

УДК 004.89

ИНСТРУМЕНТАРИЙ ДЛЯ ПРОЕКТИРОВАНИЯ СИСТЕМ ПОНЯТИЙ И БАЗ ЗНАНИЙ ПРЕДМЕТНЫХ ОБЛАСТЕЙ С ОБЪЕКТАМИ СЛОЖНОЙ СТРУКТУРЫ

К. А. Гуляева, старший преподаватель департамента ПИИИИ ИМКТ ДВФУ, Владивосток, Россия;
orcid.org/0000-0003-0226-5072, e-mail: karinagulyaeva8@gmail.com

И. Л. Артемьева, д.т.н., профессор департамента ПИИИИ ИМКТ ДВФУ, Владивосток, Россия;
orcid.org/0000-0003-2088-5259, e-mail: artemeva.il@dvfu.ru

Системы понятий и базы знаний предметных областей со сложно-структурированными объектами быстро увеличиваются в объеме. Трудоемкую работу по проектированию базы знаний, которая включает в себя выбор ее схемы, искусный подбор промежуточных понятий, избрание вариантов ее наполнения, в настоящее время выполняют эксперты предметных областей. Возникает вопрос о необходимости автоматизации. Можно ли предъявить специализированный набор средств поддержки экспертов? Целью работы являются разработка и исследование программного инструментария для структуризации систем прикладных понятий и основанных на них баз знаний для предметных областей с объектами сложной структуры (на примере задачи определения реакционных способностей химических соединений). В представленном исследовании предлагается для решения этого вопроса использовать гибридный редактор, работающий в двух режимах автоматизации, который позволяет эксперту при разработке системы понятий и базы знаний либо затратить минимальное количество усилий, либо максимально настроить процесс. Основными элементами гибридного редактора являются редактор метаграфа объекта сложной структуры, генератор метапонятий и конструктор функций и предикатов. С помощью редактора метаграфа создаются объекты сложной структуры. Подсистемами таких объектов могут быть системы четырех структурных типов: элементарная система, дескриптивная система, система-подмножество и система-процесс. Генератор метапонятий автоматически формирует множество метапонятий, соответствующих промежуточным понятиям онтологии. Схемы определения функций и предикатов задаются экспертом посредством конструктора функций и предикатов с произвольным набором аргументов и конструктора функций и предикатов метапонятий. Последний автоматически подставляет часть допустимых для выбора аргументов функций и предикатов при проектировании их сигнатур экспертом предметной области, тем самым упрощая его работу при разработке систем понятий и баз знаний указанного типа.

Ключевые слова: база знаний, интеллектуальная система, онтология, предметная область, система сложной структуры, прикладная логическая теория, промежуточные понятия онтологии, реакционная способность.

DOI: 10.21667/1995-4565-2024-87-62-77

Введение

Меронимические отношения (или отношения «часть – целое») – одна из основ сложности и разнообразия окружающего нас мира. Профессионалы конкретных предметных областей, системные аналитики, логики (в частности, мереологи), онтологи, разработчики экспертных систем, философы, специалисты в области семантики посвящают свои работы различным аспектам данного отношения [1-6]. Неоднократно было показано (Аристотель, Г.В.Ф. Гегель и др.), что сложное внутреннее устройство свойственно как объектам, так и процессам. Далее объекты и процессы, находящиеся на различных уровнях иерархии в отношении «часть – целое», для удобства изложения будем называть термином «сложно-структурированные объекты (системы)».

Одним из важных примеров предметных областей, в которых отношение меронимии играет существенную роль, является задача определения реакционных способностей химиче-

ских 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 «что», «с чем» и «как» реагирует, химики ищут ответы, применяя как методы квантовой химии, так и проверенные эмпирические правила (англ. «rule of thumb»). Насколько прогноз будет соответствовать действительности, зависит от знаний конкретного эксперта. Интенсификация процедур получения новых данных и знаний в указанной 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, показал, что в них содержится информация различных уровней детализации о внутрисистемных, межсистемных связях и связях объектов указанной предметной области с внешней средой. Она зависит от типа решаемой ими задачи. В обзоре были проанализированы: базы данных химических соединений и химических реакций Reaxys [7], PubChem [8], Merck Index Online [9], Molbase [10], PDB [11], ChemSpider [12], NIST Chemistry Webbook [13], SciFinder [14], SPRESI [15], ZINC [16] и др., позволяющие осуществлять поиск химических соединений по подструктуре, суперструктуре, подобию; пакеты молекулярной динамики GROMACS [17], LAMMPS [18], NAMD [19], BIOVIA Discovery Studio [20], в которых временная эволюция системы взаимодействующих атомов или частиц описывается интегрированием их уравнений движения; пакеты, реализующие методы квантовой химии Gaussian [21], GPAW [22], в частности, реализующие методы теории функционала плотности; QSAR-дескрипторы хемиинформатики, включающие 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. В результате обзора было выявлено, что частая несовместимость входных и выходных форматов, проприетарность лицензий многих продуктов, а также – наиболее важно – высокая степень сращения декларативных и процедурных знаний предметной области препятствуют 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 этого 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 меронимическими отношениями 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 свойства самой системы при взаимодействии со средой, свойства каждой из ее подсистем и свойства каждой из ее подсистем при взаимодействии со средой, свойства 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. Наличествует и проблема отладки баз знаний, сформированных на основе выбранных систем понятий. В соответствии с современными подходами участие в формировании систем понятий принимают только эксперты совместно с программистами [23-28]. Упомянутая трудоемкая работа – одна из главных обязанностей экспертов предметной области. Для того чтобы эксперт затрачивал минимальное количество умственных и физических

усилий, требуются технологии, максимально автоматизирующие разработку и проектирование систем понятий и основанных на них баз знаний указанного типа.

