



“НАЦІОНАЛЬНА СИСТЕМА ЗАБЕЗПЕЧЕННЯ ЯКОСТІ
І ВЗАЄМНОЇ ДОВІРИ В СИСТЕМІ ВИЩОЇ ОСВІТИ (TRUST)”

Проект скерований на розробку нової стратегії забезпечення якості вищої освіти та підходів до прозорої та неупередженої оцінки якості освіти, в основу яких покладено європейські практики, що базуються на «трикутнику» освіта – дослідження – інновації згідно з Лісабонською стратегією. Це дозволить створити національне інноваційне середовище з університетами в ролі основних елементів системи забезпечення якості.

“TOWARDS TRUST IN QUALITY ASSURANCE SYSTEMS (TRUST)”

The project “Towards Trust in Quality Assurance Systems” will provide a unified transparent framework for the quality assurance of the Ukrainian higher education system which is in line with the common European academic culture and supports both the national reform of higher education and integration of Ukraine into the European Higher Education Area.



ПРОГРАМА ТЕМПУС – освітня програма Європейського Союзу, яка підтримує модернізацію системи вищої освіти та створює простір для співпраці в країнах-партнерах ЄС через університетські проекти. Заснована в 1990 році, програма Темпус охоплює сьогодні 27 країн: на Західних Балканах, у Східній Європі, Центральній Азії, Північній Африці та на Близькому Сході. Програма Темпус фінансує міжуніверситетську співпрацю у сфері розробки та вдосконалення навчальних програм, управління університетами, взаємодії науковців і громадянського суспільства, партнерство освіти і бізнесу, а також структурні реформи у сфері вищої освіти.

TEMPUS is the European Union’s programme which supports the modernisation of higher education in the Partner Countries of Eastern Europe, Central Asia, the Western Balkans and the Mediterranean region, mainly through university cooperation projects. The overall objective of Tempus is to contribute to the creation of an area of cooperation in the field of higher education between the European Union and Partner Countries in the countries neighbouring the EU.



Темпус-проект 516935-TEMPUS-1-2011-1-FITEMPUS-SMGR
“Towards Trust in Quality Assurance Systems”

Національний портал забезпечення якості вищої освіти

ІНСТРУКЦІЯ ДЛЯ РОЗРОБНИКІВ СТОРОННІХ СИСТЕМ ЩОДО ІНТЕГРАЦІЇ ЇХ З НАЦІОНАЛЬНИМ ПОРТАЛОМ ЗАБЕЗПЕЧЕННЯ ЯКОСТІ ВИЩОЇ ОСВІТИ



Шевченко Олександр Юрійович
Сокол Вадим Вікторович
Шевченко Олена Леонідівна

Національний портал забезпечення якості вищої освіти

**ІНСТРУКЦІЯ ДЛЯ РОЗРОБНИКІВ
сторонніх систем щодо інтеграції
їх з національним порталом
забезпечення якості вищої освіти**

Львів 2014

**Видання підготували представники учасників консорціуму Темпус-проекту
516935-TEMPUS-1-2011-1-FITEMPUS-SMGR “Towards Trust in Quality Assurance Systems”.**

Учасники проекту:

Університет м. Ювяскюля, Фінляндія – координатор проекту
Університет м. Коїмбра, Португалія
Технічний університет м. Кошице, Словаччина
Європейська асоціація з питань забезпечення якості вищої освіти, Бельгія
Харківський національний університет радіоелектроніки, Україна
Чернівецький національний університет ім. Ю. Федьковича, Україна
Український католицький університет, м. Львів, Україна
Національна академія керівних кадрів культури і мистецтв, м. Київ, Україна
Міністерство освіти і науки України
Українська асоціація студентського самоврядування, м. Київ, Україна
Державна інспекція навчальних закладів, м. Київ, Україна
Державна акредитаційна комісія, м. Київ, Україна
Департамент атестації кадрів вищої кваліфікації МОН України
Національна академія педагогічних наук України, м. Київ, Україна
Інститут інноваційних технологій і змісту освіти, м. Київ, Україна
Парламентський комітет з питань освіти та науки, м. Київ, Україна
(консультативний партнер)

Ця інструкція є результатом діяльності консорціуму Темпус-проекту “Національна система забезпечення якості і взаємної довіри в системі вищої освіти (TRUST)”. Вона призначена для розробників сторонніх систем щодо інтеграції їх з Національним порталом забезпечення якості вищої освіти – сучасним інтернет-ресурсом, який виконуватиме роль платформи для відкритого розміщення та збереження інформації про освітні та наукові досягнення академічної спільноти України, що сприятиме створенню культури громадського контролю і впливу на якість вищої освіти та появи довіри суспільства до системи вищої освіти.

Ми вдячні експертам з Університету м. Ювяскюля (Фінляндія), Університету м. Коїмбра (Португалія), Технічного університету м. Кошице (Словаччина), Європейської асоціації із забезпечення якості вищої освіти (Бельгія) за їх внесок у розробку порталу. З повним списком експертів та наданих матеріалів можна ознайомитись за адресою: <http://dovira.eu/experts.html>

We are grateful to the experts from University of Jyväskylä (Finland), University of Coimbra (Portugal), Technical University of Cosice (Slovakia), The European Association for Quality Assurance in Higher Education (ENQA, Belgium) for their support, which made essential contribution to developing the portal. The complete list of the contributors and the provided materials is available in: <http://dovira.eu/experts.html>

*This project has been funded with support from the European Commission.
This publication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.*

*Проект фінансується за підтримки Європейської Комісії. Ця публікація відображає
винятково погляди автора, і Європейська Комісія не відповідає за будь-яке використання
матеріалів, що містяться в цьому виданні.*

ЗМІСТ

| | |
|---|----|
| ВСТУП, ЦІЛІ ТА ЗАВДАННЯ ПРОЕКТУ | 5 |
| 1. ТЕХНІЧНИЙ ОПИС ПОРТАЛУ..... | 8 |
| 1.1 Технології, що були використані при реалізації порталу... 8 | |
| 1.2 Структура порталу | 9 |
| 1.3 Взаємодія модулів порталу..... | 10 |
| 1.4 Універсальні редактори ресурсів (smart property) | 11 |
| 1.5 Технічні особливості модуля реєстрації ресурсів | 12 |
| 1.6 Технічні особливості модуля формування систем цінностей..... | 12 |
| 1.7 Технічні особливості модуля розрахунку рейтингів..... | 13 |
| 1.8 Публікація рейтингів і систем цінностей в порталі..... | 13 |
| 1.9 Особливості використання бази знань Virtuoso | 14 |
| 2. ІНТЕГРАЦІЯ ПОРТАЛУ ІЗ ЗОВНІШНІМИ ІНФОРМАЦІЙНИМИ СИСТЕМАМИ..... | 15 |
| 2.1 Інтеграція зовнішніх систем із порталом..... | 15 |
| 2.2 Використання онтологічного порталу як загальнодоступного сховища інформації | 16 |
| 3. ОНТОЛОГІЯ І МОДЕЛІ ДАНИХ..... | 19 |
| 3.1 Основні елементи сервісної онтології | 20 |
| 3.2 Онтологія предметної області | 29 |
| 3.3 Методи зміни онтології предметної області з можливістю автоматичної зміни інтерфейсів порталу..... | 29 |
| 4. ТЕХНІЧНЕ ОБСЛУГОВУВАННЯ ТА РОЗГОРТАННЯ СИСТЕМИ..... | 31 |

