

УДК 004.891.22

DOI: 10.25559/SITITO.14.201803.654-662

СБОР И АНАЛИЗ НАУЧНО-ТЕХНИЧЕСКОЙ ИНФОРМАЦИИ НА ОСНОВЕ ДИНАМИЧЕСКОЙ ОНТОЛОГИИ

А.П. Гагарин¹, В.В. Сердюков^{1,2}¹ Московский авиационный институт (национальный исследовательский университет), г. Москва, Россия² ООО «МПС», г. Москва, Россия

DYNAMIC ONTOLOGY BASED RETRIEVAL AND ANALYSIS OF INFORMATION ON SCIENCE AND ENGINEERING

Andrey P. Gagarin¹, Vladimir V. Serdyukov^{1,2}¹ Moscow Aviation Institute (National Research University), Moscow, Russia² «MPS» Ltd, Moscow, Russia

© Гагарин А.П., Сердюков В.В., 2018

Ключевые слова

Технический продукт; инновационный цикл; онтология; информационный ресурс; язык Resource Definition Language (RDF); JavaScript Object Notation (JSON); триплет; информационный поиск; сеть Интернет; интернет-бот.

Аннотация

Предлагается модель ассоциаций инновационных циклов наукоёмкой технической продукции в виде онтологии на языке RDF, выступающей как структурная основа персонализированного архива научно-технической информации. Инновационный цикл конкретного артефакта отображается в последовательность триплетов RDF, причём артефакт выступает субъектом триплетов, а процессы цикла – объектами, приобретая статус классов, которые в дальнейшем заполняются информационными ресурсами, относящимися к этим процессам. Артефакты в составе онтологии выстраиваются в иерархию, в которой «младшие» артефакты являются частным случаем «старших». В целом онтология образует верхний уровень архива информационных ресурсов, хранящегося в виде набора файлов. Реализовано клиент-серверное приложение, образующее программную оболочку такого архива. Когда приложение работает, онтология загружается в оперативную память в формате JSON как компонент сервера и доступна как для просмотра, так и редактирования. Онтология может быть показана в виде графа. Для «ручного» перебора и редактирования подграфы онтологии отображаются в таблицах. Работа в таблицах на уровне отдельных артефактов и процессов их жизненных циклов не требует от пользователя знания языка RDF. Пользователь имеет возможность вызвать автоматическое формирование поисковых запросов к интересующим его сайтам Интернета, которые будут опрашиваться по заданному графику. Опрос осуществляется интернет-ботом, встроенным в приложение. Результаты выполнения этих запросов доступны пользователю, который может вносить их в онтологию в виде новых артефактов или информационных ресурсов, привязанных к процессам жизненных циклов. Таким образом, приложение служит оболочкой онтологии, которая обеспечивает её автоматизированное обогащение за счёт релевантного контента, поступающего из сети.

Keywords

Technical product; innovation cycle; ontology; information resource; Resource Definition Language (RDF); JavaScript Object Notation (JSON); triple; information search; Internet; net-crawler.

Abstract

RDF ontologies are proposed as modelling pattern for associations of innovation cycles being performed by science-intensive technical products. An appropriate ontology is applied as framework of personalized archives of scientific and engineering information. Innovation cycle of a given artifact is mapped into a sequence of RDF triples, the artefact is a subject of triples thereby, and the processes of the cycle become subjects, each of them gains status of a class. Later, the classes are populated by information resources pertaining to the life cycle processes. Artefacts contained in the ontology build a hierarchy. «Junior» artefacts being special case of a «senior». As a whole, ontology represents an upper level in the archives of information

Об авторах:

Гагарин Андрей Петрович, кандидат технических наук, профессор, Московский авиационный институт (национальный исследовательский университет) (125993, Россия, г. Москва, Волоколамское ш., д. 4), ORCID: <http://orcid.org/0000-0002-0929-2834>, gagarin_ay@outlook.com

Сердюков Владимир Владимирович, магистр, Московский авиационный институт (национальный исследовательский университет) (125993, Россия, г. Москва, Волоколамское ш., д. 4); старший разработчик, ООО «МПС», ORCID: <http://orcid.org/0000-0002-2307-3291>, vl.ukov@yandex.ru



resources, residing as a set of files. A client/server application is implemented constituting program shell of the archives. If the application runs, the ontology is loaded into the operative memory in JSON format on server side and is accessible for perusal and editing. It may be displayed as a graph, and its subgraphs may be mapped into tables for manipulation and editing. Handling the ontology in tables on the level of distinct artefacts and processes of their life cycles does not assume user understanding of the RDF. User may invoke automated generation of search request on Internet sites of his interest in polling them according to a given schedule. The polling is performed by a net-crawler built in the application. The user can access the results of the search and may insert them into the ontology as new artefacts or as information resources, associated with life cycle processes. That is, the application appears to be a shell for the ontology enabling its automated enrichment by the relevant content coming from the Internet.

Введение

В области компьютерной обработки научно-технической информации, получаемой из глобальных сетей, далеко не решена задача накопления и анализа её профессиональной части на столе научно-технического работника в его индивидуальных целях, в том числе, для расширения его кругозора, поиска партнёров, отслеживания деятельности конкурентов, оценки перспективности и выбора конкретных направлений личной инженерной и научной практики.

