

Научная статья
УДК 004.438Python+004.92
<https://doi.org/10.35266/1999-7604-2026-1-6>



Визуализация онтологического графа с использованием библиотеки pydot Python

Кирилл Валерьевич Святков, Игорь Александрович Щукарев✉

Ульяновский государственный технический университет, Ульяновск, Россия

Аннотация. При попытке визуализации русскоязычной онтологии с помощью библиотеки pydot в Python получается граф знаний, который содержит неправильно отображающиеся кириллические символы. Задание шрифта с поддержкой русского языка в процессе описания графа никак не влияет на итоговое изображение. Решить данную проблему предлагается путем явного задания кодировки в момент экспорта изображения в файл командой `.write_png('output.png', encoding='utf8')`. Вследствие этого появляется возможность построения графа знаний с поддержкой кириллических символов средствами языка программирования Python. Предложенный в статье способ визуализации русскоязычных онтологий средствами Python может быть полезен ИТ-специалистам, которые принимают участие в разработке информационных систем на основе таких онтологий, либо преподавателям, работающим с онтологиями в рамках образовательного процесса.

Ключевые слова: библиотека pydot, язык python, информационная система, граф знаний, русскоязычная онтология, кириллические символы, кодировка utf8

Для цитирования: Святков К. В., Щукарев И. А. Визуализация онтологического графа с использованием библиотеки pydot Python // Вестник кибернетики. 2026. Т. 25, № 1. С. 64–69. <https://doi.org/10.35266/1999-7604-2026-1-6>.

Original article

Ontology graph visualization using pydot library in Python

Kirill V. Svyatov, Igor A. Shchukarev✉

Ulyanovsk State Technical University, Ulyanovsk, Russia

Abstract. Visualization of a Russian-language ontology using the pydot library in Python leads to the creation of a knowledge graph with incorrectly displayed Cyrillic characters. Assignment of a Russian-language supporting font while defining the graph has no influence on the final image. The study proposes a solution to this problem which lies in setting an explicit encoding when exporting the image as a file via the `.write_png('output.png', encoding='utf8')` command. This procedure enables the creation of a knowledge graph with Cyrillic script support. The introduced method of Russian-language ontologies visualization in Python can be applied by IT specialists in developing information systems based on ontologies under consideration or by educators using the ontologies in the teaching process.

Keywords: pydot library, Python, information system, knowledge graph, Russian-language ontology, Cyrillic characters, utf8 encoding

For citation: Svyatov K. V., Shchukarev I. A. Ontology graph visualization using pydot library in Python. *Proceedings in Cybernetics*. 2026;25(1):64–69. <https://doi.org/10.35266/1999-7604-2026-1-6>.

ВВЕДЕНИЕ

В настоящее время графическая визуализация используется для представления различной структурированной информации. Выбор

конкретного инструмента для разного рода диаграмм, абстрактных графиков или сетей зависит от цели. Например, для документов отлично подойдет статичное изображение,

для мониторинга можно использовать интерактивную схему, а для детального анализа – топологию. Визуализировать информацию, которая представима в виде объектов и связей между ними, можно также и с помощью графов. Графовые модели представляют собой наглядную схему данных, связанных друг с другом. Такое графическое представление можно встретить, например, в разработке баз данных и программного обеспечения, в сетевых технологиях, веб-дизайне машинном обучении и во многих других областях [1].

На сегодняшний день существует несколько программ для работы с базами знаний. Среди них можно выделить Protégé и Fluent Editor. Инструмент Protege, который является Java-программой, содержит редактор онтологий и предполагает использование языка OWL, а также Manchester Syntax. Фреймворк Protege поддерживает визуализацию онтологий, работу с машиной логического вывода, создание иерархической структуры классов, работу с различными типами свойств и т.д. [2, 3]. Программа Fluent Editor от компании Cognitum представляет собой приложение, которое дает возможность редактировать базы знаний, поддерживает логический вывод и создание графического представления связей между понятиями и их экземплярами. Его отличительной особенностью является то, что он позволяет разрабатывать базы знаний без глубокого понимания синтаксиса языка OWL, так как для создания онтологий используется естественный язык, а именно английский, к которому применены определенные правила и ограничения [4]. Python является высокоуровневым языком программирования с открытым исходным кодом, который поддерживает подходы объектно-ориентированного и структурного программирования. Python можно использовать для решения разнообразных задач в различных сферах деятельности [5]. В Python для работы с онтологиями используется библиотека `owlready2`. Данный модуль поддерживает создание и редактирование онтологий, импорт и экспорт в формате OWL, управление клас-

сами, экземплярами и свойствами онтологии как объектами языка Python, использование машины логического вывода (reasoner HemiT или Pellet) и многое другое [6].

