

ОНТОЛОГІЧНЕ МОДЕЛЮВАННЯ БАЗИ ЗНАНЬ «ОПЕРАЦІЙНІ СИСТЕМИ»

¹Вінницький національний технічний університет

У сучасних умовах розвитку інформаційної екосистеми вищої освіти та систем генеративного штучного інтелекту особливого значення набуває впровадження онтологічних баз знань. Застосування технологій Semantic Web дозволяє полегшити цей процес, забезпечуючи машинне розуміння контенту, інтероперабельність даних та їхню безшовну інтеграцію між різними освітніми платформами. У статті проведено онтологічне моделювання предметної області «Операційні системи» як основи для інтелектуалізації освітнього процесу. Актуальність дослідження зумовлена стрімким ускладненням архітектури сучасного системного ПЗ, що вимагає переходу від фрагментарного опису до машинозчитуваної формалізації. Описуються переваги використання стандарту *Ontology Web Language* для опису функціональних залежностей між компонентами ОС. Метою роботи є систематизація взаємозв'язків та генеалогії операційних систем за допомогою онтологічного інжинірингу. Описано процес створення моделі в середовищі *Protégé*, вибраному завдяки його надійній підтримці сучасних систем логічного виведення. Структура бази знань охоплює компоненти ОС, системне ПЗ та класифікацію операційних систем. Особливу увагу приділено моделюванню спадкоємності та архітектурних відгалужень, що є критично важливим для розуміння еволюції сучасних ОС. Практична цінність полягає у верифікації онтології через запити *SPARQL*, які дозволяють аналізувати зв'язки між типами ядер, файловими системами, мережевими стеками та засобами віртуалізації. Такий підхід дозволяє обмеження статичних репозиторіїв та забезпечує динамічне виведення нових знань. Результати дослідження підтверджують ефективність моделі для інтеграції в освітнє середовище, зокрема для формування контексту LLM-моделей під час оцінювання знань студентів. Розробка відкриває перспективи для інтелектуальних систем підтримки навчання, виходячи за межі традиційних методів викладання дисципліни «Операційні системи», що забезпечує студентам глибше розуміння архітектурних принципів побудови складних програмних продуктів.

Ключові слова: онтологія, онтологічна база знань, операційні системи, системні компоненти, системне програмне забезпечення, генеалогія операційних систем.

Вступ

Концепція сучасного університету передбачає запровадження змін в усі освітні процеси та комунікації. Серед таких змін — активне використання освітніх інформаційних технологій. При цьому саме комплексний та системний підхід забезпечує зменшення рівня фрагментарності використання цифрових інструментів та допомагає створити ефективну інформаційну екосистему [1], [2]. Однією з актуальних задач для створення такої екосистеми є організація доступу учасників освітнього процесу до онтологічно-структурованих баз знань на рівні Semantic WEB [3], ініціативи консорціуму W3C, метою якого є інтелектуалізація всієї мережі Інтернет. У рамках проекту Semantic WEB, який створено для аналізу семантики інформаційних ресурсів, створено і досі працює група WEB Ontology [4], оскільки саме онтологічний підхід є основою для подання знань про різні предметні області. Саме цією групою реалізовано проект зі створення нової мови подання онтологій OWL (*Ontology Web Language*). Власне мова OWL позиціонується як мова опису онтологій у Web. Під онтологією в цьому випадку розуміється сукупність термінів і понять, що використовуються в певній області знань або діяльності. До того ж онтологія також містить формальні, а це значить зрозумілі як для людей, так і для комп'ютерів, описи ключових понять їх властивостей і взаємозв'язків між ними. Засоби опису і аналізу онтологій, що закладені в OWL, дозволяють коректно співпрацювати з онтологіями безлічі програм, що не тільки комунікують з даними та знаннями, а опрацьовують та аналізують вміст інформаційних ресурсів. Стандарт OWL [5], [6] визначає три підмови, що

відрізняються рівнем своїх можливостей: OWL Lite, OWL DL і OWL Full. Важливо також відмітити їх відкритість і незалежність від областей застосування і видів знань.

Другою складовою цього процесу є впровадження в освітньому середовищі інформаційних технологій менеджменту знань, серед яких найперспективнішою є освітній онтологічний інжиніринг [7], що полягає в моделюванні, розробці, накопиченні та використанні спеціалізованих онтологічних баз знань як освітніх компонентів. Саме тому це дослідження ставить перед собою одну з цілей онтологічне моделювання бази знань в предметній області «Операційні системи».

Актуальність дослідження зумовлена безперервним ускладненням архітектури сучасного системного програмного забезпечення та високою динамікою розгалуження генеалогії операційних систем. Попри існування масштабних відкритих баз знань, більшість з них забезпечують лише фактологічний опис інформаційних технологій, залишаючи поза увагою глибинні функціональні залежності між типами ядер, файловими системами, мережевими стеками та середовищами віртуалізації. У зв'язку з цим виникає наукова та освітня потреба у застосуванні методів інженерії знань для переходу від плоского енциклопедичного представлення до глибокої машинозчитуваної формалізації.

