

# КОНВЕРТАЦІЯ ФОРМАТІВ ЗБЕРІГАННЯ ГРАФОВИХ БАЗ ДАНИХ І ЇЇ РОЛЬ В ОНТОЛОГІЧНОМУ ІНЖИНІРИНГУ НА ПРИКЛАДІ СИСТЕМИ СУПРОВОДЖЕННЯ РЕАБІЛІТАЦІЇ

*Каверинський В. В.*

*Інститут проблем матеріалознавства ім. І.М. Францевича  
Національної академії наук України, м. Київ, Україна*

 <https://orcid.org/0000-0002-6940-579X>

**Для кореспонденції:** Каверинський Владислав Володимирович, кандидат технічних наук, старший науковий співробітник відділу № 36 ІПМ НАН України, 67, вул. Метрологічна 56, м. Київ, 03143, Україна; e-mail: [insamhlaythe@gmail.com](mailto:insamhlaythe@gmail.com); контактний тел.: +38 (050) 212 17 24.

Графові бази даних є зручним інструментом для представлення онтологій та інших випадків даних з природною графовою структурою. Графові бази даних онтологічного типу були задіяні при розробці інтелектуальної інформаційно-аналітичної системи супроводження процесів реабілітації (TISP). Зокрема, в подібній формі було представлено такі розділи як «Біла книга», «Гастроентерологія», «Ішемічна хвороба серця», «Курортологія», «Медичне право», «Фізична реабілітація» та ін..

Одним з найбільш поширених і універсальних форматів зберігання онтологій є OWL, зокрема у синтаксисі RDF/XML. Проблемою є відсутність доступного і водночас потужного і зручного графічного редактору схем онтології, що працює з подібними форматами. Створення ж складної схеми онтології, наприклад, у редакторі, вбудованому у "Protégé" є складним і незручним процесом. Більш того, можливості побудови онто-графу у "Protégé" є суттєво обмеженими.

Зручним інструментом для візуального розроблення, перегляду і редагування онто-графів є, розроблений в Інституті кібернетики НАН України КІТ «Полієдр», що поєднує дружній візуальний інтерфейс і широкі функціональні можливості з простотою роботи і освоєння. Попре це, особливістю КІТ «Полієдр» є специфічний формат зберігання графів. Натомість OWL, що є прийнятним для багатьох графових СУБД як основний чи проміжний формат зберігання і завантаження даних наразі не підтримується у КІТ «Полієдр». Таким чином, актуальною задачею стало створення конвертера, що переводить XML-файли з формату КІТ «Полієдр» в OWL у синтаксисі RDF/XML, також у зворотному напрямку.

Метою досліджень було побудувати програмну систему для переведення файлів з формату зберігання онтологічного графу КІТ «Полієдр» ("Graph Editor") до загальноприйнятого RDF/XML синтаксису зберігання OWL-онтологій.

В результаті роботи було проведено аналіз особливостей зберігання даних онтологічних графів у форматі "Graph Editor", структури відповідних XML-файлів і можливості максимально повного представлення як значущої так і допоміжної (технічної) інформації доступними засобами RDF/XML синтаксису. З іншого боку, розроблено способи представлення стандартних OWL сутностей і конструкторів засобами "Graph Editor". Розроблена система програмного мета-представлення даних зчитаних з відповідного XML чи OWL-файлу та реалізовано способи її серіалізації в формат OWL чи .XML-"Graph Editor". Відповідну програмну систему було втілено у вигляді он-лайн сервісу.

В роботі було показано, що вершини онтологічного графу, побудованого у КІТ «Полієдр» при конвертації можуть бути приведені до стандартних OWL-сутностей – класів, властивостей або екземплярів класів (individuals). Якщо тип явно не вказано, за замовчуванням вершини стають класами. Також можливо представлення не типізованих вершин іншими OWL-типами, враховуючі специфічні типи зв'язків (за їх наявності). Не типізовані зв'язки можуть опційно бути інтерпретованими як «Subclass» або «Default». Особливу увагу приділено представленню даних, що містяться у вершинах і зв'язках графу. За замовчуванням вони стають label, але, якщо вказано специфічний тип, вони можуть стати будь-яким з варіантів «анотацій» відповідної OWL-сутності. З даних зв'язків утворюються специфічні технічні властивості (ObjectProperty), що поєднують відповідні вершини, до них у вигляді label чи інших annotations приєднуються дані. Передбачено створення різноманітних, як стандартних (підклас, еквівалентність та ін..), так і довільних зв'язків-посилань між вершинами онтологічного графу, якщо такі означені у вихідному файлі. Вкладні дані представляються посиланнями через атрибути тегів, що вказують власний ідентифікатор та ідентифікатор батьківського data-тегу. Зворотна конвертація створює вершини онто-графу відповідних типів з класів, властивостей і екземплярів класів онтології, за виключенням технічних (за умов конвертації попередньо перетвореного файлу), які використовуються у такому разі для відновлення структури прив'язаних даних. Теги, що містять технічну інформацію, не перетворюються у прив'язані дані, але використовуються для ініціалізації характеристик вершин (форма, колір, позиція тощо).