Так, авторами работы [7] был построен граф онтологии для наглядной демонстрации онтологического представления основной образовательной программы в редакторе Protégé. В статье [8] рассматривается подход с использованием инструмента Ontodia для визуализации онтологических графов, содержащих пустые узлы (Blank nodes) и используемых для представления объектов, не имеющих постоянных идентификаторов, то есть IRI. В работе [9] авторами разработан прототип программного обеспечения, позволяющий визуализировать графовую модель на основе имеющихся данных. Интерактивный граф, построенный на основе данных, полученных из базы данных, выводится в формате html. В статье [10] для визуализации онтологии был выбран алгоритм послойной отрисовки графа, а сама реализация выполнена средствами библиотеки Javascript Dagre. Для рендера графа, то есть получения изображения по модели, был использован React-Flow, который является инструментом на основе языка Javascript и фреймворка React.

При разработке ИС, в основе которых лежат онтологии, желательно использовать один инструмент, позволяющий работать как с кодом, так и с онтологиями без потери функциональности. Использование языка Python при разработке простейших ИС с онтологиями более чем оправдано из-за наличия модуля для работы с базами знаний – `owlready2`, и библиотеки для разработки графического интерфейса пользователя – `tkinter` [11]. Библиотека `Pydot` является интерфейсом Python к языку программирования DOT, который представляет собой текстовое описание структуры графа, а именно вершин, связей между ними, а также атрибутов для их визуального оформления. Он позволяет разработчикам программно генерировать, обрабатывать и отображать описания графов. Таким образом, исследование способа визуализации русско-

язычного графа средствами библиотеки pydot языка Python является актуальной задачей.

МАТЕРИАЛЫ И МЕТОДЫ

Зачастую в процессе разработки может потребоваться визуально просмотреть изменения, внесенные в онтологию. Для этого необходимо экспортировать онтологию и с помощью Protégé или Fluent Editor получить граф знаний. Однако выполнить такую цепочку действий при каждом изменении или при тестировании не очень удобно, а возмож-

ность визуализации онтологии в библиотеке owlready2 отсутствует. С другой стороны, для визуализации графов можно использовать другую библиотеку Python, а именно pydot. Этот инструмент позволяет визуализировать графы и создавать различные диаграммы. При создании графов пакет ищет шрифты в различных каталогах по умолчанию, а также в известных системных каталогах шрифтов. Тем не менее модуль корректно визуализирует только графы с англоязычными символами (рис. 1, 2).

```
1 import pydot
2 graph = pydot.Dot(graph_type='digraph', rankdir='TB', fontname='Arial')
3 graph.set_size('50,50!')
4 node1 = pydot.Node(name="Node 1", shape="circle", color="blue", fontname='Arial')
5 node2 = pydot.Node(name="Node 2", shape="box", color="red", fontname='Arial')
6 graph.add_node(node1)
7 graph.add_node(node2)
8 edge = pydot.Edge(node1, node2, label='distance between 1 and 2', fontname='Arial')
9 graph.add_edge(edge)
10 graph.write_png('output.png')
```

Рис. 1. Фрагмент кода для построения англоязычного графа

Примечание: составлено авторами.

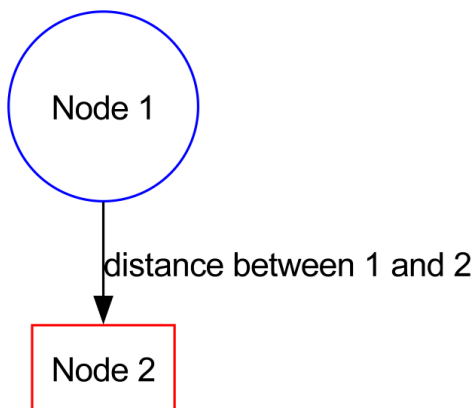


Рис. 2. Внешний вид англоязычного графа

Примечание: составлено авторами.

Если же использовать русский текст при описании графа, то на выходе получаются нечитаемые данные. Задание в явном виде шрифтов, которые поддерживают кириллические символы, также не дает никакого результата (рис. 3, 4).