Мета дослідження — підвищення ефективності систематизації міжкомпонентних взаємозв'язків у структурі операційних систем та їхньої генеалогії за допомогою апарату онтологічного моделювання. Для досягнення поставленої мети потрібно вирішити такі завдання:

- систематизація основних концепцій навчальної дисципліни «Операційні системи»;
- розробка онтологічної бази знань у предметній області «Операційні системи» з використанням платформи Protégé;
- верифікація онтології за допомогою SPARQL-запитів до резонерів.

Аналіз останніх досягнень та публікацій

Серед наукових публікацій, щодо безпосередньо онтологій операційних систем, можна виділити роботу [8]. Онтологія розглядає два сімейства окремих мобільних ОС: Android та Symbian, визначаючи ключові класи та методи для передачі об'єктів між ними, а також використовують діаграми властивостей та механізми автоматичного логічних висновків для побудови загальної ієрархії понять. Автори у своїй роботі не розглянули ще одну мобільну операційну систему – iOS, яка охоплює велику долю ринку мобільних пристроїв iPhone. До того ж їхня онтологія робить фокус на функціонуванні смартфона та ролі операційної системи у цьому процесі, не приділяючи уваги іншим не мобільним платформам. До того ж дані про операційні системи містяться також у відкритих репозиторіях, таких як DBpedia [9] та Wikidata [10]. Дані з цих репозиторіїв містять переважно енциклопедичні відомості про операційні системи, проте не містять генеалогічних даних про операційні системи, незважаючи на те, що у статтях Вікіпедії все таки зустрічаються згадки про відгалуження нових операційних систем чи дистрибутивів GNU/Linux. Також у даних репозиторіях відсутні відомості про компоненти операційних систем та вибір дистрибутивами GNU/Linux [11] тих чи інших реалізацій системних компонентів.

Для розробки нової онтології, орієнтованої на використання у освітньому просторі, насамперед необхідно враховувати концепції, важливі для розвитку практичних навичок у студентів. У роботі [12] акцентовано на вбудованих UNIX-подібних операційних системах для Інтернету речей (IoT). Книга демонструє роботу утиліт командного рядка для управління операційною системою, що не враховувалось у наявних онтологіях та репозиторіях даних. Навчальний посібник «Операційні системи» [13], розроблений для студентів спеціальності «Комп'ютерна інженерія» в КПІ ім. Ігоря Сікорського, наголошує на поглибленому вивченні методологічних та теоретичних принципів побудови і функціонування сучасних ОС. Особливу увагу в ньому приділено практичному засвоєнню базових механізмів керування комп'ютерними ресурсами, зокрема плануванню процесів, управлінню пам'яттю, файловими системами та підсистемами введення-виведення. Стаття [14] акцентує на використанні засобів віртуалізації у вивченні операційних систем, пропонуючи дидактичну модель застосування технологій віртуалізації UNIX-подібних ОС. Запропонований перелік компетентностей є важливим для подальшої систематизації основних концепцій навчальної дисципліни «Операційні системи». У статті визначено системне програмування мовою C++ із застосуванням API-інтерфейсу ОС як один з ключових показників оволодіння практичними навичками з операційних систем.

Попри наявність окремих наукових розробок та загальнодоступних баз знань, у предметній області «Операційні системи» зберігається дефіцит комплексних онтологічних рішень, які б поєднували глибинний аналіз архітектурних компонентів з генеалогією систем. Наявні підходи здебільшого обмежу-

ються або вузькоспеціалізованими сегментами, такими як мобільні платформи, або ж надають виключно енциклопедичну інформацію, залишаючи поза увагою динамічні функціональні залежності та практичні аспекти системного програмування. Отже, подальший розвиток онтології є важливим для створення інтелектуальних освітніх інструментів, забезпечуючи базу для глибокого вивчення методологічних принципів побудови ОС, оцінювання практичних навичок студентів та автоматизованої підтримки складних інтегрованих систем навчання.

