

УДК 004.89

DOI 10.52575/2687-0932-2024-51-2-425-443

## **Инструментальные средства поддержки процессов создания систем понятий и баз знаний для предметных областей со сложно-структурированными объектами (на примере задачи определения реакционных способностей химических соединений)**

**Гуляева К.А., Артемьева И.Л.**

Дальневосточный федеральный университет,  
Россия, 690922, Приморский край, г. Владивосток, о. Русский, п. Аякс, 10  
E-mail: karinagulyaeva8@gmail.com, artemeva.il@dvfu.ru

**Аннотация.** Системы понятий предметных областей с холо-меронимическими отношениями и основанные на них базы знаний представляют научный интерес ввиду важности прикладных задач, решаемых с их использованием. Выбор схемы базы знаний, искусный подбор промежуточных понятий – данная трудоемкая работа остается за экспертами предметных областей после закономерного ухода из процесса инженеров знаний. Возникает вопрос, можно ли автоматизировать сам процесс создания меронимической системы понятий и предъявить набор инструментальных средств поддержки данного процесса. В настоящем исследовании предлагается для решения данного вопроса использовать гибридный редактор, работающий в двух режимах (максимальной и минимальной) автоматизации процесса формирования меронимических систем понятий и баз знаний и позволяющий эксперту как затратить минимальное количество усилий, так и максимально настроить процесс. Основными элементами гибридного редактора являются редактор метаграфа сложно-структурированной системы, генератор метапонятий и конструктор функций и предикатов. С помощью редактора метаграфа создается сама сложно-структурированная система, подсистемами которой могут быть системы четырех структурных типов: элементарная система, дескриптивная система, система-подмножество и система-процесс. Генератор метапонятий автоматически формирует множество метапонятий, соответствующих промежуточным понятиям. В свою очередь, эксперт задает схемы определения функций и предикатов с помощью конструкторов: конструктора функций и предикатов метапонятий и конструктора функций и предикатов с произвольным набором аргументов. При разработке баз знаний указанного типа работа эксперта предметной области заметно упрощается – конструктор функций и предикатов метапонятий подставляет часть аргументов автоматически.

**Ключевые слова:** холо-меронимические отношения, меронимическая система понятий, база знаний, система понятий, промежуточные понятия онтологии, редактор, метаграф, система сложной структуры, процесс, метапонятие, онтология, реакционная способность

**Благодарности:** результаты, изложенные в данной статье, получены в рамках проекта РФФИ №19-37-90137.

**Для цитирования:** Гуляева К.А., Артемьева И.Л. 2024. Инструментальные средства поддержки процессов создания систем понятий и баз знаний для предметных областей со сложно-структурированными объектами (на примере задачи определения реакционных способностей химических соединений). Экономика. Информатика. 51(2): 425–443. DOI 10.52575/2687-0932-2024-51-2-425-443

---



# Toolkits for Concept System and Knowledge Base Creation Processes for Complex-Structured Object Domains (Based on the Chemical Compound Reaction Capacity Identification Problem)

**Karina A. Gulyaeva, Irina L. Artemieva**

Far Eastern Federal University

10 Ajax Bay, Russky Island, Vladivostok, 690922, Russia

E-mail: karinagulyaeva8@gmail.com, artemeva.il@dvfu.ru

**Abstract.** Concept systems of application domains with meronym relation and knowledge bases founded on them are of particular research interest due to the importance of applied problems solved using them. The choice of knowledge base scheme and savvy selection of interim concepts – this time-taking work is due to application domain experts to do after knowledge engineers have expectably left the process. The question is whether it is possible to automate the very process of meronym concept system creation and present the developed toolkits to support this process. This research suggests that the hybrid editor be used. This editor works in both regimes of maximum and minimum meronym concept system and knowledge base formation process automation. It allows application domain expert to make minimum effort or to customize the process, whichever he or she prefers. Basic elements of hybrid editor are complex-structured system metagraph editor, metaconcept generator, and function and predicate constructor. Complex-structured system is created with the help of metagraph editor. The subsystems of complex-structured system can be of four structural types: elementary system, descriptive system, system-subset, and system-process. Metaconcept generator automatically forms the set of metaconcepts that match with interim concepts. In turn, application domain expert sets function and predicate signatures with the help of the constructors: function and predicate metaconcept constructor and function and predicate constructor with arbitrary argument list. Expert work during knowledge base development is facilitated because function and predicate metaconcept constructor automatically sets some of the arguments.

**Keywords:** meronym relation, meronym concept system, knowledge base, concept system, ontology interim concept, editor, metagraph, complex-structured system, process, metaconcept, ontology, reaction capacity

**Acknowledgements:** The reported study was funded by RFBR, project number 19-37-90137.

**For citation:** Gulyaeva K.A., Artemieva I.L. 2024. Toolkits for Concept System and Knowledge Base Creation Processes for Complex-Structured Object Domains (Based on the Chemical Compound Reaction Capacity Identification Problem). *Economics. Information technologies.* 51(2): 425–443 (in Russian). DOI 10.52575/2687-0932-2024-51-2-425-443

## Введение

Холо-меронимические отношения (или отношения партитивности) – одна из основ сложности и разнообразия окружающего нас мира. Из-за большого прикладного значения не только профессионалы конкретных предметных областей, но и системные аналитики, логики (в частности, метеорологи), онтологи, разработчики экспертных систем, философы, специалисты в области семантики посвящают свои работы различным аспектам данного отношения [Aitken et al., 2004; Ahmad et al., 2008; Кузьменко, 2015; Morales-González et al., 2015; Колодько, 2016; Искандарова, 2019], причем иерархическое устройство зачастую свойственно не только объектам со сложным внутренним устройством, но и процессам (далее объекты и процессы, находящиеся на различных уровнях иерархии в отношении партитивности, для удобства изложения будем называть единым термином «сложно-структурированные объекты (системы)»).

Одним из важных примеров предметных областей, в которых данное отношение играет существенную роль, является задача определения реакционных способностей химических соединений. Несмотря на то, что исходное разнообразие объектов (электрон, атом, функциональная группа, химическое соединение) и процессов (химическая реакция) предметной области невелико, их конфигурации и комбинации сложны и многочисленны. На

вопросы того, «что», «с чем» и «каким образом» прореагирует, химики ищут ответы, применяя как методы квантовой химии, так и проверенные эмпирические правила (“rule of thumb”). То, насколько прогноз будет соответствовать действительности, зависит от знаний конкретного эксперта. Интенсификация процедур получения новых данных и знаний в данной предметной области делает эти знания всё более сложными для восприятия человеком и требует компьютерной поддержки. Обзор информационных и программных продуктов и проектов, связанных в той или иной степени с задачей определения реакционных способностей химических соединений: базы данных химических соединений и химических реакций ChemSpider [ChemSpider], Merck Index Online [Merck Index Online], Molbase [Molbase], NIST Chemistry Webbook [NIST], PDB [Protein Data Bank], PubChem [PubChem], Reaxys [Reaxys], SciFinder [SciFinder], SPRESI [SPRESI], ZINC [ZINC] и др., позволяющие осуществлять поиск химических соединений по подструктуре, суперструктуре, подобию; пакеты молекулярной динамики BIOVIA Discovery Studio [BIOVIA Discovery Studio], GROMACS [GROMACS], LAMMPS [LAMMPS], NAMD [NAMD], в которых временная эволюция системы взаимодействующих атомов или частиц описывается интегрированием их уравнений движения; пакеты, реализующие методы квантовой химии Gaussian [GAUSSIAN], GPAW [GPAW], в частности, реализующие методы теории функционала плотности; QSAR-дескрипторы хемоинформатики, в частности, топологические (отражающие связность атомов), геометрические (описывающие структуру молекулы как объект химического пространства), электростатические (описывают электростатическое поле молекулы), дескрипторы колец (описывают их количество и расположение кольца как фрагмента в молекуле), дескрипторы на основании матрицы смежности и 3D структур и др. – показал, что в зависимости от типа решаемой ими задачи в них содержится информация различных уровней детализации о внутрисистемных, межсистемных связях и связях сложно-структурированного объекта с внешней средой. Частая несовместимость их входных и выходных форматов, проприетарность лицензий многих продуктов, а также – наиболее важно – высокая степень сращения декларативных и процедурных знаний предметной области препятствует полноценному использованию их функциональных возможностей для организации многоступенчатого логического вывода, имитирующего процесс рассуждения профессионального химика, знания которого приобретаются за десятилетия теоретической и практической работы.