Для решения поставленной задачи были проанализированы различные способы задания кодировки и этапы, на которых ее мож-

но указывать, для визуализации графа знаний, благодаря чему и было найдено предлагаемое в статье решение. Для корректного отображения русского языка на итоговой картинке необходимо в явном виде указать кодировку UTF-8 в момент экспорта в файл png следующим образом:

```
graph.write_png('output.png', encoding='utf8')
```

```
1 import pydot
2 graph = pydot.Dot(graph_type='digraph', rankdir='TB', fontname='Arial')
3 graph.set_size('50,50!')
4 node1 = pydot.Node(name="Узел 1", shape="circle", color="blue", fontname='Arial')
5 node2 = pydot.Node(name="Узел 2", shape="box", color="red", fontname='Arial')
6 graph.add_node(node1)
7 graph.add_node(node2)
8 edge = pydot.Edge(node1, node2, label='расстояние между 1 и 2', fontname='Arial')
9 graph.add_edge(edge)
10 graph.write_png('output.png')
```

Рис. 3. Фрагмент кода для построения русскоязычного графа

Примечание: составлено авторами.

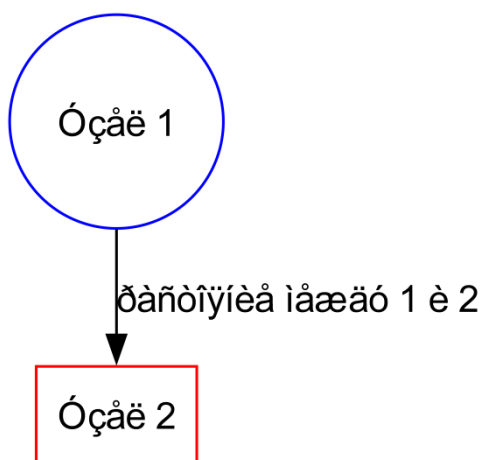


Рис. 4. Внешний вид русскоязычного графа

Примечание: составлено авторами.

РЕЗУЛЬТАТЫ И ИХ ОБСУЖДЕНИЕ

Для проверки работы предложенного метода средствами библиотеки owlready2 Python была разработана русскоязычная фармацев-

тическая онтология (рис. 5) и с помощью библиотеки pydot написан код для получения графа знаний (рис. 6, 7).

```
onto = get_ontology("http://test.org/onto.owl")

with onto:
    # class()
    class Аптека(Thing): pass 1 usage
    class Привет(Thing): pass
    class Лекарственные_средства(Аптека): pass 23 usages
    class Условия_отпуска_из_аптек(Лекарственные_средства): pass 2 usages
    class По_рецепту(Условия_отпуска_из_аптек): pass 1 usage
    class Без_рецепта(Условия_отпуска_из_аптек): pass 1 usage
    class Показания_к_применению(Лекарственные_средства): pass 1 usage
    class лечится(ObjectProperty):
        domain = [Показания_к_применению]
        range = [Лекарственные_средства]
```

Рис. 5. Фрагмент кода фармацевтической онтологии с помощью модуля Owlready2

Примечание: составлено авторами.

```
285 for ind in onto.individuals():
286     graph.add_node(pydot.Node(str(ind.name), shape="ellipse", style="filled", fillcolor="lightgreen"))
287     for cls in ind.is_a:
288         if isinstance(cls, ThingClass):
289             graph.add_edge(pydot.Edge(str(ind.name), str(cls.name), label="instance_of", style="dotted"))
290
291 for ind in onto.individuals():
292     for prop in onto.data_properties():
293         values = prop[ind]
294         for val in values:
295             literal_node_name = f"{ind.name}_{prop.name}_{val}"
296             graph.add_node(pydot.Node(literal_node_name, shape="note", style="filled", fillcolor="lightyellow"))
297             graph.add_edge(pydot.Edge(str(ind.name), literal_node_name, style="dashed"))
298
299 graph.write_png('output.png', encoding='utf8')
```

Рис. 6. Фрагмент кода для построения русскоязычного графа на основе русскоязычной онтологии с использованием предложенного способа

Примечание: составлено авторами.