Словарный поиск в сети Интернет с помощью современных браузеров эту задачу не решает не только потому, что в результате поиска выдаются большие объёмы информации с повторениями и привязкой к лексике, но и потому, что никак не обеспечивается сохранение этой информации в виде, пригодном для дальнейшего использования.

Онтологии в поисковых системах углубляют поиск, выводя его за пределы словарного, но требуют, в качестве предпосылки успешного применения, наличия актуальной онтологии для конкретной научно-технической области, а разработка и развитие любой онтологии остается трудоемкой инженерной задачей.

Не отказываясь от автоматического поиска и анализа, но не получая при этом удовлетворительного результата по сроку или содержанию, пользователь заинтересован в частичной автоматизации «персональной информационной службы», которая позволила бы с минимальной детальностью управлять селективным поиском и презентацией внешних источников и анализировать собранную информацию «ручным способом» в соответствии с личным профессиональным интересом.

Ключевым решением для такой частичной автоматизации является выбор модели данных для локального архива собранной информации. Эта модель должна быть благоприятна как для автоматизации поиска, так и сквозного просмотра, а также графической визуализации. Представляемая для анализа информация должна быть ограничена по объёму, поскольку анализ является для работника фоновой деятельностью, но допустить оперативную проверку, уточнение и, желательно, автоматически обновляться.

Постановка задачи

Необходима модель данных, способная служить основой «локального» внутри-компьютерного архива научно-технической информации, отражающей состояние и процессы развития фрагмента техносферы, в котором можно производить автоматический поиск, как в глобальной сети, и, наряду с этим, «видеть» хранящуюся информацию в целом и по частям, различать связи между частями и осуществлять визуальный поиск, следуя этим связям.

Эта модель должна допускать построение программной оболочки архива, обеспечивающей его пополнение и доступ к нему пользователя. Кроме чтения контента архива, пользователю должны быть предоставлены возможности (сценарии) задавать тематику автоматически повторяющегося поиска в конкретных источниках сети Интернет и производить отбор собранной информации для занесения в архив.

Подход к решению задачи

Предлагается строить искомую модель данных на основе концепции «инновационного цикла». Эта концепция давно и широко используется в экономической науке. В [1] можно найти её современную трактовку и обстоятельную библиографию. В [2, 3] это понятие использовано для имитационного моделирования процессов развития информационных технологий.

Инновационный цикл представляет собой замкнутую последовательность процессов, точный состав и характеристики которых варьируют у разных авторов [4]. В данной статье принята упрощённая версия инновационного цикла, показанная на Рис.1: в частности, не представлены процессы производства и технического сопровождения, внедрение объединено с эксплуатацией. Эти упрощения допущены в связи с тем, что в качестве продуктов предполагается рассматриваются преимущественно программные.

Установлено, что жизненный цикл состоит из процессов: «ОКР» - разработка продукта, «Внедрение» - процесс ввода продукта в эксплуатацию и дальнейшая эксплуатация, «Оценки» - процесс оценки продукта, «НИР» - процесс решения проблем,



отмеченных в оценке продукта, с целью усовершенствования продукта или перехода к разработке иного, нового продукта.

Предлагаемая модель является графом, каждый узел которого представляет жизненный цикл некоторого технического изделия, далее – «артефакта». Артефакт, в свою очередь, представляется четвёркой триплетов – предложений языка RDF [5]. Каждый RDF-триплет состоит из трех компонентов: субъекта, предиката и объекта.

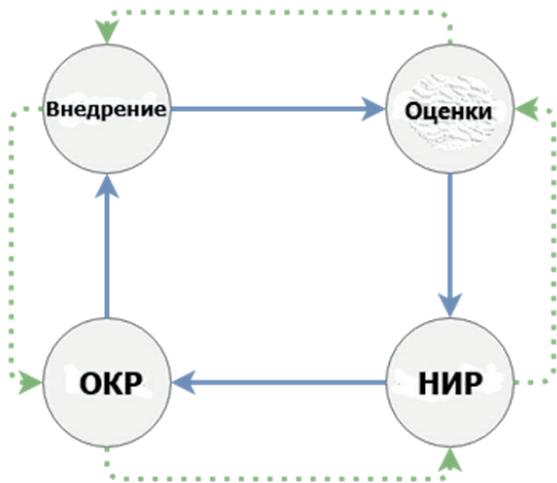


Рис. 1. Схема инновационного цикла
Fig. 1. Innovation Cycle Scheme

Во всех триплетях артефакта субъект один и тот же – обозначение артефакта. Предикат также единый – «is». В качестве объектов указываются обозначения процессов жизненного цикла. То есть, процессы жизненного цикла рассматриваются как формальные свойства артефакта. Это даёт основание в дальнейшем именовать «артефактом» всю четвёрку триплетов RDF. Совокупность таких четвёрок образует онтологию, моделирующую некоторый фрагмент техносферы. Схематическое изображение артефакта приведено на Рис. 2.

