Программный модуль формирования цифрового математического пространства на основе графов знаний

В.И. Гурьянов[1], А.М. Елизаров[1,2]

[1] Казанский федеральный университет, ул. Кремлевская, д. 35, г. Казань, 420008

[2] Национальный исследовательский центр (НИЦ) «Курчатовский институт». пл. Академика Курчатова, д. 1, Москва, 123182

Аннотация. Современное информационное пространство содержит множество данных, однако они зачастую слабо структурированы, трудно находимы и не всегда корректны. Это создаёт дополнительные трудности при исследованиях, поэтому в настоящее время формируются цифровые пространства научных знаний, в частности, на основе графов знаний. Для обеспечения качества информации такие графы часто наполняются данными вручную, что требует больших затрат времени. Поэтому создание инструмента, предоставляющего возможность автоматического наполнения графа данными, а также обеспечивающего контроль их качества, позволит упростить и ускорить процесс формирования цифровых пространств научных знаний. Предложены методы автоматизации наполнения графа данными, обеспечивающие параллельный контроль их целостности. На основе предложенных методов разработан программный модуль, описаны механизмы его функционирования и его архитектура.

Ключевые слова: цифровое пространство научных знаний, формирование цифровых пространств научных знаний, графы знаний, автоматизация построения графов знаний

Software module for forming digital mathematical space based on knowledge graphs

V.I. Gurianov[1], A.M. Elizarov[1,2]

[1] Kazan Federal University, ul. Kremlyovskaya, 35, Kazan, 420008

[2] National Research Center (NRC) "Kurchatov Institute". Academician Kurchatov Square, 1, Moscow, 123182

Abstract. The modern information space contains a lot of data, but they are often poorly structured, difficult to find and not always correct. This creates

additional difficulties during researches, so digital spaces of scientific knowledge are currently being formed, in particular, based on knowledge graphs. To ensure the quality of information, such graphs are often filled with data manually, which is time-consuming. Therefore, the creation of a tool that provides the ability to automate process of filling a graph with data, as well as ensures data quality, will simplify and speed up the process of forming digital spaces of scientific knowledge. Methods for automating the filling of the graph with data are proposed, including parallel control of their integrity. Based on the proposed methods, a software module has been developed, the mechanisms of its functioning and its architecture are de-scribed.

Keywords: digital space of scientific knowledge, formation of digital spaces of scientific knowledge, knowledge graphs, automation of knowledge graph construction

Введение

Современное информационное пространство содержит множество данных из разнообразных предметных областей, однако эта информация часто слабо структурирована, трудно находима и к тому же может оказаться некорректной. Всё это значительно осложняет поиск и обработку необходимой информации. Поэтому актуальной задачей является создание цифровых пространств научных знаний, позволяющих систематизировать доступ к данным, упростить их обработку и использование, а также гарантировать их корректность (см. также [1, 2]).

Проблема обеспечения исследователей качественными, проверенными и непротиворечивыми данными давно изучается многими исследователями как в России, так и в других странах. В зарубежных публикациях этот вопрос обычно рассматривается в рамках более общего понятия – создания исследовательских инфраструктур [3–5]. Отечественные авторы часто рассматривают этот вопрос с позиции создания цифровых пространств научных знаний [1, 2, 6, 7]. Оба этих понятия подразумевают создание некоторой компьютерной среды, объединяющей научные знания, а также предоставляющей дополнительные инструменты для их поиска и использования. В последнее время для создания такой среды изучают возможности применения онтологий и графов знаний (например, [8–10]).

Как разновидность научных знаний, в науке активно изучают математическое знание. Построение цифровых математических пространств и графов математических знаний рассматривают многие авторы, например, [8, 11–14]. Это актуально, поскольку ещё не создан граф, полностью охватывающий всё математическое знание.

Существует множество подходов к созданию графов знаний, каждый из которых сопряжён со своими преимуществами и недостатками. Первый подход – автоматизированное создание графа на основе программного извлечения и преобразования данных. Примером использования такого

метода является граф DBpedia [15], построенный путём преобразования информации, собранной в рамках проекта Wikipedia, в структурированные данные. Преимуществом программного наполнения этого графа является возможность обработки большого объёма данных. Однако очевидны и недостатки: для применения метода необходимо существование веб-ресурса или базы данных, уже содержащих необходимую информацию. Кроме того, при сборе информации из разных источников для каждого может потребоваться разработка отдельного сборщика данных, и существенным становится вопрос доверия к извлекаемым данным.

