УДК 004.43 doi: 10.15827/0236-235X.142.223-227 2023. Т. 36. № 2. С. 223–227

Особенности работы с русскоязычными онтологиями с помощью библиотеки Owlready2 на языке Python

И.А. Щукарев

Ссылка для цитирования
Щукарев И.А. Особенности работы с русскоязычными онтологиями с помощью библиотеки Owlready2 на языке Python // Программные продукты и системы. 2023. Т. 36. № 2. С. 223–227. doi: 10.15827/0236-235X.142.223-227
Информация о статье
Поступила в редакцию: 30.01.2023 После доработки: 16.02.2023 Принята к публикации: 03.03.2023

Аннотация. При работе в библиотеке Owlready2 языка Python с онтологиями, в которых изначально классы, индивидуумы и отношения написаны кириллицей, машина логического вывода reasoner выдает некорректные данные. Вследствие сбоя в кодировке Owlready2 дублирует онтологию, а вместо текста кириллицы появляются нечитаемые символы. Решить данную проблему предлагается путем явного задания кодировки выходных данных в файле reasoning.py, а именно cp1251, то есть стандартной 8-битной кодировки для русских версий Microsoft Windows. Сам файл находится в корневой папке программы – \Python\Python311\Lib\site-packages\owlready2\ для версии Python 3.11.0. Он и рассматривается в данной статье. Для поиска решения использован метод сравнительного анализа различных версий библиотеки Owlready2 и ее предшественника – библиотеки Owlready. Решение найдено путем сравнения команд работы с исходными данными в различных версиях библиотек Owlready. После внесения соответствующего изменения упрощается работа с онтологиями в Python, особенно при многократных запусках машины логического вывода reasoner. Становится возможным использование огромного функционала библиотеки Owlready2 для работы с русскоязычными онтологиями, например, для создания соответствующих русскоязычных баз знаний. Предложенный в статье способ может быть полезен для ИТ-специалистов, разрабатывающих информационные системы на основе онтологий предметных областей, а также при работе с онтологиями в рамках образовательного процесса в вузе.

Ключевые слова: онтология, информационная система, Python, Protege, Owlready2, русскоязычные онтологии

При решении прикладных задач в различных областях науки и техники активно используются возможности искусственного интеллекта. В информатике для формализованного представления знаний о закономерностях различных исследуемых предметных областей сегодня успешно применяются онтологии, под которыми в данном случае понимается способ целостной формализации знаний о какой-либо предметной области с помощью понятий и отношений между ними [1–3]. Онтологию можно построить как некую иерархическую структуру классов, связанных понятиями, выразив ее в графическом виде. Связь реализуется с помощью триплета и имеет следующий вид: «субъект–отношение–объект». Принцип применения онтологий предметных областей для разработки информационных процессов и систем получает все более широкое распространение. На основе онтологий возможно создание баз знаний, являющихся необходимой составляющей большинства информационных систем. Например, один из прикладных вариантов использования онтологий – их применение в образовании, поскольку систематизация и упорядочение знаний в числе главных целей образовательного процесса [4, 5].

Инструментарий для построения онтологий

Существует большое количество различных программных средств, модулей и библиотек для создания онтологий и работы с ними. Наиболее популярными являются Protégé, Fluent Editor и модуль Owlready2 для языка программирования Python [6–8].

Так, Protégé представляет собой Java-программу, включающую редактор онтологий, который позволяет проектировать онтологии, создавая иерархическую структуру классов. Protégé поддерживает язык OWL и дает возможность генерировать HTML-документы, отражающие структуру онтологии [9], а сложные ограничения реализуются с помощью Manchester Syntax [10]. Python, в свою очередь, является высокоуровневым языком программирования общего назначения с открытым исходным кодом, который поддерживает подходы объектно-ориентированного и структурного программирования. Python отлично подходит для решения проблем, возникающих в различных сферах человеческой деятельности [11].