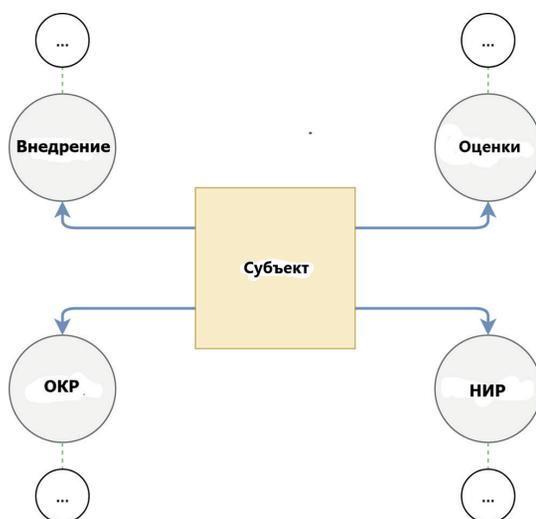


Рис. 2. Схема артефакта
Fig. 2. Artifact Scheme

Кроме указанных триплетов, в эту онтологию входят триплеты, «привязывающие» к процессам релевантные информационные ресурсы.

В фрагменте онтологии на Рис.3 в качестве артефактов определены программные продукты Node.js [6] и webVOWL [7], среда разработки Express [8], пакеты fs-js [9] и cheerio [10], менеджер пакетов npmjs [11], библиотеки puppeteer [12] и N3.js [13].

Предикат «а» определён в онтологии и выражает принадлежность субъекта, первого элемента триплета – классу, идентифицированному как его объект. Данная онтология представлена в нотации TriG [14]. В этой нотации триплеты с одним и тем же субъектом и предикатом могут объединяться в конструкции, в которых объекты перечисляются через запятую. В частности, третье предложение онтологии указывает, что Node.js является средой runtime – времени выполнения. Node.js как артефакт является частным случаем абстрактного (обобщённого) артефакта runtime. Далее показано, что пакеты fs-js и cheerio выполняются в среде Node.js. Объекты триплетов с суффиксами #USAGE, #REVIEW, #OKP, #NIR идентифицируют классы, через которые к артефакту «прикрепляются» информационные ресурсы. Например, URL <https://www.npmjs.com/package/fs-js> прикреплен к артефакту node.js через класс fs-js_package#OKP.

```
<Node.js> а <Node.js#NIR>, <Node.js#OKP>, <Node.js#REVIEW>,
<Node.js#USAGE>, <runtime>.
<webvowl_application> а <webvowl_application#REVIEW>,
<webvowl_application#OKP>,
<webvowl_application#NIR>,
<IT>.
<Express> а <Express#NIR>, <Express#OKP>, <Express#REVIEW>,
<Express#USAGE>, <IDE>.
<IDE> а <IDE#NIR>, <IDE#OKP>, <IDE#REVIEW>, <IDE#USAGE>, <IT>.
<fs-js_package> а <fs-js_package#USAGE>, <fs-js_package#REVIEW>,
<fs-js_package#OKP>, <fs-js_package#NIR>, <Node.js>.
<https://www.npmjs.com/package/fs-js> а <fs-js_package#OKP>.
<cheerio> а <cheerio#USAGE>, <cheerio#REVIEW>, <cheerio#OKP>,
<cheerio#NIR>, <Node.js>.
<https://www.npmjs.com/package/cheerio> а <cheerio#OKP>.
<npmjs> а <npmjs#USAGE>, <npmjs#REVIEW>, <npmjs#OKP>,
<npmjs#NIR>, <Node.js>.
<JavaScript_package_manager_for_Nodejs> а <Node.js#REVIEW>.
<https://www.npmjs.com/> а <Node.js#NIR>.
<puppeteer_library> а <puppeteer_library#USAGE>, <puppeteer_library#REVIEW>,
<puppeteer_library#OKP>,
<puppeteer_library#NIR>, <Node.js>.
<control_of_Chrome_or_Chromium_over_the_DevTools_Protocol> а
<puppeteer_library#REVIEW>.
<https://pptr.dev/> а <puppeteer_library#OKP>.
<Notation3_(N3)> а <Notation3_(N3)#USAGE>, <Notation3_(N3)#REVIEW>,
<Notation3_(N3)#OKP>, <Notation3_(N3)#NIR>, <RDF/OWL>.
<N3.js_library_(NPM-package)> а <N3.js_library_(NPM-package)#USAGE>,
<N3.js_library_(NPM-package)#REVIEW>,
<N3.js_library_(NPM-package)#OKP>,
<N3.js_library_(NPM-package)#NIR>,
<Node.js>.
```

Рис. 3. Фрагмент онтологии
Fig. 3. Fragment of ontology



Распределение информационных ресурсов по элементам жизненного цикла артефакта и группировка соответствующих триплетов способствует их эффективному нахождению как при автоматизированном, так и визуальном поиске в тексте антологии.

Программные продукты, пакеты и библиотеки из рассмотренного фрагмента онтологии составили технологическую базу, на которой разработана информационно-поисковая и архивная система ТРИАДА+, демонстрирующая применение предложенного подхода к решению задачи, поставленной в данной публикации.

Апробация предложенного подхода

Система ТРИАДА+ состоит из частей:

- сбора внешних данных в сети Интернет в автоматическом режиме, а также по запросам пользователя, как в обычном браузере;
- архивной;
- оперативной.

Оперативная часть даёт пользователю возможность управлять поиском данных в сети, просматривать собранные данные и отбирать их для архива, обращаться в архив для анализа и дальнейшей обработки.