Одним из решений данного вопроса способно стать создание развитой системы понятий предметной области, в терминах которой можно было бы ставить и решать не только упомянутую (масштаб которой обширен), но и прикладные задачи смежных классов. Системы понятий предметных областей с холо-меронимическими отношениями (меронимические системы понятий) и основанные на них базы знаний должны описывать структуру сложно-структурированной системы, критерии вхождения экземпляров подсистем в экземпляр сложно-структурированной системы, свойства самой системы и свойствах самой системы при взаимодействии со средой, свойствах каждой из ее подсистем и свойства каждой из ее подсистем при взаимодействии со средой, свойствах сочетаний каждой из ее подсистем и свойства сочетаний каждой из ее подсистем при взаимодействии со средой и др. Разработка системы понятий в целом способна помочь создавать на их основе монотонно улучшаемые базы знаний. Однако по-прежнему существуют такие сложности, как выбор схемы базы знаний, искусный подбор промежуточных понятий, которые бы позволили эксперту сформировать исчерпывающие наборы основных понятий предметной области, а также наличие проблемы отладки сформированных на основе выбранных систем понятий баз знаний. После закономерного ухода инженеров знаний из процесса формирования систем понятий (в соответствии с современными подходами участие в формировании систем понятий принимают программисты совместно с экспертами [Клещёв, Орлов, 2006; Загорулько, Боровикова, 2007; Грибова et al., 2010; Голенков, Гулякина, 2014; Musen, 2015; Грибова et al., 2016]), данная трудоемкая работа осталась одной из главных обязанностей экспертов. Для того чтобы эксперт затрачивал минимальное количество умственных и физических усилий, требуются технологии, максимально автоматизирующие сам процесс создания системы понятий.

В работе [Гуляева, Артемьева, 2023] предлагается технология автоматизации процесса формирования систем понятий и баз знаний для предметных областей с холо-



меронимическими отношениями, заключающаяся в формировании промежуточных понятий на основании метаграфа системы сложной структуры, процедуры исследования окрестностей вершин метаграфа и введения специального множества метапонятий. *Цель настоящей работы* – предъявить набор разработанных инструментальных средств поддержки указанных процессов для предметных областей со сложно-структурированными объектами (на примере задачи определения реакционных способностей химических соединений) и обсудить ключевые особенности их разработки. Основой данного набора инструментов служит *гибридный редактор*, работающий в двух режимах (максимальной и минимальной) автоматизации процесса формирования систем понятий и баз знаний и позволяющий эксперту (в отличие от редактора [АСРaaS], требующего обязательного участия программиста для генерации участков подсети при автоматизации процесса формирования промежуточных понятий) по своему выбору – как затратить минимальное количество усилий при создании промежуточных понятий и меронимической системы понятий в целом (количество фактически затраченных усилий при взаимодействии эксперта с интерфейсом может быть рассчитано методами GOMS, KLM [John, Kieras, 1996], при этом редактор в режиме работы с максимальной автоматизацией подставляет обязательные аргументы функций и предикатов автоматически при определении экспертом их сигнатуры, что сокращает количество элементов в списке с доступными для выбора аргументами и количество соответствующих элементарных действий эксперта; подробности технологии изложены далее) – так и максимально настроить процесс.

### Объекты и методы исследования

Объектом исследования настоящей работы являются системы понятий предметных областей с холо-меронимическими отношениями (меронимические системы понятий) и основанные на них базы знаний.

В исследовании использовались различные методы научного познания, включающие методы системного анализа и моделирования, инженерии знаний, теории графов, технологии программирования и искусственного интеллекта, которые были направлены на разработку инструментальных средств поддержки процессов создания меронимических систем понятий и баз знаний (на примере задачи определения реакционных способностей химических соединений).

### Результаты и их обсуждение

Автоматизация процессов создания меронимических систем понятий и основанных на них баз знаний основана на использовании набора инструментальных средств, именуемых далее *гибридный редактор*, основными элементами которого являются редактор метаграфа сложно-структурированной системы, генератор метапонятий и конструктор функций и предикатов. Каждый инструмент, входящий в гибридный редактор, обсуждается далее отдельно.

### Редактор метаграфа сложно-структурированной системы

Сложно-структурированным объектом (системой) будем называть объект (систему) предметной области (с холо-меронимическими отношениями), в котором можно выделить подсистемы следующих типов: *элементарная система*, *дескриптивная система*, *система-подмножество* и *система-процесс*. *Элементарная система* (обозначение – Elementary System (E)) подсистем не имеет и может иметь только один экземпляр. Ее прямая надсистема имеет два числа из множества  $\mathbb{N} \cup \{0\}$ , ассоциированные с минимальным и максимальным количествами ее единственного экземпляра. Максимальное количество может быть не ограничено (и обозначается элементом *inf*, добавленным в указанное множество:  $\mathbb{N} \cup \{0\} \cup \{inf\}$ ). *Дескриптивная система* (обозначение – Descriptive Element System (D)) не является элементарной, и ее экземпляры имеют элементами экземпляры иных, чем она, подсистем, которые вместе образуют определенную целостность. *Система-подмножество* (обозначение – Descriptive Subset System (S)) не является элементарной, и все ее элементы являются и

элементами ее прямой надсистемы. Система-подмножество может быть отождествлена со своей прямой надсистемой, что означает, что ее экземпляр может рассматриваться и как экземпляр ее прямой надсистемы (например, экземпляры “Бензольное кольцо” или “Циклопентадиен” системы “Функциональная группа”, которая является прямой подсистемой системы “Химическое соединение”, могут рассматриваться и как экземпляры системы “Химическое соединение”). Система-процесс (обозначение – Process (P)) не является элементарной и обозначает совокупность взаимосвязанных или взаимодействующих видов деятельности, преобразующих входы в выходы [ГОСТ Р ИСО 9000-2001]. Для системы-процесса должен быть задан хотя бы один участник процесса (ее прямая подсистема) с указанием роли и максимальное количество шагов процесса. Для каждого типа системы, кроме элементарной, для каждой из ее подсистем должны быть заданы два числа из множества  $\mathbb{N} \cup \{0\}$ , ассоциированные с минимальным и максимальным количествами каждой ее подсистемы. Максимальное количество может быть не ограничено (и обозначается элементом  $inf$ , добавленным в указанное множество:  $\mathbb{N} \cup \{0\} \cup \{inf\}$ ).

Для представления сложно-структурированного объекта (системы) используется направленный ациклический метаграф  $(G, \mu_G)$ , который называется *метаграфом* сложно-структурированной системы, где  $G$  – направленный ациклический граф,  $\mu_G$  – разметка графа  $G$ . Разметка  $\mu_G: Terminal \cup Nonterminal \rightarrow SystemNames$ ;  $\mu_G: R \rightarrow \{\text{Являться прямой надсистемой (подсистемой)}\}$ . Далее разметка дуг “Являться прямой надсистемой (подсистемой)” опускается, подразумеваясь существующей.  $SystemNames$  включает имена систем четырех описанных структурных типов. Пример метаграфа сложно-структурированной системы “Химическая реакция” приведен на рис.1.