Результати дослідження

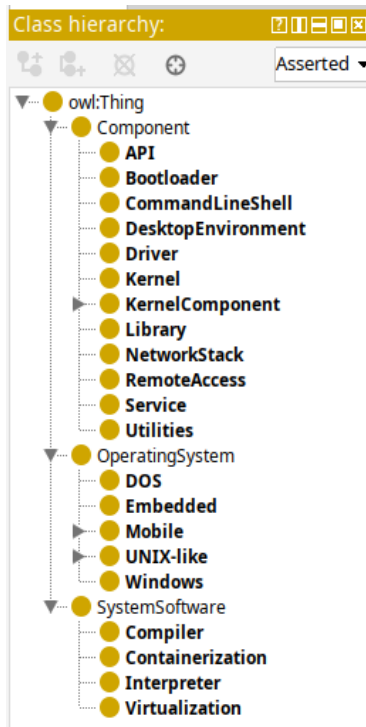


Рис. 1. Ієрархія класів онтологічної моделі

Для ефективного моделювання онтологій потрібно мати досить потужні і функціональні інструменти. Сьогодні доступно більше двох десятків інструментів для створення та редагування онтологій. Серед них є як комерційні продукти, так і редактори вільного доступу. Найпопулярнішими з них є Classic, CYC, Protégé, WebProtégé та TopBraid Composer. Protégé має для цих задач беззаперечні переваги: вільне розповсюдження, відкриту багаторівневу архітектуру, використання мов OWL, RDF, XML, величезну бібліотеку різноманітних онтологій та плагінів. Protégé [15] використано як базову платформу для моделювання насамперед тому, що він має найбільшу серед інших кількість підтримуваних ризонерів, зокрема HermiT та Pellet, а також тому, що на відміну від згаданих аналогів до нього можна підключити нестандартні плагіни з репозиторіїв Github. Першим етапом моделювання буде визначення області та масштабів онтології «Операційні системи». Під масштабуванням розуміється рівень деталізації під час розгляду властивостей та атрибутів класів і екземп-

рки онтологія презентує предметну область операційних систем з навчальною метою, вона повинна включати базовий тезаурус, різновиди, класифікацію, структури, суттєві відмінності, еволюційні корені. Початковий набір основних класів ієрархічної моделі показаний на рис. 1.

Структура класів онтологічної моделі пропонує розподіл класів на такі три категорії: компоненти ОС (Component), системне програмне забезпечення (SystemSoftware) та власне операційні системи (OperatingSystem). Згідно зі структурою онтологічної моделі, операційні системи складаються з компонентів ОС, які є їх невід'ємною частиною. Обов'язковими для всіх операційних систем є такі компоненти як завантажувач, ядро, драйвери, системні бібліотеки, служби, оболонка командного рядка та утиліти (рис. 2). Більшість сучасних операційних систем містить також мережевий стек, який зазвичай є частиною ядра ОС, проте у багатьох історичних операційних систем початково не було мережевого стеку, оскільки мережеві протоколи IP та IPX з'явилися пізніше. Також мережевий стек може бути опціональним у такій категорії ОС як вбудовані операційні системи (рис. 3). У разі їх використання на мікроконтролерах та промислових пристроях за відсутності мережевої взаємодії мережевий стек може не бути включеним у склад ОС, зокрема тому, що вбудовані операційні системи часто встановлюються на платформи з дуже малим об'ємом постійної та/або оперативної пам'яті.

Графічна оболонка є присутньою у операційних системах для персональних комп'ютерів та мобільних пристроїв. Одним з виключень є операційна система Windows Server, яка використовує графічну оболонку для віддаленого робочого столу (служба віддалених терміналів). Служба віддалених терміналів відноситься до інструментів віддаленого доступу. Крім віддаленого доступу в режимі робочого столу до багатьох ОС можна підключитись в режимі текстового терміналу, зокрема за допомогою таких протоколів як Telnet та SSH.

Крім компонентів операційної системи, які є її частиною багатоопераційних систем, можуть містити також додаткові опціональні пакети програмного забезпечення універсального призначення, зокрема інтерпретатори, компілятори, а також інструменти для віртуалізації та контейнеризації (рис. 4). Інтерпретатори часто розвивались окремо від операційних систем незалежними компаніями та спільнотами розробників, як наприклад Python та ECMAScript. Історичні ОС мали у своєму

складі лише інтерпретатор команд (як один із режимів командної оболонки). Проте з появою таких ОС як Windows 9x та Windows NT (рис. 5) такі інтерпретатори як Windows Script Host почали входити в склад ОС, надаючи можливість запуску скриптів Visual Basic та JavaScript без встановлення стороннього ОС.