Для работы с онтологиями в Python предназначен программный модуль Owlready2, который может управлять онтологиями и графами знаний [12]. К основным возможностям библиотеки Owlready2 можно отнести следующие:

– импорт и экспорт онтологий OWL 2.0 в форматах NTriples, RDF/XML или OWL/XML;

– управление классами онтологий, экземплярами и свойствами как обычными объектами Python;

– использование рассуждений (Reasoning) HermiT или Pellet;

– поддержка оптимизированных запросов SPARQL и т.д.

Постановка задачи

Сегодня востребованы как англоязычные, так и русскоязычные онтологии [13]. Наиболее известным инструментом для создания онтологий является Protégé [14]. Если, например, онтология создается лишь для образовательного процесса в качестве визуального представления необходимой предметной области и не предполагает дальнейшую обработку в какой-либо программе, например в Python, то можно ограничиться только средствами Protégé.

Можно также создать русскоязычную онтологию на основе англоязычной благодаря таким возможностям Protégé, как annotation properties и rdfs:label. Однако этот метод не подходит для работы с большими онтологиями, поскольку требует значительных временных затрат на присвоение каждому объекту онтологии на латинице русскоязычного аналога. В данной статье предлагается метод, позволяющий работать с онтологиями, например в Python, для создания которых первоначально использовалась лишь кириллица.

При разработке какой-либо информационной системы на основе онтологии в большинстве случаев требуется пользовательский русскоязычный интерфейс будущего приложения, который можно создать, например, с помощью GUI MATLAB [15]. Однако использование в таком случае языка программирования Python предпочтительнее из-за наличия мощного модуля для работы как с онтологиями (библиотека Owlready2), так и с графикой (библиотека Tkinter). Для конечного пользователя важным является и наличие русского языка в программе. Поэтому актуально создание русскоязычных онтологий, которые могут быть положены в основу русскоязычных информационных систем.

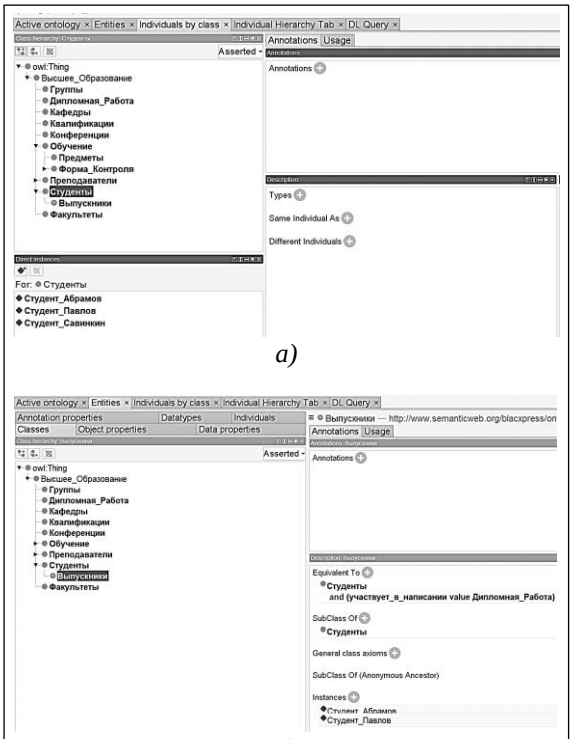
Модуль Owlready2 позволяет создавать и вносить необходимые изменения в уже существующие и вновь создаваемые онтологии. Основным инструментом для поиска новых знаний или фактов, находящихся в онтологии, является машина логического вывода, или reasoner. Инструмент reasoner автоматически классифицирует онтологию и проверяет ее на согласованность, то есть непротиворечивость. Кроме того, reasoner осуществляет автоматическое построение иерархий классов, классификацию индивидов на основе заданных или введенных в онтологию данных, ограничений и описаний индивидов. Эти действия позволяют получать новые факты в онтологиях для их дальнейшего использования [16].