Система ТРИАДА+ организована на принципах клиент-сер-

верной архитектуры. На серверной стороне применен компонент Node.js. Он позволяет использовать готовые пакеты NPM и выполнять программы, написанные на языке JavaScript.

Модуль nw.exe представляет систему ТРИАДА+ как приложение в операционной среде Windows, запускает сервер системы и открывает страницы, в виде которых функционирует оперативная часть системы. В качестве полноценного web-сервера используется Express.js. В качестве клиентского программного обеспечения может быть также использован обычный веб-браузер.

Архивная часть системы состоит из онтологии и директорий, содержащих информационные ресурсы в виде файлов.

Онтология хранится в виде rdf-модулей, и обновляется на стороне сервера в памяти приложения, причём создаётся и поддерживается хеш-дерево с информацией о всех артефактах онтологии. Это позволяет существенно ускорить скорость работы с ней. Создание и пополнение онтологии, полностью физическ в ней осуществляются на сервере «управляющим слоем» с помощью ресурсов пакетов N3.js, fs-js и cheerio по обращению от клиентских страниц.

Библиотека пакета N3.js реализует большую часть спецификаций RDF, в частности, поддерживает синтаксис TriG. Пакет fs-js обеспечивает работу с файлами, а cheerio – выборку по запросам на языке JQuery [15]. Рис. 4 передаёт вид страницы, в которой пользователь работает с онтологией.

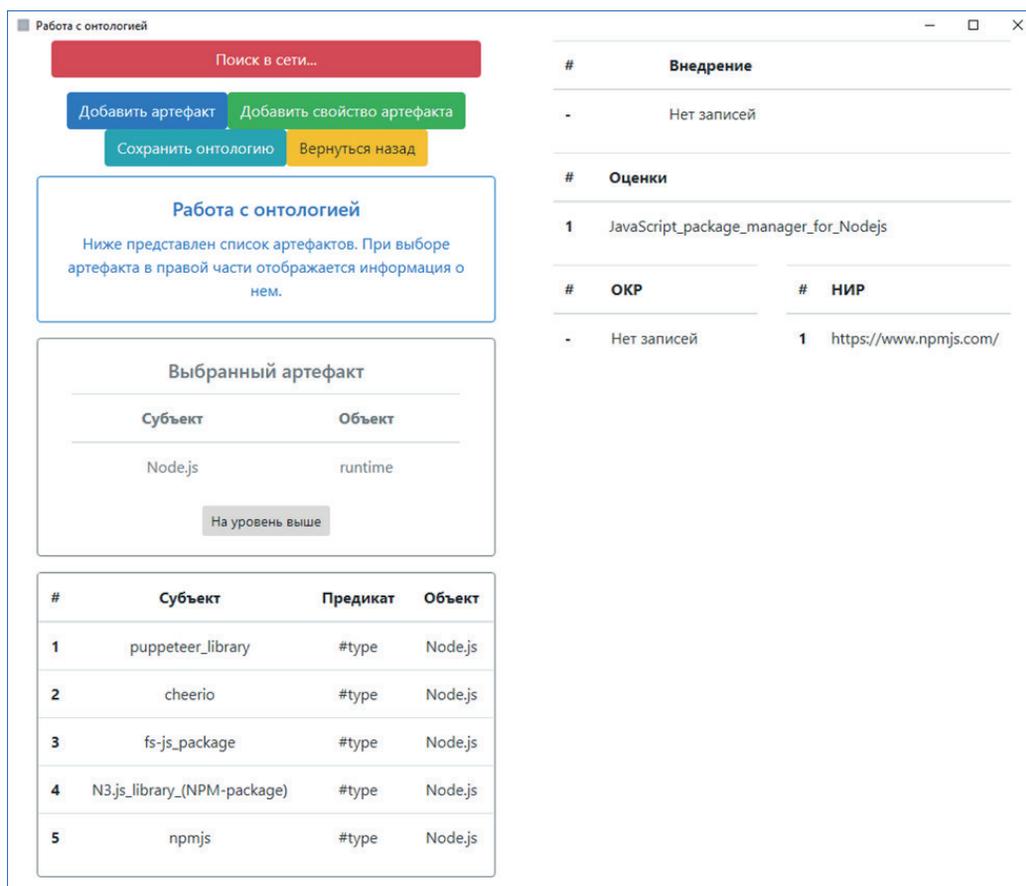


Рис. 4. Страница для работы с онтологией
Fig. 4. Page for working with ontology



Основной блок для просмотра и выбора текущего рабочего артефакта онтологии находится в левой части страницы.

Под кнопками и рамкой с инструктивной информацией находится однострочная таблица, в которой отображаются субъект текущего (выбранного) артефакта и как его объект – субъект артефакта, частным случаем которого является текущий, в примере – Node.js и runtime. По умолчанию субъектом является “IT”. Здесь же находится кнопка для перемещения на следующий более высокий уровень текущего графа онтологии, в данном случае – к артефакту runtime.

В правой части страницы находится таблица информационных ресурсов выбранного артефакта, разделённая на 4 части по принадлежности ресурса к этапу его жизненного цикла. Ниже слева располагается таблица с артефактами следующего более

низкого уровня онтологии. Выбор строки в этой таблице мышью перемещает выбранный артефакт в вышележащую таблицу и происходит полная смена текущего контекста остальных таблиц. Вверху страницы находятся кнопки, вызывающие добавление артефакта, сохранение, а также переход к поиску в сети и визуальному представлению онтологии.