Розроблену програму для конвертації було використано при переведенні до OWL-формату онтології за «Білою книгою», з яким працює RDFlib у складі діалогової підсистеми TISP. Також було успішно здійснено конвертацію до OWL-формату ряду інших медичних онтологій у рамках проекту TISP. Аналіз результатів показав відсутність втрати інформації при прямій та зворотній конвертації. Отримані успішно файли сприймаються парсерами RDF/XML формату, зокрема "Protégé", RDFlib, Neo4j (завантаження OWL-онтології за допомогою плагіна Neosemantics), до них можуть бути використані формальні запити на мові SPARQL. У разі переведення отриманого OWL-файлу до Neo4j до онтології може бути застосована також мова Cypher.

В результаті роботи було створено зручний програмний інструмент для переведення онтологічних графів, створених за допомогою зручної, наочної і потужної системи онтологічного інжинірингу КІТ «Полієдр» до OWL-онтологій у синтаксисі RDF/XML. Це суттєво розширює практичну придатність створених у такий спосіб онтологій, дає змогу використовувати їх у довідкових системах, системах збору і обробки інформації, з використанням стандартних засобів роботи з RDF/XML-форматом (безпосередньо чи у якості вихідного). Можливість зворотної конвертації дає змогу застосовувати КІТ «Полієдр» для перегляду і редагування сторонніх OWL-онтологій, набутих у мережі Інтернет або і інших спосіб

**Ключові слова:** онтологія, графова база даних, конвертація форматів, OWL, RDF/XML, онтологічний інжиніринг.

**Вступ.** Для машинного представлення зв'язків між поняттями предметної області зручним засобом є графові бази даних. Такі представлення називають онтологіями [1]. Область застосування графових баз даних не обмежуються онтологіями, вони також прийнятні і для інших випадків даних з природною графовою структурою, як то: опис взаємодії між організаціями та/чи окремими людьми, транспортних, телекомунікаційних зв'язків, мовних конструкцій, тощо. Для доменів з природною графовою структурою даних графові бази даних можуть значно перевищувати реляційні по продуктивності, а також мати переваги в наочності уявлення і простоті внесення змін до схеми бази даних [2].

Слід зазначити, що саме графові бази даних онтологічного типу були широко задіяні при розробці інтелектуальної інформаційно-аналітичної системи супроводження процесів реабілітації (TISP). Зокрема, в подібній формі було представлено такі розділи як «Біла книга», «Гастроентерологія», «Ішемічна хвороба серця», «Курортологія», «Медичне право», «Фізична реабілітація» та ін.

Існує багато варіантів представлення і зберігання графових баз даних. Часто графову модель даних розглядають як узагальнення RDF-моделі даних [3]. Отже, не дивно, що одним з найбільш поширених і універсальних форматів зберігання онтологій є OWL, зокрема у синтаксисі RDF/XML. Також існують такі формати, як, наприклад, “Manchester OWL Syntax”, “OWL Functional Syntax”, “Turtle” та ін. Можливість конвертації між цими форматами реалізована і доступна у таких застосунках як “Protégé”, або через API он-лайн сервісів. Таким чином, не мало принципового значення, який саме варіант OWL-синтаксису буде обрано у якості робочого варіанту у нашій роботі – якщо потрібно, наявні інструменти дозволяють привести формат файлу згідно конкретних вимог.

Графові СУБД і інструменти візуалізації онтографів існують і розвиваються. Графічне наочне представлення бази даних пропонує, наприклад, Neo4j. У “Protégé” вбудовано графічний модуль, що дозволяє переглядати, а також редагувати та створювати онтологічні графи. Попре це, на жаль, функціональність широкодоступних інструментів обмежена. Таким чином, постає проблема відсутності потужного і зручного графічного редактору схем онтологій, що працює з OWL. Так, наприклад, створення складної схеми онтології у редакторі, вбудованому, наприклад, у тому ж “Protégé” є складним і незручним процесом. Більш того, можливості побудови онто-графу у “Protégé” є суттєво обмеженими, наприклад, відсутня можливість вказати декілька батьківських класів. Теж стосується і численних он-лайн редакторів онтологій. Neo4j не працює напряму з OWL чи RDF/XML моделлю, а графічний інтерфейс пропонує лише перегляд графів або їх частин, створення і редагування ж виконується через мову запитів Cypher [4].

Зручним інструментом для візуального розроблення, перегляду і редагування онтологічних графів є, розроблений в Інституті кібернетики НАН України КІТ «Поліедр» (також відомий як “Graph Editor”). Поєднання дружнього візуального інтерфейсу, широких функціональних можливостей і простоти роботи і освоєння робить КІТ «Поліедр» чи не одним з передових інструментів у сучасному онтологічному інжинірингу. Так, наприклад, одним з варіантів для створення онто-графу у КІТ «Поліедр» може бути файл MS Excel, який містить таблицю RDF-трійок, що може бути створена чи то на розуміння розробника онтологій, чи збиратися автоматично, наприклад, за допомогою аналізу інформації. Зручним і інтуїтивно зрозумілим є і суто графічний інтерфейс, що дозволяє з легкістю створювати, видаляти та редагувати вершини графу і зв'язки між ними. Вершини і зв'язки графу можуть раціонально позиціонуватися автоматично чи вручну. Є можливість типізації вершин і зв'язків, яку можна візуально відобразити шляхом різного кольору, форми та написів. Реалізовано можливість приєднання і до вершин, і до зв'язків даних різних типів – текст, таблиця, зображення. Приєднані дані можуть бути вкладеними і довільно типізуватися (як то «назва», «визначення», «характеристика», «властивість», тощо).