В результате после создания онтологии необходимой предметной области в Protégé для всех дальнейших действий, например, для создания простейшей информационной системы, можно пользоваться лишь инструментами и библиотеками Python. Существует несколько видов reasoner в Owlready2: модифицированные версии резонера HermiT и Pellet, аналогичные решения есть и в Protégé. Актуальны версии используемых в данной статье программ: Protégé 5.5.0, Python 3.11.0 (модуль Owlready2 0.40), JDK 19.0.1 (64-bit), PyCharm Community Edition 2022.2.4

Программы Protégé и Python с библиотекой Owlready2 с англоязычными онтологиями работают корректно. Файл с онтологией читается и отображается без нарушений кодировки. После добавления плагина OntoGraf в Protégé структура онтологии также может представляться в виде графа, что очень удобно.

В качестве примера составим в Protégé русскоязычную онтологию, например, обучение студента в вузе (рис. 1). Согласно заданным условиям, после запуска reasoner в Protégé получаем новый факт: если студенты пишут дипломную работу, то они являются выпускниками, хотя сам этот факт исходно не задан. Как и при работе с англоязычной онтологией, сбоя в кодировке нет и файл отображается без видимых проблем (рис. 1б).

Если аналогичные действия выполнить через библиотеку Owlready2 в Python, то есть также запустить reasoner и результат сохранить в файл, то будут получены некорректные данные, с которыми без дополнительных действий дальнейшая работа в Python или Protégé не представляется возможной. Происходит это из-за сбоя кодировки вследствие работы reasoner. Вместо внесения изменений, как это делает Protégé, в уже существующую и загруженную русскоязычную онтологию Owlready2 в Python дублирует онтологию, а вместо текста кириллицы появляются нечитаемые символы. Однако reasoner Owlready2 даже после сбоя кодировки правильно классифицирует необходимых индивидов (студентов из приведенного примера), но работать с информацией в таком виде в дальнейшем все равно не получится



а)

б)

Рис. 1. Вид онтологии в Protégé: а) до запуска reasoner, б) после запуска

Fig. 1. The ontology view in Protégé: a) before and б) after running a reasoner

(рис. 2). Сравнивая рисунки 1б и 2, можно сделать вывод, что машина логического вывода в Owlready2 правильно находит новое знание в онтологии, но неверно отображает и записывает его в файл. Причем простое изменение кодировки конечного файла онтологии в формате .owl в целом средствами Python или, например, Notepad++ на UTF-8 и Windows-1251 приводит к полной потере читаемости файла. Любые команды, которые позволяют обращаться к объектам онтологии в Python, начинают выдавать ошибку.

Возможность корректной работы с русскоязычными онтологиями

Для решения поставленной задачи применялся метод сравнительного анализа [17]. Были проанализированы различные версии Owlready2 и ее предшественника Owlready.

Для примера загрузим в Python созданную в Protégé русскоязычную онтологию образования студента в вузе – obrazovanie.owl, графическое представление которой дано на рисунке 1. Запустить reasoner с помощью модуля Owlready2, проверить онтологию obrazovanie.owl

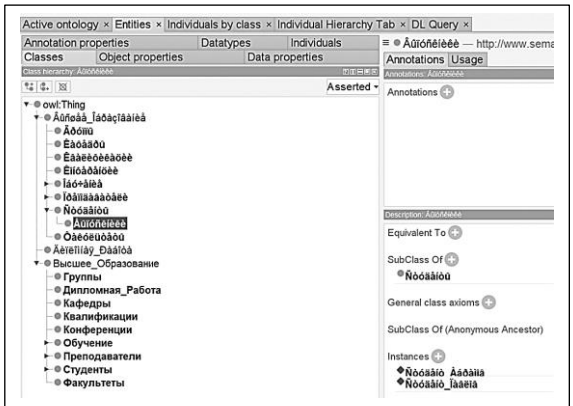


Рис. 2. Некорректное отображение русскоязычной онтологии

Fig. 2. An incorrect display of a Russian-language ontology in Protégé

на непротиворечивость и получить из нее новые знания позволяет следующий код:

```
from owlready2 import *
onto_path.append("/Programs/
OneDrive/Sample/")
onto = get_ontology("obrazovanie.
owl").load()
with onto:
    sync_reasoner(debug=0)
onto.save("tempo.owl")
for j in onto.Выпускники.instances():
    print(j.name)
```

Видно, что сначала указывается путь до папки с файлами онтологий, затем загружается сама онтология, запускается reasoner и для удобства онтология с новыми найденными знаниями сохраняется в новый файл с названием tempo.owl для графической проверки в Protégé.

