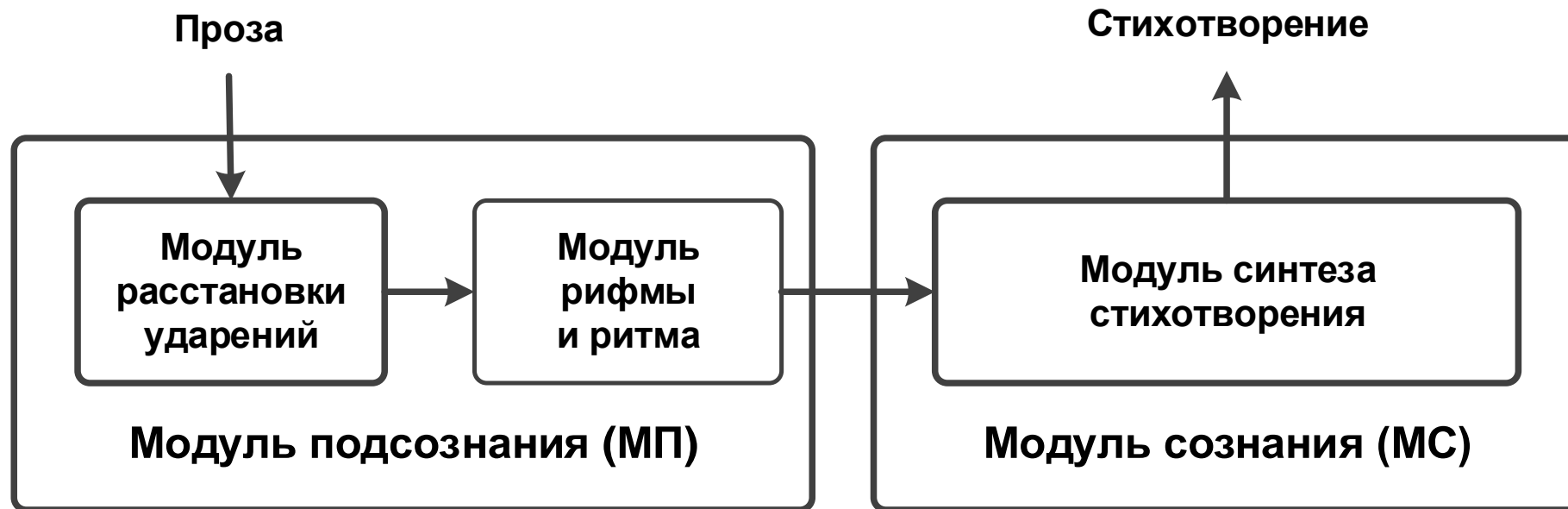
- В случае использования стратегии dropout структура нейросети должна быть изменена. При использовании данной стратегии сначала применяется метаграфовый агент $ag_{NN_DROPOUT}$ (в отличие от агентов-функций показан пунктирной стрелкой) который на основе правил модифицирует NN и формирует новую структуру **NN^{DROPOUT}** (индекс i показывает, что может быть сформировано несколько структур для различных вариантов dropout). Далее для **NN^{DROPOUT}** создается соответствующая структура **TM^{DROPOUT}** используемая для обучения сети.
- Альтернативой данному подходу является встраивание правил в агент ag_{TM} , которые будут использовать dropout уже в TM-структуре без необходимости модификации исходной структуры NN.
- Таким образом:
 - Нейронная сеть может быть представлена в виде метаграфа данных.
 - С использованием агентов-функций можно трансформировать структуру нейросети.
 - С использованием метаграфовых агентов можно динамически изменять структуру нейросети, моделируя различные алгоритмы обучения.

Пример 3. Гибридная интеллектуальная информационная система для генерации стихотворений

Maria Taran, Georgiy Revunkov, Yuriy Gapanyuk. The Hybrid Intelligent Information System for Poems Generation. NEUROINFORMATICS 2019: Advances in Neural Computation, Machine Learning, and Cognitive Research III. pp 78-86.