Але особливістю КІТ «Поліедр» є досить специфічний формат зберігання графів у XML-файлах. Цей формат (надалі формат “Graph Editor”) сприймається виключно КІТ «Поліедр». Натомість OWL, що є прийнятним для багатьох графових СУБД як основний чи проміжний формат зберігання і завантаження даних, наразі не підтримується у КІТ «Поліедр». Так, наприклад, OWL-онтології можна переглядати і редагувати у застосунку “Protégé”, вони добре переводяться у список RDF-трійок за допомогою RDFlib, у якій також повноцінно реалізована робота зі SPARQL-запитами. Використання додаткового модулю «Neosemantics» [5] надає змогу завантажувати інформацію з OWL-онтології (та інших RDF/XML представлень) до потужної графової СУБД Neo4j.

Таким чином, актуальною задачею стало створення конвертеру, що переводить XML-файли з формату “Graph Editor” в OWL у синтаксисі RDF/XML, також у зворотному напрямку.

**Мета** роботи полягала у розробці програмної системи для ефективного переведення файлів з формату зберігання онтологічного графу КІТ «Поліедр» (“Graph Editor”) до загальноприйнятого RDF/XML синтаксису зберігання OWL-онтологій.

**Матеріали і методи.** проведено аналіз особливостей зберігання даних онтологічних графів у форматі “Graph Editor”, структури відповідних XML-файлів і можливості максимально повного представлення як значущої так і допоміжної (технічної) інформації доступними засобами RDF/XML синтаксису. З іншого боку, розроблено способи

представлення стандартних OWL сутностей і конструкції засобами “Graph Editor”. Розроблена система програмного мета-представлення даних зчитаних з відповідного XML чи OWL-файлу та реалізовано способи її серіалізації в формат OWL чи . XML-“Graph Editor”. Відповідну програмну систему було втілено у вигляді он-лайн сервісу з використанням мови програмування Python і фреймворку Flask.

**Результати.** Було розроблено алгоритм переведення, що створює структуру даних у синтаксисі RDF/XML, аналогічну за змістом і представленими сутностями до вихідного XML-файлу формату “Graph Editor”. Далі опишемо основні ідеї запропонованого алгоритму.

Вершини онтологічного графу, побудованого у KIT «Поліедр» при конвертації приводяться відповідно до стандартних OWL-сутностей – класів, властивостей або екземплярів класів (individuals). Якщо тип явно не вказано, за замовчуванням вершини стають класами. Також можливо представлення не типізованих вершин іншими OWL-типами, враховуючі специфічні типи зв’язків (за їх наявності). Так, наприклад, якщо деяка вершина приєднана до іншої за допомогою зв’язка, типізованого як «іменована сутність» (“named individual”), то для дочірньої вершини буде створено відповідний екземпляр класу. Теж стосується і зв’язків типів Domain та Range, що можуть поєднувати тільки вершини типу «властивість» (“object property”) і типу «клас».

Особливу увагу приділено представленню даних, що містяться у вершинах і зв’язках графу. За замовчуванням вони стають label, але, якщо вказано специфічний тип, вони можуть стати будь-яким з варіантів «анотацій» відповідної OWL-сутності, наприклад, “comment”, або отримати специфічний навіть нестандартний щодо OWL тип, як то, наприклад, “option”, “linked\_image”, “start\_time”, тощо.

З даних, приєднаних до зв’язків, утворюються специфічні технічні властивості (ObjectProperty), що поєднують відповідні вершини, до них у вигляді label чи інших типів так званих OWL-анотацій приєднуються дані. Вкладні дані представляються посиланнями через атрибути тегів, що вказують власний ідентифікатор та ідентифікатор батьківського data-тегу.

Передбачено створення різноманітних, як стандартних (підклас, еквівалентність, відмінність та ін.), так і довільних зв’язків-посилань між вершинами онтологічного графу, якщо такі означені у вихідному файлі. Таким чином, у вихідному RDF/XML-файлі може бути втілено довільне «горизонтальне» неієрархічне зв’язування сутностей онто-графу за допомогою довільних типів предикатів (без створення технічних властивостей, як це було втілено, наприклад у інших наших онтологіях, описаних, наприклад, у роботі [6]). Хоча, слід зазначити, що подібні зв’язки між сутностями онтології не відображаються візуально у редакторі

“Protégé” (там вони будуть поміщені до вікна відображення “Annotations”). Тим не менш, подібні типи зв’язків добре інтерпретуються до RDF-трибок за допомогою RDFlib, що робить конвертовані таким чином RDF/XML-файли придатними для роботи в інформаційних довідкових системах з використанням SPARQL. Більш того, SPARQL-запити інформації за безпосередніми довільними предикатами можуть бути простішими, а, відтак, більш швидко виконуваними, що часто є критичним моментом для RDFlib, для якої характерно доволі повільне виконання запитів зі складними умовами відбору.