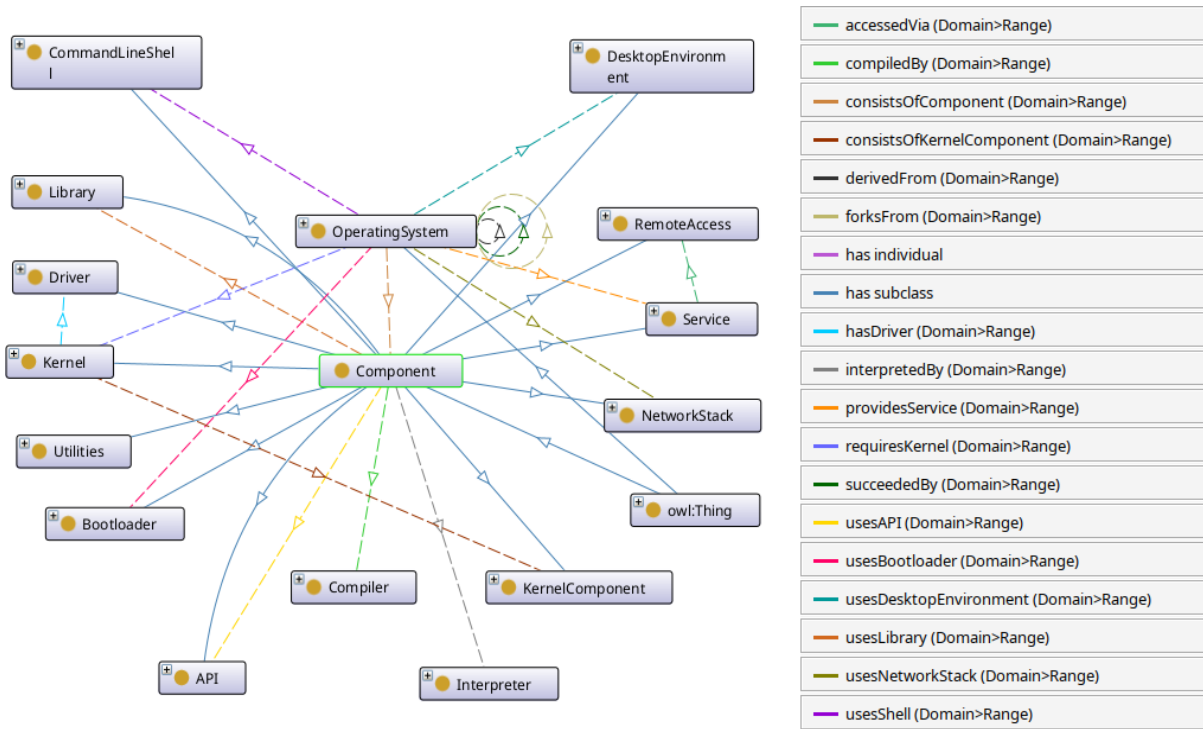


Рис. 2. Ієрархія класів компонентів операційної системи

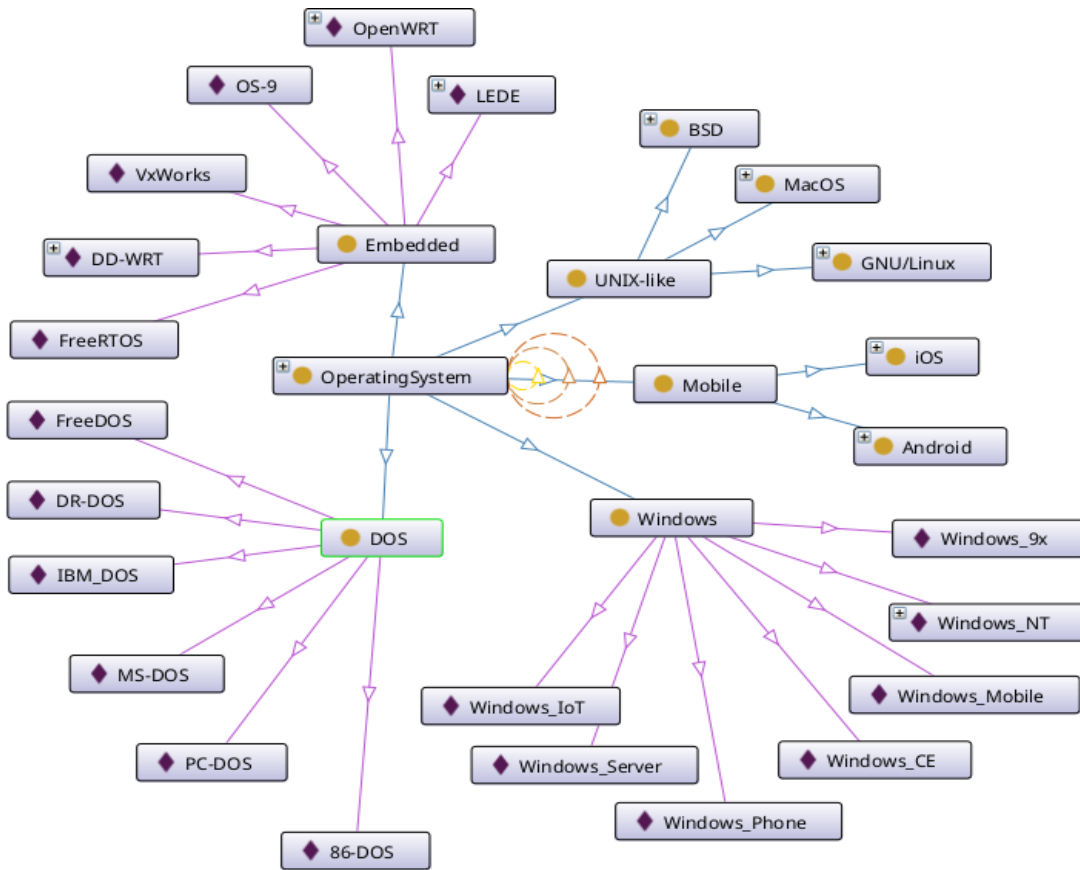


Рис. 3. Класи та екземпляри сімейств операційних систем

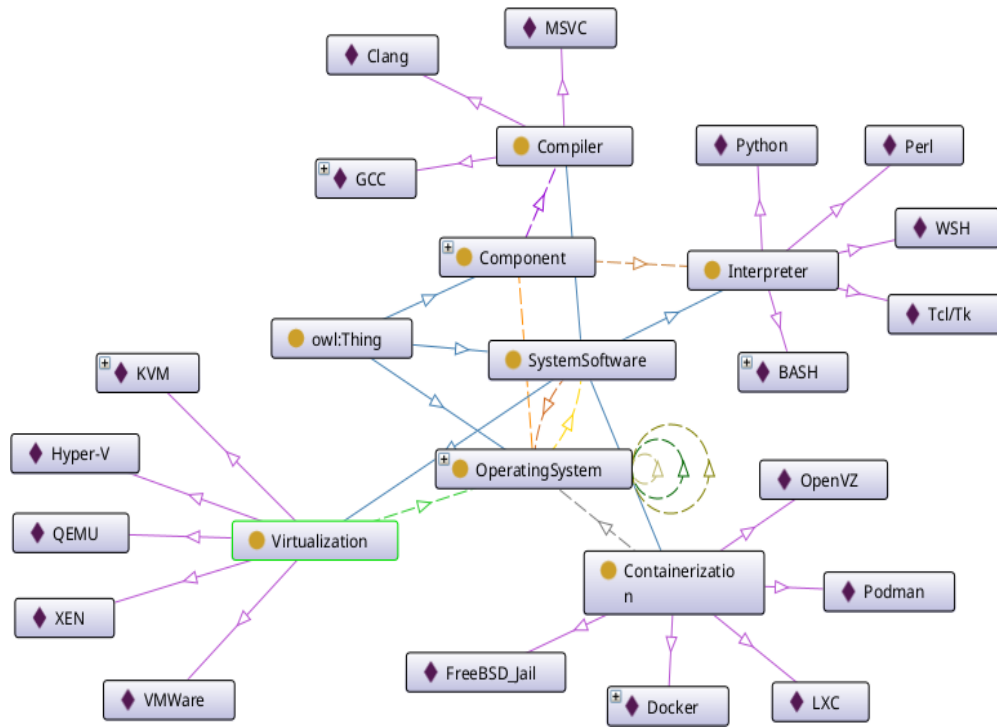


Рис. 4. Класи та екземпляри опціонального системного програмного забезпечення

Іншим опціональним пакетом ПЗ є компілятор. Першим компілятором що увійшов у склад ОС був Portable C Compiler, який входив спочатку у склад ОС Unix, а згодом увійшов у склад ОС BSD