В работе [29] предлагается технология автоматизации процесса формирования систем понятий и баз знаний для предметных областей с объектами сложной структуры, 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 разработанный программный инструмент для структуризации систем прикладных понятий и основанных на них баз знаний для предметных областей с объектами сложной структуры (на примере задачи определения реакционных способностей химических соединений) и обсудить ключевые особенности его разработки. Основой представленного набора инструментов служит гибридный редактор, работающий в двух режимах автоматизации и позволяющий эксперту либо затратить минимальное количество усилий при разработке промежуточных понятий онтологии, либо максимально настроить процесс (по своему выбору). В отличие от редактора [30], требующего обязательного участия программиста для генерации участков подсети при автоматизации процесса создания промежуточных понятий (для разработки агентов, шаблонов сообщений и др.), представленное решение поддерживает автоматическую генерацию промежуточных понятий онтологии на уровне редактора. Количество фактически затраченных усилий при взаимодействии эксперта с интерфейсом может быть рассчитано методами GOMS, KLM [31] (при этом необходимо учитывать, что результаты зависят от весов, назначенных атомарным действиям при взаимодействии с интерфейсом когнитологами). Представленный здесь редактор в режиме работы с максимальной автоматизацией подставляет обязательные аргументы функций и предикатов при определении экспертом их сигнатуры автоматически. При этом сокращается количество элементов в списке с доступными для выбора аргументами, что сокращает количество соответствующих элементарных действий эксперта (подробности технологии приведены далее).

Объекты и методы исследования

Объектом исследования настоящей работы являются системы понятий предметных областей с 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 средств поддержки проектирования меронимических систем понятий и основанных на них баз знаний (на примере задачи определения реакционных способностей химических соединений).

Результаты и их обсуждение

Автоматизация проектирования и разработки меронимических систем понятий и основанных на них баз знаний основана на использовании набора инструментальных средств, именуемых далее гибридный редактор. Его основными элементами являются редактор метаграфа системы сложной структуры, генератор метапонятий и конструктор функций и предикатов. Каждый инструмент, входящий в гибридный редактор, обсуждается далее отдельно.

Редактор метаграфа системы сложной структуры

Системой сложной структуры будем называть объект (систему) предметной области, в котором можно выделить подсистемы следующих типов: *элементарная система*, *дескриптивная система*, *система-подмножество* и *система-процесс*. *Элементарная система* [обозначение – Elementary System (аббр. Е)] подсистем не имеет и может иметь только один экземпляр. Ее прямая надсистема ассоциирует с минимальным и максимальным количествами ее единственного экземпляра два числа из множества $\mathbb{N} \cup \{0\}$. Максимальное количество мо-

жет быть не ограничено (и обозначается элементом $\inf$, добавленным в указанное множество: $N \cup \{0\} \cup \{\inf\}$). *Дескриптивная система* [обозначение – *Descriptive Element System* (аббр. 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* (аббр. 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, экземпляр «Циклопентадиен» системы «Функциональная группа», которая является прямой подсистемой системы «Химическое соединение», может рассматриваться и как экземпляр системы «Химическое соединение»). *Система-процесс* [обозначение – *Process* (аббр. P)] не является элементарной и обозначает совокупность взаимосвязанных или взаимодействующих видов деятельности, преобразующих входы в выходы. Для системы-процесса должны быть заданы хотя бы один участник процесса (ее прямая подсистема) с указанием роли и максимальное количество шагов процесса.

Для представления сложно-структурированного объекта (системы) используется направленный ациклический метаграф (G, μ_G) , который называется *метаграфом* сложно-структурированной системы, где G – направленный ациклический граф, μ_G – разметка графа G . Введем следующие обозначения: *Terminal* – множество терминальных вершин графа G , *Nonterminal* – множество нетерминальных вершин графа G , *SystemNames* – множество имен вершин метаграфа сложно-структурированной системы, R – бинарное отношение на множестве вершин $Terminal \cup Nonterminal$, R – множество ребер графа G . $R \subseteq (Terminal \cup Nonterminal) \times (Terminal \cup Nonterminal)$, где $\times$ обозначает декартово произведение множеств. R специфицирует ориентированное ребро из вершины n в другую вершину m , если $(n, m) \in R$. Транзитивное замыкание R^+ отношения R нерефлексивно (условие ацикличности направленного ациклического графа). Разметка $\mu_G : Terminal \cup Nonterminal \rightarrow SystemNames$; $\mu_G : R \rightarrow \{\text{Являться прямой надсистемой (подсистемой)}\}$. *SystemNames* включает имена систем четырех описанных структурных типов $\{Elementary System (E), Descriptive Element System (D), Descriptive Subset System (S), Process (P)\}$. Пример метаграфа сложно-структурированной системы «Химическая реакция» и подробности его формирования приведены в работе [29]. Далее приводятся лишь некоторые важные замечания.