Зворотна конвертація створює вершини онто-графу відповідних типів з класів, властивостей і екземплярів класів онтології, за виключенням технічних (за умов конвертації попередньо перетвореного файлу), які використовуються у такому разі для відновлення структури прив’язаних даних. Теги, що містять технічну інформацію, не перетворюються у прив’язані дані, але використовуються для ініціалізації характеристик вершин (форма, колір, позиція тощо).

Безпосередньо процеси конвертації суттєво відрізняються залежно від напрямку. Процес конвертації файлу з формату Graph Editor у RDF/XML можна коротко охарактеризувати наступними основними етапами:

- створення з записів вихідного файлу списків структур словникового типу, кожна з яких характеризує відповідну вершину, зв’язок, або групу зв’язків;
- створення програмного об’єктного представлення OWL-сутностей (класів, властивостей, іменованих сутностей) з вершин вихідного графу;
- формування опису типізованих структур даних, що містяться у вершинах графу, прийнятної для висловлення засобами RDF/XML;
- формування у створених програмних об’єктних представленнях OWL-сутностей зв’язків на основі інформації про зв’язки у вихідному файлі;
- формування технічних властивостей (ObjectProperty) для зберігання розширених даних зв’язків (при їх наявності);
- серіалізація створених програмних об’єктних представлень OWL-сутностей у форматі RDF/XML.

Процес зворотної конвертації протікає наступним чином:

- створюється програмне об’єктне представлення вершин графу (Node) на основі наявних OWL-сутностей;
- створення програмного об’єктного представлення зв’язків (Edge) на основі інформації, що міститься у відповідних OWL-сутностях;
- формуються структури вкладених структур даних у вершинах графу за їх наявності на основі відповідної інформації в OWL-сутностях;
- серіалізація програмного об’єктного представлення вершин, зв’язків і груп графу у форматі XML, прийнятному щодо роботи у Graph Editor.

Програмну систему було втілено у вигляді он-лайн сервісу. Принципова схема роботи програмної системи наочно представлена на рисунку 1 у вигляді UML-діаграми. Коротко роботу сервісної оболонки системи можна охарактеризувати наступним чином. Користувач обирає файл для завантаження з метою конвертації. Після цього стають активними кнопки для двох варіантів режиму конвертації. Натискання такої кнопки відправляє на сервер файл і інформацію про тип конвертації, яку потрібно виконати. Також на сервер відправляється інформація, як інтерпретувати неіменовані зв'язки у файлі формату Graph Editor (актуально тільки для нього). Остання інформація береться зі стану відміченості відповідного поля. Так, вони можуть бути інтерпретовані як «є підкласом», або як зв'язок «за замовчуванням».

Отриманий файл записується на сервері у певну директорію. Програма-клієнт отримує унікальний ідентифікатор процесу (послідовність символів), за яким він може направляти запити про стан конвертації на даний момент. Цей ідентифікатор, а також технічна інформація (ім'я файлу, час початку процесу, поточний стан конвертації – завершено або ні, список помилок і попереджень) записується у допоміжну базу даних словникового типу під управлінням Redis.

У паралельному потоці стартує відповідний процес конвертації. Запуск паралельного процесу дозволяє відправити клієнту швидку відповідь, не очікуючи завершення конвертації. Це дозволяє уникнути технічної затримки відповіді серверу. Програма-клієнт відправляє періодичні аїах-запити про стан процесу конвертації (чи його завершено?).

В ході даного процесу інформація про помилки і попередження, у разі їх виникнення записується у базу даних словникового типу під управлінням Redis. Після завершення конвертації, успішного чи ні, створюється відповідний запис у Redis-словнику. Файл з результатами, якщо його успішно отримано, записується у відповідну директорію на сервері. Вихідний тимчасовий файл видаляється з серверу.

Якщо після чергового запиту від програми-клієнта виявляється, що конвертацію завершено, клієнту відправляється повідомлення, що містить посилання на файл результату на сервері, якщо такий було успішно створено, і список помилок і попереджень, за наявності таких. Отримавши таку відповідь від сервера, програма-клієнт відображає у інтерфейсі користувача посилання для завантаження файлу, а також список помилок і попереджень, за наявності таких. Також відправляється строк дійсності посилання (фактично термін зберігання файлу результатів на сервері).

На сервері йде безперервний періодичний паралельний процес «збирання сміття». Цей процес видаляє старі файли результатів, відповідні записи у Redis, а також об'єкти завершених процесів конвертації з пам'яті комп'ютера.

Розроблену програму для конвертації було використано при переведенні до OWL-формату онтології за «Білою книгою», з яким працює RDFlib у складі діалогової підсистеми TISP. Також було успішно здійснено конвертацію до OWL-формату ряду інших медичних онтологій у рамках проекту TISP. Аналіз результатів показав відсутність втрати інформації при прямій та зворотній конвертації. Отримані успішно файли сприймаються парсерами RDF/XML формату, зокрема “Protégé”, RDFlib, Neo4j (завантаження OWL-онтології за допомогою плагіна Neosemantics), до них можуть бути використані формальні запити на мові SPARQL, що дозволяють витягнути різноманітну інформацію з онтології. У разі переведення отриманого OWL-файлу до Neo4j до онтології може бути застосована також мова Cypher.