ВСТУП, ЦІЛІ ТА ЗАВДАННЯ ПРОЕКТУ

Портал – загальнодоступний персональний портфоліо-менеджер для працівників освіти та освітніх закладів. Його користувачами є як представники галузі освіти, так і кінцеві споживачі їх праці – родини абітурієнтів, підприємці та ін.

Портал є інструментом громадського контролю і впливу на якість вищої освіти. Він покликаний допомогти освітянам відкрито поділитися власними викладацькими та науковими досягненнями, а суспільству – оцінити їх за актуальною шкалою цінностей. Ми впевнені, що досить налагодити діалог між освітянами і суспільством (органами влади, працедавцями, студентами, іншими), відновити взаємну довіру, і позитивні зрушення у вищій освіті не забаряться. Основне завдання порталу – забезпечити прозорість системи освіти та вимог споживачів освітніх результатів як фундамент для такого діалогу.

Портал – єдина точка доступу до хмарного сховища зареєстрованих даних про суб'єктів, що надають послуги у сфері освіти, та про їхні досягнення (в термінах порталу всі вони мають єдину назву «освітні ресурси»), а також документи, які підтверджують ці досягнення. Він також є інструментом для формування і зберігання персоналізованих рейтингів зареєстрованих ресурсів, а також зберігання, перегляду і порівняння систем цінностей, на підставі яких такі рейтинги були або можуть бути побудовані. Зручною можливістю порталу є також можливість динамічного генерування CV і персональних веб-сторінок суб'єктів освітньої діяльності на основі їхніх досягнень, зареєстрованих у порталі.

Для користувачів портал доступний за адресою <http://portal.dovira.eu>.

Таким чином, портал дає наступні переваги:

- з будь-якої точки земної кулі мати необмежений 7/24/365 on-line доступ до інформації як про себе, так і про інші освітні ресурси, зареєстровані в порталі;
- зберігати і управляти структурованою інформацією про себе і динамічно генерувати та поновлювати на основі наданої інформації свої CV, звіти і персональні веб-сторінки;
- здійснювати пошук зареєстрованих освітніх ресурсів, навіть не маючи повної назви ресурсу, а тільки її фрагменти, завдяки використаній технології Semantic Web;
- знаходити потрібні ресурси за категоріями, використовуючи розширений фільтр;
- знаходити завдяки зареєстрованим досягненням (статті, гранти, авторські курси тощо) експертів у відповідній галузі. Для всіх зареєстрованих в порталі статей обчислюється імпаکت-фактор на основі рівня видання, в якому стаття надрукована. Рівень видання отримується із зовнішнього ресурсу (джерела), рекомендованого європейськими експертами. Ним є список міжнародних видань, ретельно складений і рекомендований для публікації фінською академією наук. Цей список регулярно поновлюють європейські експерти, що призводить до його автоматичного поновлення в порталі. До списку увійшло близько дев'ятнадцяти з половиною тисяч видань (наукових журналів і матеріалів конференцій), реферованих в базах Scopus і Web of Science з відповідним імпаکت-фактором. Згідно з рекомендаціями фінської академії наук, всі видання мають рівень від «1» до «3», після реєстрації статті/доповіді її імпакт-фактор обчислюється, базуючись на зазначеному рівні видання. Команда проекту ввела 4-й рівень, найнижчий, – для видань, які не ввійшли у список, рекомендований європейськими експертами;
- доступ до інформації про зареєстровані освітні ресурси: про співробітників, їхні досягнення, зібрані не у вигляді статистичних даних, а з можливістю перегляду і оцінки будь-якого досягнення, включаючи документи, що його підтверджують;

- можливість прозорого менеджменту процесів, у результаті яких мають визначитися кращі показники за обраними критеріями, тобто такі, які мають на меті рейтингування: відбір кандидатів на вакантну посаду, мотивація кращих співробітників, підрозділів, розподіл матеріальних ресурсів;
- можливість гнучкого налаштування системи цінностей, за якою здійснюється рейтингування, залежно від поточних цілей і завдань, а також зберігання систем цінностей і можливість порівняння результатів рейтингів, зроблених у різний час за різними показниками для відстеження динаміки змін. Ця можливість є базовою для внутрішньої системи забезпечення якості при визначенні стратегії розвитку підрозділу/вишу і відповідності системи забезпечення якості обраній стратегії, її цілям, завданням і критеріям успіху;
- знаходження експертів у відповідній галузі, визначення рівня вченого за його роботами, базуючись на зареєстрованих досягненнях, зокрема статтях і доповідях на конференціях, імпаکت-фактор яких обчислюється за наведеним вище принципом;
- портал надає унікальні можливості для розробників стороннього програмного забезпечення. Він дозволяє на власній платформі мати доступ до усієї публічної інформації, що опублікована в онтологічному порталі. Також він надає можливості для розширення онтологічної бази власними даними.

Портал є прикладом нового типу ядра розподіленого програмного забезпечення. Таке ядро може бути вдосконалене лише за рахунок модифікації моделі даних навіть на етапі використання без внесення жодних змін до початкового коду.

1. ТЕХНІЧНИЙ ОПИС ПОРТАЛУ

1.1 Технології, що були використані при реалізації порталу

При створенні порталу був використаний набір технологій, які, з одного боку, спрощують розробку порталу та будь-яку подальшу його доробку, а з іншого боку – дозволяють іншим розробникам швидше і простіше зрозуміти, як працює портал.

За базову технологію було вибрано технологію JavaEE, яка є промисловою технологією і в основному використовується у високпродуктивних проектах, в яких необхідна надійність, масштабованість і гнучкість. Ця технологія використовується для реалізації великих проектів, наприклад, для організації таких складних веб-порталів, як Trust, особливо якщо необхідно забезпечити безперебійну роботу багатьох користувачів. JBoss Application Server (JBoss AS) використовується як сервер прикладних програм JavaEE, а в якості контейнера сервлетів JBoss AS використовує Tomcat.