Другим подходом является предоставление доступа к открытому редактированию графа знаний широкой группе пользователей. Например, таким способом был реализован проект Wikidata [16]. Преимуществом такого способа является постепенное наполнение графа данными без необходимости дополнительного вмешательства со стороны команды проекта: постепенно пользователи будут сами интегрировать в граф информацию из различных источников. Недостатком является необходимость дополнительного привлечения людей для наполнения графа, стоит также вопрос доверия к вносимым данным и даже имеется вероятность злого умысла в виде внесения заведомо ложной информации.

Третий подход – ручное наполнение графа данными. В этом случае некоторая группа квалифицированных специалистов наполняет граф знаний проверенными данными. Например, таким способом были реализованы проекты MathAlgoDB [17] и OntoMathPro [12, 13]. Названный подход схож со вторым, однако круг редакторов графа заведомо ограничивается доверенными квалифицированными людьми. Преимуществами являются высокое качество и структурированность данных. Однако такой подход приводит к большим затратам времени квалифицированных специалистов. Также в процессе ручного создания графа появляется вероятность возникновения ошибок по причине человеческого фактора, что может негативно отразиться на итоговом качестве.

Так как качество данных является приоритетом при создании графов научных знаний, в процессе их формирования часто применяют методы ручного наполнения данными [17].

Исходя из вышеизложенного, можно утверждать, что существует потребность в инструменте, предоставляющем не только возможность интеграции систем автоматизации сбора информации и её добавления в граф, но и параллельного контроля качества вносимых данных. По этой причине нами был разработан инструмент, обладающий следующими свойствами:

- Возможность расширения функционала;
- Контроль типа вносимых объектов;
- Возможность автоматического назначения типа объекта;

- Применимость к уже построенному графу;
- Наличие функций для вызова SPARQL-запросов из программного кода;
- Построение подграфа на основе SPARQL-запроса;
- Открытый исходный код.

Существующие инструменты, используемые при построении графов знаний

В настоящее время существует несколько инструментов, реализующих различные аспекты реализации поставленной задачи, например, Protégé [18]; RDFLib [19]; SPARQLWrapper [20]. Некоторое сравнение этих инструментов представлено в таблице 1.

Таблица 1. Сравнение альтернативных решений

	Protégé	RDFLib	SPARQLWrapper
Возможность взаимодействия через язык программирования	-	+	+
Взаимодействие с базой данных	-	-	+
Контроль целостности данных	+	-	-
Открытый исходный код	+	+	+

Каждый из представленных инструментов решает различные задачи, но не соответствует всем необходимым критериям (см. таблицу 1).

Protégé – редактор онтологий с открытым исходным кодом. Он предоставляет множество инструментов для редактирования онтологий и контроля целостности данных. При этом вся работа осуществляется через графический интерфейс, что, хотя и является удобным для ручной модификации графа, но накладывает значительные ограничения на организацию процесса автоматизации работы.

RDFLib – библиотека для языка Python с открытым исходным кодом. Содержит функции для работы с RDF-документами и поддерживает многие часто используемые стандарты, такие как RDF, RDFS, OWL, FOAF. Однако функционал этой библиотеки подразумевает полную загрузку графа в оперативную память, что неприемлемо при работе с крупными графами.

SPARQLWrapper – библиотека для языка Python, способная к взаимодействию с базой данных через точку доступа SPARQL. Это

позволяет работать напрямую с графом и не требует загрузки всех данных в оперативную память. Однако названная библиотека дает только функционал для выполнения текстового SPARQL-запроса и не содержит какого-либо другого функционала взаимодействия с графом, например, проверки целостности.

Логика контроля целостности данных

Основными элементами взаимодействия между модулями созданной нами программы являются Triple и TripleItem.

Triple представляет собой реализацию RDF-триплета и состоит из субъекта, предиката и объекта, каждый из которых является элементом типа TripleItem.

В соответствии со стандартом RDF [21], «субъект» и «предикат» должны представлять собой некий идентификатор (IRI или идентификатор пустого узла), а «объект» может быть как идентификатором, так и просто значением.

Для контроля целостности данных все объекты были разделены на два типа – Entity (сущность) и Predicate (предикат). Каждому из них соответствует отдельная таблица в реляционной базе данных, содержащая Id и IRI объекта. Объект может быть добавлен в граф только после того, как будет внесён в соответствующую ему таблицу (регистрация).

Решение использовать реляционную базу данных для определения типа объектов было принято по той причине, что применение для решения рассматриваемой задачи только графовой базы данных потребует либо рекурсивного обхода части узлов, либо создания отдельного графа, содержащего только структурную информацию. Оба этих метода менее оптимальны, чем использование реляционной базы данных.

Для регистрации объекта существует специальная функция. Ей нужно передать его тип – Entity или Predicate. Также существует возможность дополнительно передать Id или IRI, которые будут обозначать создаваемый объект. Сначала проводится проверка существования объекта с таким IRI/Id в соответствующей таблице базы данных. Если объект существует, то информация о нём извлекается из базы данных. Если IRI не был передан, то он формируется на основе Id – переданного из базы данных или нового. После этого объект также добавляется в граф.

