



17. Положення про дистанційне навчання, затверджене наказом МОН України від 25.04.2013 № 466 // Офіц. вісн. України. – 2013. – № 36. – С. 202.

18. Garrison D.R. Theoretical Challenges for Distance Education in the 21st century: a Shift from Structural to Transactional Issues / D. Garrison // International Review of Research in Open and Distance Learning. – 2000. – Vol. 1, № 1. – P. 1–17.

19. Вишнівський В.В. Організація дистанційного навчання. Створення електронних навчальних курсів та електронних тестів : навч. посібник / В.В. Вишнівський, М.П. Гніденко та ін. – Київ : ДУТ, 2014. – 140 с.

20. Краевский В.В. Основы обучения. Дидактика и методика / В.В. Краевский, А.В. Хуторской. – М. : Академия, 2007. – 352 с.

УДК 378.016+004.82

ПРОГРАМНІ ЗАСОБИ РОЗРОБКИ ОНТОЛОГІЙ У ПРОЦЕСІ ПІДГОТОВКИ ІНЖЕНЕРІВ-ПЕДАГОГІВ КОМП'ЮТЕРНОГО ПРОФІЛЮ

Козіброда С.В., аспірант
кафедри комп'ютерних технологій
Тернопільський національний педагогічний університет
імені Володимира Гнатюка

У статті досліджено проблему вибору програмних засобів розробки онтологій у процесі підготовки інженерів-педагогів комп'ютерного профілю. Розглянуто найбільш відомі інструменти інженерії онтологій – Apollo, OntoStudio, Protégé, Swoop, TopBraid. Проаналізовано застосування онтологій комп'ютерних систем і програмних засобів їх розробки у різних сферах та виділено основні критерії вибору програмного забезпечення для професійної діяльності майбутнього інженера-педагога комп'ютерного профілю.

Ключові слова: інженер-педагог, онтологія, база знань, предметна галузь, комп'ютерні системи, програмні засоби.

В статье исследована проблема выбора программных средств разработки онтологий в процессе подготовки инженеров-педагогов компьютерного профиля. Рассмотрены наиболее известные инструменты инженерии онтологий – Apollo, OntoStudio, Protégé, Swoop, TopBraid. Проанализировано применение онтологий компьютерных систем и программных средств их разработки в разных сферах и выделены основные критерии выбора программного обеспечения для профессиональной деятельности будущего инженера-педагога компьютерного профиля.

Ключевые слова: инженер-педагог, онтология, база знаний, предметная область, компьютерные системы, программные средства.

Kozibroda S.V. SOFTWARE TOOLS OF ONTOLOGY DEVELOPMENT IN TRAINING PROCESS OF FUTURE ENGINEERS-TEACHERS

The article deals with the problem of choosing the software for computer ontologies during training and skills development of future engineers and teachers. The most famous ontology engineering tools such as Apollo, OntoStudio, Protégé, Swoop, TopBraid were characterized in the article. The use of computer systems ontologies in different areas and their software development was analyzed as the research highlights the main criteria for choosing the software for the professional activity of the future teachers of computer engineering.

Key words: engineer-teacher, ontology, knowledge base, Specialization, computer systems, software.

Постановка проблеми. Розвиток інформаційних технологій останньої чверті минулого століття характеризувався розробкою технологій баз даних і масовим їх впровадженням. На початку цього століття подібні процеси спостерігаються у розробці технологій, пов'язаних з онтологічним моделюванням. У більшості дослідників уже сформувався розуміння того, що використання бібліотек онтології в організації інформаційних процесів у найближчий час буде поширене так, як використання

баз даних зараз. У процесі підготовки інженерів-педагогів комп'ютерного профілю «онтологічні» методи, як правило, застосовуються для побудови моделей процесів. Інженерна модель процесу і є онтологією, точніше, вона є описом цієї моделі. Такі описи моделей зазвичай формальні, тобто створені на мові, спеціально розробленій для цієї мети, конструкції якої завжди інтерпретуються точно і однозначно.