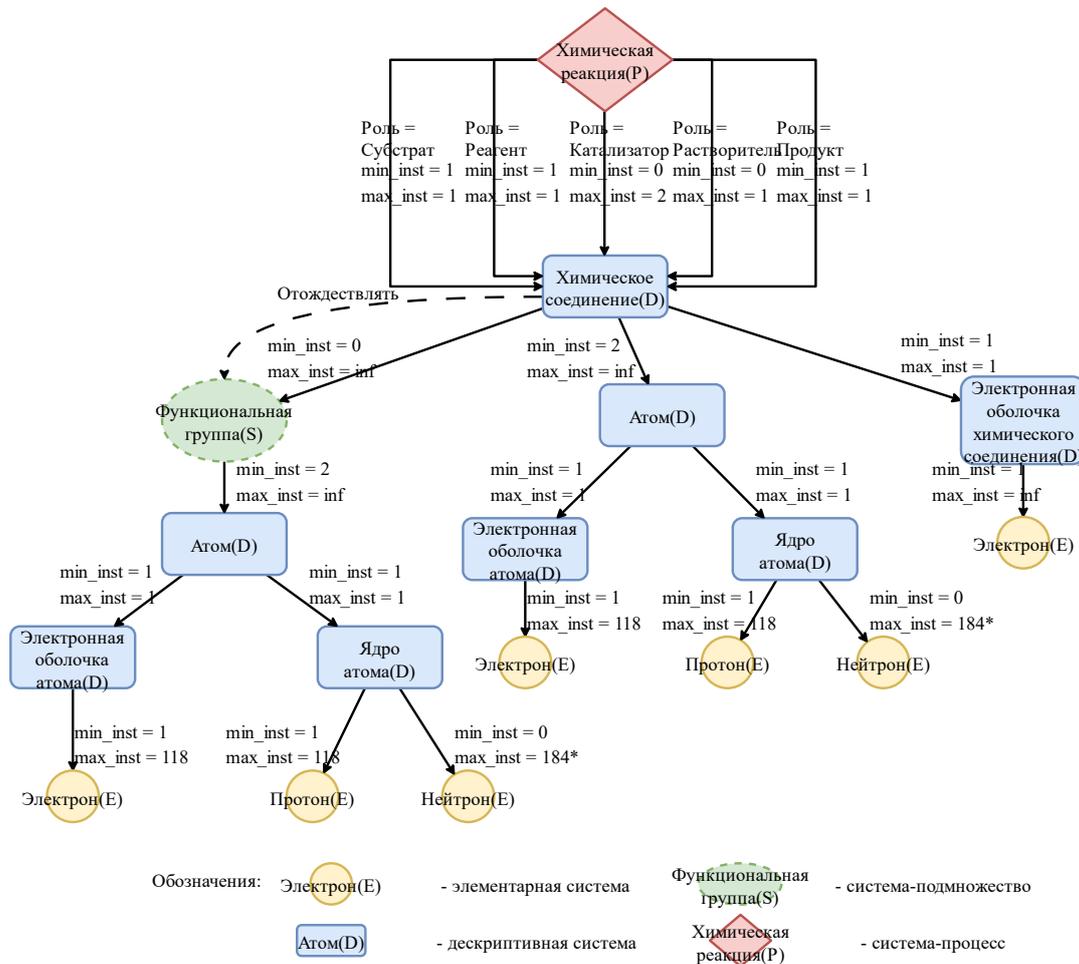


Рис. 1. Пример метаграфа сложно-структурированной системы “Химическая реакция”  
 Fig. 1. Complex-structured system “Chemical reaction” metagraph example

Метаграф сложно-структурированной системы является основой меронимической системы понятий. Метаграф формируется “снизу-вверх” (сначала должны быть созданы элементарные системы, а затем они могут быть указаны в качестве прямых подсистем систем какого-либо неэлементарного структурного типа). Высота сложно-структурированной системы определяется автоматически (высота элементарной системы равна 0, а при присоединении прямой подсистемы высота сложно-структурированной системы переопределяется). Система с наибольшей высотой называется *образующей* системой метаграфа. Все недостижимые из образующей системы подсистемы автоматически помечаются в качестве таковых и в дальнейшем в формировании меронимической системы понятий участия не принимают.

Редактор метаграфа сложно-структурированной системы должен позволять ввести название системы, тип системы (SysStructuralType  $\in$  {Elementary System (E), Descriptive Element System (D), Descriptive Subset System (S), Process (P)}), а также указать прямые подсистемы системы, не включающие создаваемую систему. По событию выбора типа системы в элементе пользовательского интерфейса “Выпадающий список” должно срабатывать действие создания форм ввода при учёте следующих аспектов. Если создаётся система типа Elementary System (E), то все прямые подсистемы должны быть недоступны для выбора (реализуется, например, с помощью CSS команды “display: none;”). Если создаётся система типа Process (P), то в прямых подсистемах, доступных для выбора, все системы типа Process (P) должны быть скрыты, а максимальное число шагов процесса, напротив, отображено. Если создаётся система, прямая подсистема которой имеет тип Descriptive Subset System (S), то необходимо отобразить checkbox “Отождествить с прямой надсистемой”. Если эксперт активирует данный checkbox, то для пары ([Система], [прямая подсистема типа (S)]) будет автоматически создан предикат внутреннего устройства системы “Являться прямой подсистемой([Система], [прямая подсистема типа (S)])”. Далее программистом может быть добавлен код подпрограммы для вычисления значения данного предиката в базе знаний. На рис. 2 представлена форма для создания экспертом предметной области системы типа Process (P) с помощью редактора метаграфа сложно-структурированной системы. Дополнительно с создаваемой системой может быть ассоциирована подпрограмма проверки дополнительных ограничений, которая будет запускаться на исполнение при создании экземпляров созданной системы.

Рис. 2. Редактор метаграфа сложно-структурированной системы. Пример формы для создания системы типа Process (P)

Fig. 2. Complex-structured system meta-graph editor. Form instance for Process (P) type system creation

После того, как экспертом задана образующая система и все ее достижимые подсистемы определены, можно переходить к шагу генерации метапонятий.

### Генератор метапонятий

*Метапонятие* может рассматриваться как промежуточное понятие системы понятий. Обозначим множество метапонятий *Metaconcepts*. Любое метапонятие из множества *Metaconcepts* характеризуется текстом инициализирующего шаблона, пользовательским названием, множеством систем узла, а также может характеризоваться множеством *сквозных* понятий (будем выделять множество метапонятий без учета сквозных понятий *M* и множество метапонятий с учетом сквозных понятий *MOut*).  $Metaconcepts = M \cup MOut$ . Будем называть *сквозными* понятиями те понятия, которые необходимы для описания объективной реальности и условий возможности изменения окружающего мира, но не входящие во множество объектов предметной области (например, сквозными понятиями могут быть “давление”, “температура”, “энергия” (или ее различные виды), “моменты наблюдения процесса” и др.). Каждое сквозное понятие характеризуется множеством единиц измерения сквозного понятия. Множество сквозных понятий может быть пусто.

Множество метапонятий меронимической системы понятий может быть автоматически сгенерировано по метаграфу сложно-структурированной системы. Для этого дерево образующей системы обходится одним из алгоритмов обхода деревьев (например, в глубину с постпорядком). При этом окрестность каждой еще не исследованной вершины метаграфа сложно-структурированной системы может быть исследована с помощью процедуры исследования окрестности вершины “ProcedureNodeNeighborhoodExamination”, предложенной в [Гуляева, Артемьева, 2023]. Эксперту предлагается выбрать глубину обхода перед началом процедуры. Процедура исследования вершины запускается для каждой вершины, при этом глубина обхода и фактическая высота системы влияют на типы метапонятий, которые будут сгенерированы для каждой исследованной вершины. При фактической высоте системы больше либо равной двум и выборе экспертом глубины обхода, равной двум (обозн. “Deep”), будут сгенерированы все пять типов метапонятий (рис. 3), равной единице (обозн. “Medium”) – метапонятия типов Individual, Hierarchy Individual, Collective, равной нулю (default) – метапонятия типа Individual (названия типов метапонятий выбраны произвольно, но отражают, по мнению авторов, тип обхода узла метаграфа сложно-структурированной системы).

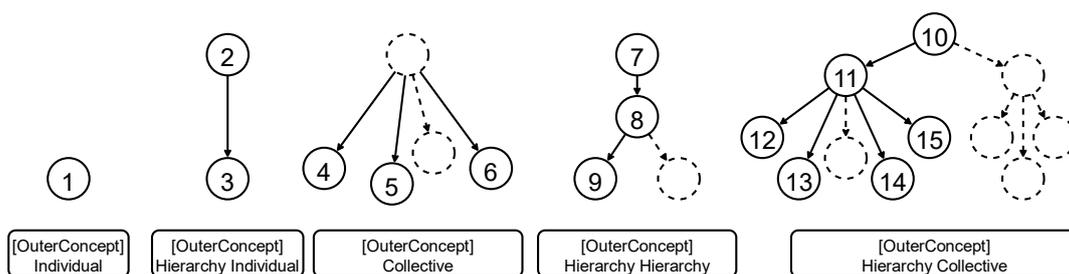


Рис. 3. Типы метапонятий и системы узла  
Fig. 3. Metaconcept types and node systems

Каждое метапонятие задает целый набор понятий, схема определения которых соответствует его узлу, определяя, таким образом, набор отношений экземпляров систем узла. Метапонятие определяет онтологическую структуру группы свойств систем узла. Генератор метапонятий обходит дерево сложно-структурированной системы и с каждым созданным метапонятием ассоциирует системы узла (их названия используются также в инициализирующих шаблонах метапонятий), инициализирующий шаблон метапонятия, пользовательское название метапонятия и множество сквозных понятий (если сгенерировано метапонятие из множества *MOut*). Это позволяет эксперту на следующем этапе – в



конструкторе предикатов и функций метапонятий – создавать функции и предикаты, аргументы-системы и аргументы-сквозные понятия которых уже заданы. Эксперту потребуется только выбрать, будет ли в функцию или предикат передаваться множество экземпляров конкретного аргумента-системы или же один экземпляр.

Инициализирующий шаблон метапонятия представляет внутреннюю структуру групп свойств системы, соответствующей вершине метаграфа, исследуемой с помощью процедуры исследования окрестности вершины. Эксперту предлагается переименовать название, предложенное в инициализирующем шаблоне. Так, например, для инициализирующего шаблона “Совместные свойства сочетания следующих систем: Протон, Нейтрон” эксперт может задать пользовательское название “Свойства нуклонов”, а для инициализирующего шаблона “Свойства системы Атом как прямой подсистемы системы Химическое соединение как прямой подсистемы системы Химическая реакция” – пользовательское название “Свойства атомов реакционного центра”. Инициализирующие шаблоны различных типов метапонятий представлены в таблице.