Для визуализации используется компонент WebVOWL [11], поддерживающий RDF-нотацию и позволяющий выводить артефакты и связи между ними в виде дерева графов. На Рис.5 в виде графа показан пример онтологии, фрагмент текста которой рассмотрен выше в данной публикации. Кружки представляют элементы онтологии, являющиеся OWL-классами. Для выделенного элемента информационные ресурсы перечисляются под заголовком Individuals в правом поле страницы.

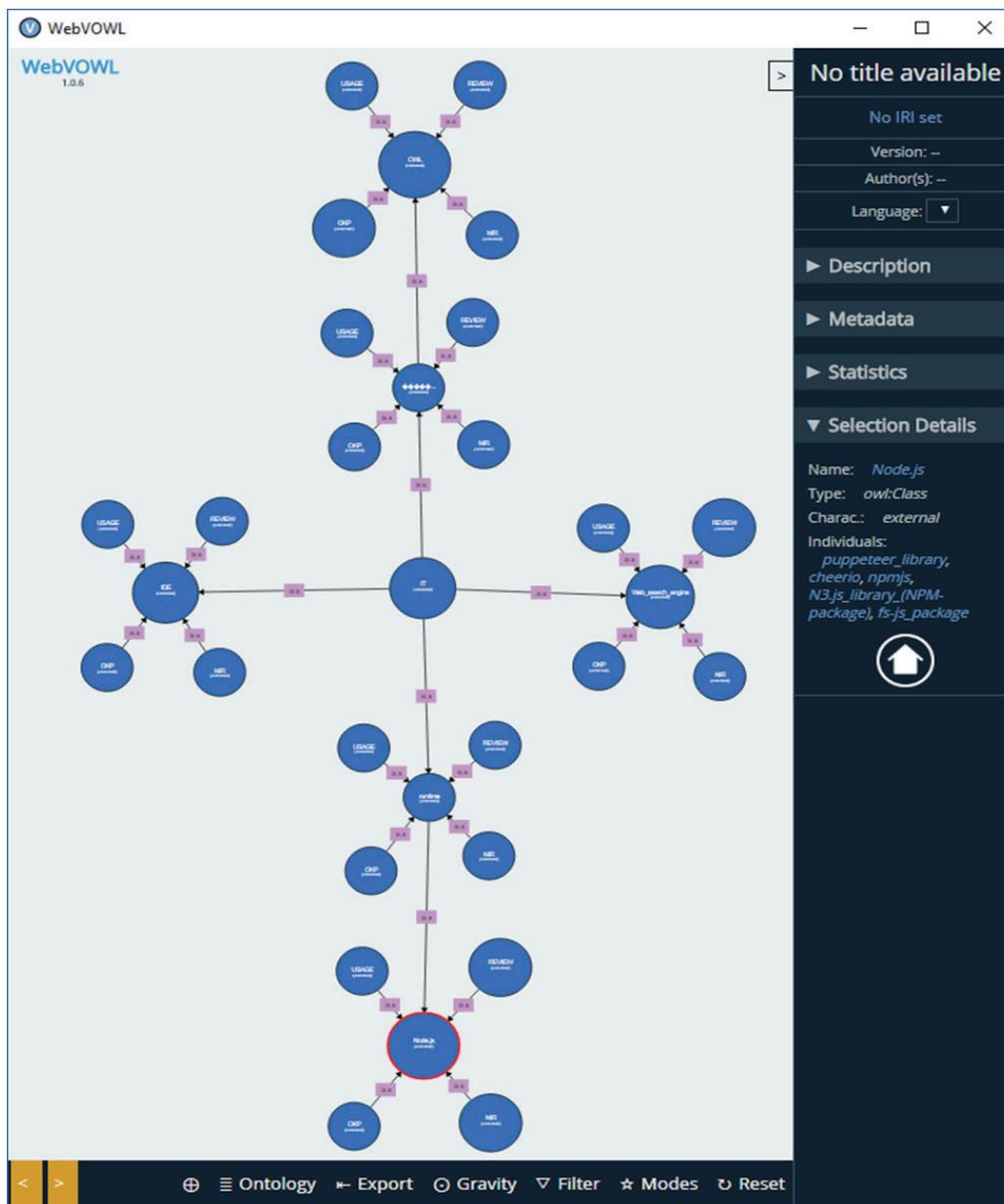


Рис. 5. Визуальное представление онтологии посредством программы WebVOWL

Fig. 5. Visual representation of ontology through the program WebVOWL



Поиск информации в сети

Для автоматического сбора информации в сети используется компонент Puppeteer [12]. Это фреймворк для использования браузера Chrome без пользовательского интерфейса. Он позволяет использовать протокол Chrome DevTools [16] для автоматизированного управления программой. В отличие от подобных решений, в данном случае присутствует поддержка выполнения JavaScript кода, так как это, по сути, полноценный браузер. С его помощью осуществляется сбор информации по запросу с доверенных ресурсов. Для каждого из таких ресурсов необходимо реализовать своего интернет-бота, для которого создается свой собственный профиль поведения и правила для извлечения информации из страницы с результатом запроса.