Подсознание – нейронная сеть

Сознание – система на правилах

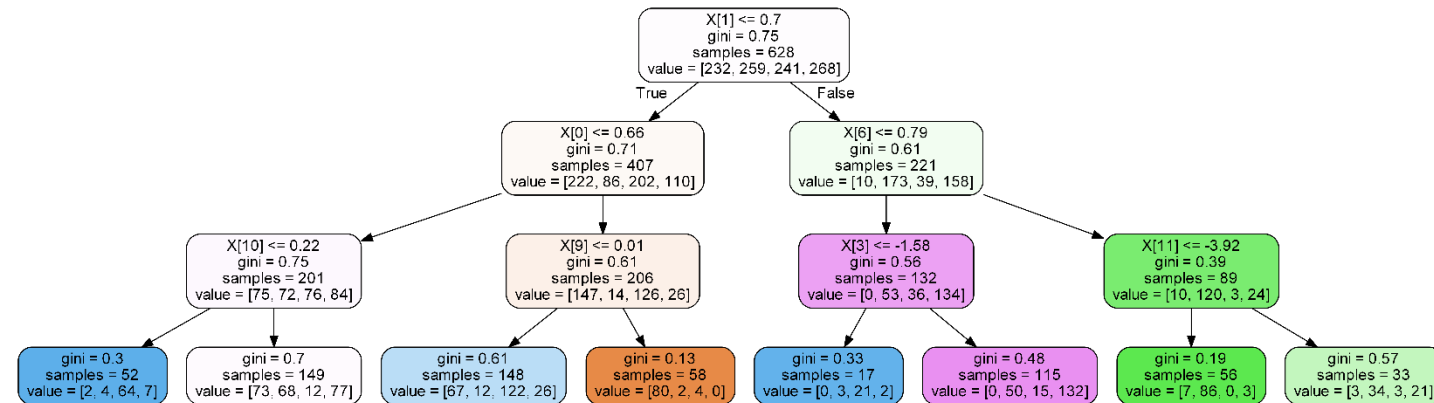
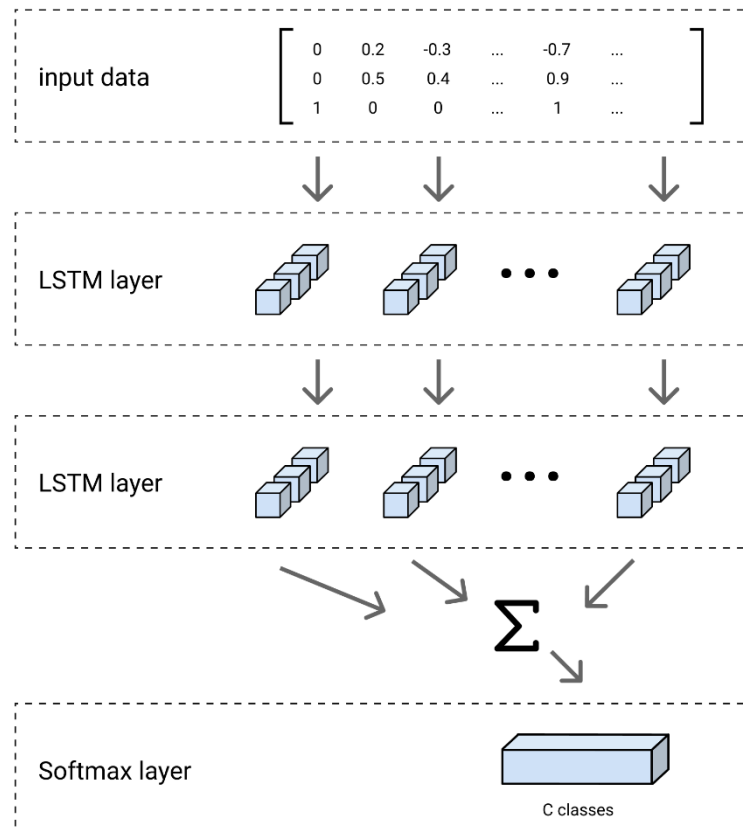


Пример 4. Гибридная интеллектуальная информационная система для классификации музыки

Aleksandr Stikharnyi, Alexey Orekhov, Ark Andreev, Yuriy Gapanyuk. The Hybrid Intelligent Information System for Music Classification. NEUROINFORMATICS 2019: Advances in Neural Computation, Machine Learning, and Cognitive Research III. pp 71-77.