В граф разрешено добавлять только связи вида:

- Сущность – Предикат – Значение (например, Теорема Ферма – Сформулирована – 1637 год);
- Сущность – Предикат – Сущность (например, Алгебра – Относится к – Математика);
- Предикат – Предикат – Значение (Относится к – Добавлен – Дата добавления);

• Предикат – Предикат – Предикат (Относится к – Класс – Predicate).

Исключением является возможность установки отношения эквивалентности между предикатом и сущностью. Это может быть полезным, если необходимо указать связь между предикатом и реальными объектом или понятием. Например, для предиката «Имя», устанавливающего связь между человеком и его именем, может потребоваться организация связи с понятием «Имя», объясняющим на основе связей, что такое имя имеется у человека.

Добавление триплета в граф реализуется специальной функцией. По умолчанию для переданного RDF-триплета проверяется, соответствует ли он одному из четырёх ранее описанных видов связей, а для каждого его элемента определяется, был ли он ранее зарегистрирован.

Блок-схема алгоритма проверки целостности показана на рис. 1. Если требования к качеству данных не излишне строгие, то функция сама может проводить регистрацию недостающих объектов. Для этого достаточно передать этой функции соответствующий параметр. Вывод о том, какой тип должен быть присвоен незарегистрированному элементу RDF-триплета, основан на разрешённых связях.

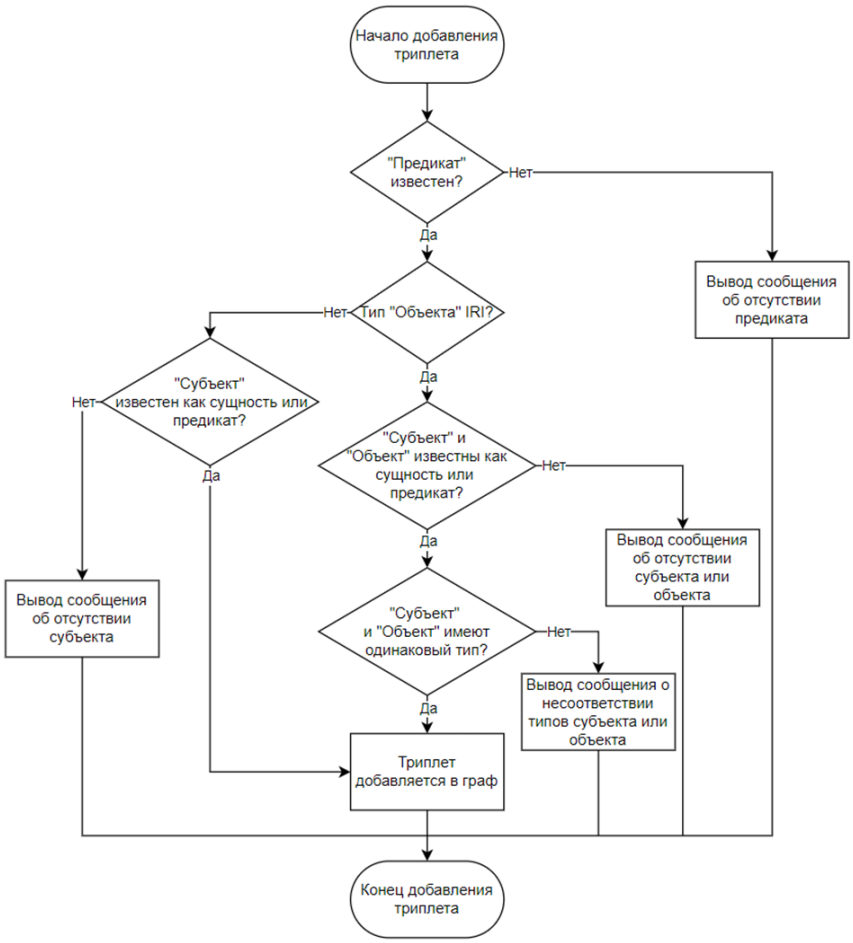

Рис. 1. Алгоритм добавления триплета с проверкой целостности

Рассмотрим правила автоматической регистрации недостающих объектов. Пусть RDF-триплет представляет собой тройку (s, p, o). Обозначим множество зарегистрированных объектов через R, множества сущностей, предикатов и значений – соответственно через E, P и L. Тогда:

- «Предикат» триплета не зарегистрирован, следовательно, его тип – Predicate ($p \notin R \Rightarrow p \in P$).
- «Субъект» («Объект») не зарегистрирован, но «Объект» («Субъект») зарегистрирован, следовательно, тип «Субъекта» («Объекта») равен типу «Объекта» («Субъекта»)

$$(s \notin R, o \in E \Rightarrow s \in E;\ s \notin R, o \in P \Rightarrow s \in P; o \notin R, s \in E \Rightarrow o \in E;\ o \notin R, s \in P \Rightarrow o \in P).$$

- «Субъект» и «Объект» не зарегистрированы, следовательно, их тип – Entity ($s \notin R, o \notin E \Rightarrow s \in E, o \in E$).
- «Субъект» не зарегистрирован, а «Объект» является значением, следовательно, «Субъект» – это Entity ($s \notin R, o \in L \Rightarrow s \in E$).

Последние два решения выполнены на основе того, что операция добавления сущности в граф статистически гораздо более частая, чем добавление предиката. Блок-схема алгоритма для этого случая показана на рис. 2.

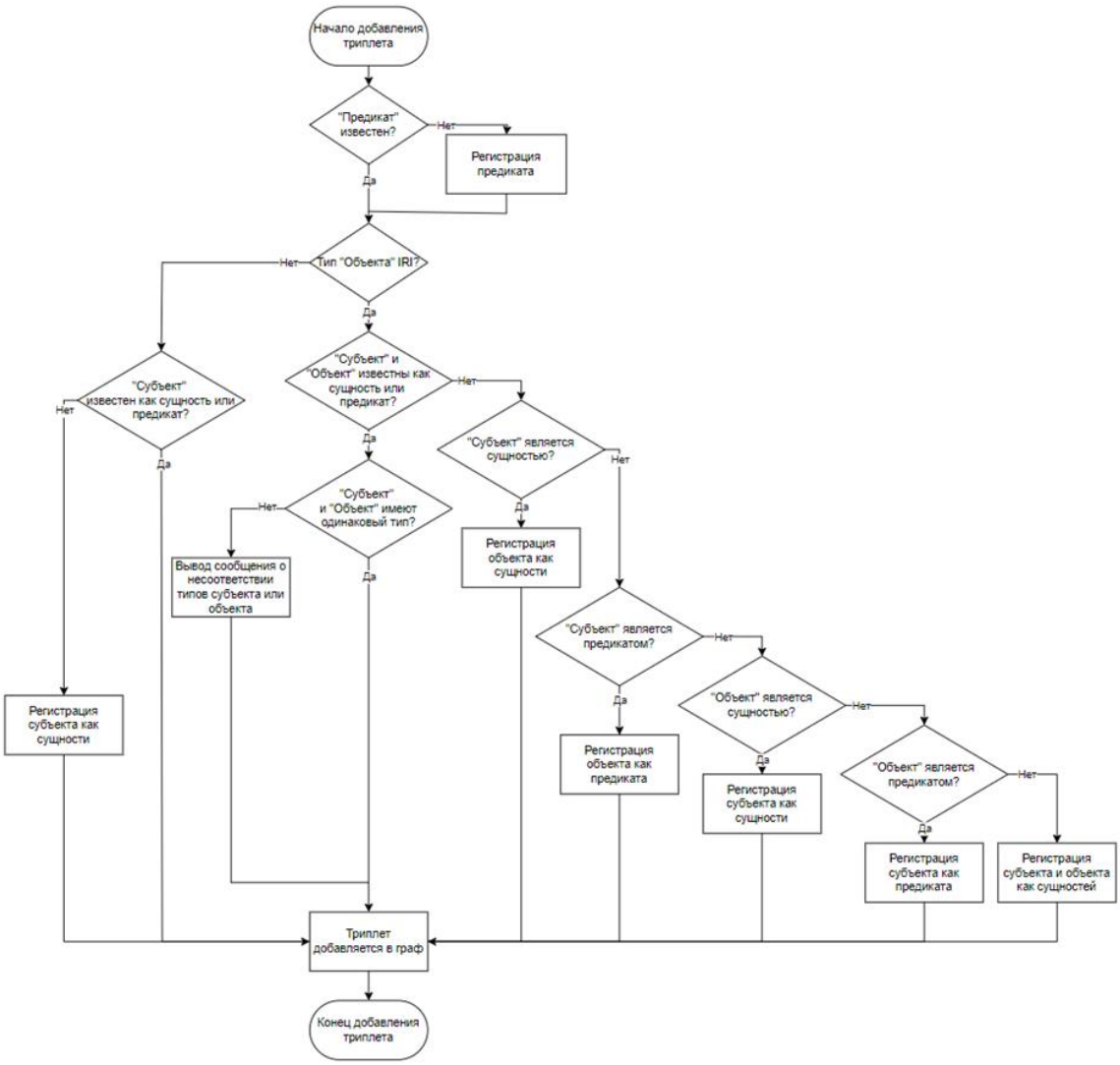

Рис. 2. Алгоритм добавления триплета с автоматической регистрацией

Архитектура разработанного модуля

Для взаимодействия с системой управления данными Virtuoso Universal Server [22] был создан класс, содержащий функции-конструкторы SPARQL-запросов. Конструкторы были разделены на два типа в зависимости от выполняемых операций: вносящие изменения в базу данных (операции записи) и применяемые для получения информации из базы данных (операция чтения). Сконструированные запросы выполняются с применением библиотеки SPARQLWrapper. На данный момент поддерживаются следующие операции:

- SELECT – для получения данных из графа;
- CREATE GRAPH – для создания графа;
- DROP GRAPH – для удаления графа;
- INSERT – для добавления триплетов в граф;
- DELETE – для удаления триплетов из графа.

Сконструированные запросы на изменение графа выполняются не сразу, а только после вызова специальной функции Commit. Это позволяет объединить несколько запросов в один и уменьшить время ожидания их выполнения.

В зависимости от цели результат Select запроса можно получить в формате JSON или в виде массива RDF-триплетов. Библиотека SPARQLWrapper, выполняющая сформированный SPARQL запрос, возвращает значения в формате JSON. Если результат требуется получить в виде массива RDF-триплетов, то он формируется на основе WHERE и OPTIONAL части запросов. Для этого каждая переменная в WHERE и OPTIONAL части запросов заменяется её значением, возвращённым в формате JSON.

Получение результата SPARQL SELECT запроса в виде массива RDF-триплетов важно, так как позволяет получить подграф в формате, поддерживаемом другими разработанными модулями, такими как модуль визуализации и модуль взаимодействия с графом.

Взаимодействие с графами проводится через специальный класс, содержащий функции, объединяющие взаимодействие с реляционной и графовой базами данных. Основными функциями этого класса являются:

- Регистрация объекта;
- Добавление триплета в граф;
- Копирование графа.

Функции регистрации объекта и добавления триплета в граф являются реализацией логики контроля целостности данных, описанной выше.

Функция копирования графа позволяет с помощью конструктора SPARQL-запросов получить полный граф по переданной точке доступа SPARQL и построить на его основе новый граф вызовом предыдущих двух функций. Для этого сначала регистрируются все предикаты триплетов, а тип остальных объектов определяется на основе их связей по ранее описанному алгоритму. Эта функция позволяет интегрировать ранее созданные графы и применить к ним разработанную логику контроля целостности данных.

Результаты

Разработанный программный модуль был применён для решения ряда задач, описанных ниже. С сайта Казанского федерального университета была собрана информация о сотрудниках Института информационных технологий и интеллектуальных систем (ИТИС) и записана в граф. На основе полученного списка был проведён поиск публикаций сотрудников ИТИС в Научной электронной библиотеке Elibrary (https://www.elibrary.ru/). Полученная информация также была записана в граф и связана с ранее полученными данными обо всех сотрудниках Института ИТИС. Часть построенного графа представлена на рисунке 3.

Также разработанный программный модуль был успешно применён для работы с онтологией OntoMathPro (https://github.com/CLLKazan/OntoMathPro/) (см. также [12, 13]), что даёт больше возможностей взаимодействия с ней.