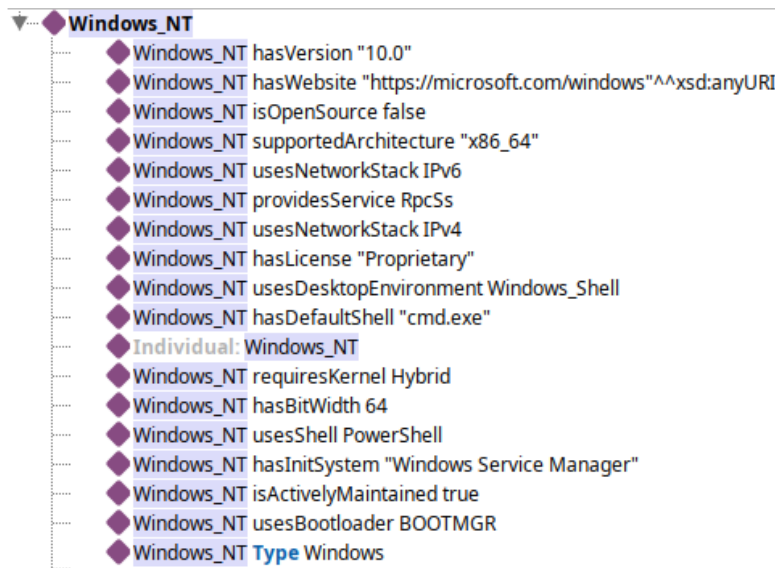


Рис. 5. Властивості екземпляру ОС Windows NT

Unix. Іншим компілятором, який входив у склад ОС був GNU C Compiler (GCC), як частина ОС GNU. Самі по собі операційні системи Unix, BSD та GNU не вимагали обов'язкової наявності компілятора для нормальної роботи, проте він потрібен для встановлення програмного забезпечення, яке розповсюджувалось у вигляді поточного коду.