Аналіз останніх досліджень і публікацій. У процесі підготовки інженерів-педаго-



гів у галузі комп'ютерних технологій у ВНЗ значне місце посідає вивчення інтелектуальних систем, які розглядали В.В. Литвин, В.В. Пасічник, Ю.В. Яцишин [3], де онтології використовуються для формальної специфікації понять і зв'язків, що притаманні певній сфері знань. Оскільки комп'ютер не може розуміти, як людина, стан речей у світі, то йому необхідно подавати всю інформацію у формальному вигляді, тому побудова онтології не завжди є кінцевою метою, переважно її використовують в інших інтелектуальних системах для вирішення практичних цілей – на цьому ґрунтуються дослідження С. Ніренбурга і В. Раскіна [7]. Детально процес розробки і використання онтології в загальному вигляді розглянуто в роботі Н. Ноя [8]. Проблеми онтологій і використання їх у комп'ютерних системах досліджував В. Лапшин [2]. Методику створення інтерфейсу на основі онтології у середовищі WEB-порталу вивчали О.Є. Стрижак, М.А. Попова, К.В. Ляшук [5]. Оглядом інструментів інженерії онтологій займалися О.М. Овдей, Г.Ю. Проскудіна [4].

У сучасних умовах інженер-педагог у галузі комп'ютерних технологій повинен чітко розуміти методику розробки онтологій з тієї чи іншої предметної галузі. Та для ефективного загального використання інформації йому потрібно розв'язати певні технічні проблеми. Зокрема, це вибір програмних засобів розробки онтологій, оскільки в останні роки кількість програмних засобів для розробки та використання онтологій значно збільшилась. Тому в деяких ситуаціях на різних стадіях розробки доцільно було би використовувати інше програмне забезпечення, а от де краще застосовувати той чи інший програмний засіб, практично ніде не розглядається.

Постановка завдання. Метою статті є обґрунтування вибору ефективних програмних засобів розробки і використання онтологій у процесі підготовки та майбутньої діяльності інженерів-педагогів комп'ютерного профілю.

Інженерно-педагогічна освіта як складова частина професійно-педагогічної стає однією із ключових ланок, яка визначає якість і перспективи розвитку всієї системи освіти.

Концепція навчання інженерів-педагогів характеризується глибокою інтеграцією інженерної, виробничої та психолого-педагогічної підготовки. Майбутній інженер-педагог повинен бути готовим до виконання різних професійно-педагогічних видів діяльності, таких як професійне навчання; методична та виробничо-технологічна робота; розвиток автоматизації; удосконалення техніки і

технологій (ІКТ), що впливає зі структури й інтеграційної природи власне інформаційно-комунікаційних технологій. А володіння компонентним складом педагогічних та інформаційних знань про онтології є основою для формування професійних умінь та забезпечує відповідність рівня професійної підготовки майбутніх фахівців щодо створення онтології потребам і вимогам ринку праці. Усе це передбачає інтеграцію в особі викладача професійно-технічної освіти загальної і професійно-педагогічної культури [1, с. 8].

Т.Є. Фіногеева вважає, що інженерно-педагогічна освіта є синтезом двох основних рівноцінних компонентів – гуманітарно-соціального і професійно-технічного. Змістом гуманітарно-соціальної, тобто педагогічної діяльності інженера-педагога, підготовленого в системі професійної освіти, є професія як відносно постійний вид трудової діяльності. Він характеризується спеціальними знаннями й уміннями, а також способами і характером взаємодії людини з різноманітними технологіями, тобто це діяльність, спрямована на реалізацію педагогічної, спеціально організованої взаємодії. Якісною оцінкою цієї діяльності є вміння організувати цей процес відповідно до педагогічної та виробничої технології та якомога точніше її реалізувати [6, с. 123].