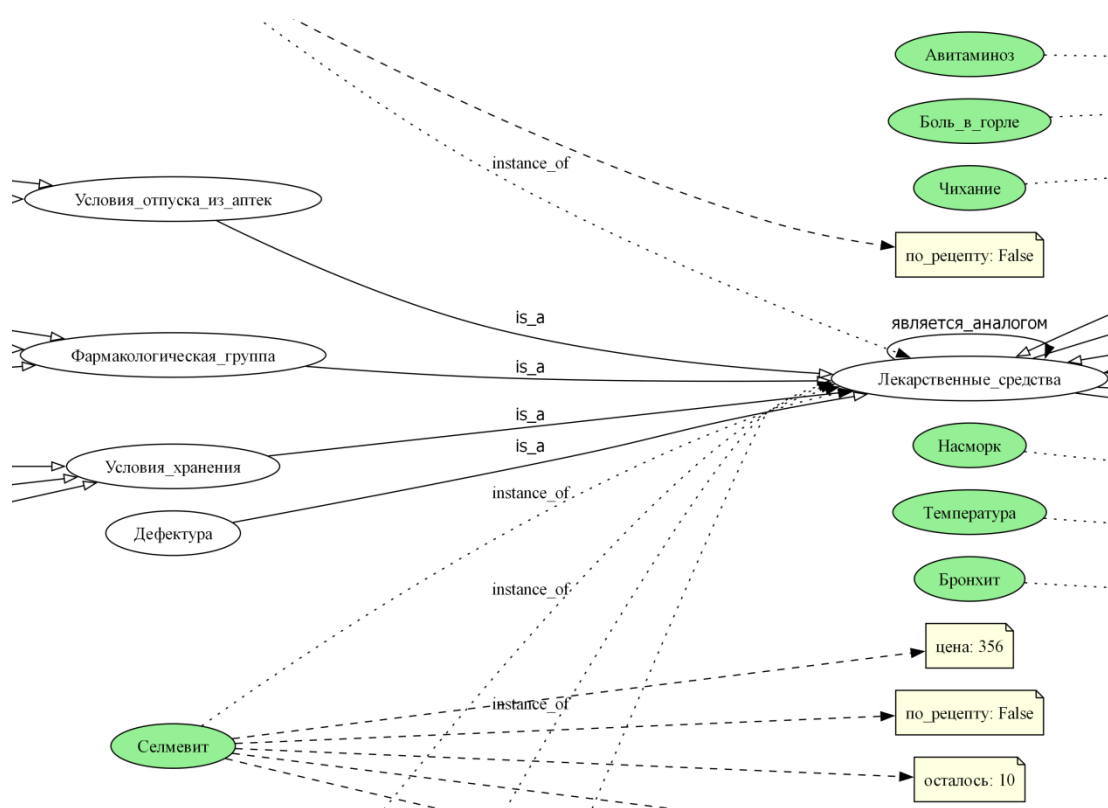


Рис. 7. Корректное отображение кириллических символов на фрагменте графа на основе русскоязычной онтологии в pydot

Примечание: составлено авторами.

Как видно из рис. 7, после выполнения данной команды на итоговом изображении русский язык отображается корректно и появляется возможность получать графы знаний русскоязычных онтологий.

ЗАКЛЮЧЕНИЕ

Итак, в статье решена проблема корректного отображения кириллических символов на графе знаний русскоязычной онтологии. Пу-

тем указания кодировки UTF-8 в момент экспорта в файл в Python появляется возможность визуализации русскоязычных онтологий с помощью библиотеки pydot. Предложенный в статье способ может быть полезен ИТ-специалистам, которые участвуют в разработке информационных систем на основе таких онтологий, либо преподавателям, работающим с онтологиями в рамках образовательного процесса.