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). 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 $\{Elementary System (E), Descriptive Element System (D), Descriptive Subset System (S), Process (P)\}$, 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, тип которых отличен от *Descriptive Element System (D)*, должны быть скрыты, а максимальное число состояний процесса отображено. Если создаётся система, прямая подсистема которой имеет тип *Descriptive Subset*

System (S), 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 этого предиката в базе знаний. На рисунке 1 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.

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

RAPIDiS | Активные проекты | Сквозные понятия | Единицы измерения | Представления | Стратегии | Решатели | Приветствую, Fedor_Privalov! | Выход

Название системы*

Химическая реакция

Тип системы*

Process

Прямые подсистемы

☐ Ядро атома [D]

☐ Электронная оболочка атома [D]

☐ Электронная оболочка химического соединения [D]

☐ Атом [D]

☒ Химическое соединение [D]

Подсистема-участник процесса №1

MIN количество экземпляров: 1

MAX количество экземпляров: 1 Не ограничено ☐

Роль подсистемы в процессе:

Субстрат

Подсистема-участник процесса №2

MIN количество экземпляров: 1

MAX количество экземпляров: 1 Не ограничено ☐

Роль подсистемы в процессе:

Реагент

Подсистема-участник процесса №3

MIN количество экземпляров: 1

MAX количество экземпляров: 1 Не ограничено ☐

Роль подсистемы в процессе:

Продукт

Подсистема-участник процесса №4

MIN количество экземпляров: 0

MAX количество экземпляров: 2 Не ограничено ☐

Роль подсистемы в процессе:

Катализатор

Подсистема-участник процесса №5

MIN количество экземпляров: 0

MAX количество экземпляров: 1 Не ограничено ☐

Роль подсистемы в процессе:

Растворитель

Подсистема-участник процесса №6

MIN количество экземпляров: 0

MAX количество экземпляров: 1 Не ограничено ☒

Роль подсистемы в процессе:

Побочный продукт

Добавить участника процесса с другой ролью | Удалить участника

Максимальное число состояний процесса: 20

OK | Сохранить

Рисунок 1 – Пример формы для создания системы типа Process (P) посредством редактора метаграфа сложно-структурированной системы
Figure 1 – Form instance for Process (P) type system creation with the help of complex-structured system metagraph editor

Генератор метапонятий

Необходимо отметить, что существует несколько подходов к определениям «метаонтология» и «метапонятие». Наиболее интуитивным считается подход, ведущий свое начало из философии, при котором метапонятие – это понятие более высокого уровня абстракции, которое используется для описания, анализа или структурирования других понятий, а метаонтология – это онтология, которая оперирует метапонятиями. Существует также и другой подход, основанный на многоуровневых системах понятий, при котором метаонтологиями называются онтологии уровня выше второго (при этом база знаний имеет уровень 1). В этом подходе уровень понятия обычно задается натуральным числом (e.g. «понятие уровня 3»), которое помогает обнаружить конкретную прикладную логическую теорию, где оно было

определено. Ввиду того, что термин «метапонятие» при таком подходе остался свободным, он стал использоваться для обозначения промежуточных понятий онтологии, которые автоматически генерируются по предложенной в работе [29] технологии. Фактически данные понятия являются промежуточными для системы понятий в целом, так как для их генерации необходимы понятия, которые являются денотатами систем (иерархическое устройство которых формируется с помощью вышеописанного редактора метаграфа системы сложной структуры) и сквозных понятий (определены в следующем абзаце). Далее посредством сгенерированных метапонятий задаются конкретные функции и предикаты, частью аргументов которых являются системы, входящие в «узел» метаграфа системы сложной структуры (описание приведено в [29]).

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*). $Metaconcepts = MUMOut$. 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) и др.

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 [29]. 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 (рисунок 2), равной единице (обозн. «*Medium*») – метапонятия типов *Individual*, *Hierarchy Individual*, *Collective*, равной нулю (default) – метапонятия типа *Individual* (если говорить о метапонятиях множества *M*). Названия типов метапонятий выбраны произвольно, но отражают, по мнению авторов, тип обхода узла метаграфа сложно-структурированной системы.

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». Инициализирующие шаблоны различных типов метапонятий представлены в таблице 1. Определенные метапонятия могут быть затем «выключены» и не учитываться при организации некоторой стратегии логического вывода.

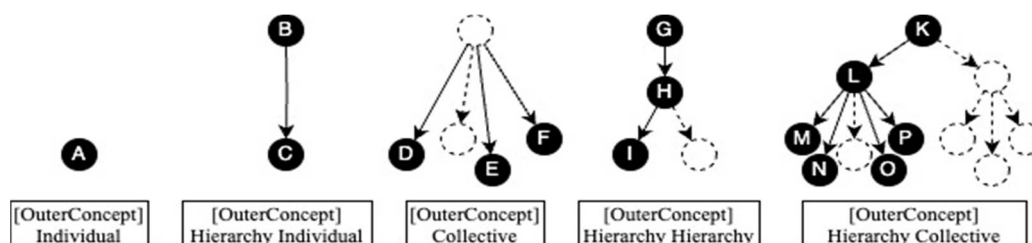


Рисунок 2 – Типы метапонятий и системы узла
Figure 2 – Metaconcept types and node systems