Метапонятия различных типов и их инициализирующие шаблоны  
 Metaconcept initialization templates

Тип метапонятия	Пример текста инициализирующего шаблона метапонятия (f-string)
[OuterConcept] Individual	f“Индивидуальные свойства подсистемы {1} [при учете следующих сквозных понятий {OuterConceptSet}]”
[OuterConcept] Hierarchy Individual	f“Свойства системы {3} как прямой подсистемы системы {2} [при учете следующих сквозных понятий {OuterConceptSet}]”
[OuterConcept] Collective	f“Совместные свойства сочетания следующих систем: {4, 5, 6} [при учете следующих сквозных понятий {OuterConceptSet}]”
[OuterConcept] Hierarchy Hierarchy	f“Свойства системы {9} как прямой подсистемы системы {8} как прямой подсистемы системы {7} [при учете следующих сквозных понятий {OuterConceptSet}]”
[OuterConcept] Hierarchy Collective	f“Совместные свойства сочетания следующих систем: {12, 13, 14, 15} – как прямых подсистем системы {11} как прямой подсистемы системы {10} [при учете следующих сквозных понятий {OuterConceptSet}]”

Генератор метапонятий позволяет эксперту создать исчерпывающий набор метапонятий, соответствующих промежуточным понятиям меронимической системы понятий, автоматически. Это дает эксперту возможность систематически исследовать наборы отношений экземпляров систем узла. Для этой цели ему необходимо задать схемы определения функций и предикатов метапонятий с помощью конструктора.

### Конструктор функций и предикатов

После запуска генератора метапонятий по метаграфу сложно-структурированной системы будет создан исчерпывающий набор метапонятий, которые будут доступны эксперту для выбора и переименования. Для каждого выбранного метапонятия эксперт может задать набор функций и предикатов посредством конструктора. Конструктор функций и предикатов работает в двух режимах: конструктор функций и предикатов с произвольным набором аргументов (режим минимальной автоматизации) и конструктор функций и предикатов метапонятий (режим максимальной автоматизации).

#### *Конструктор функций и предикатов с произвольным набором аргументов*

Эксперт указывает название функции или предиката и задает схему определения аргументов и компонентов возвращаемого значения (для функции). Аргументы и компоненты возвращаемого значения (для функции), доступные эксперту для выбора, включают в себя стандартные типы значений, типы значений, связанные с системами метаграфа, типы

значений, связанные со сквозными понятиями и типы значений, связанные со вспомогательными понятиями базы знаний.

Стандартными значениями можно считать комплексные, действительные, целые значения, имена (скалярные значения) и логические значения. Аргументом может быть либо один экземпляр указанного стандартного значения, либо множество (может быть специфицирован интервал допустимых стандартных значений).

Типы значений, связанные с системами метаграфа, формируются при обходе метаграфа и имеют следующий вид: “[Система]\_KB”, “[Система]\_KB”, “[Система]\_PH”, “[Система]\_PH”. “[Система]\_KB” означает, что аргументом или компонентой возвращаемого значения может быть единственный экземпляр системы [Система], “[Система]\_KB” означает множество экземпляров системы [Система], “[Система]\_PH” означает единственный *фантом* экземпляра системы [Система], “[Система]\_PH” означает множество *фантомов* экземпляра системы [Система].

*Фантомом экземпляра системы* называется специальный экземпляр системы, у которого все свойства совпадают со свойствами экземпляра системы, на основании которого фантом порожден, но имеются свои уникальные свойства, из-за которых фантом может быть аргументом или компонентой возвращаемого значения (для функций) определенных функций и предикатов, аргументом или компонентой возвращаемого значения (для функций) которых не может быть экземпляр системы, на основании которого фантом порожден. Фантомы характерны для предметной области определения реакционных способностей химических соединений, но могут использоваться и в других областях со сложно-структурированными объектами. Часть свойств, которые являются неизменными для всех конфигураций определенного химического соединения, целесообразно считать свойствами данного экземпляра системы “Химическое соединение”, а свойства экземпляра системы, которые характерны для конкретной конфигурации экземпляра системы (различные конфигурации химических соединений могут иметь одинаковый состав и матрицу смежности атомов, но различное пространственное расположение данных атомов), необходимо считать свойствами фантома экземпляра системы. Так, у диастереомеров состав атомов и матрица смежности атомов (отражающая структуру соединения) будут совпадать, но взаимодействие с плоскополяризованным светом, биологические активности, некоторые физико-химические свойства могут существенно отличаться в зависимости от пространственной конфигурации. Свойства смесей из химических соединений в разной пространственной конфигурации также имеют особенности. Такие свойства, как температура плавления, теплота плавления и растворимость рацематов (рацемических смесей) в твердом состоянии отличаются от тех же свойств чистых энантиомеров. Так, (+) и (-)-энантиомеры винной кислоты плавятся при 170°C, а рацемическая винная кислота – при 204–206°C [Реутов et al., 2021]. В то же время физико-химические свойства R/S(D/L)-энантиомеров будут совпадать почти полностью в газовой фазе и ахиральной жидкой среде, за исключением знака вращения. Если же хирален реагент, растворитель или катализатор в жидкой среде, то скорости реакций у энантиомеров будут различаться.

Типы значений, связанные со сквозными понятиями, имеют следующий вид: “[Сквозное понятие]”, “[Сквозное понятие]”. “[Сквозное понятие]” означает, что аргументом или компонентой возвращаемого значения (для функций) может быть один экземпляр сквозного понятия, а “[Сквозное понятие]” означает множество экземпляров сквозных понятий. Например, задание экспертом в схеме определения функции аргумента “Момент наблюдения” означает, что при заполнении базы знаний будет проверено, что в данную функцию передан в качестве аргумента единственный момент наблюдения. Напротив, задание экспертом в схеме определения функции аргумента “[Сквозное понятие]” означает, что при заполнении базы знаний будет проверено, что в данную функцию передано в качестве аргумента множество моментов наблюдения. Сквозные понятия могут быть включены в схему определения функций и предикатов, если они заданы для создаваемого проекта в целом.

Типы значений, связанные со вспомогательными понятиями базы знаний, имеют следующий вид: “[Вспомогательное понятие]”, “[Вспомогательное понятие]” (по аналогии со

сквозными понятиями предыдущего абзаца: аргумент предиката “Тип гибридизации” в схеме определения предиката означает, что в базе знаний для данного аргумента предиката, заданного экспертом, должен быть выбран единственный элемент множества  $\{sp, sp2, sp3\}$  (если, конечно, именно такие значения были определены для вспомогательного понятия “Тип гибридизации” при его определении), “{Тип гибридизации”, напротив, означает, что в базе знаний должно быть задано подмножество множества  $\{sp, sp2, sp3\}$ ). Вспомогательные понятия могут быть локальными или глобальными и должны быть полностью заданы до выбора их в качестве аргумента или компонентов возвращаемого значения в конструкторе функций и предикатов.

На рис. 4 показан конструктор функций с произвольным набором аргументов. Аргументы сконструированной экспертом с помощью данного конструктора функции – экземпляр вспомогательного понятия “Тип гибридизации” и единственный экземпляр системы “Атом”, а возвращаемое значение – действительное число из отрезка  $[0, 180]$ . После сохранения функция станет доступной в базе знаний.

Проект: Определение реакционных способностей ХимСоед  
Модуль проекта: Свойства химических соединений

Конструктор функций с произвольным набором аргументов

Название функции\*

Угол между гибридными орбиталями

Описание функции

Значение размера угла между гибридными орбиталями атома

Выбор аргументов функции

Номер аргумента\*

1

Тип значения аргумента\*

Тип гибридизации [module: Свойства химических соединений]

Добавить аргумент

Выбор компонентов возвращаемого значения функции

Номер компоненты возвращаемого значения\*

1

Тип значения компоненты возвращаемого значения\*

[R, R]

Левая граница для номера 1: \*

0

Правая граница для номера 1: \*

180

Добавить компоненту возвращаемого значения Сохранить

Рис. 4. Функция “Угол между гибридными орбиталями”, определяемая экспертом в конструкторе функций с произвольным набором аргументов  
Fig. 4. Function “Angle between hybrid orbitals” declared by the expert with the help of function constructor with arbitrary argument list

### Конструктор функций и предикатов метапонятий

Данный конструктор отличается от конструктора с произвольным набором аргументов тем, что аргументы, связанные с системами метаграфа, и аргументы, связанные со сквозными понятиями, определяются метапонятием однозначно и являются обязательными аргументами конструируемой функции или предиката. Эксперту только необходимо указать один из четырех вариантов альтернатив, состоящих в том, будет ли аргументом один экземпляр системы узла метапонятия, либо множество экземпляров системы, либо один фантом экземпляра системы, либо множество фантомов экземпляра системы. Сквозные понятия, если они указаны для метапонятия (если метапонятие является элементом множества  $MOut$ ), также

становятся обязательными аргументами в схеме определения функции или предиката метапонятия. Остальные типы аргументов (доступные для выбора в конструкторе с произвольным набором аргументов) являются необязательными и могут быть выбраны экспертом для определенных функций и предикатов. На рис. 5 представлена схема формирования обязательных аргументов функции или предиката метапонятия без учета сквозных понятий (метапонятие является элементом множества  $M$ ), а на рис. 6 представлена схема формирования необязательных аргументов функции или предиката метапонятия без учета сквозных понятий (метапонятие является элементом множества  $M$ ).