### Обговорення.

Слід зазначити, що задача, що вирішується за допомогою даного програмного засобу є специфічною до формату “Graph Editor” і його роль повинна розглядатися у сукупності з такою для КІТ «Поліедр». В свою чергу, КІТ «Поліедр» є фактично унікальним інструментом, що пропонує широкі можливості для зручної побудови схеми онтологічного графа. Інші засоби для візуалізації і редагування онтологічних графів існують. Наприклад [7] або [8] надають он-лайн платформу для перегляду та редагування онтологій. Стосовно [7] – WebVOWL Editor – слід зазначити, що наразі публічний сервіс WebVOWL було дезактивовано 1 жовтня 2021 року, хоча вихідний код проекту ще знаходиться у відкритому доступі на GitHub. WebVOWL Editor – то скоріше інструмент для перегляду і редагування вже існуючих онтологій, аніж ручного створення нових з нуля. Окрім того він не надає широких можливостей контекстного наповнення різним форматом даних вершини і зв'язки графу, що передбачено у “Graph Editor”. Іншим прикладом візуалізатора онтологій є OWLGrEd ontology visualizer, описаний у роботі [8]. Це також, як і WebVOWL Editor, застосунок для перегляду і незначного редагування онтологій. Прикладом редактору, створеного безпосередньо для візуальної розробки онтологічних графів може слугувати “Eddy” [9]. Як заявляють автори, “Eddy” спеціально розроблений для створення онтологій шляхом повністю візуальної онтологічної мови, в “Eddy” онтології легко малюються як діаграми, а не записуються як набори формул, як це зазвичай відбувається у популярних середовищах онтологічного дизайну та інженерії. “Eddy” надає інтуїтивно зрозумілі функції для визначення діаграм, гарантує їх синтаксичну правильність та дозволяє експортувати їх у стандартному синтаксисі OWL 2. Проте, не дивлячись на поєднання простоти і можливостей графічного редактора онтологій “Eddy”, ми можемо відзначити ряд переваг притаманних саме “Graph Editor”, зокрема, наприклад можливість створення онтологічного графу з набору RDF-