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. Далее для этой цели ему необходимо задать схемы определения функций и предикатов метапонятий с помощью конструктора

Таблица 1 – Инициализирующие шаблоны различных типов метапонятий
Table 1 – Metaconcept initialization templates

Тип метапонятия	Пример текста инициализирующего шаблона метапонятия (f-string)
[OuterConcept] Individual	f"Индивидуальные свойства подсистемы {A} [при учете следующих сквозных понятий {OuterConceptSet}]"
[OuterConcept] Hierarchy Individual	f"Свойства системы {C} как прямой подсистемы системы {B} [при учете следующих сквозных понятий {OuterConceptSet}]"
[OuterConcept] Collective	f"Совместные свойства сочетания следующих систем: {D, E, F} [при учете следующих сквозных понятий {OuterConceptSet}]"
[OuterConcept] Hierarchy Hierarchy	f"Свойства системы {I} как прямой подсистемы системы {H} как прямой подсистемы системы {G} [при учете следующих сквозных понятий {OuterConceptSet}]"
[OuterConcept] Hierarchy Collective	f"Совместные свойства сочетания следующих систем: {M, N, O, P} – как прямых подсистем системы {L} как прямой подсистемы системы {K} [при учете следующих сквозных понятий {OuterConceptSet}]"

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 функции. Аргументы и компоненты возвращаемого значения 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. Аргументом или компонентой возвращаемого значения может быть либо один экземпляр указанного значения, либо множество (в случае числового множества может быть специфицирован интервал, полуинтервал, сегмент допустимых значений).

Стандартными значениями можно считать комплексные, действительные, целые значения, имена (скалярные значения) и логические значения.

Типы значений, связанные с системами метаграфа, формируются при обходе метаграфа. Для предметной области определения реакционных способностей химических соединений используются также фантомы экземпляров систем. *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 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 атомов), необходимо считать свойствами фантома экземпляра системы. Так, у диастереомеров состав атомов и матрица смежности атомов будут совпадать, но взаимодействие с плоскополяризованным светом, биологические активности, некоторые физико-химические свойства могут существенно отличаться в зависимости от пространственной конфигурации [32].

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]». Например, аргумент предиката «Тип гибридизации» в схеме определения предиката означает, что в базе знаний для этого 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. 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 рисунке 3 показан конструктор функций и предикатов с произвольным набором аргументов. Аргументами сконструированной экспертом с помощью данного конструктора функции будут один экземпляр системы «Химическая реакция» и один экземпляр системы «Атом», а возвращаемым значением – множество действительных чисел. После задания названия функция станет доступной в базе знаний.

Конструктор функций и предикатов метапонятий. Этот 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.

Функции:

Название функции*

Номер аргумента*

1

Тип значения аргумента*

Химическая реакция_KB

Номер аргумента*

2

Тип значения аргумента*

Атом_KB

Добавить аргумент Удалить аргумент

Номер компоненты возвращаемого значения*

1

Тип значения компоненты возвращаемого значения*

{R

Добавить компоненту Сохранить

Добавленные понятия:

Рисунок 3 – Функция, определяемая экспертом в конструкторе функций и предикатов с произвольным набором аргументов

Figure 3 – Function declared by the expert with the help of function and predicate constructor with arbitrary argument list

Остальные типы аргументов (доступные для выбора в конструкторе) являются необязательными и могут быть выбраны экспертом для определенных функций и предикатов. На рисунке 4 представлена схема формирования обязательных аргументов функции или предиката метапонятия без учета сквозных понятий, а на рисунке 5 представлена схема формирования необязательных аргументов функции или предиката метапонятия без учета сквозных понятий.

На рисунке 6 представлен интерфейс эксперта при задании схемы определения предиката с помощью конструктора предиката метапонятия (без учета сквозных понятий).

Генератор предикатов внутреннего устройства систем

Для тех систем, для которых прямыми подсистемами являются системы типа *Descriptive Subset System* (S), 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, приведенным в работе [29], 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))».

Программный код функций и предикатов

Если в базе знаний вычислять значения функций и предикатов не требуется, 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 возвращаемых значений функции только компоненты заданного экспертом типа, что исключает возможность ввода пользователем ошибочных данных в базу знаний.