В силу того, что при проведении самой процедуры рассуждения в явном виде не указывается кодировка выходных данных, reasoner в библиотеке Owlready2 в Python изменяет кодировку файла онтологии. Файл, отвечающий за reasoner, находится в корневой папке программы Python, а именно по пути \Python\Python311\Lib\site-packages\owlready2\reasoning.py для версии Python 3.11.0, который и рассматривается в данной статье. Чтобы после вызова reasoner через командную строку Python выходные данные имели строго определенную и корректно читаемую Protégé кодировку, необходимо указать ее в явном виде. Для этого изменим строки в файле reasoning.py с *output = _decode(output).replace("\r","")* на *output = output.decode('cp1251').replace("\r","")*. Тем самым для работы reasoner устанавливается стандартная 8-битная кодировка для русских версий Microsoft Windows. Эта замена

упрощает дальнейшую работу с онтологиями в Python, особенно при многократных запусках reasoner. После такой замены новые знания, получаемые из онтологии вследствие работы reasoner Python, являются корректными и адекватно отображаются в Protégé (см. http://www.swsys.ru/uploaded/image/2023-2/2023-2-dop/11.jpg).

Заключение

Решена проблема работы с русскоязычными онтологиями в Python. Путем замены одной строки в файле, отвечающем за работу reasoner, становится возможной реализация в Python всего богатого функционала библиотеки Owlready2 для работы с русскоязычными онтологиями. Предложенный в статье способ может быть полезен для ИТ-специалистов, занимающихся разработкой информационных систем на основе онтологий предметных областей, а также при работе с онтологиями в рамках образовательного процесса в вузе.

Список литературы

1. Грибова В.В., Паршкова С.В., Федорищев Л.А. Онтологии для разработки и генерации адаптивных пользовательских интерфейсов редакторов баз знаний // Онтология проектирования. 2022. Т. 12. № 2. С. 200–217. doi: 10.18287/2223-9537-2022-12-2-200-217.
2. Антонов А.А., Быков А.Н., Чернышев С.А. Обзор существующих способов формирования онтологии предметной области при моделировании // Междунар. журнал информационных технологий и энергоэффективности. 2021. Т. 6. № 4. С. 12–17.
3. Сартабанова Ж.Е., Димитров В.Т., Сарсимбаева С.М. Применение базы знаний слабостей CWE в проектировании программного обеспечения // Вестн. КазНУ. Сер.: Математика, механика, информатика. 2020. Т. 108. № 4. С. 72–80. doi: 10.26577/JMMCS.2020.v108.i4.06.
4. Каленов Н.Е. Об одном подходе к формированию предметных онтологий различных областей науки // Научный сервис в сети интернет: тр. конф. 2020. № 22. С. 276–285. doi: 10.20948/abrau-2020-14.
5. Прохорова А.М. Онтологическая модель образовательного портала вуза // Мягкие измерения и вычисления. 2022. Т. 58. № 9. С. 53–61. doi: 10.36871/2618-9976.2022.09.005.
6. Дедков Д.А. Программные продукты для создания онтологий. Система Protege // Тенденции развития Интернет и цифровой экономики: тр. III Всерос. науч.-практич. конф. 2020. С. 205.
7. Weichbroth P. Fluent editor and controlled natural language in ontology development. IJAIT, 2019, vol. 28, no. 04, art. 1940007. doi: 10.1142/s0218213019400074.
8. Иванов П.И., Мышкина И.Ю., Грудцына Л.Ю. Обзор библиотеки Owlready2 для работы с онтологиями на языке Python // Науч.-технич. вестн. Поволжья. 2022. № 12. С. 139–141.
9. Кузин Д.А., Даниленко И.Н. Онтологическая модель проектной деятельности // Современная наука: актуальные проблемы теории и практики. Сер.: Естественные и технические науки. 2020. № 10. С. 77–82. doi: 10.37882/2223-2966.2020.10.15.
10. Ovcinnikova J. Ontology export patterns in OWLGrEd editor. Baltic J. Modern Computing, 2020, vol. 8, no. 3, pp. 444–460. doi: 10.22364/bjmc.2020.8.3.04.
11. Савина А.Г., Уханов Д.В., Савин Д.А. Сферы и перспективы применения языка программирования Python // Научные записки ОрелГИЭТ. 2021. Т. 38. № 2. С. 24–28.
12. Larny J.-B. Ontologies with Python: Programming Owl 2.0 Ontologies with Python and Owlready2. NY, Apress, 2020, 364 p.
13. Уткин Д.В., Шульга Т.Э., Сытник А.А. Разработка веб-сервиса отображения русскоязычных онтологий // Проблемы управления в социально-экономических и технических системах: матер. XVII Междунар. науч.-практич. конф. 2021. С. 77–80.
14. Musen M.A. The Protégé project: A look back and a look forward. AI Matters, 2015, vol. 1, no. 4, pp. 4–12. doi: 10.1145/2757001.2757003.
15. Щукарев И.А., Маркова Е.В. Разработка генератора паролей с использованием GUI MATLAB // Программные продукты и системы. 2022. Т. 3. № 3. С. 197–206. doi: 10.15827/0236-235X.139.413-419.
16. Kolchin M., Klimov N., Andreev A. Ontologies for web of things: A pragmatic review. In: CCIS. Proc. KESW, 2015, vol. 518, pp. 102–116. doi: 10.1007/978-3-319-24543-0_8.
17. Кудж С.А. Методы сравнительного анализа // Славянский форум. 2019. № 3. С. 140–150.