Например, для ресурса eLibrary.ru поисковый бот настраивается таким образом, чтобы повторять действия обычного пользователя [17]. Для этого выполняется ряд действий:

- открывается начальная страница;
- заполняется поле для поиска;

- выбираются дополнительные параметры поиска;
- происходит ожидание ответа от интернет-ресурса;
- проверяется факт того, что ответ получен;
- выделяется только нужная часть ответа;
- результат отображается в пользовательском интерфейсе.

Аналогичным образом формируется алгоритм действий для других поисковых ресурсов. Результаты обработки ответа от сервера сохраняются в виде JSON-структур данных [18]. Далее эта информация используется для сбора аналитической информации о самом ресурсе. Стандартный сценарий предполагает осуществление поисковых запросов в сети на основании данных от пользователя и о текущем рабочем наборе, выбранном в режиме редактирования онтологии. Для этого пользователь выбирает конкретный артефакт онтологии или его свойства. В случае, показанном на Рис. 6, это node.js, «вокруг» которого происходит формирования запроса – объединение имени данного артефакта и его свойства. На Рис. 7 отражены частичные результаты поиска по данному запросу.

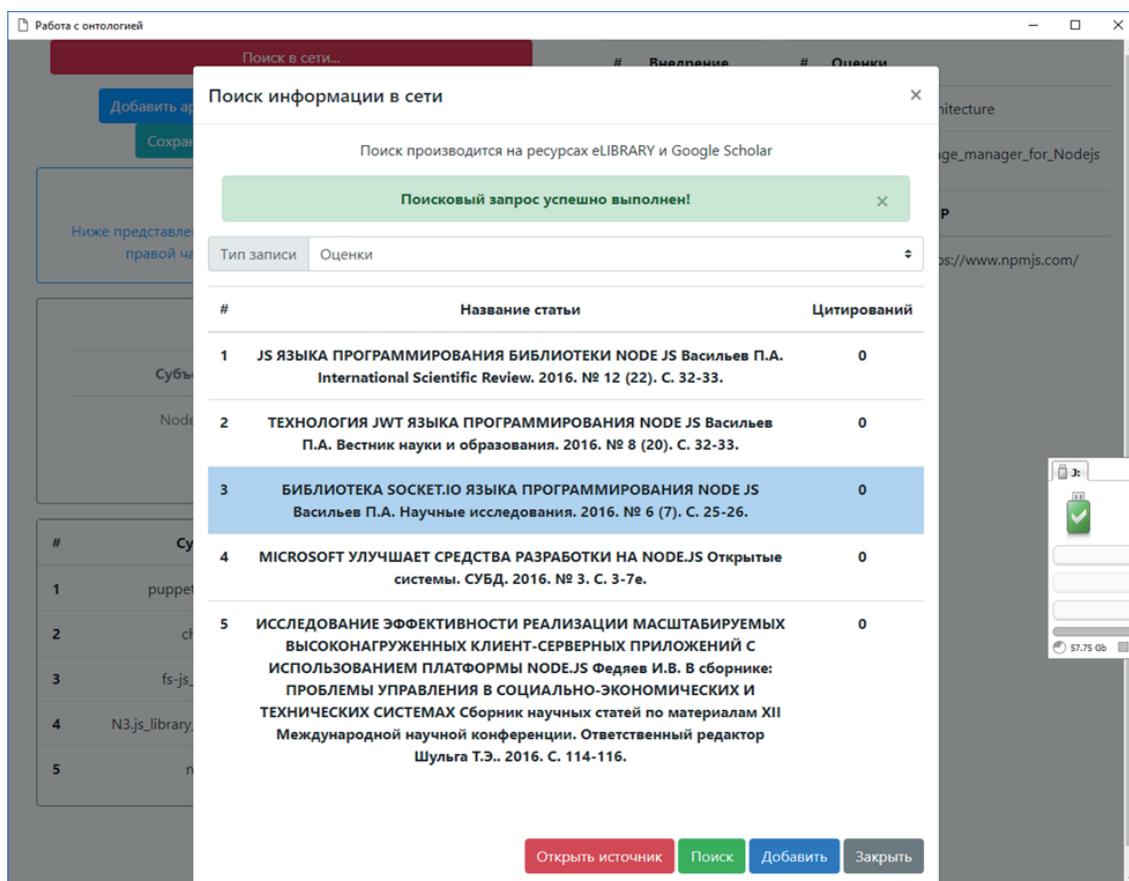


Рис. 6. Визуальное представление онтологии посредством программы WebVOWL
 Fig. 6. Visual representation of ontology through the program WebVOWL

После успешного получения ответа на поисковый запрос пользователь может приступить к просмотру и добавлению записи. При нажатии на строку в результатах запроса можно перейти к содержимому этой записи на исходном ресурсе. Предоставляется также возможность открыть полноэкранное представление этой записи по данному адресу URL в браузере по

умолчанию. Далее пользователь может добавить его содержимое в онтологию с привязкой к текущему рабочему артефакту (CPU).

На Рис. 6 показан результат занесения в онтологию информационного ресурса, найденного в сети, с привязкой к артефакту, послужившего основой поиска.