Для створення інтерфейсу користувача (веб-сторінок) використовується технологія JSP. Перевага, яку дає технологія JSP в порівнянні з іншими веб-технологіями, полягає в тому, що JSP є технологією для розробки веб-програм, яка є платформонезалежною, такою, яку легко переносити і яка легко може бути розширена.

Для того, щоб сторонні системи мали змогу отримувати дані з порталу і передавати дані в портал, використовуються веб-служби, які створені за допомогою JAX WS.

OpenLink Virtuoso Server використовується як сховище бази знань для зберігання онтологічних даних (класів, властивостей і екземплярів) у порталі. Це окреме програмне забезпечення, для

взаємодії з яким використовується бібліотека Jena. Вона надає API для вилучення даних або для додавання даних в RDF-графи. Графи є абстрактні моделі. Для моделі можуть бути задані джерела даних з файлів, баз даних, URL або інші. В порталі як джерело даних використовується OpenLink Virtuoso Server.

1.2 Структура порталу

При реалізації порталу була використана архітектура, показана на рисунку 1.

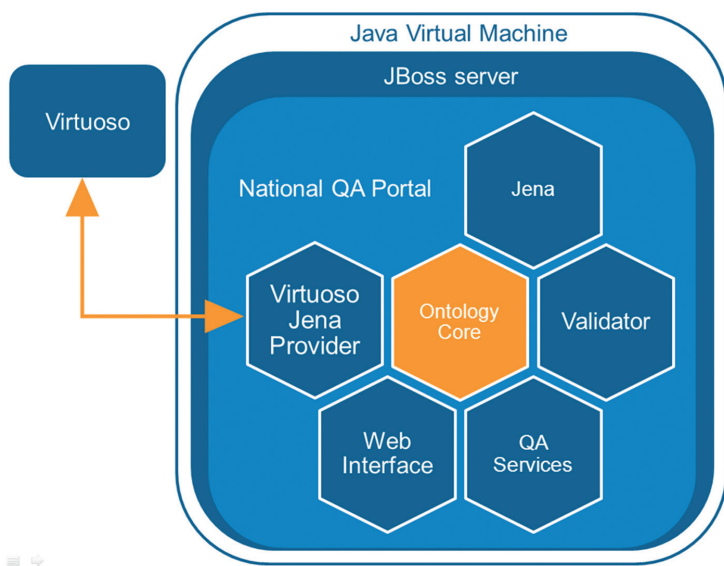


Рисунок 1. Архітектура порталу

У структурі порталу можна виділити дві великі частини:

- 1) Virtuoso – сховище бази знань, яке являє собою окрему прикладну програму, що працює на тому ж або іншому комп'ютері. Доступ до Virtuoso здійснюється за допомогою спеціальної бібліотеки Virtuoso Jena Provider.
- 2) National QA Portal – безпосередньо сам портал, що працює як окрема прикладна програма в сервері прикладних програм JBoss server, який, у свою чергу, є Java-програмою, тобто працює під управлінням Java Virtual Machine.

1.3 Взаємодія модулів порталу

В порталі Trust при архітектурному проектуванні було використано MVC-шаблон. Стандартно є три компоненти: модель (яка має в собі усю бізнес-логіку), вигляд (відповідальний за представлення даних користувачеві) і контролер (який поєднує перші дві частини між собою).

В порталі Trust модель (Model) складається з механізму основної взаємодії, яка працює з об'єктами моделі (ресурсами), а також із зберіганням та завантаженням їх із джерела даних (у нашому випадку база знань Virtuoso). Загальні компоненти: ranking manager (управління рейтингами), sov manager (управління системами цінностей), resource manager (управління ресурсами), user manager (керування обліковими записами користувачів) – для управління профілями користувачів, filter (фільтр) – використовується для пошуку ресурсів, автоматичного створення SPARQL-запитів тощо), achievement registration (реєстрація досягнень) – зверніть увагу, що цей модуль не має інтерфейсу користувача.

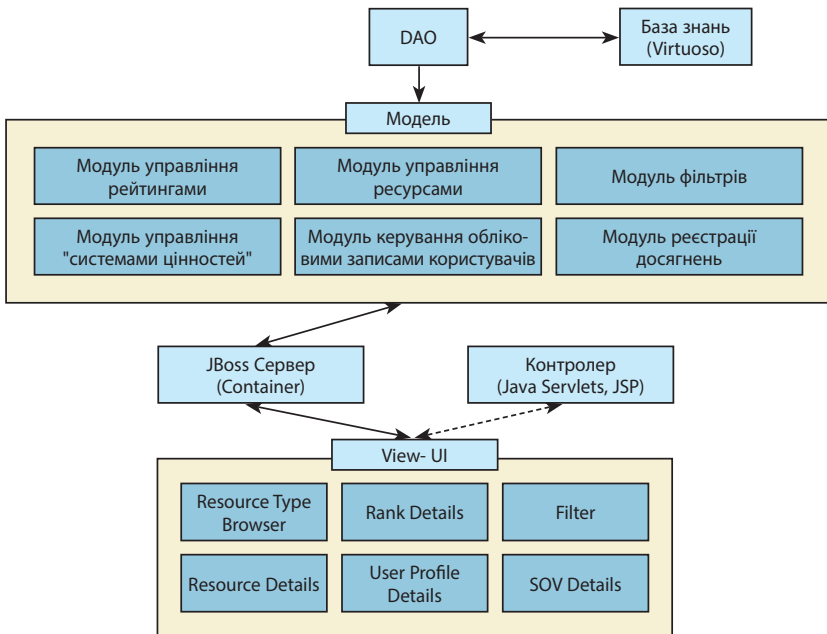


Рисунок 2. Взаємодія модулів порталу

Взаємодія між моделлю та джерелом даних буде виконуватися за допомогою спеціального об'єкта (DAO), що реалізує в собі всю взаємодію, яка є специфічною для джерела даних, наприклад, ініціалізація драйвера або з'єднання з базою даних.

Сервер прикладних програм (Application Server) являє собою контейнер для порталу як прикладної програми. Цей сервер надає інформацію від користувача, отриману через Інтернет, і виступає як базова платформа для постійної роботи порталу. В цьому випадку буде використовуватися JBoss application server версії 7.1.

Сервер прикладних програм безпосередньо запускає всі контролери порталу, тобто сервлети (для обробки HTTP-запитів та створення відповідей), jsp-сторінки та іншу взаємодію на стороні сервера між виглядом і моделлю.