Software & Systems doi: 10.15827/0236-235X.142.223-227 2023, vol. 36, no. 2, pp. 223–227

Features of working with Russian-language ontologies using the Owlready2 library in Python

Igor A. Shchukarev

For citation
Shchukarev, I.A. (2023) ‘Features of working with Russian-language ontologies using the Owlready2 library in Python’, *Software & Systems*, 36(2), pp. 223–227 (in Russ.). doi: 10.15827/0236-235X.142.223-227
Article info
Received: 30.01.2023 After revision: 16.02.2023 Accepted: 03.03.2023

Abstract. The use of domain ontologies to create information systems is currently becoming more and more widespread. Based on ontologies, it is possible to create so-called knowledge bases, which are essential components of most information systems. To work with ontologies, there are various software products, such as Protégé or the Owlready2 module for the Python programming language. As a rule, to search for new knowledge or facts that are in the ontology, a logical inference machine or reasoner is used, which checks it for consistency, i.e. consistency. When working in the Owlready2 library of the Python language with Russian-language ontologies, i.e. Ontologies in which initially all classes, individuals and relationships are written in Cyrillic, reasoner gives incorrect data that is simply unreadable. Due to a failure in the encoding during the operation of reasoner owlready2, firstly, it duplicates the ontology, and, secondly, unreadable characters appear instead of the Cyrillic text. With such data, further actions in Python or Protégé are not possible without additional actions. The article proposes a way to solve this problem by explicitly setting the encoding of the output file after reasoner's work, namely cp1251 encoding, i.e. standard 8-bit encoding for Russian versions of Microsoft Windows. As a result, when working with Russian-language ontologies, it becomes possible to use the full potential of the Owlready2 library of the Python programming language. Therefore, the creation of Russian-language ontologies that can be used as the basis for Russian-language information systems is an urgent task at the present time. The method proposed in the article can be useful for IT specialists involved in the development of information systems based on ontologies of subject areas and when working with ontologies as part of the educational process at a university.
Keywords: ontology, information system, python, protege, owlready2, russian-language ontologies

Reference List

1. Gribova, V.V., Parshkova, S.V., Fedorischev, L.A. (2022) ‘Ontologies for development and generation adaptive user interfaces of knowledge base editors’, *Ontology of Designing*, 12(2), pp. 200–217. doi: 10.18287/2223-9537-2022-12-2-200-217 (in Russ.).