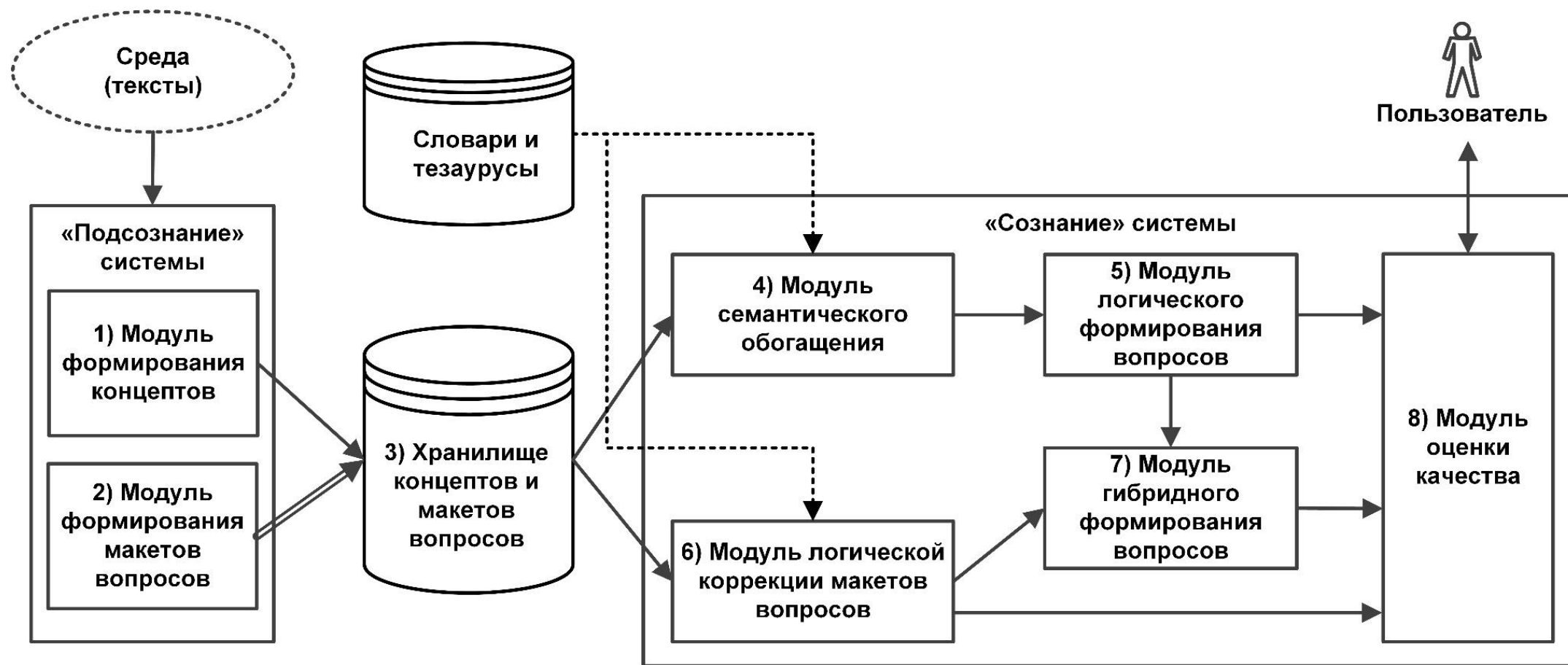
Подсознание – нейронная сеть LSTM

Сознание – решающее дерево



Пример 5. Гибридная интеллектуальная информационная система формирования вопросов по тексту

Belyanova M.A., Andreev A.M., Gapanyuk Y.E. NEURAL TEXT QUESTION GENERATION FOR RUSSIAN LANGUAGE USING HYBRID INTELLIGENT INFORMATION SYSTEMS APPROACH. Studies in Computational Intelligence. 2022. T. 1008 SCI. C. 217-223.



Перспективные направления исследований

- Связанные с применением подхода ГИИС и сложных графов
 - Исследование и проектирование интеллектуальных систем на основе подхода ГИИС.
 - Исследование и проектирование интеллектуальных систем, использующих сложные графы для описания и обработки различных видов данных:
 - Одномодальные системы (анализ и обработка текстов).
 - Мультимодальные системы (текст, изображения, видео, ...).
 - Реализация семантического [СЕР](#) в системах интернета вещей.
- Связанные с реализацией и развитием подхода ГИИС и сложных графов
 - Реализация метаграфовых агентов и эффективных способов обработки правил (развитие алгоритма RETE).
 - Реализация метаграфового хранилища, эффективных способов хранения и кодирования метаграфовых данных, языков запросов и обработки метаграфов.
 - Интеграция миварного подхода и подхода ГИИС.