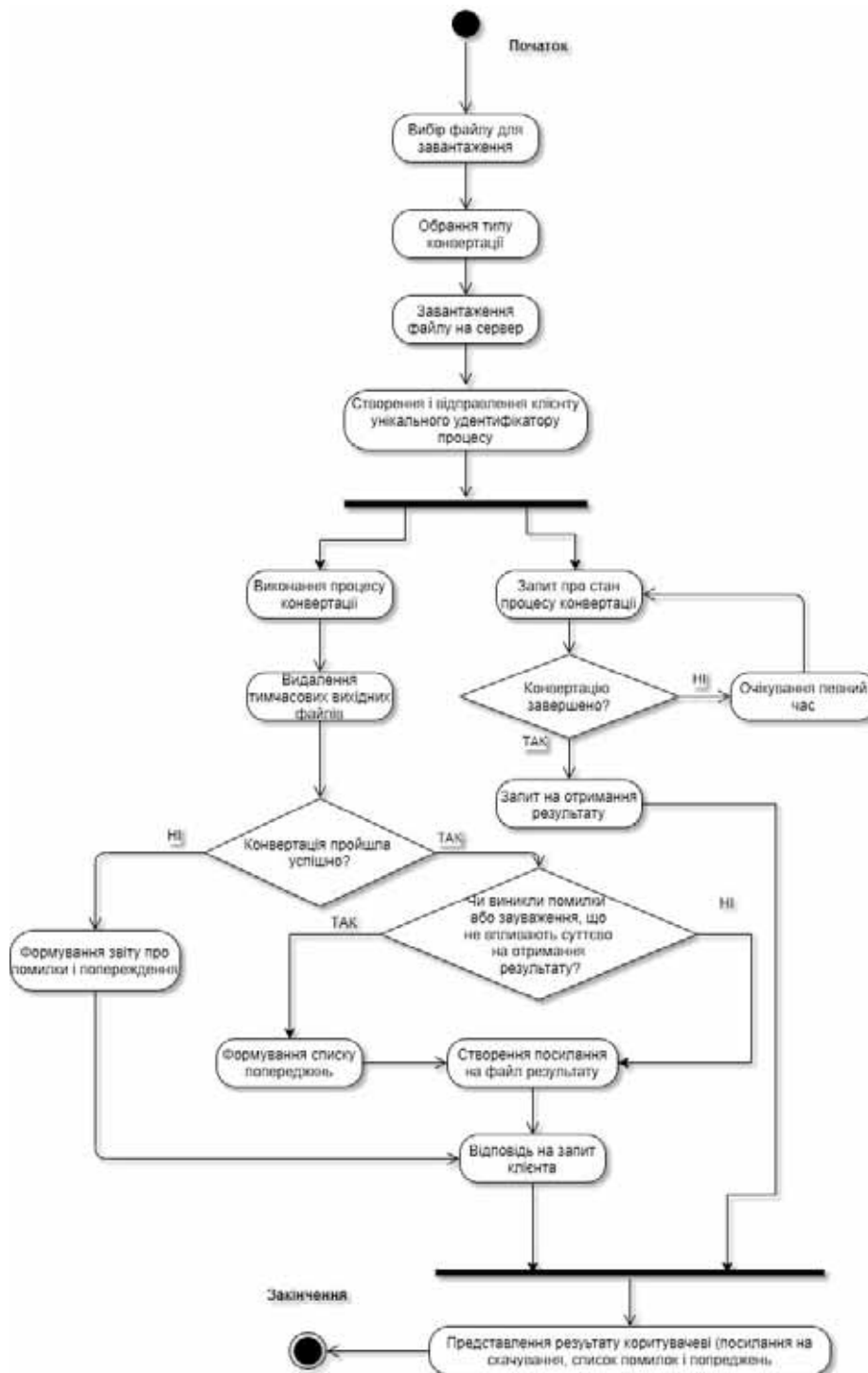


Рис. 1. UML-діаграма дій, що описує загальну процедуру конвертації у системі

трибок, представлених у файлі MS Excel. Тобто, користувач, чи програма може просто складати інформацію, наприклад з аналізованого джерела, у вигляді «сутність-1» - «предикат» - «сутність-2», навіть не замислюючись над якоюсь структурою, а граф буде автоматично побудованим. Окрім того, КІТ «Поліедр» являє в цілому набагато ширші можливості, ніж просто редактор графів. Когнітивні засоби «Поліедр» забезпечують наочне інтуїтивно зрозуміле

відображення трансдисциплінарних онтологічних образів, семантично пов'язаних контекстів їх описів та результати обміну інформацією між розподіленими трансдисциплінарними інформаційними ресурсами (системами, прикладними програмами, базами даних тощо). Можливості даного засобу включають автоматизовану індуктивну побудову онтологій на основі невпорядкованих тверджень про об'єкти та їх властивості з виявленням нових знань – перетинів

описів об'єктів. Це дозволяє вирішувати задачу кластеризації – знаходження груп схожих об'єктів та подій та виявляти нові класи об'єктів у процесі побудови онтології. Перспективною є можливість автоматичного виявлення контекстного зв'язку між термінами з різних документів, що дозволяє виявляти приховані зв'язки між об'єктами та подіями, які описані у різних документах. Цього не надають інші редактори онтологій.

Важливим слабким місцем КІТ «Поліедр» є відсутність підтримки OWL, а натомість свій власний XML формат зберігання даних графів. Розроблений і представлений в наявній роботі сервіс вирішує зазначену проблему, а саме робить можливим зіставлення КІТ «Поліедр» з OWL-онтологіями.

Слід сказати, що засоби конвертації XML в OWL розроблялися і існують [10]. Але вони мали істотно іншу мету – переведення схеми XML-документа до сукупності пов'язаних OWL класів і властивостей. Тобто, це фактично є створення своєрідної онтології XML-документа. В нашому випадку задача суто специфічна – відтворення засобами OWL онтологічної схеми даних поданої засобами “Graph Editor”. Таким чином, розроблена нами система не просто представляє схему XML-документа в OWL, це є засіб, направлений на «розуміння» семантики онтології, поданої в форматі “Graph Editor” і створення на її основі нової онтології, яка використовує для відтворення аналогічної семантики доступні інструменти OWL. Подібним чином діє і зворотній конвертер – його дія направлена на відтворення за допомогою програмних об'єктів саме семантики поданої OWL-документом, а вже потім створення відповідного XML-файлу, з використанням засобів, зрозумілих для “Graph Editor”.

При роботі над проектом «Трансдисциплінарна інтелектуальна інформаційно-аналітична система супроводження процесів реабілітації при пандемії (TISP)» виникла гостра потреба у створенні онтологій з різних предметних областей, так чи інакше пов'язаних з процесом реабілітації. Для створення цих онтологій було використано КІТ «Поліедр». На його ж базі виконується і безпосереднє застосування онтологій у різних частинах робочого проекту. Однак, деякі частини розроблюваного програмно-інформаційного комплексу потребували саме OWL-формату, бажано у синтаксисі RDF/XML. Це обумовлено наявними програмними інструментами, що виконують роль графових СУБД і здійснюють формальні запити до онтологій. До таких частин відносяться «Діалогова підсистема “Білої книги”» та «Словник “Білої книги” – інтерактивна довідкова система». У наявній на даний момент реалізації перша використовує RDFlib, що безпосередньо завантажує дані з OWL і працює з мовою запитів SPARQL; друга використовує СУБД Neo4j, в цьому разі OWL-документ слугує як вихідний матеріал для створення графової бази даних власного формату. Сфера використання засобів, що так чи інакше

працюють з OWL-форматом у наших подальших планованих проектах буде тільки поширюватися. Так, зокрема, нещодавно було створено робочий прототип діалогової системи «Guidance on core indicators for entity reporting on contribution towards implementation of the Sustainable Development Goals».