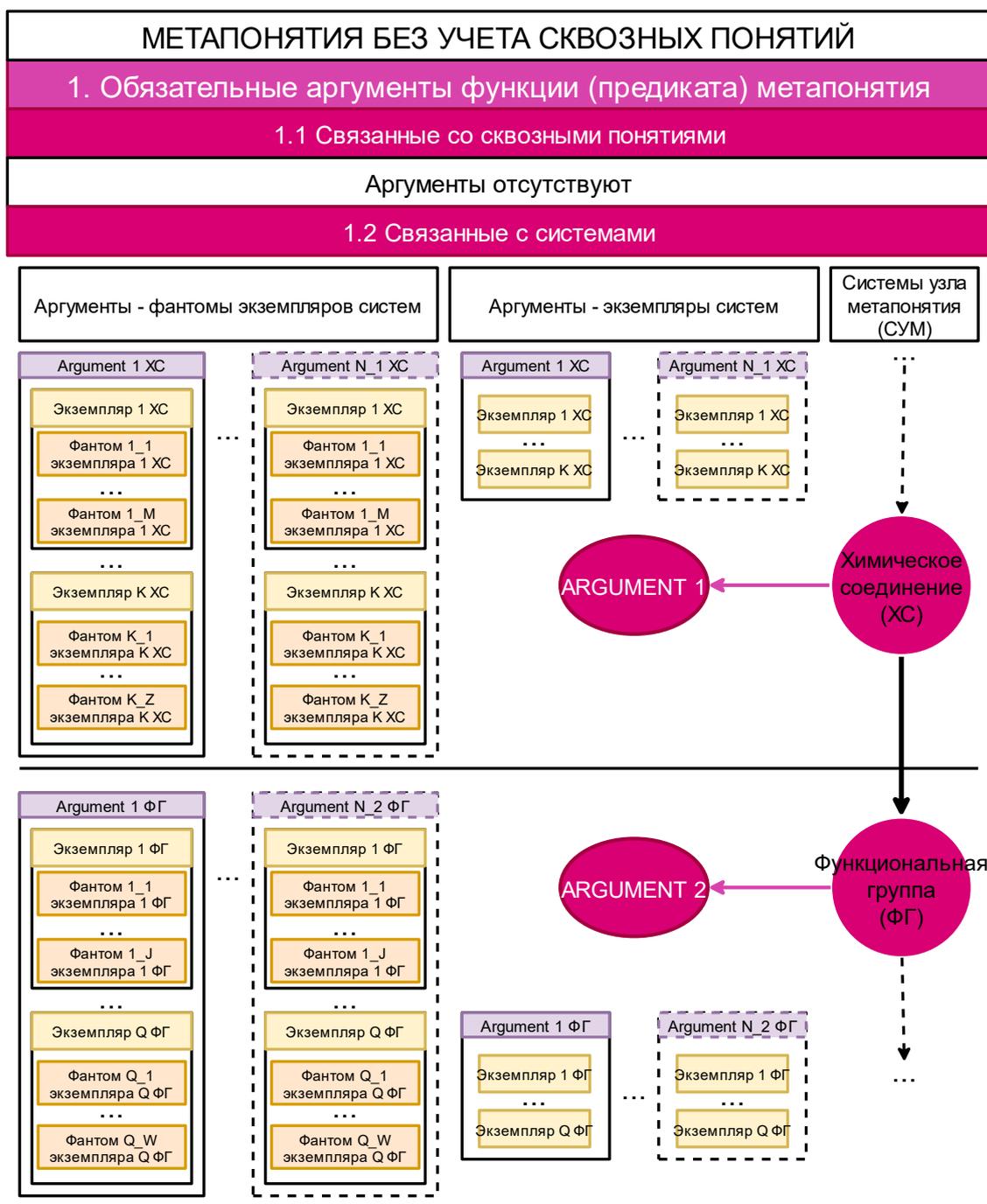


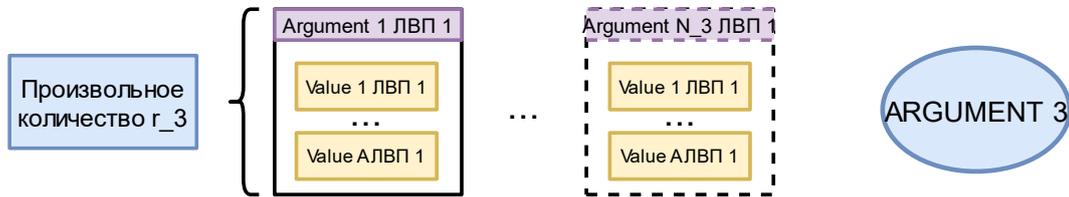
Рис. 5. Схема формирования аргументов функции/предиката метапонятия без учета сквозных понятий (обязательные аргументы)

Fig. 5. Metaconcept function/predicate argument formation scheme without outer concepts (required arguments)

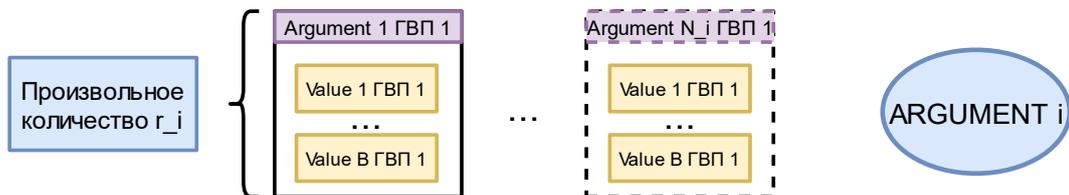
## 2. Необязательные аргументы функции (предиката) метапонятия

### 2.1 Связанные с вспомогательными понятиями

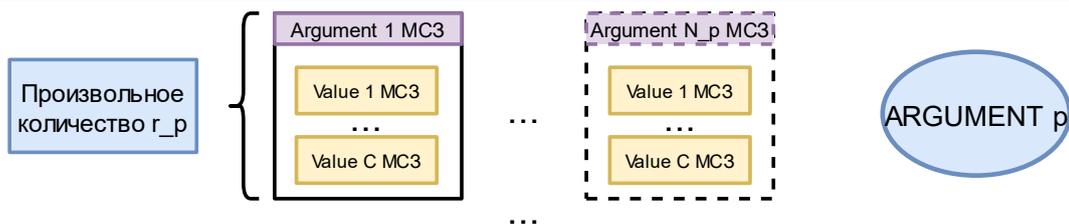
#### 2.1.1 Связанные с локальными вспомогательными понятиями (ЛВП)



#### 2.1.2 Связанные с глобальными вспомогательными понятиями (ГВП)



#### 2.2 Связанные со стандартными значениями. Множества стандартных значений (МСЗ): имён (N), логических значений (L), целых чисел (I), действительных чисел (R), комплексных чисел (C)\*



\* - язык прикладной логики

Рис. 6. Схема формирования аргументов функции/предиката метапонятия без учета сквозных понятий (необязательные аргументы)

Fig. 6. Metaconcept function/predicate argument formation scheme without outer concepts (optional arguments)

На рис. 7 представлен интерфейс эксперта при задании схемы определения предиката с помощью конструктора предиката метапонятия (с учетом сквозных понятий).

### Генератор предикатов внутреннего устройства систем

Для тех систем, для которых экспертом был активирован checkbox “Отождествить с прямой надсистемой”, автоматически создаются предикаты внутреннего устройства систем “Являться подсистемой экземпляра системы([Система], [Прямая подсистема])” и “Являться подсистемой фантома экземпляра системы([Система], [Прямая подсистема])”. Например, в соответствии с метаграфом (см. рис. 1) будут сформированы предикаты внутреннего устройства систем “Являться подсистемой экземпляра системы(Химическое соединение (D), Функциональная группа (S))” и “Являться подсистемой фантома экземпляра системы(Химическое соединение (D), Функциональная группа (S))”.