Перспективные направления разработок

- Связанные с применением подхода ГИИС и сложных графов
 - Анализ «больших графов» публикаций с использованием метаграфового подхода.
 - Реализация систем моделирования (имитационного, мультиагентного) на основе метаграфового подхода.
 - Реализация семантической файловой системы на основе метаграфового подхода.
 - Реализация «табличного редактора» для метаграфовых данных.
- Связанные с реализацией и развитием подхода ГИИС и сложных графов
 - Реализация библиотеки визуализации метаграфов.
 - Реализация метаграфового хранилища, эффективных способов хранения и кодирования метаграфовых данных, языков запросов и обработки метаграфов.
 - Реализация метаграфовых агентов и эффективных способов обработки правил.

Выводы - 1

- Среди когнитивных моделей в искусственном интеллекте традиционно важную роль играют графовые модели представления знаний. В настоящее время повышенное внимание привлекают модели на основе сложных сетей или сложных графов.
- Гиперграф не позволяет моделировать сложные иерархические зависимости и не является полноценной «сложной сетью с эмерджентностью».
- Гиперсеть является «послойным» описанием графов и состоит из разнородных элементов (гиперграфов и отображений или гиперграфов и гиперсимплексов).
- Метаграфовая модель позволяет с помощью метавершин обеспечивать связь как между элементами одного уровня, так и между элементами различных уровней (при этом не обязательно соседних).
- Метаграфовый подход позволяет рассматривать сеть не только в виде «горизонтальных» слоев, но и в виде «вертикальных» колонок.
- Многомерно-метаграфовая модель включает меры, измерения, факты, правила агрегации и метаграфовые процессы.
- Метаграфовые процессы могут быть использованы для агрегации метаграфовых данных (в этом случае они являются аналогами метаграфовых правил агрегации), для обработки данных на одном уровне многомерно-метаграфовой модели, для детализации данных с переходом на нижние уровни.

Выводы - 2

- Основные свойства грануляции информации выполняются для метаграфовой модели.
- Метаграфовое исчисление полностью соответствует определению атомарности, и элементы модели метаграфа, преобразованные с использованием исчисления, можно считать атомарными.
- Все основные элементы метаграфовой модели могут быть представлены в виде метавершин специального вида. Поскольку метавершины используют холоническую организацию, то метаграф всегда является холонически организованным.
- Утончение, огрубление и частичный порядок, отношение сходства, разбиения и покрытия для метаграфовой модели могут быть описаны естественным образом с использованием особенностей метаграфовой модели.
- В метаграфовой модели «информационные гранулы высшего порядка» реализуются как организация результатов предыдущей организации. Используя оператор конструирования (организации), предложенный Л. Заде и реализованный в метаграфовом исчислении, можно рекурсивно включать метавершины более низкого уровня в метавершины более высокого уровня.
- Таким образом, метаграфовую модель можно рассматривать как один из вариантов практической реализации абстрактного подхода грануляции информации.

Выводы - 3

- Традиционная метаграфовая модель содержит три вида элементов: вершины, ребра и метавершины. Метавершина, в дополнение к свойствам вершины, включает вложенный фрагмент метаграфа.
- Особенностью аннотированной метаграфовой модели является то, что одинаковый набор вершин и ребер может быть включен в несколько различных метавершин, которые могут представлять различные ситуации и быть аннотированы различными атрибутами.
- Поскольку в текущей версии метаграфовой модели недостаточно детально рассматривается вопрос о том, как элементы модели соединяются в единый метаграф, то для соединения элементов в данной работе предлагается использовать протографовый подход.
- Классический протограф задается множеством элементов и матрицей соседства, в данной работе предлагается альтернативное определение протографа на основании введенных операций ненаправленного прилегания «*» и направленного прилегания «**».
- Архиграф позволяет разбивать множество элементов протографа на непересекающиеся классы, которые могут быть интерпретированы как цвета раскраски протографа. Операция расширения протографа (архиграфа) позволяет встраивать в существующий архиграф новые классы, задавая прилегание соответствующих элементов классов.
- В случае использования подхода на основе «информационных элементов метаграфа» (ИЭМ), ИЭМ изоморфен элементу протографа, а роль ИЭМ (соответствующая вершине, ребру, или метавершине) аналогична раскраске протографа, то есть задает архиграф из трех классов.
- Предложенная в данной работе метаграфовая модель на основе точек соединения использует в качестве базовых концепций протографовый (архиграфовый) подход, а также понятие ИЭМ.
- Точка соединения позволяет объединить два ИЭМ, задавая для объединения дополнительные атрибуты, которые могут интерпретироваться как мера нечеткости или вероятность вхождения ИЭМ друг в друга.
- Таким образом, предложенный подход позволяет расширить метаграфовую модель, строя за счет использования атрибутов точек соединения нечеткие или вероятностные метаграфы, и при этом не изменяя основные элементы модели (вершины, метавершины, ребра).