Для верифікації онтології використано SPARQL-запити до резервів. Запити виконуються під час роботи різернів Hermit та Pellet. Запити сформульовано таким чином, щоб можна було визначити як властивості операційних систем, так і зв'язки між їх компонентами.

Досить часто у різних сімействах ОС спостерігається спадкоємність, коли нова операційна система (або її дистрибутив у випадку з GNU/Linux) переймала архітектурні особливості та базу поточного коду у іншій ОС (або дистрибутиві).

В той час як в окремих операційних системах рідше утворювались відгалуження через патентні обмеження, у дистрибутивів операційної системи GNU/Linux відгалуження спостерігаються досить часто.

Розроблена онтологічна модель дозволяє за допомогою SPARQL-запиту прослідкувати спадкоємність між окремими ОС або їх дистрибутивами (рис. 6 — на прикладі дистрибутиву GNU/Linux Debian).

На можливість спадкоємництва ОС також впливає закритість чи відкритість ліцензії операційної системи. Для цього для онтологічної моделі розроблено SPARQL-запит який дозволяє їх відокремити (рис. 7).

SPARQL query: 🔍 📄 🗑️

```
SELECT DISTINCT ?os1 ?os2 ?os3
WHERE {
  ?os1 rdf:type/rdfs:subClassOf* <ostamp-2026#OperatingSystem> .
  ?os1 <ostamp-2026#derivedFrom> ?os2 .
  ?os2 <ostamp-2026#derivedFrom> ?os3 .
  FILTER ( ?os1 != ?os3 )
}
ORDER BY ?os1
```

os1	os2	os3
Kali_Linux	Ubuntu	Debian

Рис. 6. SPARQL-запит визначення похідних дистрибутивів

SPARQL query: 🔍 📄 🗑️

```
SELECT ?os ?lic
WHERE {
  ?os rdf:type/rdfs:subClassOf* <ostamp-2026#OperatingSystem> .
  ?os <ostamp-2026#hasLicense> ?lic .
  FILTER ( REGEX(STR(?lic), "Proprietary") )
}
ORDER BY ?os
```

os	lic
MS-DOS	"Proprietary"^^<http://www.w3.org/2001/XMLSchema#string>
Windows_9x	"Proprietary"^^<http://www.w3.org/2001/XMLSchema#string>
Windows_NT	"Proprietary"^^<http://www.w3.org/2001/XMLSchema#string>
macOS	"Proprietary"^^<http://www.w3.org/2001/XMLSchema#string>

Рис. 7. SPARQL-запит визначення операційних систем з закритим поточним кодом

Розроблена онтологічна модель дозволяє також за допомогою іншого SPARQL-запиту відслідкувати наявність пакетів у різних дистрибутивах (рис. 8).

Розроблена онтологічна модель також дозволяє визначити який протокол використовує той чи інший мережевий компонент операційної системи. Це можна дізнатись за допомогою спеціального SPARQL-запиту (рис. 9).

SPARQL query: 🔍 📄 🗑️

```
SELECT ?os ?sw
WHERE {
  ?os rdf:type/rdfs:subClassOf* <ostamp-2026#OperatingSystem> .
  ?os <ostamp-2026#isOpenSource> "true"^^xsd:boolean .
  ?os <ostamp-2026#consistsOfSoftware> ?sw .
}
ORDER BY ?os
```

os	sw
Ubuntu	GCC

Рис. 8. SPARQL-запит визначення допоміжного ПЗ

SPARQL query: 🔍 📄 🗑️

```
SELECT ?os ?svc ?ra
WHERE {
  ?os rdf:type/rdfs:subClassOf* <ostamp-2026#OperatingSystem> .
  ?os <ostamp-2026#providesService> ?svc .
  ?svc <ostamp-2026#accessedVia> ?ra .
}
ORDER BY ?os
```

os	svc	ra
Ubuntu	sshd	SSH
Windows_NT	TermService	RDP

Рис. 9. SPARQL-запит визначення служб для віддаленого доступу у різних ОС

Для підтвердження коректності та цілісності розробленої онтології проведено її верифікацію шляхом виконання SPARQL-запитів до резонерів. Розроблені SPARQL-запити можуть бути використані для створення контексту LLM-моделей для динамічного створення тестових питань для оцінювання знань студентів. Також сучасні LLM-моделі можуть використовувати семантику розробленої онтологічної моделі для оцінювання знань студентів. Перевагою розробленої онтологічної бази знань є наявність генеалогічних зв'язків між операційними системами, чого немає у наявних репозиторіях знань — як у операційних систем в цілому [9], [10], так і серед дистрибутивів GNU/Linux [11]. Іншою перевагою розробленої онтології над наявними [8] є охоплення різних платформ, зокрема простоїв Інтернету речей (IoT), персональних комп'ютерів, серверів та хмар. У подальшому розроблена онтологічна база знань може бути інтегрована у освітнє середовище [1] для вивчення навчальної дисципліни «Операційні системи».