RAPIDIS | Активные проекты | Сквозные понятия | Единицы измерения | Представления | Стратегии | Решатели | Приветствую, Fedor\_Privalov! | Выход

**Проект: Определение реакционных способностей ХимСоед**  
 Модуль проекта: Термодинамические характеристики

Метапонятие: Свойства функциональных групп реакционного центра субстрата при изменении температуры в химических реакциях

**Конструктор предиката метапонятия (с учетом сквозных понятий)**

Название предиката\*

Описание предиката

Тип возвращаемого значения  
L

**Выбор аргументов предиката**  
 Обязательные аргументы-системы узла метапонятия:

Номер аргумента\*  
1

Тип значения аргумента\*  
{Функциональная группа\_КВ}

Номер аргумента\*  
2

Тип значения аргумента\*  
Химическое соединение\_КВ

Номер аргумента\*  
3

Тип значения аргумента\*  
Химическая реакция\_КВ

Обязательные аргументы-сквозные понятия узла метапонятия:

Номер аргумента\*  
4

Температура  
 {}Температура

Необязательные аргументы:

Номер аргумента\*  
5

Тип значения аргумента\*  
Тип гибридизации [module: Свойства химических соединений]

Добавить аргумент-вспомогательное понятие | Добавить аргумент-стандартное значение | Удалить аргумент | Сохранить

Рис. 7. Интерфейс эксперта при задании схемы определения предиката с помощью конструктора предиката метапонятия (с учётом сквозных понятий)

Fig. 7. System look-and-feel while defining predicate signature with the help of metaconcept predicate constructor (with outer concepts) by the chemist-expert

### *Программный код функций и предикатов*

Если в базе знаний вычислять значения функций и предикатов (включая предикаты внутреннего устройства систем) не требуется, а требуется только указывать значения, то участие программиста в формировании базы знаний не требуется. В базе знаний при выборе определенной функции или предиката по названию, пользователю, заполняющему базу знаний, будут доступны для ввода в поля аргументов только аргументы заданного экспертом (при определении схемы функции или предиката) типа. Аналогично, при выборе пользователем из множества {"Указать", "Вычислить"} элемента "Указать", ему будут доступны для ввода в поля компонентов возвращаемого значения функции только компоненты заданного экспертом типа, что полностью исключает возможность ввода пользователем ошибочных данных в базу знаний (исключение составляют только возможные ошибки эксперта в самих значениях).

Если в базе знаний требуется вычислить значение функции или предиката (включая предикаты внутреннего устройства систем), то пользователем из группы “Программист” предварительно выбираются типы представлений для аргументов (при необходимости) функции или предиката и скачивается шаблон подпрограммы. Для добавления кода подпрограммы, в загруженный с сервера шаблон программист добавляет свой код в функцию `actionFunction` (вместо комментария ““INSERT YOUR CODE HERE””), причем ему необходимо в коде тела функции с названием `actionFunction` изменить на “SUCCESS” переменную `self.finalResultsDict['Results']['res'] = “ERROR”`, которая означает успешную остановку вычисления. На странице добавления подпрограммы программист указывает название и прикрепляет готовый файл с заполненным шаблоном. Происходит валидация файла, проверка его на наличие запрещенных операторов (например, “`exec`”, “`import os`”, и т. п.). Если проверка пройдена, файл будет загружен на сервер в директорию со статическими файлами и далее скопирован в директорию с исполняемыми файлами (содержит файл `__init__.py`). Далее подпрограмма может быть вызвана на исполнение при заполнении базы знаний. При попытке программистом загрузки файла с кодом подпрограммы, содержащей запрещенные операторы, будет выдано сообщение об ошибке.

На рис. 8 приведен пример результата работы подпрограммы экземпляра предиката внутреннего устройства систем “Являться подсистемой экземпляра системы(Химическое соединение (D), Функциональная группа (S))” для экземпляра “Амитриптилин” системы “Химическое соединение (D)” и экземпляра “Бензол” системы “Функциональная группа (S)”. Подпрограмма реализует алгоритм, изначально предложенный J. Ullmann для графов без специальной разметки (не содержала молекулярную информацию) [Ullmann, 1976], но модифицированный для поиска изоморфного вложения подграфа в граф молекулы (молекулы представлены файлами в формате BIOVIA Molfile V2000 [BIOVIA Discovery Studio CTfile formats]). Возвращенное после исполнения подпрограммы представление – рис. 9. Распознавание профессиональным химиком функциональных групп в структуре химического соединения является одним из примеров интеллектуальной деятельности человека-специалиста предметной области, которую интеллектуальная система определения реакционных способностей химических соединений призвана имитировать.

**Создание экземпляров предикатов внутреннего устройства систем:**

Предикат\*

Метод получения значения предиката\*

Система\*

Подсистема\*

Истина

Сохранен экземпляр предиката [Являться подсистемой экземпляра системы(Химическое соединение\_KB, Функциональная группа\_KB)] с результатом [true]  
 Результат предиката успешно вычислен.

**Дополнительные представления:**

Дополнительное представление номер 1  
 Тип представления: BIOVIA Molfile V2000. Ссылка: [Файл](#)

Созданный экземпляр предиката внутреннего устройства систем:

Рис. 8. Результат работы подпрограммы, вычисляющей значение экземпляра предиката внутреннего устройства систем “Являться подсистемой экземпляра системы(Химическое соединение (D), Функциональная группа (S))” для экземпляра “Амитриптилин” системы “Химическое соединение (D)” и экземпляра “Бензол” системы “Функциональная группа (S)”

Fig. 8. The result returned by the subprogram that calculate the value of the system inner predicate instance “To be subsystem of the system instance(Chemical compound (D), Functional group (S))” for “Amitriptyline” instance of the system “Chemical compound (D)” and “Benzene” instance of the system “Functional group (S)”

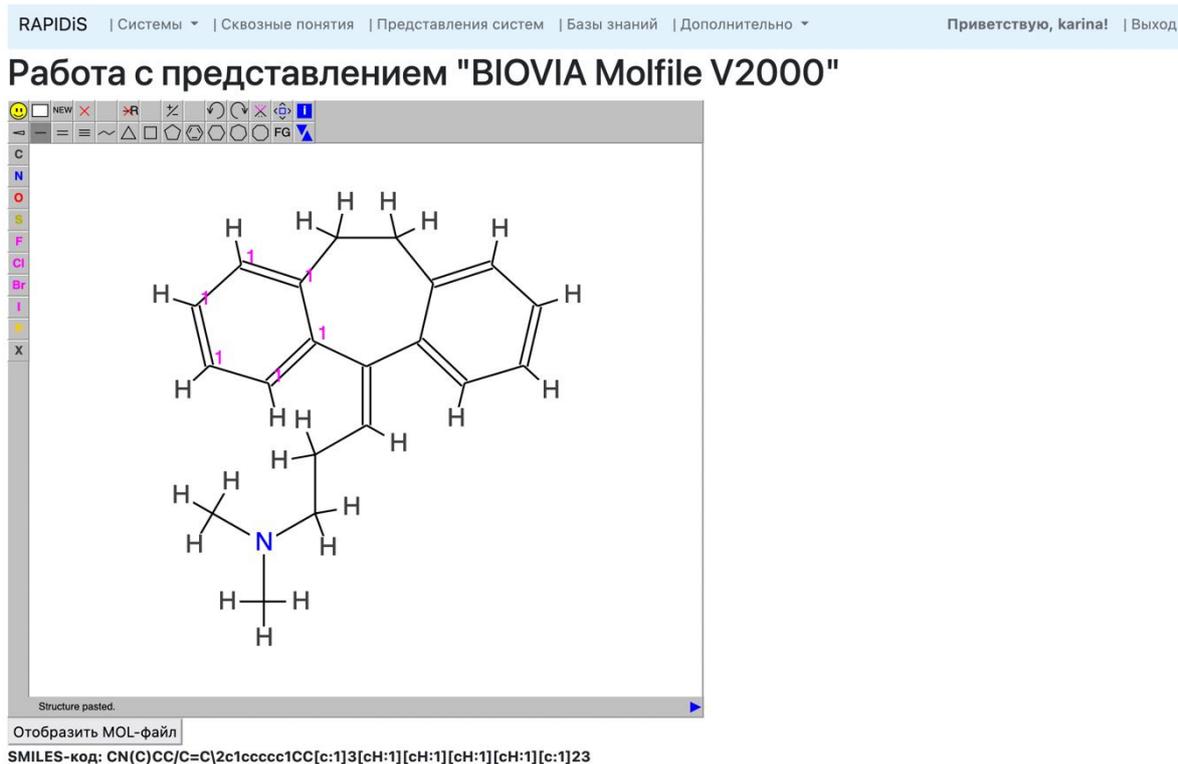


Рис. 9. Дополнительное представление, возвращенное подпрограммой, вычисляющей значение экземпляра предиката внутреннего устройства систем “Являться подсистемой экземпляра системы(Химическое соединение (D), Функциональная группа (S))” при поиске изоморфного вложения бензола в молекулу amitриптилина представления BIOVIA Molfile V2000