**Висновки:** Створено зручний програмний інструмент для переведення онтологічних графів з формату XML-“Graph Editor” (що є власним для системи онтологічного інжинірингу КІТ «Поліедр»), до стандартних OWL-онтологій у синтаксисі RDF/XML. Це суттєво розширює практичну придатність створених у такий спосіб онтологій, дає змогу використовувати їх у довідкових системах, системах збору і обробки інформації, з використанням стандартних засобів роботи з RDF/XML-форматом (безпосередньо чи у якості вихідного). Можливість зворотної конвертації дає змогу застосовувати КІТ «Поліедр» для перегляду і редагування сторонніх OWL-онтологій, набутих у мережі Інтернет або і іншим способом.

**Перспективи подальших досліджень.** У подальшій роботі планується використовувати розроблений конвертер для переведення створених у людино-зрозумілій формі онтологій у КІТ «Поліедр» і серіалізованих у відповідному форматі до машино-зрозумілого RDF/XML-синтаксису, що дозволить їм працювати під управлінням таких графових СУБД, як RDFlib, Neo4j, Virtuoso, тощо. Подальший розвиток самої системи конвертації може передбачати врахування більшого спектру можливостей як OWL так і КІТ «Поліедр» і їх більш точного зіставлення, створення можливостей більш гнучких налаштувань параметрів конвертації користувачем, додавання підтримки інших варіантів синтаксису представлення онтологій, як то “Manchester OWL Syntax”, “Turtle”, тощо.

#### **Додаткова інформація.**

Опубліковані матеріали на мають конфлікту інтересів.

**Подяка.** Дослідження виконано при підтримці гранту Національного фонду досліджень України за договором від 07.05.2021 р. № 159/01/0245 «Трансдисциплінарна інтелектуальна інформаційно-аналітична система супроводження процесів реабілітації при пандемії (TISP)»

#### **Список літератури**

1. Guarino, N. Formal ontology in information systems. Proceedings of FOIS'98, Trento, Italy. 1998 June 6-8: 3-15.
2. Robinson I., Webber J., Eifrem E. Graph databases: O'Reilly Media: USA. 219 p.
3. Hebel J., Fisher M., Blace R., Perez-Lopez A. Semantic Web Programming: John Wiley & Sons; 2009. 648 p.
4. Goel A. Neo4j Cookbook: Packt Publishing: Birminhem, UK; 2015. 209 p.
5. Neosemantics: Neo4j RDF & Semantics toolkit. Available from: <https://neo4j.com/labs/neosemantics/>
6. Litvin A. A., Velychko V. Yu., Kavarynskyi V. V. Method

of information obtaining from ontology on the basis of a natural language phrase analysis. *Problems in programming* 2020; 2-3. 2020: 322 – 330. doi: 10.15407/pp2020.02-03.322

7. Wiens V., Lohmann S., Auer S. WebVOWL Editor: Device-Independent Visual Ontology Modeling. Proceedings of the ISWC 2018 Posters & Demonstrations, Industry and Blue Sky Ideas Tracks. CEUR Workshop Proceedings, vol. 2180. 2018.

8. Liepins R., Grasmanis M. OWLGeEd ontology visualiser. ISWC 2014 Developers Workshop. 2014: 37 – 42.

9. Lembo D., Santarelli V., Savo D. F. Drawing OWL 2 ontologies with Eddy the aditor. *Ai Communciations* 2018; 31 (9-10): 1 – 17. doi: 10.3233/AIC-180751

10. Kramer T., Schlenoff C., Kootbally Z., Pietromartire A. Software Tools for XML to OWL Translation. National Institute of Standards and Technology. June 2015. 34 p.

### References

1. Guarino, N. Formal ontology in information systems. Proceedings of FOIS'98, Trento, Italy. 1998 June 6-8: 3-15.

2. Robinson I., Webber J., Eifrem E. Graph databases: O'Reilly Media: USA. 219p.

3. Hebel J., Fisher M., Blace R., Perez-Lopez A. Semantic Web Programming: John Wiley & Sons; 2009. 648 p.

4. Goel A. Neo4j Cookbook: Packt Publishing: Birminhem, UK; 2015. 209 p.

5. Neosemantics: Neo4j RDF & Semantics toolkit. Available from: <https://neo4j.com/labs/neosemantics/>

6. Litvin A. A., Velychko V. Yu., Kaverynskyi V. V. Method of information obtaining from ontology on the basis of a natural language phrase analysis. *Problems in programming* 2020; 2-3. 2020: 322 – 330. doi: 10.15407/pp2020.02-03.322

7. Wiens V., Lohmann S., Auer S. WebVOWL Editor: Device-Independent Visual Ontology Modeling. Proceedings of the ISWC 2018 Posters & Demonstrations, Industry and Blue Sky Ideas Tracks. CEUR Workshop Proceedings, vol. 2180. 2018.

8. Liepins R., Grasmanis M. OWLGeEd ontology visualiser. ISWC 2014 Developers Workshop. 2014: 37 – 42.

9. Lembo D., Santarelli V., Savo D. F. Drawing OWL 2 ontologies with Eddy the aditor. *Ai Communciations* 2018; 31 (9-10): 1 – 17. doi: 10.3233/AIC-180751

10. Kramer T., Schlenoff C., Kootbally Z., Pietromartire A. Software Tools for XML to OWL Translation. National Institute of Standards and Technology. June 2015. 34 p.