Вигляд (View) складається з декількох модулів інтерфейсу користувача. Ці модулі до певної міри схожі до менеджерів. Є модуль перегляду типу ресурсу (resource type browser), модуль докладного перегляду ресурсу та його редагування (resource details), модуль докладного перегляду рейтингу та його редагування (rank details), фільтр пошуку ресурсів (filter), модуль докладного перегляду профілю користувача та його редагування (user profile details) та модуль докладного перегляду системи цінностей та її редагування (sov details). Всі компоненти інтерфейсу користувача незалежні і можуть бути використані в будь-якому контексті (напр., вікно пошуку з фільтром при редагуванні значення властивості ресурсу).

1.4 Універсальні редактори ресурсів (smart property)

При формуванні частин веб-сторінок, які відповідають за редагування об'єктів моделі даних порталу, використовується спеціально розроблений універсальний редактор. Його універсальність полягає в динамічному формуванні форми для редагування кожного об'єкта в залежності від його типу. Перелік елементів управління на формі редагування та їх поведінка цілком визначається переліком та типом властивостей класу онтології, екземпляром якої є об'єкт, що підлягає редагуванню. Ще одною особливістю редактора є здатність відображати додаткові властивості

об'єктів, які не зберігаються для нього в моделі, але можуть бути обраховані за потреби в реальному часі на основі інших його властивостей. Наприклад, коли потрібно відобразити вік, тоді можна використовувати властивість «дата народження» або «дата створення» і на її основі сформуванати нову властивість – «вік», яка не може бути відредагована. Універсальний редактор «розумних властивостей» при ініціалізації зв'язується з процедурою перетворення однієї властивості в іншу і на сторінках детального перегляду автоматично відображає її в полях, які не можна редагувати. Прикладом такої властивості є властивість «вік об'єкта».

1.5 Технічні особливості модуля реєстрації ресурсів

Модуль реєстрації ресурсів створює всі свої інтерфейси на підставі аналізу онтології предметної області, використовуючи ключові класи та властивості з сервісної онтології. Таким чином він адаптується під поточний стан онтології.

Кожен користувач отримує можливість переглядати інформацію про об'єкти онтології. Після авторизації на порталі він також отримує можливість змінювати і створювати нові об'єкти.

Користувач має можливість через налаштування зв'язати свій профіль користувача порталу з ресурсом, успадкованим від класу `ts:ActiveResource`. В такому разі параметри цього ресурсу блокуються від змін іншими користувачами, і тільки цей користувач або адміністратор може вносити зміни в цей об'єкт.

Від базових класів сервісної онтології всі класи ресурсів успадкували можливість створювати посилання на свої електронні сторінки, а також можливість додавати один або декілька підтверджуючих документів в електронній формі. Для класів, успадкованих від `ts:ActiveResource`, є можливість додавати фотографії ресурсу.

1.6 Технічні особливості модуля формування систем цінностей

Модуль формування систем цінностей дозволяє користувачеві скласти для будь-якого ресурсу, що описаний в порталі, систему, за якою цей ресурс буде оцінюватися. Інтерфейс цього модуля

також генерується динамічно в залежності від того, які класи і властивості були додані в онтологію предметної області. В онтології предметної області внесено набір досягнень, які враховуються порталом. При виборі типу ресурсу портал аналізує, які можуть бути досягнення, і вибирає всі, які можуть стосуватися обраного типу ресурсу. Для кожного з досягнень користувач може вказати, наскільки воно важливе в цій системі цінностей, вибравши значення ваги досягнення в діапазоні від 0 до 100 одиниць. Також в систему цінностей можуть бути включені числові параметри, характерні для цього типу досягнень, їх важливість також вказується в діапазоні від 0 до 100 одиниць. Модуль формування систем цінностей містить гнучкий фільтр, який дозволяє з класу об'єктів виділити групу, яку необхідно рейтингувати. Оцінка кожного досягнення ресурсу може бути сформована на підставі одного з двох взаємовиключних факторів: в першому випадку, на основі кількості таких досягнень без урахування їх якості, в другому – з урахуванням їх впливу на систему освіти, що був розрахований за додатковими ознаками, наприклад, важливість статті може бути оцінена на основі рейтингу журналу, в якому вона була надрукована. Всі системи цінностей можуть бути збережені в порталі і за ними можуть бути розраховані рейтинги.

1.7 Технічні особливості модуля розрахунку рейтингів

Модуль розрахунку рейтингів дозволяє за сформованою системою цінностей обчислити поточний рейтинг об'єктів обраного класу. Пораховані рейтинги можуть бути збережені окремо і включають в себе всю детальну інформацію про розрахунок і значення кожного параметра сформованого рейтингу. Крім того, кожен розрахований рейтинг посилається на детальну інформацію про систему цінностей, за якою цей рейтинг був розрахований.

1.8 Публікація рейтингів і систем цінностей в порталі

Зазвичай всі системи цінностей і розраховані рейтинги зберігаються в приватному просторі користувача, який їх створив, і є недоступні для інших користувачів системи. Щоб мати змогу

посилатися на рейтинги з інших систем, передбачена можливість їх публікації на порталі. Після публікації збережена інформація стає доступна всім користувачам portalу. Також формується постійне посилання, за яким будь-який користувач інтернет-простору зможе перейти на розрахований рейтинг і побачити систему цінностей, на основі якої він був розрахований.

1.9 Особливості використання бази знань Virtuoso

OpenLink Virtuoso server використовується як база знань для зберігання даних онтології (класи та екземпляри) в порталі. Початкова онтологія (сервісна онтологія та онтологія предметної галузі) може бути завантажена через веб-інтерфейс. Оскільки вважається, що сервісна онтологія (класи і властивості) не повинна змінюватись протягом функціонування portalу, то вона була розроблена до його запуску окремо. Сервісну онтологію зручно розробляти з використанням спеціальних онтологічних інструментів, в даному випадку використовувалося Protege.

Взаємодія між порталом та Virtuoso здійснюється за допомогою бібліотеки Jena, яка надсилає запити в Virtuoso, при цьому використовується Virtuoso Jena provider, що надається розробниками Virtuoso.

2. ІНТЕГРАЦІЯ ПОРТАЛУ ІЗ ЗОВНІШНІМИ ІНФОРМАЦІЙНИМИ СИСТЕМАМИ

2.1 Інтеграція зовнішніх систем із порталом

Система призначена для побудови рейтингів ресурсів, специфічних для системи вищої освіти України. Система надає користувачеві можливість отримати незалежну та адекватну інформацію про освітні ресурси в Україні, а також можливість порівнювати їх між собою.