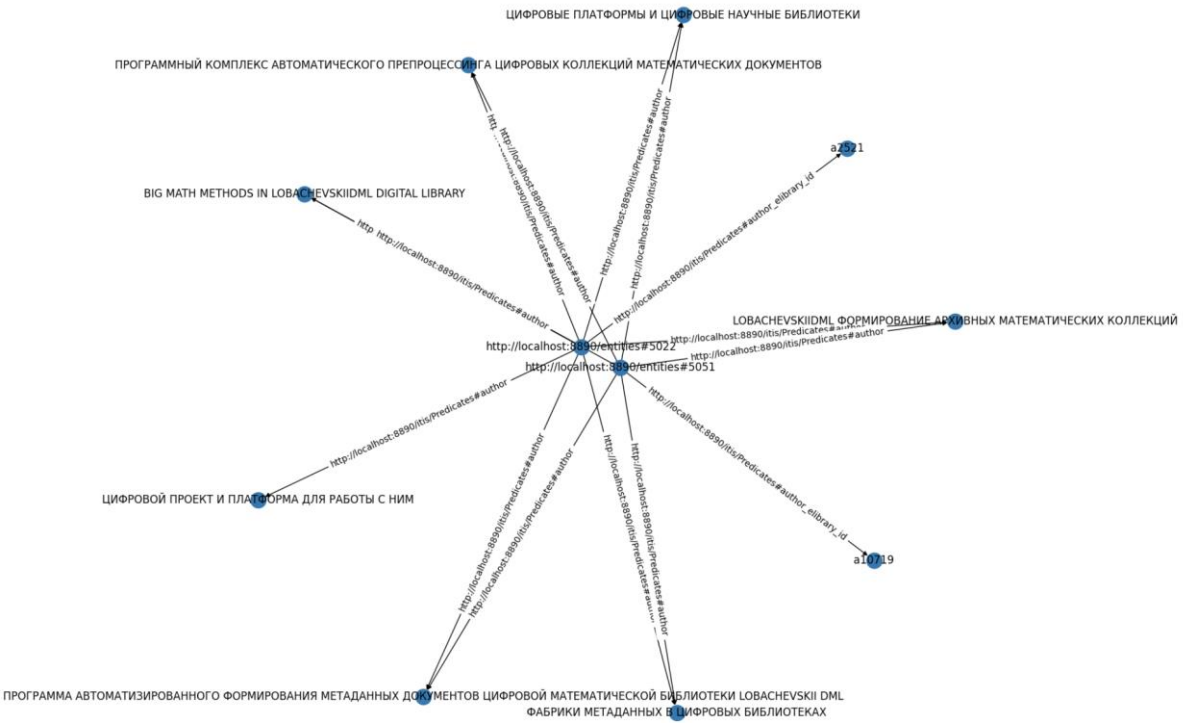

Рис. 3. Подграф графа публикаций сотрудников ИТИС

Заключение

Существует множество подходов для формирования цифровых пространств научных знаний и построения графов знаний. В настоящее время нет инструмента, находящегося в свободном доступе и предоставляющего возможности автоматизации наполнения графа данными и параллельного контроля их целостности. Потребность в таком инструменте есть у исследователей, занимающихся построением графов научных знаний. Предложенный в статье метод контроля целостности данных позволил на программном уровне отслеживать тип объектов, вносимых в граф, и корректность добавляемых триплетов. Соответствующий разработанный инструмент с описанным функционалом его использования размещён в репозитории GitHub [23].

Литература

1. Ataeva O., Kalenov N., Serebryakov V., Sotnikov A. Informational Infrastructure of the Common Digital Space of Scientific Knowledge // International Conference «Common Digital Space of Scientific Knowledge», November 10–12, 2020, Moscow, Russia / CEUR Workshop Proceedings,

2021. — Vol. 2990. — P. 1–10. — https://doi.org/10.51218/1613-0073-2990-1-10
2. Атаева О.М., Каленов Н.Е., Серебряков В.А. Об основных понятиях Единого цифрового пространства научных знаний // Научный сервис в сети Интернет: труды XXII Всероссийской научной конференции (21–25 сентября 2020 г., онлайн). М.: ИПМ им. М.В. Келдыша, 2020. — С. 29–40. — https://doi.org/10.20948/abrau-2020-18
3. Fecher B., Kahn R., Sokolovska N., Völker T., Nebe P. Making a Re-search Infrastructure: Conditions and Strategies to Transform a Service into an Infra-structure // Science and Public Policy, 2021. — Vol. 48, No. 4. — P. 499–507. — https://doi.org/10.1093/scipol/scab026
4. Kohn Rådberg K., Löfsten H. Developing a knowledge ecosystem for large-scale research infrastructure // The Journal of Technology Transfer, 2023. — Vol. 48, No. 1. — P. 441–467. — https://doi.org/10.1007/s10961-022-09945-x
5. Papon P. European scientific cooperation and research infrastructures: Past tendencies and future prospects // Minerva, 2004. — Vol. 42, No. 1. — P. 61–76. — https://doi.org/10.1023/B:MINE.0000017700.63978.4a
6. Антопольский А.Б., Каленов Н.Е., Серебряков В.А., Сотников А.Н. О едином цифровом пространстве научных знаний // Вестник Российской академии наук, 2019. — Т. 89, №7. — С. 728–735. — https://doi.org/10.31857/S0869-5873897728-735
7. Elizarov A., Lipachev E. Lobachevskii Digital Library in the Scientific Space of Mathematical Knowledge, Scientific and Technical Information Processing, 2023. — Vol. 50, No. 1. — P. 35–39. — https://doi.org/10.3103/S0147688223010021
8. Serebryakov V.A., Ataeva O.M. Ontology Based Approach to Modeling of the Subject Domain ‘‘Mathematics” in the Digital Library // Lobachevskii Journal of Mathematics, 2021. — Vol. 42, No. 8. — P. 1920–1934. — https://doi.org/10.1134/S199508022108028X
9. Lange C. Ontologies and languages for representing mathematical knowledge on the Semantic Web // Semantic Web, 2013. — Vol. 4, No. 2. — P. 119–158. — https://doi.org/10.3233/SW-2012-0059
10. Wang J. Math-KG: Construction and Applications of Mathematical Knowledge Graph, arXiv:2205.03772, 2022. — P. 1–5. — https://doi.org/10.48550/arXiv.2205.03772.
11. Муромский А.А., Тучкова Н.П. Представление математических понятий в онтологии научных знаний // Онтология проектирования, 2019. — Т. 9, №1 (31). — С. 50–69. — https://doi.org/10.18287/2223-9537-2019-9-1-50-69
12. Елизаров А.М., Кириллович А.В., Липачёв Е.К., Невзорова О.А. Онтология математического знания OntoMathPRO // Доклады Российской академии наук. Математика, информатика, процессы