Fig. 9. Additional representation returned by the subprogram that calculate the value of the system inner predicate instance “To be subsystem of the system instance(Chemical compound (D), Functional group (S))” during subgraph isomorphism search for benzene in amitriptyline represented by BIOVIA Molfile V2000

## Заключение

В настоящей работе предьявлен набор инструментальных средств поддержки процесса формирования меронимических систем понятий и баз знаний для предметных областей со сложно-структурированными объектами (системами). В качестве иллюстративного примера предметной области такого типа выбрана задача определения реакционных способностей химических соединений. Данные системы понятий и основанные на них базы знаний характеризуются тем, что содержат информацию о структуре сложно-структурированной системы (подсистемы которых могут быть элементарного типа, дескриптивного типа, системой-подмножеством и системой-процессом), критериях вхождения экземпляров подсистем в экземпляр сложно-структурированной системы, свойствах самой системы и свойствах самой системы при взаимодействии со средой, свойствах каждой из ее подсистем и свойствах каждой из ее подсистем при взаимодействии со средой, свойствах сочетаний каждой из ее подсистем и свойствах сочетаний каждой из ее подсистем при взаимодействии со средой и т. д.

В связи с закономерным уходом инженеров знаний из процесса формирования систем понятий, выбор схемы базы знаний, искусный подбор промежуточных понятий, которые бы позволили сформировать исчерпывающие наборы основных понятий предметной области, – остались за экспертами предметных областей, что сделало актуальными исследования, позволяющие экспертам снизить количество усилий, затрачиваемых на всех этапах процесса создания систем понятий и баз знаний на их основе.

В работе [Гуляева, Артемьева, 2023] предложена технология автоматизации процесса формирования систем понятий и баз знаний для предметных областей с холомеронимическими отношениями, заключающаяся в формировании промежуточных понятий



на основании метаграфа системы сложной структуры, процедуры исследования окрестностей вершин метаграфа и введения специального множества метапонятий. Настоящая работа предлагает набор разработанных инструментальных средств поддержки указанного процесса для предметных областей со сложно-структурированными объектами (на примере задачи определения реакционных способностей химических соединений).

Основой данного набора инструментов служит гибридный редактор, работающий в двух режимах (максимальной и минимальной) автоматизации процесса формирования систем понятий и баз знаний указанного типа и позволяющий эксперту (в отличие от редакторов орграфовых моделей, требующих обязательного участия программиста для написания кода агентов, шаблонов сообщений для генерации участков подсети при автоматизации процесса формирования промежуточных понятий) по своему выбору – как затратить минимальное количество усилий при создании промежуточных понятий и меронимической системы понятий в целом (редактор в режиме работы с максимальной автоматизацией подставляет обязательные аргументы функций и предикатов автоматически при определении экспертом их сигнатуры, что сокращает количество элементов в списке с доступными для выбора аргументами и суммарное количество соответствующих элементарных действий эксперта) – так и максимально настроить процесс.

Основными элементами гибридного редактора являются редактор метаграфа сложно-структурированной системы, генератор метапонятий и конструктор функций и предикатов.

С помощью редактора метаграфа сложно-структурированной системы экспертом создается сама сложно-структурированная система, подсистемами которой на различных уровнях иерархии могут быть системы четырех описанных структурных типов.

Генератор метапонятий автоматически формирует множество метапонятий, соответствующих промежуточным понятиям меронимической системы понятий. Это дает эксперту возможность систематически исследовать наборы отношений экземпляров систем узла.

Эксперт задает схемы определения функций и предикатов посредством конструктора функций и предикатов с произвольным набором аргументов и конструктора функций и предикатов метапонятий. Последний автоматически подставляет аргументы, соответствующие системам и сквозным понятиям, упрощая работу эксперта предметной области при формировании баз знаний указанного типа.

Набор разработанных инструментальных средств позволяет создавать декларативные и процедурные компоненты интеллектуальной системы определения реакционных способностей химических соединений, имитирующей процесс логического вывода профессионального химика, распознающего функциональные группы и их свойства, обосновывающего свой прогноз относительно реакционной способности отдельного химического соединения на основании развитой базы знаний, накопленной за десятилетия теоретической и практической работы в предметной области.

Набор разработанных инструментальных средств реализован в программе [Программа “RAPIDiS”]. Данные инструменты также могут быть использованы и в других предметных областях со сложно-структурированными объектами (системами).

### Список источников

- ГОСТ Р ИСО 9000-2001 Системы менеджмента качества. Основные положения и словарь. ИПК Издательство стандартов. 2001. 68 с.
- Программа для ЭВМ “Программа “RAPIDiS” для создания систем с базами знаний в предметных областях, объекты которых являются системами сложной структуры”. Свидетельство Федеральной службы по интеллектуальной собственности (РОСПАТЕНТ) о государственной регистрации программы для ЭВМ № 2023669863. (Зарегистрировано в Реестре программ для ЭВМ 21 сентября 2023 г.).

## Список литературы

- Голенков В.В., Гулякина Н.А. 2014. Проект открытой семантической технологии компонентного проектирования интеллектуальных систем. Часть 1: Принципы создания. Онтология проектирования, 1(11): 42–64.
- Грибова В.В., Клещев А.С., Москаленко Ф.М., Тимченко В.А., Шалфеева Е.А., Федорищев Л.А. 2016. Методы и средства разработки жизнеспособных интеллектуальных сервисов. Вестник ДВО РАН. Владивосток: Изд. «Дальнаука», 4: 133–141.
- Грибова В.В., Клещев А.С., Шалфеева Е.А. 2010. Управление интеллектуальными системами. Известия РАН. Теория и системы управления, 6: 122–137.
- Гуляева К.А., Артемьева И.Л. 2023. Технология автоматизации процесса формирования систем понятий и баз знаний для предметных областей с объектами сложной структуры. Вестник Рязанского государственного радиотехнического университета, 3(85): 65–81.
- Загорулько Ю.А., Боровикова О.И. 2007. Технология построения онтологий для порталов научных знаний. Вестник НГУ. Серия: Информационные технологии, 5(2): 42–52.
- Искандарова Г.Р. 2019. Исследование меронимической лексики в современном немецком и английском языке. Известия ВГПУ. Филологические науки, 4: 186–189. <https://doi.org/10.3726/978-3-653-03561-2>
- Клещев А.С., Орлов В.А. 2006. Компьютерные банки знаний. Универсальный подход к решению проблемы редактирования информации. Информационные технологии, 5: 25–31.
- Колодыко Д.А. 2016. Меронимические отношения как проявление системности лексики. Вестник Самарского университета. История, педагогика, филология, 3.2(22): 270–275.
- Кузьменко Н.В. 2015. О меронимических связях в лексической системе языка (на материале наименований частей тела современного английского языка). Вестн. Вят. гос. гуманит. ун-та, 10: 58–64.
- Реутов О.А., Курц А.Л., Бутин К.П. 2021. Органическая химия в 4 ч. Ч. 2. 10-е Изд. М., Лаборатория знаний, 623 с.
- Ahmad M. N., Colomb R. M., Sadiq S. 2008. A Relevant Portion of an Ontology: Defining a System of ED Rules Using a Part-Whole Relationship. 2008 Second Asia International Conference on Modelling & Simulation (AMS), Kuala Lumpur, Malaysia: 46–51. <https://doi.org/10.1109/AMS.2008.28>
- Aitken J.S., Webber B.L., Bard J.B.L. 2004. Part-of relations in anatomy ontologies: a proposal for RDFs and OWL formalisations. Pacific Symposium on Biocomputing, 9:166–177.
- BIOVIA Discovery Studio CTfile formats. Режим доступа: <https://discover.3ds.com/ctfile-documentation-request-form>
- BIOVIA Discovery Studio. Режим доступа: <https://discover.3ds.com/discovery-studio-visualizer-download>
- ChemSpider. Режим доступа: <http://www.chemspider.com>
- GAUSSIAN. Режим доступа: <https://gaussian.com/>
- GPAW. Режим доступа: <https://wiki.fysik.dtu.dk/gpaw/>
- GROMACS. Режим доступа: <https://www.gromacs.org/>
- IACPaaS. Режим доступа: <https://iacpaas.dvo.ru/>
- John В.Е., Kieras D.E. 1996. The GOMS family of user interface analysis techniques: Comparison and contrast. ACM Transactions on Computer-Human Interaction, 3: 320–351.
- LAMMPS. Режим доступа: <https://www.lammps.org/>
- Merck Index Online. Режим доступа: <http://www.rsc.org/Merck-Index/>
- Molbase. Режим доступа: <https://www.molbase.com/>
- Morales-González A., Fernández-Reyes F.C., Keet C.M. 2015. ONTOPARTS: A Tool to Select Part-Whole Relations in OWL Ontologies. In: Simperl, E., et al. The Semantic Web: ESWC 2012 Satellite Events. ESWC 2012. Lecture Notes in Computer Science, vol 7540. Springer, Berlin, Heidelberg: 452–457. [https://doi.org/10.1007/978-3-662-46641-4\\_44](https://doi.org/10.1007/978-3-662-46641-4_44)
- Musen M. 2015. The Protégé Project: A Look Back and a Look Forward. AI Matters., 1(4): 4–12.
- NAMD. Режим доступа: <https://www.ks.uiuc.edu/Research/namd/>
- NIST Chemistry Webbook. Режим доступа: <https://webbook.nist.gov/chemistry/>
- Protein Data Bank (PDB). Режим доступа: <http://www wwptdb.org/>
- PubChem. Режим доступа: <http://pubchem.ncbi.nlm.nih.gov/>
- Reaxys. An expert-curated chemistry database. Режим доступа: <https://www.elsevier.com/solutions/reaxys>
- SciFinder. Режим доступа: <http://scifinder.cas.org>
- SPRESI. Режим доступа: <http://www.spresi.com>