З одного боку, портал дозволяє користувачам отримувати інформацію через веб-програму. Але, крім того, він надає іншим системам можливість взаємодіяти з даними порталу через спеціальний програмний інтерфейс. Використання даних порталу в сторонніх прикладних програмах дозволяє останнім не створювати своє сховище даних зі складною структурою і численними зв'язками, а використовувати вже існуючі дані з порталу. Це значно спрощує розробку зовнішніх прикладних програм, а використання стандартних структур опису даних спрощує взаємодію між прикладними програмами.

Крім того, використання єдиного центру зберігання інформації різними прикладними програмами дозволить накопичувати великий обсяг даних (знань). І навіть якщо прикладні програми зроблять невеликий внесок, то сумарно в результаті вийде досить велика база знань, доповнена з різних прикладних програм, але з тою самою структурою даних. Це добре як для порталу, так і для зовнішніх прикладних програм.

Використання однієї і тієї ж бази знань набагато спрощує взаємодію між різними прикладними програмами, оскільки структура даних однакова.

Якщо це буде необхідно, то структура порталу може бути доповнена, і тоді нова версія онтології буде доступна відразу ж у всіх прикладних програмах.

2.2 Використання онтологічного порталу як загальнодоступного сховища інформації

Сторонні розробники мають змогу взаємодіяти з порталом. Для цього є ряд методів (API порталу), використовуючи які можна отримати або зберегти дані в порталі.

Таблиця 1.

Перелік методів для отримання та збереження даних із зовнішніх систем

| № | метод | параметри | відповідь | опис |
|---|------------------------|----------------------|--------------------|--|
| 1 | 2 | 3 | 4 | 5 |
| 1 | auth | login, password | auth_token | Автентифікація та авторизація користувача в системі, необхідно для виконання подальших запитів |
| 2 | run_sparql | auth_token, query | sparql_result | Виконати SPARQL-запит на онтології порталу |
| 3 | get_public_rating_list | auth_token | public_rating_list | Отримати список рейтингів, які є публічно доступні |
| 4 | get_public_sov_list | auth_token | public_sov_list | Отримати список систем цінностей, які є публічно доступні |
| 5 | get_own_rating_list | auth_token | own_rating_list | Отримати список рейтингів, які створив цей користувач |
| 6 | get_own_sov_list | auth_token | own_sov_list | Отримати список систем цінностей, які створив цей користувач |

| 1 | 2 | 3 | 4 | 5 |
|---|--------------------|-----------------------|----------------|--|
| 7 | get_rating_details | auth_token, rating_id | rating_details | Отримати детальну інформацію про рейтинги |
| 8 | get_sov_details | auth_token, sov_id | sov_details | Отримати детальну інформацію про систему цінностей |
| 9 | calculate_rating | auth_token, sov_id | rating_details | Розрахувати рейтинг за зазначеною системою цінностей |

Кожного разу, коли зовнішня система ініціює роботу з порталом, вона мусить виконати запит автентифікації та авторизації – для цього необхідно викликати метод `auth`, в який передаються логін і пароль користувача portalу. Логіном може бути email користувача, зареєстрованого в порталі. Якщо користувач з таким email-ом та паролем існує, то у відповідь буде відправлений параметр `auth_token`, який в подальшому повинен передаватися як один із параметрів запиту зовнішньою системою у всіх наступних запитах до portalу.

Для отримання або оновлення даних онтології необхідно викликати метод `run_sparql` і передати в якості параметрів `auth_token` (цей параметр приходить у відповіді на запит `auth`) і сам SPARQL-запит.

Існує кілька методів API portalу для роботи з системами цінностей. Це: метод для отримання списку систем цінностей, які є публічно доступні (`get_public_sov_list`), і метод для отримання списку систем цінностей, які були створені конкретним користувачем (`get_own_sov_list`). Дані методи повертають викликаючій стороні тільки базову інформацію по кожному об'єкту: назва, ідентифікатор, опис, автор і т. д. Повну інформацію можна отримати використовуючи спеціально створений для цього метод – `get_sov_details`.

Схожими є методи для роботи з рейтингами. Є метод, який дозволяє отримати список публічно доступних рейтингів (`get_public_rating_list`), і метод, що дозволяє отримати список рейтингів, які були створені конкретним користувачем (`get_own_rating`).

list). Для отримання повної інформації по рейтингу існує окремий метод – `get_rating_details`.

Також є можливість розрахувати рейтинг на основі існуючої системи цінностей. Щоб це зробити, необхідно викликати спеціальний метод (`calculate_rating`), передавши як параметр ідентифікатор системи цінностей, на основі якої необхідно побудувати рейтинг.

Всі запити від зовнішніх систем заносяться в журнал, це дозволить відслідковувати частоту використання цієї функціональності, визначати, які дані користуються більшою популярністю, а також збирати статистичні дані для майбутніх звітів.

При обробці запитів від зовнішніх систем існує декілька обмежень, які накладаються на запити:

1. Забороняється змінювати структуру онтології. За допомогою запитів від зовнішніх систем не можна створювати, оновлювати або видаляти класи або властивості в онтології.
2. Персональні дані користувачів не повертаються. За допомогою запитів від зовнішніх систем не можна отримати детальну інформацію про користувачів або про облікові записи користувачів.

3. ОНТОЛОГІЯ І МОДЕЛІ ДАНИХ

Всі дані у порталі Trust зберігаються виключно у вигляді онтології, як система управління цією онтологією використовується Virtuoso. Онтологічна інформація є найважливіша складова системи, вона керує частиною структури сторінок, переліком об'єктів, що можуть бути зареєстровані, переліком їх властивостей і т. ін. Онтологія є повністю відкрита для інших систем. Дані, які зберігаються в онтології, представлені в універсальному вигляді, це дає змогу легко використовувати їх в інших програмах, а також доповнювати їх результатами роботи інших програм.

Онтологія portalу складається з трьох основних гілок:

- 1) *сервісна онтологія* містить основні класи, через які ядро portalу «бачить» всю іншу онтологію. Сервісна онтологія містить ключові класи та властивості, нащадками яких є класи предметної галузі;
- 2) *онтологія предметної галузі* містить класи об'єктів, що відносяться до предметної галузі portalу. Ядро portalу має доступ до об'єктів онтології предметної галузі лише через класи-предки, визначені в сервісній онтології. Такий підхід дозволяє дуже глибоко міняти онтологію предметної галузі без жодних змін у вихідному коді portalу, що робить можливим розвивати і модифікувати портал лише за рахунок зміни моделі даних предметної галузі, якою він оперує;
- 3) *онтологія, яка містить конкретні екземпляри класів предметної галузі*. Ця онтологія була виділена для спрощення процесу модифікації наповнення portalу конкретними

даними. Екземпляри всіх класів додаються лише в цю гілку онтології.