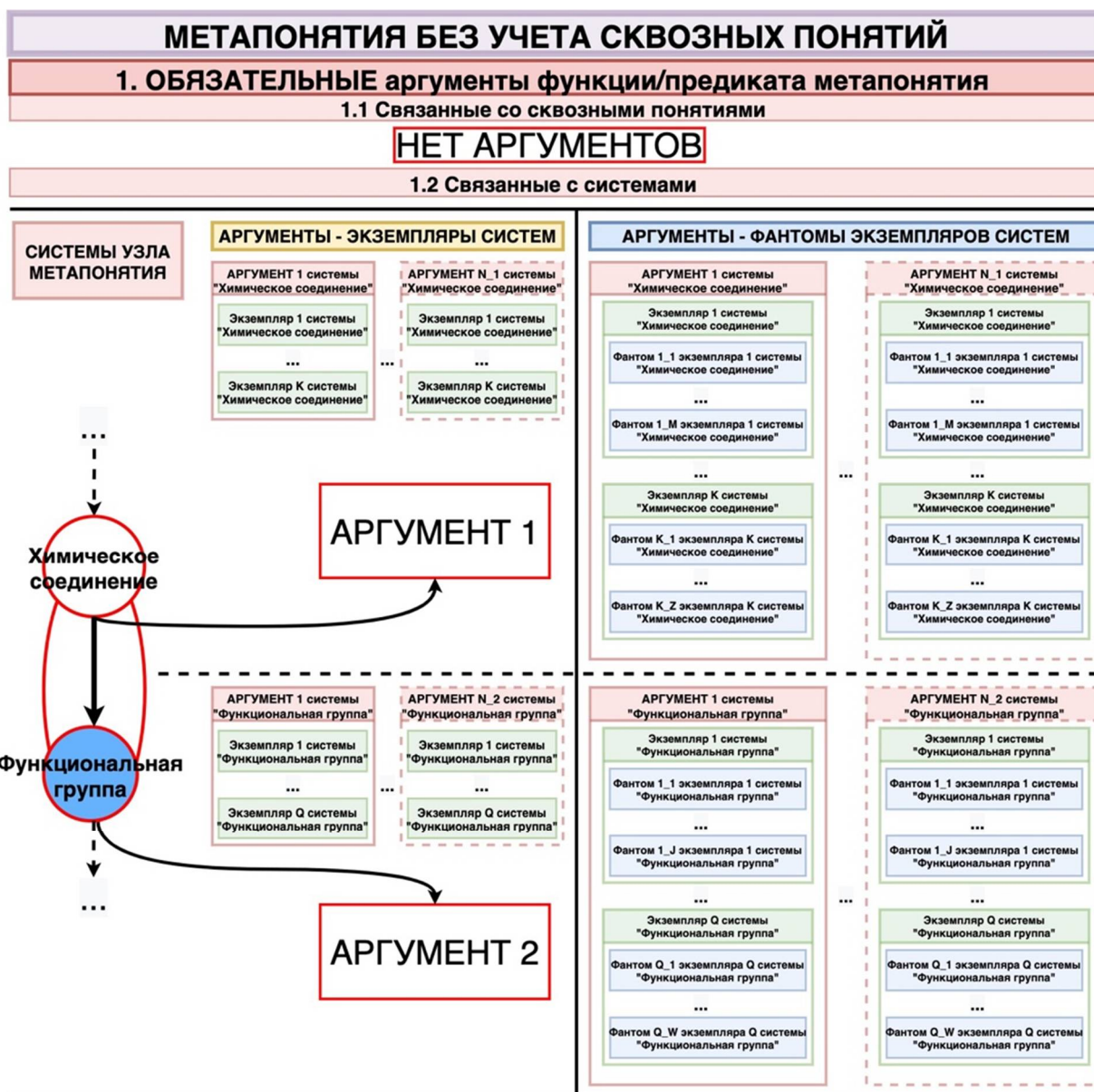


Рисунок 4 – Схема формирования аргументов функции/предиката метапонятия без учета сквозных понятий (обязательные аргументы)

Figure 4 – Metaconcept function/predicate argument formation scheme without outer concepts (required arguments)

Если в базе знаний требуется вычислить значение функции или предиката (включая предикаты внутреннего устройства систем), то пользователем из группы «Программист» предварительно добавляется программный код подпрограммы.

Заключение

В настоящей работе предъявлен программный инструментарий для структуризации систем прикладных понятий и основанных на них баз знаний для предметных областей с объектами сложной структуры. В качестве иллюстративного примера выбрана задача определения реакционных способностей химических соединений. Рассматриваемые 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 системы и свойствах 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.



Рисунок 5 – Схема формирования аргументов функции/предиката метапонятия без учета сквозных понятий (необязательные аргументы)
Figure 5 – Metaconcept function/predicate argument formation scheme without outer concepts (optional arguments)

Трудоемкая работа по проектированию логической структуры и организации наполнения базы знаний лежит на плечах экспертов предметных областей. Подбор промежуточных понятий онтологии, которые бы позволили сформировать исчерпывающие наборы основных понятий, усложняется при разрастании систем понятий и баз знаний для предметных областей со сложно-структурированными объектами. Это сделало актуальными и практически полезными исследования, которые позволяют снизить количество умственных и физических усилий, затрачиваемых экспертами на всех этапах проектирования и разработки систем понятий и баз знаний на их основе.

В работе [29] предложена технология автоматизации процесса формирования систем понятий и баз знаний для предметных областей с объектами сложной структуры, 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 средств поддержки указанного процесса (на примере задачи определения реакционных способностей химических соединений).

Основой предложенного инструментария служит гибридный редактор, работающий в двух режимах автоматизации (максимальной и минимальной). В отличие от редакторов орграфовых моделей, требующих обязательного участия программиста для написания кода агентов, шаблонов сообщений для генерации участков подсети при автоматизации создания промежуточных понятий, в рассматриваемом здесь инструментарии генерация промежуточных понятий поддерживается на уровне редактора.