2. Antonov, A.A., Bykov, A.N., Chernyshev, S.A. (2021) ‘Review of existing methods of forming the ontology of the scope in modeling’, *Int. J. of Inform. Technology and Energy Efficiency*, 6(4), pp. 12–17 (in Russ.).

3. Sartabanova, Zh.E., Dimitrov, V.T., Sarsimbayeva, S.M. (2020) ‘Applying the knowledge base of CWE weaknesses in software design’, *J. of Math., Mechanics and Comput. Sci*., 108(4), pp. 72–80. doi: 10.26577/JMMCS.2020.v108.i4.06.

4. Kalenov, N.E. (2020) ‘About one approach to the formation of subject ontologies for science various fields’, *Proc. Conf. Sci. Services & Internet*, (22), pp. 276–285. doi: 10.20948/abrau-2020-14 (in Russ.).

5. Prokhorova, A.M. (2022) ‘Ontological model of the educational portal of the university’, *Soft Measurements and Computing*, 58(9), pp. 53–61. doi: 10.36871/2618-9976.2022.09.005 (in Russ.).

6. Dedkov, D.A. (2020) ‘Software products for creating ontologies. Protege system’, *Proc. Conf. Trends in the Development of the Internet and the Digital Economy*, pp. 205 (in Russ.).

7. Weichbroth, P. (2019) ‘Fluent editor and controlled natural language in ontology development’, *IJAIT*, 28(04), art. 1940007. doi: 10.1142/s0218213019400074.

8. Ivanov, P.I., Myshkina, I.Y., Grudcyna, L.Y. (2022) ‘Overview of the Owlready2 library for working with Python ontologies’, *Sci. and Tech. Volga Region Bull.*, (12), pp. 139–141 (in Russ.).

9. Kuzin, D.A., Danilenko, I.N. (2020) ‘Ontological model of project activity’, *Modern Science: Actual Problems of Theory & Practice. Ser. Natural and Tech. Sci.*, (10), pp. 77–82. doi: 10.37882/2223-2966.2020.10.15 (in Russ.).

10. Ovcinnikova, J. (2020) ‘Ontology export patterns in OWLGrEd editor’, *Baltic J. Modern Computing*, 8(3), pp. 444–460. doi: 10.22364/bjmc.2020.8.3.04.

11. Savina, A.G., Ukhanov, D.V., Savin, D.A. (2021) ‘Areas and prospects of application the python programming language’, *Sci. J. of OrelSIET*, 38(2), pp. 24–28 (in Russ.).

12. Larny, J.-B. (2020) *Ontologies with Python: Programming Owl 2.0 Ontologies with Python and Owlready2*, NY: Apress, 364 p.

13. Utkin, D.V., Shulga, T.E., Sytnik, A.A. (2021) ‘Development of a web service for displaying Russian-language ontologies’, *Proc. Int. Conf. Problems of Management in Socio-economic and Technical Systems*, pp. 77–80 (in Russ.).

14. Musen, M.A. (2015) ‘The Protégé project: A look back and a look forward’, *AI Matters*, 1(4), pp. 4–12. doi: 10.1145/2757001.2757003.

15. Shchukarev, I.A., Markova, E.V. (2022) ‘Developing a password generator using GUI MATLAB’, *Software & Systems*, 35(3), pp. 197–206. doi: 10.15827/0236-235X.139.413-419 (in Russ.).

16. Kolchin, M., Klimov, N., Andreev, A. (2015) ‘Ontologies for web of things: A pragmatic review’, in *CCIS. Proc. KESW*, 518, pp. 102–116. doi: 10.1007/978-3-319-24543-0_8.

17. Kudzh, S.A. (2019) ‘Methods for comparative analysis’, *Slavic Forum*, (3), pp. 140–150 (in Russ.).

Авторы

Щукарев Игорь Александрович [1], к.ф.-м.н., доцент, blacxpress@gmail.com

[1] Ульяновский государственный технический университет, г. Ульяновск, 432027, Россия

Authors

Igor A. Shchukarev [1], Ph.D. (Physics and Mathematics), Associate Professor, blacxpress@gmail.com

[1] Ulyanovsk State Technical University, Ulyanovsk, 432027, Russian Federation