У процесі експорту з порталу кожна з перелічених трьох гілок онтології зберігається в окремому файлі, який потім може бути відредагований в будь-якому редакторі онтології і знову імпортований назад до порталу. При такому підході є можливість змінити структуру онтології без втрати введених даних і без внесення змін у вихідний код порталу.

3.1 Основні елементи сервісної онтології

Ця онтологія є кореневою і визначає основний функціонал порталу. Вихідний код порталу жорстко пов'язаний з кожним класом або атрибутом класу цієї онтології, тому будь-які зміни сервісної онтології тягнуть за собою зміни у вихідному коді ядра порталу. Вихідний код порталу не залежить від будь-яких інших гілок онтології. Доступ з коду до класів, атрибутів та екземплярів інших гілок онтології здійснюється за допомогою відношення спадкування класів та атрибутів сервісної онтології класами і атрибутами онтології предметної галузі. Структуру класів сервісної онтології представлено на рисунку 3.

Як видно на рисунку, всі класи сервісної онтології знаходяться в просторі імен (namespace) `ts` (trust service) – це є характерна ознака для ядра онтологічного порталу. Найзагальнішим класом онтології є клас `owl:Thing`, він є базовим для всіх інших класів системи. Від нього успадковуються групи класів, які впливають на функціональність системи. Клас `ts:Resource` є базовим класом основних класів онтологічного порталу, таких як `ts:ActiveResource` – базовий клас всіх ресурсів, здатних створювати інші ресурси (до таких класів відноситься і клас «людина»); `ts:Beneficiary` – базовий клас для всіх ресурсів, які можуть отримувати бали в рейтингу від досягнень своїх співробітників (організації, кафедри, підрозділи і т.д.); `ts:ValuableResource` – базовий клас для ресурсів, створення яких може бути досягненням об'єктів спадкоємців класу `ts:ActiveResource` (публікації, дисертації, премії, гранти тощо). Перемикачем режимів відображення та аналізу ресурсів

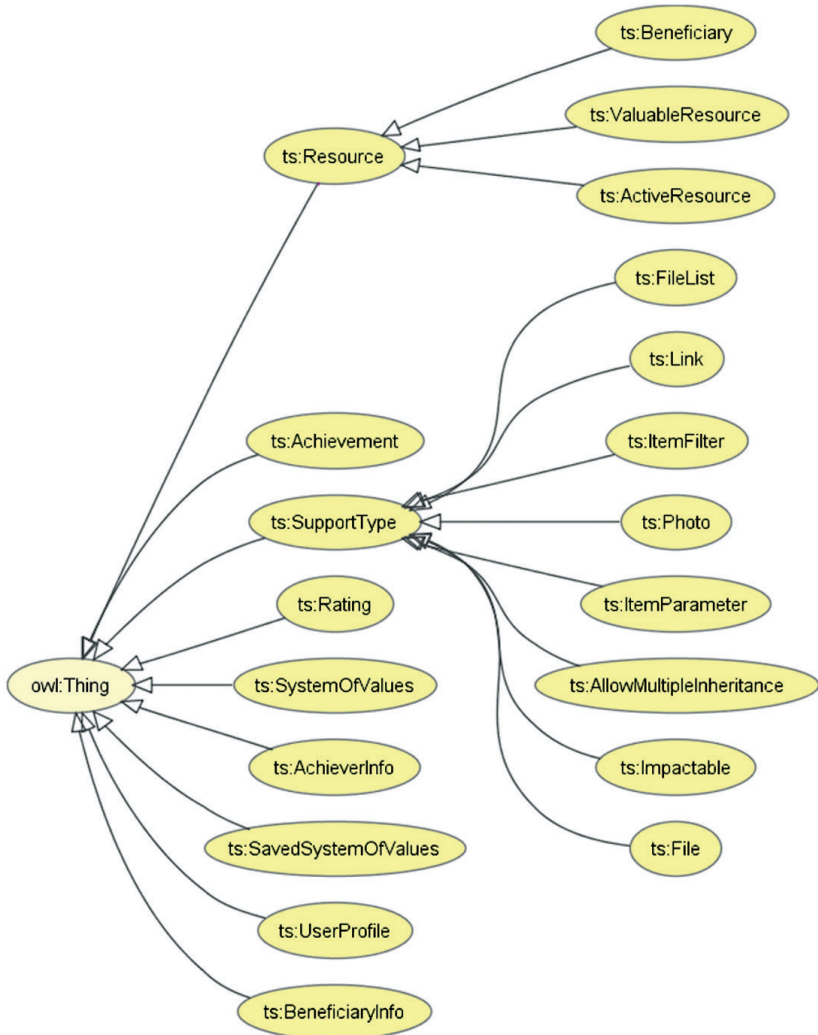


Рисунок 3. Структура класів сервісної онтології

для онтологічного порталу є факт успадкування класу об'єкта від одного з трьох різних базових класів.

Також у сервісній онтології знаходиться ряд абстрактних об'єктів, які є базовими класами для різних класів онтологічного порталу, їх призначення наведено в таблиці 2.

Таблиця 2.

Призначення класів сервісної онтології

| ідентифікатор класу | назва | призначення |
|------------------------|--|--|
| 1 | 2 | 3 |
| ts:Achievement | Досягнення | Базовий клас для об'єктів, які зберігають інформацію про досягнення |
| ts:AchieverInfo | Інформація про досяжника | Базовий клас для об'єктів, які зберігають інформацію про об'єкти, які мають досягнення |
| ts:BeneficiaryInfo | Інформація про організацію або підрозділ | Базовий клас для об'єктів, які зберігають інформацію про організації та підрозділи |
| ts:Rating | Рейтинг | Базовий клас для об'єктів, які зберігають інформацію про розраховані рейтинги |
| ts:Resource | Ресурс | Базовий клас для об'єктів типу «ресурс» |
| ts SavedSystemOfValues | Збережена система цінностей | Базовий клас для об'єктів, які зберігають інформацію про збережені системи цінностей, на основі яких були розраховані рейтинги |
| ts:SupportType | Додаткові типи даних | Базовий клас для об'єктів додаткових властивостей, які підтримуються в онтологічному порталі |

| 1 | 2 | 3 |
|-------------------|---------------------|--|
| ts:SystemOfValues | Система цінностей | Базовий клас для об'єктів, які зберігають інформацію про збережені системи цінностей |
| ts:UserProfile | Профіль користувача | Базовий клас для об'єктів, які зберігають інформацію про зареєстрованих у порталі користувачів |

Важливим класом сервісної онтології є клас `ts:SupportType` – від нього успадковуються всі класи додаткових властивостей, які підтримуються в онтологічному порталі, але не є стандартними для онтологічних систем; це – фотографії, файли, посилання на інтернет-ресурси тощо. До кожного з цих типів додані свої редактори, які дозволяють на різних сторінках portalу переглядати, редагувати і додавати інформацію таких типів.