RAPIDiS Активные проекты | Сквозные понятия | Единицы измерения | Представления | Решатели | Приветствую, Fedor_Privalov! | Выход

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

Метапонятие: Свойства системы Атом как прямой подсистемы системы Химическое соединение как прямой подсистемы системы Химическая реакция

Конструктор предиката метапонятия (без учета сквозных понятий)

Название предиката*

Описание предиката

Тип возвращаемого значения

Выбор аргументов предиката
 Обязательные аргументы-системы узла метапонятия:

Номер аргумента*

Тип значения аргумента

Номер аргумента*

Тип значения аргумента

Тип значения аргумента

Необязательные аргументы:

Рисунок 6 – Интерфейс эксперта при задании схемы определения предиката с помощью конструктора предиката метапонятия (без учета сквозных понятий)
Figure 6 -- System look-and-feel while defining predicate signature with the help of metaconcept predicate constructor (without outer concepts) by the chemist-expert

Основными элементами предложенного гибридного редактора являются редактор метаграфа системы сложной структуры, генератор метапонятий и конструктор функций и предикатов.

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

Генератор метапонятий автоматически формирует множество метапонятий, соответствующих промежуточным понятиям 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 при определении экспертом сигнатур функций и предикатов автоматически подставляет допустимые аргументы, соответствующие системам и сквозным понятиям. Количество элементов в списке с доступными для выбора аргументами сокращается, что уменьшает суммарное количество элементарных действий эксперта при взаимодействии с интерфейсом программной системы. Это может упростить работу эксперта предметной области при формировании баз знаний указанного типа.

Набор разработанных средств поддержки позволяет создавать декларативные и процедурные компоненты интеллектуальной системы определения реакционных способностей химических соединений, имитирующей процесс логического вывода профессионального химика, обосновывающего свой прогноз относительно реакционной способности отдельного химического соединения на основании развитой базы знаний, накопленной за десятилетия теоретической и практической работы в предметной области. Представленный здесь инструментарий может быть использован и в других предметных областях с объектами сложной структуры.

Результаты, изложенные в статье, получены в рамках проекта РФФИ №19-37-90137.

Библиографический список

1. **Aitken J.S., Webber B.L., Bard J.B.L.** Part-of relations in anatomy ontologies: a proposal for RDFs and OWL formalisations. Pacific Symposium on Biocomputing, 2004, no. 9, pp.166-177.
2. **Ahmad M.N., Colomb R.M., Sadiq S.** 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, 2008, pp.46-51. <https://doi.org/10.1109/AMS.2008.28>
3. **Кузьменко Н.В.** О меронимических связях в лексической системе языка (на материале наименований частей тела современного английского языка) / Н.В. Кузьменко // *Вестник Вятского государственного гуманитарного университета*. 2015. №10. С. 58-64.
4. **Morales-González A., Fernández-Reyes F.C., Keet C.M.** 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, 2015, pp. 452-457. https://doi.org/10.1007/978-3-662-46641-4_44
5. **Колодько Д.А.** Меронимические отношения как проявление системности лексики / Д.А. Колодько // *Вестник Самарского универс. История, педагогика, филология*. 2016. № 3.2(22). С. 270-275.
6. **Искандарова Г.Р.** Исследование меронимической лексики в современном немецком и английском языке / Г.Р. Искандарова // *Известия ВГПУ. Филологические науки*. 2019. – №4. – С. 186-189. <https://doi.org/10.3726/978-3-653-03561-2>
7. **Reaxys.** – URL: <https://www.elsevier.com/solutions/reaxys> (дата обращения: 09.06.2023).
8. **PubChem.** – URL: <http://pubchem.ncbi.nlm.nih.gov/> (дата обращения: 09.06.2023).
9. **Merck Index Online.** – URL: <http://www.rsc.org/Merck-Index/> (дата обращения: 09.06.2023).
10. **Molbase.** – URL: <https://www.molbase.com/> (дата обращения: 09.06.2023).
11. **Protein Data Bank (PDB).** – URL: <http://www.wwpdb.org/> (дата обращения: 09.06.2023).
12. **ChemSpider.** – URL: <http://www.chemspider.com> (дата обращения: 09.06.2023).
13. **NIST Chemistry Webbook.** – URL: <https://webbook.nist.gov/chemistry/> (дата обращения: 09.06.2023).
14. **SciFinder.** – URL: <http://scifinder.cas.org> (дата обращения: 09.06.2023).
15. **SPRESI.** – URL: <http://www.spresi.com> (дата обращения: 09.06.2023).
16. **ZINC.** – URL: <http://zinc.docking.org> (дата обращения: 09.06.2023).
17. **GROMACS.** – URL: <https://www.gromacs.org/> (дата обращения: 09.06.2023).
18. **LAMMPS.** – URL: <https://www.lammps.org/> (дата обращения: 09.06.2023).
19. **NAMD.** – URL: <https://www.ks.uiuc.edu/Research/namd/> (дата обращения: 09.06.2023).
20. **BIOVIA Discovery Studio.** – URL: <https://discover.3ds.com/discovery-studio-visualizer-download> (дата обращения: 09.06.2023).
21. **GAUSSIAN.** – URL: <https://gaussian.com/> (дата обращения: 09.06.2023).
22. **GPAW.** – URL: <https://wiki.fysik.dtu.dk/gpaw/> (дата обращения: 09.06.2023).
23. **Клещёв А.С.** Компьютерные банки знаний. Универсальный подход к решению проблемы редактирования информации / А.С. Клещёв, В.А. Орлов // *Информационные технологии*, 2006. – №5. – С. 25-31.
24. **Загорулько Ю.А.** Технология построения онтологий для порталов научных знаний / Ю.А. Загорулько, О.И. Боровикова // *Вестник НГУ. Серия: Информационные технологии*, 2007. – №5 (2). – С. 42-52.
25. **Грибова В.В.** Управление интеллектуальными системами. / В.В. Грибова, А.С. Клещёв, Е.А. Шалфеева // *Известия РАН. Теория и системы управления*, 2010. – №6. – С.122-137.