Список источников

1. Русаков А. М., Юшкова Н. А., Селиванова Е. А. Современные средства визуализации графов и графовых моделей для Python // *НаукоСфера*. 2022. № 11 (2). С. 258–267.
2. Напольских Д. Л. Представление экономической концепции кластера с помощью специализированной онтологии // *Программные продукты и системы*. 2024. Т. 37, № 2. С. 155–163.
3. Ovcinnikova J. Ontology export patterns in OWL-GrEd editor // *Baltic Journal of Modern Computing*. 2020. Vol. 8, no. 3. P. 444–460.
4. Сосинская С. С., Сивушков И. Д., Дубинин Д. А. и др. База знаний о состоянии здоровья обучающихся и клиентское приложение для взаимодействия с пользователем // *Онтология проектирования*. 2022. Т. 12, № 3 (45). С. 325–335.
5. Дучаев М. М., Гордеев-Бургвиц Д. М., Гордеев-Бургвиц М. А. Python: преимущества и роль в наше время // *Научно-технический вестник Поволжья*. 2025. № 4. С. 51–54.
6. Jean-Baptiste L. *Ontologies with Python: Programming OWL 2.0 ontologies with Python and Owlready2*. New York : Apress, 2020. 344 p.
7. Назаров Е. В. Онтологическое представление основной образовательной программы // *Вестник кибернетики*. 2021. № 3 (43). С. 51–59.
8. Раздьяконов Д. С., Морозов А. В., Павлов Д. С. и др. Подход к обработке пустых узлов при порционной визуализации данных на примере инструмента *Ontodia* // *Программирование*. 2020. № 6. С. 16–29.
9. Карандеев А. А., Яшин Н. А., Кудряшев Р. А. Визуализация графовой модели медицинских препаратов и их классов с учетом несовместимости // *Научная визуализация*. 2023. Т. 15, № 5. С. 64–72.
10. Лисин В. А., Серый А. С., Сидорова Е. А. Модель представления онтологии предметных областей на основе графовых баз данных // *Вестник НГУ. Серия: Информационные технологии*. 2022. Т. 20, № 4. С. 24–38.
11. Крамаренко Т. А., Безрук И. С. Сравнительный анализ GUI библиотек популярных языков программирования для разработки бизнес-приложений // *Вестник Адыгейского государственного университета. Серия: Естественно-математические и технические науки*. 2025. № 2 (361). С. 57–63.

Информация об авторах

К. В. Святов – кандидат технических наук, доцент;
<https://orcid.org/0000-0001-8978-8567>,
k.svyatov@gmail.com

И. А. Щукарев – кандидат физико-математических наук;
<https://orcid.org/0000-0002-8860-8103>,
blacxpress@gmail.com✉

References

1. Rusakov A. M., Yushkova N. A., Selivanova E. A. Modern graph visualization tools and graph models for Python. *Naukosfera*. 2022;(11):258–267. (In Russ.).
2. Napolskikh D. L. Developing a specialized ontology to represent the economic concept of a cluster. *Software & Systems*. 2024;37(2):155–163. (In Russ.).
3. Ovcinnikova J. Ontology export patterns in OWL-GrEd editor. *Baltic Journal of Modern Computing*. 2020;8(3):444–460.
4. Sosinskaya S. S., Sivushkov I. D., Dubinin D. A. et al. Knowledge base on the health status of students and a client application for user interaction. *Ontology of designing*. 2022;12(3):325–335. (In Russ.).
5. Duchaev M. M., Gordeev-Burgwitz D. M., Gordeev-Burgwitz M. A. Python: Advantages and role in our time. *Nauchno-tekhnicheskiiy vestnik Povolzhya*. 2025;(4):51–54. (In Russ.).
6. Jean-Baptiste L. *Ontologies with Python: Programming OWL 2.0 ontologies with Python and Owlready2*. New York: Apress; 2020. 344 p.
7. Nazarov E. V. Ontological representation of basic educational program. *Proceedings in Cybernetics*. 2021;(3):51–59. (In Russ.).
8. Razd'yakonov D. S., Morozov A. V., Pavlov D. S. et al. Approach to blank node processing in incremental data visualization by the example of *ontodia*. *Programming and Computer Software*. 2020;(6):16–29. (In Russ.).
9. Karandeev A. A., Iashin N. A., Kudryashev R. A. Visualization of a graph model of medical drugs and their classes with subject to incompatibility. *Scientific Visualization*. 2023;15(5):64–72. (In Russ.).
10. Lisin V. A., Sery A. S., Sidorova E. A. Domain ontology representation model based on graph databases. *Vestnik NSU. Series: Information Technologies*. 2022;20(4):24–38. (In Russ.).
11. Kramarenko T. A., Bezruk I. S. Comparative analysis of GUI libraries for popular programming languages in business application development. *Bulletin of Adyge State University. Series: Natural-Mathematical and Technical Sciences*. 2025;(2):57–63. (In Russ.).

About the authors

K. V. Svyatov – Candidate of Sciences (Engineering), Docent;
<https://orcid.org/0000-0001-8978-8567>,
k.svyatov@gmail.com

I. A. Shchukarev – Candidate of Sciences (Physics and Mathematics);
<https://orcid.org/0000-0002-8860-8103>,
blacxpress@gmail.com✉