The screenshot shows the WebVOWL application interface. At the top, there is a search bar and buttons for adding artifacts and properties. Below this, a section titled "Работа с онтологией" (Working with ontology) provides instructions. A "Выбранный артефакт" (Selected artifact) section shows a table with columns "Субъект" (Subject) and "Объект" (Object), displaying "Node.js" and "runtime". Below this is a main table with columns "#", "Субъект", "Предикат", and "Объект", listing several Node.js related artifacts. On the right side, there are two panels: "Внедрение" (Introduction) and "Оценки" (Reviews), both showing "Нет записей" (No records). Below these are panels for "ОКР" (OKR) and "НИР" (NIR), with the NIR panel displaying a list of items including "MICROSOFT УЛУЧШАЕТ СРЕДСТВА РАЗРАБОТКИ НА NODE.JS" and a URL.

#	Субъект	Предикат	Объект
1	puppeteer_library	#type	Node.js
2	cheerio	#type	Node.js
3	fs-js_package	#type	Node.js
4	N3.js_library_(NPM-package)	#type	Node.js
5	npmjs	#type	Node.js

Рис. 7. Визуальное представление онтологии посредством программы WebVOWL
Fig. 7. Visual representation of ontology through the program WebVOWL

Для решения проблемы, связанной с наличием большого количества бесполезной информации в сети Интернет применяется подход, в котором строго ограничивается список ресурсов для поиска. В первую очередь стоит рассматривать специализированные источники данных. Этот подход имеет следующие достоинства:

- контент специализированных ресурсов проходит строгую модерацию, а значит, содержит в себе полезную информацию;
- научная значимость материала на подобном ресурсе может быть достаточно легко оценена методом подсчета ссылок на него и количеством цитирований;
- обычно у пользователя существует доступ к статистическим данным выбранного ресурса для анализа информации в его интересах;
- существует возможность относительно простой публикации собственного материала на данном ресурсе.

Однако использование подобных специализированных ресурсов также имеет определенные недостатки. К ним можно отнести:

- обязательные требования по регистрации пользователя в системе;
- применение различных ограничений на доступ к информации, в том числе с применением платных вариантов подписки на сервис;
- требования по соблюдению регламента оформления работ для публикации и/или их просмотра;
- ограничение на количество просматриваемой информации за сутки.

Заключение

Предложена модель данных, основанная на отношениях «частное-общее» и «артефакт-процесс жизненного цикла артефакта» и позволяющая кластеризовать массивы слабо структурированной информации, отражающие состояние развития произвольной области техносферы, по ключам, которые представляют универсальный профессиональный интерес для инженерных и научных работников. Отображение этой модели в онтологию в одной из нотаций OWL/RDF позволило построить поисковую и информационно-справочную систему для управляемого полуавтоматического накопления научно-технической информации из выбранных источников Интернет в соответствии с интересами конкретного пользователя. Онтология применяется как для селективной презентации накопленного материала, так и для формирования поисковых запросов к сети. При этом сама онтология пополняется по мере обнаружения в результатах выполненных запросов сведений, релевантных для пользователя системы.



Список использованных источников

- [1] Баринаева В.А., Земцов С.П. Инновационный цикл как базовая модель динамики и организации инновационной деятельности // Вестник Института экономики Российской академии наук. 2016. № 1. С. 117-127. URL: <https://elibrary.ru/item.asp?id=25463313> (дата обращения: 12.05.2018).
- [2] Гагарин А.П. Гипертекстовая технология моделирования сетей научно-технических проблем // Успехи современной радиоэлектроники. 2011. № 1. С. 72-76. URL: <https://elibrary.ru/item.asp?id=16335721> (дата обращения: 12.05.2018).
- [3] Гагарин А.П., Березняк Т.В. Динамика инновационных процессов в области информационных технологий и подходы к их моделированию // Научноёмкие технологии. 2014. Т. 15, № 1. С. 107-116. URL: <https://elibrary.ru/item.asp?id=21153987> (дата обращения: 12.05.2018).
- [4] Хайруллин Р.А. Этапы инновационного процесса // Фундаментальные исследования. 2011. № 12-4. С. 809-813. URL: <http://www.fundamental-research.ru/ru/article/view?id=29485> (дата обращения: 12.05.2018).
- [5] RDF Current Status [Электронный ресурс] // The World Wide Web Consortium (W3C). URL: https://www.w3.org/standards/techs/rdf#w3c_all (дата обращения: 12.05.2018).
- [6] Node.js Foundation [Электронный ресурс] // URL: <https://nodejs.org/en/> (дата обращения: 12.05.2018).
- [7] WebVOWL: Web-based Visualization of Ontologies [Электронный ресурс] // VOWL: Visual Notation for OWL Ontologies. URL: <http://vowl.visualdataweb.org> (дата обращения: 12.05.2018).
- [8] Express [Электронный ресурс]. URL: <https://expressjs.com/ru/> (дата обращения: 12.05.2018).
- [9] fs-jobs - npm [Электронный ресурс]. URL: <https://www.npmjs.com/package/fs-jobs> (дата обращения: 12.05.2018).
- [10] cheerio - npm [Электронный ресурс]. URL: <https://www.npmjs.com/package/cheerio> (дата обращения: 12.05.2018).
- [11] puppeteer [Электронный ресурс]. URL: <https://www.npmjs.com/puppeteer> (дата обращения: 12.05.2018).
- [12] puppeteer [Электронный ресурс]. URL: <https://pptr.dev/> (дата обращения: 12.05.2018).
- [13] n3 - npm [Электронный ресурс]. URL: <https://www.npmjs.com/package/n3> (дата обращения: 12.05.2018).
- [14] RDF 1.1 TriG [Электронный ресурс] // The World Wide Web Consortium (W3C). URL: <https://www.w3.org/TR/2014/REC-trig-20140225/#bib-TURTLE> (дата обращения: 12.05.2018).
- [15] JQuery [Электронный ресурс]. URL: <https://jquery.com/> (дата обращения: 12.05.2018).
- [16] Chrome Developer Tools [Электронный ресурс]. URL: <https://developers.google.com/web/tools/chrome-devtools/> (дата обращения: 12.05.2018).
- [17] Aggarwal C.C. Information Retrieval and Search Engines / C.C. Aggarwal (Ed.) // Machine Learning for Text. Springer International Publishing, 2018. Pp. 259-304. DOI: 10.1007/978-3-319-73531-3
- [18] Standard ECMA-404. The JSON Data Interchange Syntax. 2nd Edition. December 2017 [Электронный ресурс]. URL: <http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-404.pdf> (дата обращения: 12.05.2018).