Ullmann J.R. 1976. An algorithm for subgraph isomorphism. *Journal of the Association for Computing Machinery*, 23(1): 31–42. <https://dl.acm.org/doi/pdf/10.1145/321921.321925>  
ZINC. Режим доступа: <http://zinc.docking.org>

## References

- Golenkov V.V., Guliakina N.A. 2014. Project of Open Semantic Technology of The Componental Design of Intelligent Systems. Part 1: The Principles of Creation. *Ontology of designing*, 1(11): 42–64 (in Russian).
- Gribova V.V., Kleshev A.S., Moskalenko F.M., Timchenko V.A., Shalfeeva E.A., Fedorischev L.A. 2016. Methods and means of designing viable intelligent services. *Vestnik DVO RAN. Vladivostok: Izd. "Dalnauka"*, 4: 133–141 (in Russian).
- Gribova V.V., Kleshev A.S., Shalfeeva Ye.A. 2010. Intelligent system control. *Izvestiya RAN. Teoriya i sistemi upravleniya*, 6: 122–137 (in Russian).
- Gulyaeva K.A., Artemieva I.L. 2023. The technology of automated concept system and knowledge base formation process for application domains with complex-structured objects. *Vestnik Ryazanskogo gosudarstvennogo radiotekhnicheskogo universiteta*, 3(85): 65–81 (in Russian).
- Zagorulko Yu.A., Borovikova O.I. 2007. Ontology design technology for scientific portals. *Vestnik NGU. Seriya: Informatsionnie tekhnologii*, 5(2): 42–52 (in Russian).
- Iskandarova G.R. 2019. Study of meronymic vocabulary in modern German and English. *Izvestiya VGPU. Filologicheskie nauki*, 4: 186–189 (in Russian). <https://doi.org/10.3726/978-3-653-03561-2>
- Kleshev A.S., Orlov V.A. 2006. Computer knowledge banks. Universal approach to information editing problem. *Informatsionnie tekhnologii*, 5: 25–31 (in Russian).
- Kolodko D.A. 2016. Meronymic relations as a manifestation of the systematic nature of vocabulary. *Vestnik Samarskogo universiteta. Istorija, pedagogika, filologija*, 3.2(22): 270–275 (in Russian).
- Kuzmenko N.V. 2015. On meronymic connections in the lexical system of the language (based on the body part names of the modern English language). *Vestnik Vyat. Gos. Gumanitarnogo Universiteta*, 10: 58–64 (in Russian).
- Reutov O.A., Kurtz A.L., Butin K.P. 2021. Organic chemistry in 4 vol. Vol.2. 10-ed. M., Laboratoriya znanii, 623 p (in Russian).
- Ahmad M. N., Colomb R. M., Sadiq S. 2008. A Relevant Portion of an Ontology: Defining a System of ED Rules Using a Part-Whole Relationship. 2008 Second Asia International Conference on Modelling & Simulation (AMS), Kuala Lumpur, Malaysia: 46–51. <https://doi.org/10.1109/AMS.2008.28>
- Aitken J.S., Webber B.L., Bard J.B.L. 2004. Part-of relations in anatomy ontologies: a proposal for RDFs and OWL formalisations. *Pacific Symposium on Biocomputing*, 9:166–177.
- BIOVIA Discovery Studio CTfile formats. Access mode: <https://discover.3ds.com/ctfile-documentation-request-form>
- BIOVIA Discovery Studio. Access mode: <https://discover.3ds.com/discovery-studio-visualizer-download>
- ChemSpider. Access mode: <http://www.chemspider.com>
- GAUSSIAN. Access mode: <https://gaussian.com/>
- GPAW. Access mode: <https://wiki.fysik.dtu.dk/gpaw/>
- GROMACS. Access mode: <https://www.gromacs.org/>
- IACPaaS. Access mode: <https://iacpaas.dvo.ru/>
- John B.E., Kieras D.E. 1996. The GOMS family of user interface analysis techniques: Comparison and contrast. *ACM Transactions on Computer-Human Interaction*, 3: 320–351.
- LAMMPS. Access mode: <https://www.lammps.org/>
- Merck Index Online. Access mode: <http://www.rsc.org/Merck-Index/>
- Molbase. Access mode: <https://www.molbase.com/>
- Morales-González A., Fernández-Reyes F.C., Keet C.M. 2015. ONTOPARTS: A Tool to Select Part-Whole Relations in OWL Ontologies. In: Simperl, E., et al. *The Semantic Web: ESWC 2012 Satellite Events. ESWC 2012. Lecture Notes in Computer Science*, vol 7540. Springer, Berlin, Heidelberg: 452–457. [https://doi.org/10.1007/978-3-662-46641-4\\_44](https://doi.org/10.1007/978-3-662-46641-4_44)
- Musen M. 2015. The Protégé Project: A Look Back and a Look Forward. *AI Matters.*, 1(4): 4–12.
- NAMD. Access mode: <https://www.ks.uiuc.edu/Research/namd/>
- NIST Chemistry Webbook. Access mode: <https://webbook.nist.gov/chemistry/>
- Protein Data Bank (PDB). Access mode: <http://www.wwpdb.org/>
- PubChem. Access mode: <http://pubchem.ncbi.nlm.nih.gov/>



Reaxys. An expert-curated chemistry database. Access mode: <https://www.elsevier.com/solutions/reaxys>  
SciFinder. Access mode: <http://scifinder.cas.org>  
SPRESI. Access mode: <http://www.spresi.com>  
Ullmann J.R. 1976. An algorithm for subgraph isomorphism. *Journal of the Association for Computing Machinery*, 23(1): 31–42. <https://dl.acm.org/doi/pdf/10.1145/321921.321925>  
ZINC. Access mode: <http://zinc.docking.org>

**Конфликт интересов:** о потенциальном конфликте интересов не сообщалось.

**Conflict of interest:** no potential conflict of interest related to this article was reported.

Поступила в редакцию 18.09.2023

Received September 18, 2023

Поступила после рецензирования 03.03.2024

Revised March 03, 2024

Принята к публикации 05.06.2024

Accepted June 05, 2024

## ИНФОРМАЦИЯ ОБ АВТОРАХ

## INFORMATION ABOUT THE AUTHORS

**Гуляева Карина Александровна**, старший преподаватель Департамента программной инженерии и искусственного интеллекта Института математики и компьютерных технологий, Дальневосточный федеральный университет, г. Владивосток, Россия

**Karina A. Gulyaeva**, Senior Tutor of the Department of Software Engineering and Artificial Intelligence, Institute of Mathematics and Computer Technologies, Far Eastern Federal University, Vladivostok, Russia

**Артемьева Ирина Леонидовна**, доктор технических наук, профессор Департамента программной инженерии и искусственного интеллекта Института математики и компьютерных технологий, Дальневосточный федеральный университет, г. Владивосток, Россия

**Irina L. Artemieva**, Doctor of Technical Sciences, professor of the Department of Software Engineering and Artificial Intelligence, Institute of Mathematics and Computer Technologies, Far Eastern Federal University, Vladivostok, Russia