Окрім того, враховуючи зміст умінь інженера-педагога в галузі комп'ютерних технологій, варто зазначити, що він може застосовувати онтології комп'ютерних систем у різних напрямках, таких як:

- системи нормативно-довідкової інформації (НДІ) – ведення словників, довідників і класифікаторів та підтримка системи кодування об'єктів обліку. Онтологія забезпечує неконфліктне накопичення будь-якої кількості інформації у стандартній структурі класифікації;

- моделювання організаційної структури підприємств – онтологічне уявлення знань про суб'єкти економічної діяльності, які входять до складу будь-якої системи, для об'єднання їх інформаційних ресурсів у єдиний інформаційний простір;

- семантична розмітка тексту – користувач може генерувати і зберігати релевантні знання, необхідні для різних завдань (ці системи орієнтовані на інженера, якому треба обробляти великий обсяг інформації, витягувати структуровану інформацію);

- класифікація товарів і послуг – забезпечує наявність стандартизованого представлення інформації; відбувається систематизація понять у сфері бізнесу, впорядкування їх описів;

- система питання-відповіді – видає інформацію за запитом, у видачі відповідей



система враховує закладені в неї відомості про навколишню дійсність: набір понять, відносини між ними, обмеження на відносини і список конкретних екземплярів;

– обробка природної мови – використовується у процесі побудови вибірки визначень бази довідкового матеріалу для складних процедур обробки природної мови, для розділення галузевих і загальноприйнятих знань, що полегшує вилучення інформації з різних джерел, звужує пошукові запити і поліпшує якість виданих результатів;

– інтерфейс – дає змогу візуалізувати у легкодоступній наочній формі результат процесів інтеграції та агрегації розподілених інформаційних ресурсів у процесі організації взаємодії користувачів;

– штучний інтелект – для формальної специфікації понять і зв'язків, які властиві певній сфері знань.

Через це вибір програмних засобів для розробки онтології є важливим чинником у процесі навчання і майбутньої діяльності інженерів-педагогів комп'ютерного профілю, оскільки побудова онтологій – процес складний і займає багато часу. Щоб полегшити його, в середині 90-х років почали створюватися перші середовища для процесу розробки онтологій. Вони забезпечили інтерфейси, які дали змогу виконувати концептуалізацію, реалізацію, перевірку неузгодженості і документування. За останні роки число інструментів онтологій різко зросло (сайт консорціуму W3C, наприклад, надає список понад 50 інструментів редагування).

Інженерію онтологій можна визначити як сукупність дій, що стосуються процесу розробки онтологій; життєвого циклу онтологій; методів і методологій побудови онтологій; набору інструментів і мов для їх побудови і підтримки [4].

Для створення і підтримки онтологій існує ціла низка інструментів, які, крім загальних функцій редагування і перегляду, виконують підтримку документування онтологій, імпорт і експорт онтологій різних форматів і мов, графічне редагування, управління бібліотеками онтологій тощо.

Розглянемо найбільш відомі інструменти інженерії онтологій.

Apollo є зручним для користувача додатком моделювання знань. Він дає змогу користувачеві моделювати онтологію з основних примітивів, таких як класи, екземпляри, функції, відносини тощо. Внутрішня модель – каркасна система на основі OKBC (Open Knowledge Base Connectivity protocol). База знань складається з ієрархічної організації онтологій, які можуть бути успадковані від інших і використані в тому

разі, якщо вони походять від кореневої. Кожна онтологія за замовчуванням включає в себе всі примітивні класи. Кожен клас може створити кілька примірників і екземплярів, що успадковують всі властивості класу. Apollo не підтримує режиму діаграми і розрахований на багато користувачів та можливість вилучення інформації для спільної обробки. Має потужний функціонал перевірки узгодженості, зберігання онтології (тільки XML-файли), а також імпорту та експорту форматів (plug-in для CLOS і OCML). Apollo реалізований в Java [2, с. 176].