управления, 2022. — Т. 507, № 1. — С. 29–35. — https://doi.org/10.31857/S2686954322700011
13. Елизаров А.М., Кириллович А.В., Липачёв Е.К., Невзорова О.А. Новые компоненты онтологии OntoMathPRO представления математического знания // Научный сервис в сети Интернет: труды XXV Всероссийской научной конференции (18–21 сентября 2023 г., онлайн). М.: ИПМ им. М.В. Келдыша, 2023. — С. 141–151. — https://doi.org/10.20948/abrau-2023-32
14. Невзорова О.А., Гизатуллин Б.Т. Система автоматического построения графов знаний математических документов // Ученые записки Казанского университета. Серия: Физико-математические науки, 2023. — Т. 165, № 3. — С. 264–281. https://doi.org/10.26907/2541-7746.2023.3.264-281
15. Lehmann J., Isele R., Jakob M., Jentzsch A., Kontokostas D., N. Mendes P., Hellmann S., Morsey M., Van Kleef P., Auer S., Bizer C. DBpedia – A Large-scale, Multi-lingual Knowledge Base Extracted from Wikipedia // Semantic Web, 2015. — Vol. 6, No. 2. — P. 167–195. https://doi.org/10.3233/SW-140134
16. Vrandečić D., Krötzsch M. Wikidata: a free collaborative knowledgebase // Communications of the ACM, 2014. — Vol. 57, No. 10. — P. 78–85. — https://doi.org/10.1145/2629489
17. Schembera B., Wübbeling F., Kleikamp H., Schmidt B., Shehu A., Reidelbach M., Biedinger C., Fiedler J., Koprucki T., Iglezakis D., Göddeke D. Towards a Knowledge Graph for Models and Algorithms in Applied Mathematics // arXiv:2408.10003. — https://doi.org/10.48550/arXiv.2408.10003
18. Protégé. — https://protege.stanford.edu
19. Home // RDFLib. — https://rdflib.dev
20. SPARQL Endpoint interface to Python // SPARQLWrapper documentation — https://sparqlwrapper.readthedocs.io/en/latest/main.html
21. RDF // Semantic Web Standards. — https://www.w3.org/RDF.
22. Virtuoso Homepage // OpenLink Software. — https://virtuoso.openlinksw.com.
23. VIGuryanov/Knowledge-Graphs-Builder // GitHub. — https://github.com/VIGuryanov/Knowledge-Graphs-Builder.

References

1. Ataeva O., Kalenov N., Serebryakov V., Sotnikov A. Informational Infrastructure of the Common Digital Space of Scientific Knowledge // International Conference “Common Digital Space of Scientific Knowledge”, November 10–12, 2020, Moscow, Russia / CEUR Workshop Proceedings, 2021. — Vol. 2990. — P. 1–10. — https://doi.org/10.51218/1613-0073-2990-1-10