Прикладом властивості, яка найчастіше використовується, є посилання і файли, які можна додавати в онтологічний портал. На рисунку 4 можна побачити механізм інтеграції додаткових властивостей у базові класи онтологічного portalу.

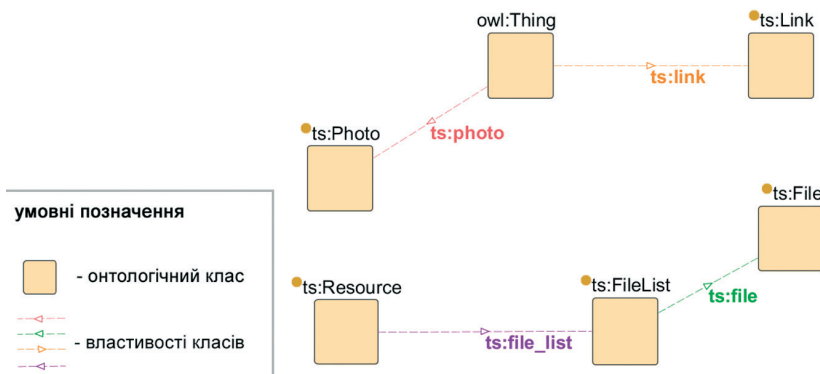


Рисунок 4. Інтеграція додаткових властивостей у базові класи онтологічного portalу

Важливими класами для реєстрації досягнень є також класи `ts:Achievement`, `ts:AchieverInfo` та `ts:BeneficiaryInfo`. Вони пов'язані між собою за допомогою агрегації, як показано на рисунку 5.

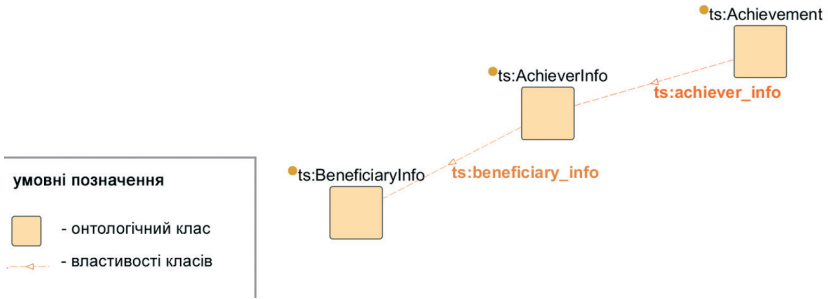


Рисунок 5. Взаємодія сервісних класів, призначених для реєстрації досягнень

Об'єкт, успадкований від `ts:Achievement`, створюється тоді, коли з'являється об'єкт досягнення. Цей клас містить масу інформації про досягнення, тип, дату створення, інформацію, до якого класу ресурсів відноситься це досягнення. Крім цього, властивість `ts:achiever_info` містить посилання на сервісний клас `ts:AchieverInfo`, призначений для зберігання інформації про те, хто отримав це досягнення. Також цей об'єкт містить властивість `ts:beneficiary_info`, яка, у свою чергу, посилається на об'єкт `ts:BeneficiaryInfo`, що описує організації та підрозділи, які брали участь у створенні цього досягнення.

Прості властивості, визначені у сервісній онтології, наведені далі в таблиці 3.

Також у сервісній онтології є об'єктні властивості. Вони дозволяють агрегувати різні об'єкти онтології. Список доступних об'єктних властивостей наведено в таблиці 4.

Таблиця 3.

Прості властивості, визначені в сервісній онтології

| ідентифікатор властивості | область | діапазон | призначення |
|---------------------------|------------------|----------|--|
| ts:achievementParam | ts:ItemParameter | string | Назва параметра, за яким виконується рейтингування |
| ts:achievers_count | ts:Achievement | int | Кількість досяжників |
| ts:activated | owl:Thing | boolean | Визначає, чи є об'єкт активним |
| ts:byAmount | ts:ItemParameter | boolean | Нормалізоване значення параметра рейтингування |
| ts:byInfluence | ts:ItemParameter | boolean | Нормалізоване значення ваги параметра |
| ts:comment | owl:Thing | string | Коментар |
| ts:condition | ts:ItemFilter | string | Умова фільтрації даних |
| ts:cv_path | ts:UserProfile | string | Посилання на кеш CV-сторінки |
| ts:cv_url | ts:UserProfile | string | Посилання на сторінку CV |
| ts:file_comment | ts:File | string | Коментар до файлу |
| ts:file_display_name | ts:File | string | Назва файлу |
| ts:file_path | ts:File | string | Розміщення файлу |
| ts:file_upload_date | ts:File | date | Дата |
| ts:has_impact_factor | owl:Thing | float | Цінність ресурсу (для статті – показник цитувань) |
| ts:language | ts:UserProfile | string | Мова користувача |

Продовження таблиці 3.

| ідентифікатор властивості | область | діапазон | призначення |
|---------------------------|--|----------|---|
| ts:link_url | ts:Link | string | Адреса посилання |
| ts:login | ts:UserProfile | string | Ідентифікатор користувача |
| ts:name | owl:Thing | string | Ім'я користувача |
| ts:normlized_weight | ts:ItemParameter | float | Вага параметра рейтингування |
| ts:numberOfYears | ts:ItemParameter | int | Кількість років, за які враховуються дані |
| ts:photo_path | ts:UserProfile | string | Розміщення фото |
| ts:property_uri | ts:Photo | string | Посилання на фотографію |
| ts:propertyParam | ts:ItemParameter | string | Ідентифікатор параметра |
| ts:rating_path | owl:Thing | string | Розміщення кешу обчисленого рейтингу |
| ts:resource_type_uri | ts:SystemOfValues ts:Rating ts:SavedSystemOfValues | string | Тип ресурсу |
| ts:sov_path | owl:Thing | string | Розміщення кешу системи цінностей, яка була використана при обчисленні рейтингу |
| ts:value1 | ts:ItemFilter | string | Перший параметр фільтра |
| ts:value2 | ts:ItemFilter | string | Другий параметр фільтра |
| ts:weight | ts:ItemParameter | float | Обчислене значення параметра |

Таблиця 4.