Поступила 12.05.2018; принята в печать 05.09.2018;
опубликована онлайн 30.09.2018.

References

- [1] Barinova V.A., Zemtsov S.P. Innovation cycle as a basic model of the dynamics and organization of innovative activity. *Vestnik Instituta Ehkonomiki Rossijskoj Akademii Nauk*. 2016; 1:117-127. Available at: <https://elibrary.ru/item.asp?id=25463313> (accessed 12.05.2018). (In Russian)
- [2] Gagarin A.P. Hypertext technology modeling networks of scientific and technical problems. *Uspechi Sovremennoj Radioelektroniki*. 2011; 1:72-76. Available at: <https://elibrary.ru/item.asp?id=16335721> (accessed 12.05.2018). (In Russian)
- [3] Gagarin A.P., Bereznjak T.V. Dynamics of innovative processes in the area of information technologies and approaches to their simulation. *Science Intensive Technologies*. 2014; 15(1):107-116. Available at: <https://elibrary.ru/item.asp?id=21153987> (accessed 12.05.2018). (In Russian)
- [4] Khayrullin R.A. Stages of Innovative Process. *Fundamentalnye issledovaniya* = Fundamental research. 2011; 12(4):809-813. Available at: <http://www.fundamental-research.ru/ru/article/view?id=29485> (accessed 09.09.2018). (In Russian)
- [5] RDF Current Status. The World Wide Web Consortium (W3C). Available at: https://www.w3.org/standards/techs/rdf#w3c_all (accessed 12.05.2018).
- [6] Node.js Foundation. Available at: <https://nodejs.org/en/> (accessed 12.05.2018).
- [7] WebVOWL: Web-based Visualization of Ontologies. VOWL: Visual Notation for OWL Ontologies. Available at: <http://vowl.visualdataweb.org> (accessed 12.05.2018).
- [8] Express. Available at: <https://expressjs.com/ru/> (accessed 12.05.2018).
- [9] fs-jobs - npm. Available at: <https://www.npmjs.com/package/fs-jobs> (accessed 12.05.2018).
- [10] cheerio - npm. Available at: <https://www.npmjs.com/package/cheerio> (accessed 12.05.2018).
- [11] Npm. Available at: <https://www.npmjs.com/> (accessed 12.05.2018).
- [12] puppeteer. Available at: <https://pptr.dev/> (accessed 12.05.2018).
- [13] n3 - npm. Available at: <https://www.npmjs.com/package/n3> (accessed 12.05.2018).
- [14] RDF 1.1 TriG. The World Wide Web Consortium (W3C). Available at: <https://www.w3.org/TR/2014/REC-trig-20140225/#bib-TURTLE> (accessed 12.05.2018).
- [15] JQuery. Available at: <https://jquery.com/> (accessed 12.05.2018).
- [16] Chrome Developer Tools. Available at: <https://developers.google.com/web/tools/chrome-devtools/> (accessed 12.05.2018).
- [17] Aggarwal C.C. Information Retrieval and Search Engines. C.C. Aggarwal (Ed.) Machine Learning for Text. Springer International Publishing, pp. 259-304, 2018. DOI: 10.1007/978-3-319-73531-3
- [18] Standard ECMA-404. The JSON Data Interchange Syntax. 2nd Edition. December 2017. Available at: <http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-404.pdf> (accessed: 12.05.2018).

Submitted 12.05.2018; revised 05.09.2018; published online 30.09.2018.



About the authors:

Andrey P. Gagarin, Candidate of Engineering Sciences, Professor on Informatics, Moscow Aviation Institute (National Research University) (4 Volokolamsk highway, Moscow 125993, Russia), ORCID: <http://orcid.org/0000-0002-0929-2834>, gagarin_ay@outlook.com

Vladimir V. Serdyukov, magister of Moscow Aviation Institute (National Research University) (4 Volokolamsk highway, Moscow 125993, Russia); senior engineer of "MPS" Ltd, ORCID: <http://orcid.org/0000-0002-2307-3291>, vl.ukov@yandex.ru



This is an Open Access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0>), which permits unrestricted reuse, distribution, and reproduction in any medium provided the original work is properly cited.