26. **Голенков В.В.** Проект открытой семантической технологии компонентного проектирования интеллектуальных систем. Часть 1: Принципы создания / В.В. Голенков, Н.А. Гулякина // *Онтология проектирования*, 2014. – №1 (11). – С.42-64.
27. **Musen M.** The Protégé Project: A Look Back and a Look Forward. *AI Matters*, 2015, no. 1(4), pp. 4-12.
28. **Грибова В.В.** Методы и средства разработки жизнеспособных интеллектуальных сервисов. / В.В. Грибова, А.С. Клещёв, Ф.М. Москаленко, В.А. Тимченко, Е.А. Шалфеева, Л.А. Федорищев // *Вестник ДВО РАН*. Владивосток: Изд. «Дальнаука», 2016. – №4. – С.133-141.
29. **Гуляева К.А.** Технология автоматизации процесса формирования систем понятий и баз знаний для предметных областей с объектами сложной структуры / К.А. Гуляева, И.Л. Артемьева // *Вестник Рязанского государственного радиотехнического университета*. 2023. № 85. С.65-81.
30. **IACPaas**. – URL: <https://iacpaas.dvo.ru/> (дата обращения: 09.06.2023).
31. **John B.E., Kieras D.E.** The GOMS family of user interface analysis techniques: Comparison and contrast. *ACM Transactions on Computer-Human Interaction*, 1996, no. 3, pp. 320-351.
32. **Реутов О.А.** Органическая химия в 4 ч. Ч. 2 / О.А. Реутов, А.Л. Курц, К.П. Бутин. – М.: Лаборатория знаний, 2021. – 623 с.

UDC 004.89

TOOLKITS FOR CONCEPT SYSTEM AND KNOWLEDGE BASE DESIGN FOR APPLICATION DOMAINS WITH COMPLEX-STRUCTURED OBJECTS

K. A. Gulyaeva, lecturer, Department of Software Engineering and Artificial Intelligence, FEFU, Vladivostok, Russia;

orcid.org/0000-0003-0226-5072, e-mail: karinagulyaeva8@gmail.com

I. L. Artemieva, Dr. Sc. (Tech.), full professor, Department of Software Engineering and Artificial Intelligence, FEFU, Vladivostok, Russia;

orcid.org/0000-0003-2088-5259, e-mail: artemeva.il@dvfu.ru

Concept systems and knowledge bases for application domains with complex-structured objects are expanded rapidly. Time-consuming operations for knowledge base design, which includes choosing its scheme, savvy selection of interim concepts, designating its filling type are performed by application domain experts. The question is whether it is possible to automate this process. Can we suggest any specialized expert decision support toolkits? The aim is to develop and research specialized program toolkits for structuring applied concept systems and knowledge bases grounded on them for application domains with complex-structured objects (based on a chemical compound reaction capacity identification problem). This research suggests that the hybrid editor be used, which works in two automation regimes. It allows any application domain expert to choose – to make minimum effort or to customize the process of concept system and knowledge base development, whichever is preferable. 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. Function and predicate signatures are set by an application domain expert with the help of function and predicate constructor with arbitrary argument list and function and predicate metaconcept constructor. The latter automatically sets some of the allowable arguments, which facilitates application domain expert work during concept system and knowledge base formation.

Keywords: knowledge base, intelligent system, ontology, application domain, complex-structured system, applied logics theory, intermediary concepts, reaction capacity identification