## CONVERSION OF GRAPH DATABASE STORAGE FORMATS AND ITS ROLE IN ONTOLOGICAL ENGINEERING ON THE EXAMPLE OF REHABILITATION SUPPORT SYSTEM.

*Kaverynskyi V. V.*

Graph databases seem like an effective tool for presenting concepts of a particular subject area and the relationships exist in it. Such representations are known as ontologies. Graph databases could be also acceptable for other cases, especially for data with a natural graph structure, for example: description of interaction between organizations and/or persons, transport, telecommunications, language structures, etc. Graph databases of an ontological type have been widely used in the development of an intelligent information-analytical system for rehabilitation monitoring processes (TISP). In particular, such sections as «White Book», «Gastroenterology», «Ischemic Heart Disease», «Medical Law», «Physical Rehabilitation», etc. were presented in this way.

There exist many options for presenting and storing graph databases, since, one of the most common and universal formats for ontologies storing is OWL, in particular with RDF/XML syntax. There are also such formats as «Manchester OWL Syntax», «OWL Functional Syntax», «Turtle» and some others. The conversion possibility between these formats is implemented and available in such applications as «Protégé», or through the API of online services. But the problem is the lack of an accessible, but powerful and convenient graphic editor for ontology schemes that works with named formats. Nevertheless, creating a complex ontology scheme in, for example, the built-in «Protégé» editor is a complex and inconvenient process. Moreover, the possibilities of constructing an ontology in «Protégé» are rather narrow. The same is also true for numerous online ontology editors.

A convenient tool for visual development, viewing and editing of onto-graphs is the KIT «Polyhedron» (also known as «Graph Editor») developed at the Institute of Cybernetics of the National Academy of Sciences of Ukraine. The combination of a friendly visual interface, wide functionality and easy going makes KIT «Polyhedron» almost one of the most advanced tools in modern ontological engineering. But the feature of KIT «Polyhedron» is a very specific format for the graphs storing in XML-files. This format (in following – the «Graph Editor» format) is perceived exclusively by the KIT «Polyhedron». However, OWL, which is acceptable for many graphical databases as the primary or intermediate format for storing and downloading data, is not currently supported in the «Polyhedron» KIT. Thus, the task was to create a converter that translates XML-files from the format «Graph Editor» to OWL in RDF/XML syntax, also in the opposite direction.

The analysis is carried out to study the data storage peculiarities for ontological graphs in «Graph Editor» format, structure of corresponding XML-files and ways of the most complete representation of both main and auxiliary (technical) information by available options of RDF/XML syntax. On the other hand, methods have been developed to represent standard OWL entities and constructs using Graph Editor. A technique has been developed for program meta-representation of data read from the corresponding OWL-entities and methods of their serialization in OWL or XML-«Graph Editor» format have been implemented. The corresponding software system was built in the form of an online service.

The nodes of the ontological graph constructed in the KIT «Polyhedron» during conversion can be moved to the standard OWL-entities like classes, properties or named individuals. If the node type is not explicitly specified, it becomes a class by default. It is also possible to represent non-typed vertices by other OWL types, taking into account specific types of relationships (if any). Non-typed relationships could optionally be interpreted as «Subclass» or «Default». A special attention is paid to the representation of the data contained in the nodes and links of the graph. By default, they become a label, but if a type is specified, they can become any of the «annotations» for the corresponding OWL entity. From these links, specific technical ontology object properties are formed that connect

the corresponding nodes, to which the data are to be attached in the form of a label or other annotations. It is possible to create various, both standard (subclass, equivalence, difference, etc.) and arbitrary links between the nodes of the ontological graph, if they are specified in the source file. Attachments are represented by links through tag attributes that indicate their own ID and the ID of the parent data tag.

The inverse conversion creates onto-graph nodes of the corresponding types from the classes, properties, and named individuals of the ontology classes, except for the technical ones, which are used in this case to restore the structure of the corresponding data. Tags that contain technical information are not converted to linked data, but are used to initialize the node characteristics (shape, color, position, etc.).

The developed program for conversion was used for the translation to the OWL-format of the ontology on the “White Book”, which using RDFlib works as a part of the dialog subsystem TISP. A number of other medical ontologies, which are the parts of the TISP project, have also been successfully converted to OWL format. Analysis of the results showed no loss of information during both direct and reverse conversion. The files are successfully accepted by RDF/XML parsers, including “Protégé”, RDFlib, Neo4j (downloading OWL ontology using the Neosemantics plug-in), and is able to be subjected to formal SPARQL queries to information extraction. Transferring the resulting OWL file to Neo4j to the ontology, the Cypher query language can also be used.

A convenient software tool has been created for translating ontological graphs created using the powerful visual ontological engineering system KIT “Polyhedron” to OWL-ontologies in RDF/XML syntax. This significantly expands the practical applicability of the ontologies created in this way, allows them to be successfully implemented into reference systems, information collection and processing systems, using standard tools for RDF/XML format processing. The reverse conversion feature allows you to use the “Polyhedron” KIT to view and edit third-party OWL ontologies acquired on the Internet or otherwise.

**Key words:** *international classification of functioning, rehabilitation, patient/personal-centered rehabilitation, multi-professional team, digital library, natural language processing, artificial intelligence, intelligent information technology.*