2. Ataeva O.M., Kalenov N.E., Serebriakov V.A. Ob osnovnykh poniatiiakh Edinogo tsifrovogo prostranstva nauchnykh znanii // Nauchnyi servis v seti Internet: trudy XXII Vserossiiskoi nauchnoi konferentsii (21–25 sentiabria 2020 g., onlain). M.: IPM im. M.V. Keldysha, 2020. — S. 29–40. — https://doi.org/10.20948/abrau-2020-18
3. Fecher B., Kahn R., Sokolovska N., Völker T., Nebe P. Making a Re-search Infrastructure: Conditions and Strategies to Transform a Service into an Infra-structure // Science and Public Policy, 2021. — Vol. 48, No. 4. — P. 499–507. — https://doi.org/10.1093/scipol/scab026
4. Kohn Rådberg K., Löfsten H. Developing a knowledge ecosystem for large-scale research infrastructure // The Journal of Technology Transfer, 2023. — Vol. 48, No. 1. — P. 441–467. — https://doi.org/10.1007/s10961-022-09945-x
5. Papon P. European scientific cooperation and research infrastructures: Past tendencies and future prospects // Minerva, 2004. — Vol. 42, No. 1. — P. 61–76. — https://doi.org/10.1023/B:MINE.0000017700.63978.4a
6. Antopolskii A.B., Kalenov N.E., Serebriakov V.A., Sotnikov A.N. O edinom tsifrovom prostranstve nauchnykh znanii // Vestnik Rossiiskoi akademii nauk, 2019. — T. 89, №7. — S. 728–735. — https://doi.org/10.31857/S0869-5873897728-735
7. Elizarov A., Lipachev E. Lobachevskii Digital Library in the Scientific Space of Mathematical Knowledge, Scientific and Technical Information Processing, 2023. — Vol. 50, No. 1. — P. 35–39. — https://doi.org/10.3103/S0147688223010021
8. Serebryakov V.A., Ataeva O.M. Ontology Based Approach to Modeling of the Subject Domain ‘‘Mathematics” in the Digital Library // Lobachevskii Journal of Mathematics, 2021. — Vol. 42, No. 8. — P. 1920–1934. — https://doi.org/10.1134/S199508022108028X
9. Lange C. Ontologies and languages for representing mathematical knowledge on the Semantic Web // Semantic Web, 2013. — Vol. 4, No. 2. — P. 119–158. — https://doi.org/10.3233/SW-2012-0059
10. Wang J. Math-KG: Construction and Applications of Mathematical Knowledge Graph, arXiv:2205.03772, 2022. — P. 1–5. — https://doi.org/10.48550/arXiv.2205.03772
11. Muromskii A.A., Tuchkova N.P. Predstavlenie matematicheskikh poniatii v ontologii nauchnykh znanii // Ontologiia proektirovaniia, 2019. — T. 9, №1 (31). — S. 50–69. — https://doi.org/10.18287/2223-9537-2019-9-1-50-69
12. Elizarov A.M., Kirillovich A.V., Lipachev E.K., Nevzorova O.A. Ontologiia matematicheskogo znaniia OntoMathPRO // Doklady Rossiiskoi akademii nauk. Matematika, informatika, protsessy upravleniia, 2022. — T. 507, № 1. — S. 29–35. — https://doi.org/10.31857/S2686954322700011
13. Elizarov A.M., Kirillovich A.V., Lipachev E.K., Nevzorova O.A. Novye komponenty ontologii OntoMathPRO predstavleniia matematicheskogo

znaniia // Nauchnyi servis v seti Internet: trudy XXV Vserossiiskoi nauchnoi konferentsii (18–21 sentiabria 2023 g., onlain). M.: IPM im. M.V. Keldysha, 2023. — S. 141–151. — https://doi.org/10.20948/abrau-2023-32
14. Nevzorova O.A., Gizatullin B.T. Sistema avtomaticheskogo postroeniia grafov znanii matematicheskikh dokumentov // Uchenye zapiski Kazanskogo universiteta. Seriia: Fiziko-matematicheskie nauki, 2023. — T. 165, № 3. — S. 264–281. https://doi.org/10.26907/2541-7746.2023.3.264-281
15. Lehmann J., Isele R., Jakob M., Jentzsch A., Kontokostas D., N. Mendes P., Hellmann S., Morsey M., Van Kleef P., Auer S., Bizer C. DBpedia – A Large-scale, Multi-lingual Knowledge Base Extracted from Wikipedia // Semantic Web, 2015. — Vol. 6, No. 2. — P. 167–195. — https://doi.org/10.3233/SW-140134
16. Vrandečić D., Krötzsch M. Wikidata: a free collaborative knowledgebase // Communications of the ACM, 2014. — Vol. 57, No. 10. — P. 78–85. — https://doi.org/10.1145/2629489
17. Schembera B., Wübbeling F., Kleikamp H., Schmidt B., Shehu A., Reidelbach M., Biedinger C., Fiedler J., Koprucki T., Iglezakis D., Göddeke D. Towards a Knowledge Graph for Models and Algorithms in Applied Mathematics // arXiv:2408.10003. — https://doi.org/10.48550/arXiv.2408.10003
18. Protégé. — https://protege.stanford.edu
19. Home // RDFLib. — https://rdflib.dev
20. SPARQL Endpoint interface to Python // SPARQLWrapper documentation — https://sparqlwrapper.readthedocs.io/en/latest/main.html
21. RDF // Semantic Web Standards. — https://www.w3.org/RDF.
22. Virtuoso Homepage // OpenLink Software. — https://virtuoso.openlinksw.com
23. VIGuryanov/Knowledge-Graphs-Builder // GitHub. — https://github.com/VIGuryanov/Knowledge-Graphs-Builder