DOI: 10.21667/1995-4565-2024-87-62-77

References

1. Aitken J.S., Webber B.L., Bard J.B.L. Part-of relations in anatomy ontologies: a proposal for RDFs and OWL formalisations. *Pacific Symposium on Biocomputing*, 2004, no. 9, pp.166-177.
2. Ahmad M.N., Colomb R.M., Sadiq S. 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, 2008, pp.46-51. <https://doi.org/10.1109/AMS.2008.28>
3. Kuzmenko N.V. O meronimicheskikh svyazyah v leksicheskoy sisteme yazyka (na materiale naimenovaniy chastej tela sovremennogo anglijskogo yazyka) [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 [Vyatski State Humanitarian University Bulletin]*, 2015, no. 10, pp. 58-64 (in Russian).
4. Morales-González A., Fernández-Reyes F.C., Keet C.M. 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, 2015, pp. 452-457. https://doi.org/10.1007/978-3-662-46641-4_44
5. Kolodko D.A. Meronimicheskie otnosheniya kak proyavlenie sistemnosti leksiki [Meronymic relations as a manifestation of the systematic nature of vocabulary]. *Vestnik Samarskogo universiteta. Istorija, pedagogika, filologija [Samarski University Bulletin. History, pedagogics, filology]*, 2016, no. 3.2(22), pp. 270-275 (in Russian).
6. Iskandarova G.R. Issledovanie meronimicheskoy leksiki v sovremennom nemeckom i anglijskom yazyke [Study of meronymic vocabulary in modern German and English]. *Izvestiya VGPU. Filologicheskie nauki [VGPU Bulletin. Filology studies]*, 2019, no. 4, pp. 186-189. <https://doi.org/10.3726/978-3-653-03561-2> (in Russian).
7. Reaxys, available at: <https://www.elsevier.com/solutions/reaxys> (accessed: 06/09/2023).
8. PubChem, available at: <http://pubchem.ncbi.nlm.nih.gov/> (accessed: 06/09/2023).
9. Merck Index Online, available at: <http://www.rsc.org/Merck-Index/> (accessed: 06/09/2023).
10. Molbase, available at: <https://www.molbase.com/> (accessed: 06/09/2023).
11. Protein Data Bank (PDB), available at: <http://www.wwpdb.org/> (accessed: 06/09/2023).
12. ChemSpider, available at: <http://www.chemspider.com> (accessed: 06/09/2023).
13. NIST Chemistry Webbook, available at: <https://webbook.nist.gov/chemistry/> (accessed: 06/09/2023).
14. SciFinder, available at: <http://scifinder.cas.org> (accessed: 06/09/2023).
15. SPRESI, available at: <http://www.spresi.com> (accessed: 06/09/2023).
16. ZINC, available at: <http://zinc.docking.org> (accessed: 06/09/2023).
17. GROMACS, available at: <https://www.gromacs.org/> (accessed: 06/09/2023).
18. LAMMPS, available at: <https://www.lammps.org/> (accessed: 06/09/2023).
19. NAMD, available at: <https://www.ks.uiuc.edu/Research/namd/> (accessed: 06/09/2023).
20. BIOVIA Discovery Studio, available at: <https://discover.3ds.com/discovery-studio-visualizer-download> (accessed: 06/09/2023).
21. GAUSSIAN, available at: <https://gaussian.com/> (accessed: 06/09/2023).
22. GPAW, available at: <https://wiki.fysik.dtu.dk/gpaw/> (accessed: 06/09/2023).
23. Kleshev A.S., Orlov V.A. Komp'yuternye banki znaniy. Universal'nyj podhod k resheniyu problemy redaktirovaniya informacii [Computer knowledge banks. Universal approach to information editing problem]. *Informatsionnie tekhnologii [Information technologies]*, 2006, no. 5, pp. 25-31 (in Russian).
24. Zagorulko Yu. A., Borovikova O.I. Tekhnologiya postroeniya ontologij dlya portalov nauchnyh znaniy [Ontology design technology for scientific portals]. *Vestnik NGU. Seriya: Informatsionnie tekhnologii [NSU Bulletin. Information technology series]*, 2007, no. 5(2), pp. 42-52 (in Russian).
25. Gribova V.V., Kleshev A.S., Shalfeeva Ye.A. Upravlenie intellektual'nymi sistemami [Intelligent system control]. *Izvestiya RAN. Teoriya i sistemi upravleniya [RAS Bulletin. Theory and control systems]*, 2010, no. 6, pp.122-137. (in Russian).
26. Golenkov V.V., Guliakina N.A. Proekt otkrytoj semanticheskoy tekhnologii komponentnogo proektirovaniya intellektualnyh sistem. Chast 1: Principy sozdaniya [Project of Open Semantic Technology of The Componental Design of Intelligent Systems. Part 1: The Principles of Creation]. *Ontology of designing*, 2014, no. 1(11), pp.42-64 (in Russian).
27. Musen M. The Protégé Project: A Look Back and a Look Forward. *AI Matters*, 2015, no. 1(4), pp. 4-12.

28. **Gribova V.V., Kleshev A.S., Moskalenko F.M., [et al.].** Metody i sredstva razrabotki zhiznesposobnyh intellektualnyh servisov [Methods and means of designing viable intelligent services]. *Vestnik DVO RAN*. Vladivostok: Izd. Dalnauka [FEB RAS Bulletin. Vladivostok. Far Eastern Science publishing house], 2016, no. 4, pp.133-141. (in Russian).

29. **Gulyaeva K.A., Artemieva I.L.** Tekhnologiya avtomatizatsii processa formirovaniya sistem ponyatiy i baz znaniy dlya predmetnyh oblastej s obektami slozhnoj struktury [The technology of automated concept system and knowledge base formation process for application domains with complex-structured objects]. *Vestnik Rjazanskogo gosudarstvennogo radiotekhnicheskogo universiteta*. 2023, no. 85, pp.65-81. (in Russian).

30. **IACPaaS**, available at: <https://iacpaas.dvo.ru/> (accessed: 06/09/2023).

31. **John B.E., Kieras D.E.** The GOMS family of user interface analysis techniques: Comparison and contrast. *ACM Transactions on Computer-Human Interaction*, 1996, no. 3, pp. 320-351.

32. **Reutov O.A., Kurtz A.L., Butin K.P.** Organicheskaya himiya v 4 ch. Ch. 2 [Organic chemistry in 4 vol. vol.2.]. M., Laboratoriya znaniy [Moscow Knowledge Laboratory], 2021, 623 p. (in Russian).