Висновки

В результаті проведеного дослідження здійснено систематизацію основних концепцій навчальної дисципліни «Операційні системи». Аналіз наявних публікацій та навчальних ресурсів дозволив визначити ключові архітектурні принципи, класифікацію компонентів та функціональні залежності, що стали теоретичним фундаментом для побудови онтологічної моделі.

Розроблено онтологічну базу знань у предметній області «Операційні системи» з використанням платформи Protégé. Створена модель охоплює багаторівневу ієрархію класів, що включає компоненти ОС, системне програмне забезпечення та генеалогічні зв'язки між сімействами операційних систем, забезпечуючи перехід від статичного опису до машинозчитуваної формалізації знань.

Виконано верифікацію онтології за допомогою SPARQL-запитів до резонерів. Розроблено чотири запити для верифікації онтології. З імпортованих даних визначено дистрибутив GNU/Linux який має два послідовних предка. За допомогою запитів знайдено чотири пропріетарних ОС з імпортованих даних. Визначено дві окремі гілки операційних систем за протоколами віддаленого доступу. Цей етап підтвердив цілісність та коректність розробленої моделі, а також продемонстрував її ефективність для аналізу складних взаємозв'язків, таких як спадкоємність дистрибутивів та функціональні особливості мережевих компонентів, що відкриває можливості для подальшого використання бази знань в інтелектуальних системах навчання.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

- [1] Р. Н. Кветний, Є. А. Паламарчук, О. В. Бісікало, і О. О. Коваленко, «Концепція сучасного університету на основі інструментів електронної екосистеми управління освітніми процесами JetIQ ВНТУ», *наук. доп. заг. зборам НАН України «Науково-методичне забезпечення цифровізації освіти України: стан, проблеми, перспективи»*, 18 листопада 2022 р., HNAESU, вип. 4, вип. 2, с. 1-7, Груд 2022, <https://doi.org/10.37472/v.naes.2022.4220>. [Електронний ресурс]. Режим доступу: <https://visnyk.naps.gov.ua/index.php/journal/article/view/314/>.
- [2] O. Kovalenko, Y. Palamarchuk, and R. Yatskovska, "Assessing the level of maturity of the automated management system of a higher education institution," in *2021 IEEE 16th International Conference on Computer Science and Information Technologies (CSIT)*, 2021, pp. 167-172, <https://doi.org/10.1109/CSIT52700.2021.9648663>. [Electronic resource]. Available: <https://ieeexplore.ieee.org/abstract/document/9648663>. (Scopus/WoS)
- [3] J. Soni, and J. Singh, "The impact of the semantic WEB on the internet world," *International Journal of Computer Science and Applications*, vol. 1, pp. 06-10, Apr. 2025, <https://doi.org/10.5281/ZENODO.14195716>. [Electronic resource]. Available: <https://zenodo.org/records/14195716>. (Scopus)
- [4] J. Nardi, M. Barcellos, and J. Almeida, "An Analysis of Ontologies for the Intellectual Property Domain," *Applied Ontology*, vol. 21, pp. 49-77, Mar. 2026, <https://doi.org/10.1177/15705838261421489>. [Electronic resource]. Available: <https://journals.sagepub.com/doi/10.1177/15705838261421489>. (Scopus/WoS)
- [5] W3C OWL Working Group, "OWL 2 Web Ontology Language Document Overview (Second Edition)," *W3C, Recommendation*, Dec. 2012. [Electronic resource]. Available: <https://www.w3.org/TR/owl2-overview/>. Accessed: 06.04.2026.
- [6] А. Долгий, і І. Шубін, «Дослідження сучасних методів та інструментів онтологічного інжинірингу в контексті створення інтелектуальних систем,» *Сучасний стан наукових досліджень та технологій в промисловості*, № 1 (31), с. 32-48, 2025, <https://doi.org/10.30837/2522-9818.2025.1.032>. [Онлайн]. Режим доступу: <https://journals.urau.ua/itssi/article/view/328658>.
- [7] Т. М. Басюк, Д. Г. Досин, і В. В. Литвин, *Онтологічний інжиніринг*. Львів, Україна: Вид-во Львівської політехніки, 2017. ISBN: 978-966-941-031-3. [Електронний ресурс]. Режим доступу: <https://nvd-nanu.org.ua/adf512fc-ad28-aa54-b94a-51a74e7128f2/>.
- [8] N. Hasni, and R. Bouallegue, "Ontology for mobile phone operating systems," *arXiv:1207.2606*, Jul. 2012, <https://doi.org/10.48550/arXiv.1207.2606>. [Electronic resource]. Available: <https://arxiv.org/abs/1207.2606>. Accessed: 06.04.2026.
- [9] DBpedia, *Operating system*, [Electronic resource]. Available: https://dbpedia.org/page/Operating_system. Accessed: 19.06.2026.
- [10] Wikidata, *Operating system (Q9135)*. [Electronic resource]. Available: <https://www.wikidata.org/wiki/Q9135>. Accessed: 19.06.2026.
- [11] DBpedia, *Linux distribution*. [Electronic resource]. Available: https://dbpedia.org/page/Linux_distribution. Accessed: 19.06.2026.
- [12] D. Comer, *Operating System Design: The XINU Approach*, 3rd ed. Boca Raton, FL, USA: CRC Press, 2025. <https://doi.org/10.1201/9781003597667>. [Electronic resource]. Available: <https://www.taylorfrancis.com/books/mono/10.1201/9781003597667/operating-system-design-douglas-comer>.
- [13] В. Г. Зайцев, і І. П. Дробязко, *Операційні системи*, Київ, Україна: КПІ ім. Ігоря Сікорського, 2019. [Онлайн]. Режим доступу: <https://ela.kpi.ua/handle/123456789/29600>. Дата звернення: 15.05.2026.
- [14] О. М. Спірін, «Застосування технологій віртуалізації UNIX-подібних систем у підготовці бакалаврів інформатики,» *Інформаційні технології і засоби навчання*, т. 65, № 3, с. 201-222, 2018. <https://doi.org/10.33407/itlt.v65i3.2055>. Режим доступу: <https://journal.iitta.gov.ua/index.php/itlt/article/view/2055>.
- [15] M. A. Musen, "The Protégé project: A look back and a look forward," *AI Matters*, vol. 1, no. 4, Jun. 2015. <https://doi.org/10.1145/2757001.2757003>.