Список литературы - 1

1. *Basu A., Robert W. Blanning. Metagraphs and their applications. – New York: Springer, 2007.*
2. *Гапанюк Ю.Е. Этапы развития метаграфовой модели данных и знаний // В сборнике: Интегрированные модели и мягкие вычисления в искусственном интеллекте (ИММВ-2021). Сборник научных трудов X-й Международной научно-технической конференции. В 2-х томах, том 1. Смоленск, 2021. С. 190-200.*
3. *Chapela V., Regino Criado, Santiago Moral, Miguel Romance. Intentional risk management through complex networks analysis. – Springer, 2015: SpringerBriefs in optimization.*
4. *Voloshin V. I. Introduction to graph and hypergraph theory. – New York: Nova Science Publishers, 2009.*
5. *Johnson J. Hypernetworks in the science of complex systems. – London, Hackensack NJ: Imperial College Press, 2013.*
6. *Попков В.К. Математические модели связности. Новосибирск: ИВ-МиМГ СО РАН, 2006.*
7. *Анохин К.В. Когнитом: гиперсетевая модель мозга // Нейроинформатика-2015. XVII Всероссийская научно-техническая конференция. Сборник научных трудов. Ч. 1. М.: МИФИ. 2015. С. 14-15.*
8. *Chernenkiy V.M., Gapanyuk Y.E., Nardid A.N., Gushcha A.V., Fedorenko Y.S. The hybrid multidimensional-ontological data model based on metagraph approach // Lecture Notes in Computer Science, Springer. – 2018. – Vol. 10742. – P. 72-87.*

Список литературы - 2

9. *Gapanyuk Y.E. Metagraph approach to the information-analytical systems development // CEUR Workshop Proceedings. – 2019. – Vol. 2514. – P. 428-439.*
10. *Providing OLAP (on-line analytical processing) to user-analysts: An IT mandate. Technical report, E.F. Codd & Associates, 1993.*
11. *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.*
12. *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*
13. *Гапанюк Ю.Е. Основные положения многомерно-метаграфовой модели данных и знаний. В сборнике: Интегрированные модели и мягкие вычисления в искусственном интеллекте ИММВ-2022. Сборник научных трудов XI Международной научно-практической конференции. В 2-х томах, том 2. Коломна, 2022. С. 28-38.*
14. *Tarassov V.B., Kaganov Y., Gapanyuk Y. The metagraph model for complex networks: definition, calculus, and granulation issues. 2021. T. 12948 LNAI. C. 135-151.*
15. *Yao, J.T., Vasilakos, A.V., Pedrycz, W.: Granular Computing: Perspectives and Challenges. IEEE Transactions on Cybernetics 43(6), pp. 1977-1989 (2013).*
16. *Кручинин С.В. О некоторых обобщениях графов: мультиграфы, гиперграфы, метаграфы, потоковые и портовые графы, протографы, архиграфы // Вопросы науки. 2017. № 3. С. 48–67.*
17. *Сухобоков А.А. Метаграфово-табличная модель данных для систем управления активами // Искусственный интеллект в автоматизированных системах управления и обработки данных. Сборник статей Всероссийской научной конференции. В 2-х томах, том 1. Москва, 2022. С. 93–99.*

СПАСИБО ЗА ВНИМАНИЕ

gapyu@bmstu.ru