Об'єктні властивості, які визначені в сервісній онтології

| ідентифікатор властивості | область | діапазон | призначення |
|---------------------------|--|---------------------|--|
| ts:a_part_of | owl:Thing | owl:Thing | Вказує на те, що клас є частиною іншого класу |
| ts:achievement_object | ts:Achievement | ts:ValuableResource | Вказує на об'єкт, який є досягненням |
| ts:achiever | ts:AchieverInfo | ts:ActiveResource | Вказує на об'єкт, який є автором досягнення |
| ts:achiever_info | ts:Achievement | ts:AchieverInfo | Вказує на об'єкт, який містить інформацію про автора досягнення |
| ts:beneficiary | ts:BeneficiaryInfo | ts:Beneficiary | Вказує на об'єкт, який є організацією, що отримує бали за досягнення |
| ts:beneficiary_info | ts:AchieverInfo | ts:BeneficiaryInfo | Вказує на об'єкт, який містить додаткову інформацію про організацію, що отримує бали за досягнення |
| ts:describing_resource | ts:UserProfile ts:Rating ts:SystemOfValues ts:SavedSystemOfValues | ts:ActiveResource | Вказує на об'єкт, що є описом ресурсу |
| ts:file | ts:FileList | ts:File | Вказує на об'єкт, що пов'язаний з файлом |

Продовження таблиці 4.

| ідентифікатор властивості | область | діапазон | призначення |
|---------------------------|------------------------|------------------------|---|
| ts:file_list | ts:Resource | ts:FileList | Вказує на список файлів |
| ts:filter_items | ts:SystemOfValues | ts:ItemFilter | Вказує на тип параметра фільтра у системах цінностей |
| ts:filter_parameters | ts:SystemOfValues | ts:ItemParameter | Вказує на значення параметра фільтра у системах цінностей |
| ts:link | owl:Thing | ts:Link | Вказує на об'єкт посилення |
| ts:photo | owl:Thing | ts:Photo | Вказує на об'єкт фотографії |
| ts:rating_created_by | ts:Rating | ts:UserProfile | Вказує на автора рейтингу або системи цінностей |
| ts:ratings_list | ts:UserProfile | ts:Rating | Вказує на об'єкт, що містить перелік усіх збережених рейтингів |
| ts:saved_sov | ts:Rating | ts:SavedSystemOfValues | Вказує на збережену систему цінностей, за якою було розраховано рейтинг |
| ts:sov_created_by | ts:SavedSystemOfValues | ts:UserProfile | Вказує на автора системи цінностей |
| ts:sov_list | ts:UserProfile | ts:SystemOfValues | Вказує на перелік збережених систем цінностей |

Використовуючи перелічені вище класи і властивості в SPARQL-запитах, які передаються як параметр `query` в метод `run_sparql`, можна отримувати будь-яку інформацію про всі об'єкти предметної області, завдяки тому факту, що всі класи і об'єкти онтології предметної області успадковуються від сервісної онтології. Онтологія предметної області може досить сильно змінюватися, але сервісна онтологія заблокована від змін, тому доступ до об'єктів онтологічного порталу бажано організувати, використовуючи класи та властивості тільки сервісної онтології, а доступ до всіх інших класів та об'єктів здійснювати як до спадкоємців класів сервісної онтології.

Також, використовуючи SPARQL-запити, можна змінювати і додавати інформацію в онтологію, тим самим збагачуючи портал новими знаннями.

Відкритість онтології і гнучкість роботи ядра порталу дозволяють йому легко адаптуватися до нових даних, які можуть бути внесені із зовнішніх систем.

Є ряд обмежень, які забезпечують цілісність даних і безпеку облікових даних користувачів. Модуль інтеграції з зовнішніми системами контролює безпеку запитів, що виконуються, і в разі спроби виконання забороненого запиту блокує його. Заборонено змінювати структуру сервісної онтології і порушувати цілісність даних онтологічної системи, існують також обмеження на отримання даних профілю користувача.

3.2 Онтологія предметної області

Онтологія предметної області ґрунтується на сервісній онтології порталу, але якщо сервісна онтологія є повністю незалежна від галузі застосування системи, то онтологія предметної області містить всю інформацію про всі класи об'єктів, які можна зареєструвати в порталі. Онтологію предметної області можна змінювати стандартними способами для онтологічних систем і потім завантажувати в портал, і він автоматично буде адаптувати інтерфейси користувача під онтологію предметної області.

3.3 Методи зміни онтології предметної області з можливістю автоматичної зміни інтерфейсів порталу

Для зміни онтології предметної області необхідно: через адмін-панель порталу <http://portal.dovira.eu/backup> зробити резервне копіювання (backup) гілки онтології, де зберігаються дані про екземпляри об'єктів. Потім за допомогою стандартного редактора онтологій внести зміни до онтології предметної області та за допомогою функцій об'єднання об'єднати онтологію та резервну копію (backup), використовуючи стандартний інструментарій редактора онтологій. Після цього результуючий файл або групу файлів необхідно імпортувати в Virtuoso за допомогою інструментарію адміністратора, доступного за адресою <http://portal.dovira.eu:8890>. Відразу після імпорту онтології портал змінить інтерфейси користувача відповідно до структури онтології предметної області.

4. ТЕХНІЧНЕ ОБСЛУГОВУВАННЯ ТА РОЗГОРТАННЯ СИСТЕМИ

Онтологічний портал встановлюється на серверній платформі під управлінням ОС Unix або Windows Server. Основою для роботи порталу є JBoss server і система управління онтологічними базами знань Virtuoso. Прикладна програма, що реалізує функціонал порталу, скомпільована у war-файл, який можна запустити на JBoss server та в Internet Cloud, сумісний з JBoss серверами. Налаштування, які використовуються під час взаємодії прикладної програми для JBoss та Virtuoso здійснюється в файлі trust-defaults.properties, що знаходиться в корені war-файлу. Також необхідно імпортувати в Virtuoso початкову версію онтології, це можна зробити за допомогою інструментарію адміністратора, який доступний за адресою <http://portal.dovira.eu:8890>.

Науково-практичне видання

Шевченко Олександр Юрійович

Сокол Вадим Вікторович

Шевченко Олена Леонідівна

Національний портал забезпечення якості вищої освіти

**ІНСТРУКЦІЯ ДЛЯ РОЗРОБНИКІВ
сторонніх систем щодо інтеграції
їх з національним порталом
забезпечення якості вищої освіти**



Національний портал забезпечення якості вищої освіти

ІНСТРУКЦІЯ ДЛЯ РОЗРОБНИКІВ СТОРОННІХ СИСТЕМ ЩОДО ІНТЕГРАЦІЇ ЇХ З НАЦІОНАЛЬНИМ ПОРТАЛОМ ЗАБЕЗПЕЧЕННЯ ЯКОСТІ ВИЩОЇ ОСВІТИ