Рекомендована кафедрою комп'ютерних наук ВНТУ

Дата надходження 11.05.2026

Дата прийняття до друку після рецензування 20.05.2026

Дата публікації 7.07.2026

Ця робота ліцензується відповідно до

[Creative Commons Attribution 4.0 International License](https://creativecommons.org/licenses/by/4.0/)

Малініч Ілля Павлович — асистент кафедри комп'ютерних наук. e-mail: malinich@vntu.edu.ua .
<https://orcid.org/0000-0002-5862-3732> ;

Слазін Олексій Віталійович — канд. техн. наук, доцент, доцент кафедри комп'ютерних наук, e-mail: avsilagin@vntu.edu.ua . <https://orcid.org/0009-0006-0089-4800> .

Вінницький національний технічний університет, Вінниця

I. P. Malinich¹
O. V. Silagin¹

Ontological Modeling of the Knowledge Base “Operating Systems”

In the current context of the development of higher education information ecosystems and generative artificial intelligence systems, the implementation of ontological knowledge bases has got particular importance. The application of Semantic Web technologies facilitates this process by enabling machine understanding of content, data interoperability, and seamless integration across various educational platforms. This article focuses on the ontological modeling of the "Operating Systems" subject domain as a foundation for the intellectualization of the educational process. The relevance of this research is driven by the rapidly increasing architectural complexity of modern system software, which necessitates a transition from fragmentary descriptions to machine-readable formalization. The paper outlines the advantages of utilizing the Web Ontology Language (OWL) standard to describe functional dependencies between OS components. The aim of this work is to systematize the interrelationships and genealogy of operating systems through ontological engineering. It details the process of creating a model within the Protégé environment, selected for its robust support of modern reasoners. The structure of the knowledge base encompasses OS components, system software, and the classification of operating systems. Special attention is given to modeling legacy succession and architectural forks, which are critical for understanding the evolution of modern operating systems. The practical value of this study lies in the verification of the ontology using SPARQL queries, enabling the analysis of relationships between kernel types, file systems, network stacks, and virtualization tools. This approach overcomes the limitations of static repositories and ensures the dynamic inference of new knowledge. The research results confirm the model's effectiveness for integration into the educational environment, particularly in providing context for LLMs during student knowledge assessment. This development opens up new prospects for intelligent learning support systems that transcend traditional teaching methods for the "Operating Systems" discipline, thereby granting students a more profound understanding of the architectural principles behind complex software products.

Keywords: ontology, ontological knowledge base, operating systems, system components, system software, operating systems genealogy.

Malinich Illia P. — Assistant of the Chair of Computer Science, e-mail: malinich@vntu.edu.ua .
<https://orcid.org/0000-0002-5862-3732> ;

Silagin Oleksiy V. — Cand. Sc (Eng), Associate Professor, Associate Professor of the Chair of Computer Science,
e-mail: avsilagin@vntu.edu.ua . <https://orcid.org/0009-0006-0089-4800>