OntoStudio побудований на базі IBM Eclipse. Його можна завантажити для безкоштовної оцінки у демо-версії. Завдяки інструменту Environment здійснюється розвиток і підтримка онтологій за допомогою графічних засобів. Він заснований на клієнт/серверній архітектурі, де онтологія регулюється центральним сервером. Різні клієнти можуть отримувати доступ і змінювати ці онтології відповідно до їхніх потреб і діяльності. OntoStudio підтримує багатомовний розвиток, тоді як модель знань в основному пов'язана зі структурою мов. Цей засіб підтримує спільну розробку онтологій. OntoStudio побудований на вершині потужної внутрішньої моделі онтологій. Інструмент дає змогу користувачеві редагувати ієрархію понять або класів. Він заснований на відкритій plug-in структурі. Внутрішня модель цих представлень може бути експортована в DAML + OIL, RDF(S), F-Logic і OXML. Крім того, онтології можуть бути експортовані реляційними базами даних через JDBC (Java DataBase Connectivity). OntoStudio може імпортувати зовнішнє представлення даних в DAML + OIL, Excel, F-Logic, RDF(S), схеми баз даних (Oracle, MS-SQL, DB2, MySQL, і OXML). Також він може імпортувати і експортувати файли OWL (Web Ontology Language). Це надає API (Application Programming Interface) можливість для доступу до онтологій об'єктно-орієнтованим способом. Значення реалізації за замовчуванням записується API онтології в основній пам'яті, а додатковий API існує для постійного зберігання. Присутній також механізм логічного висновку, який використовує OntoStudio – це OntoBroker (OntoBroker є результатом кількох років досліджень і має стати комерційним продуктом). Застосовуючи цей двигун, OntoStudio використовує всю ефективність F-Logic і може представляти виразні правила. OntoStudio підтримує спільні онтології за допомогою сервера OntoBroker Enhancement Collaborative.



Protйгй – це вільний, відкритий редактор онтологій і фреймворк для побудови баз знань. Платформа Protйгй підтримує два основних способи моделювання онтологій за допомогою редакторів Protйгй-Frames і Protйгй-OWL. Онтології, побудовані в Protйгй, можуть бути експортовані в безліч форматів, включаючи RDF (RDF Schema), OWL і XML Schema. Protйгй має відкриту архітектуру, яка легко піддається модернізації за рахунок підтримки модулів розширення функціональності та підтримується значним співтовариством, що складається з розробників і вчених, урядових і корпоративних користувачів, котрі використовують його для вирішення завдань, пов'язаних зі знаннями в таких різноманітних галузях, як біомедицина, збір знань і корпоративне моделювання. Protйгй OWL дає змогу описувати не тільки поняття, а й конкретні об'єкти та має багатий набір операторів (перетин, об'єднання, заперечення тощо). Він заснований на логічній моделі, яка дає змогу створювати визначення, відповідні неформальному опису. Таким чином, визначення складних понять можуть бути побудовані на основі простіших визначень. Крім того, логічна модель дає змогу з'ясувати, які концепції відповідають заданим визначенням, та перевірити, чи поняття і визначення в онтології взаємно узгоджуються [3, с. 313].

TopBraid – редактор онтологій, поставляється в трьох версіях:

- Free Edition (FE), який є базовою версією тільки з основним набором засобів.

- Standard Edition (SE), який включає в себе всі функції FE та функцію графічного перегляду, імпорту об'єктів, розширену підтримку рефакторингу і багато іншого.

- Maestro Edition (ME), який включає в себе всі функції SE та підтримку TopBraid Live, EVN є ключем для спільної роботи, а SPARQL містить багато інших користувацьких функцій.

Компонент TopBraid Suite є професійним інструментом розробки для семантичних моделей (онтологій). Він заснований на технології Eclipse і платформі API Jena. Це повний редактор для RDF (S) і OWL моделей, а також платформи для інших RDF на основі компонентів та сервісів. Також він може завантажувати і зберігати будь-який файл OWL2 в таких форматах, як RDF / XML або Turtle та підтримує різні міркування і послідовності для перевірки механізмів. Послідовність перевірки і налагодження підтримується за допомогою вбудованого OWL механізму логічного висновку, SPARQL (Protocol and RDF Query Language) – двигун запитів і правил опису OWL, логіка підтри-

мується за допомогою вбудованих OWL DL двигунів, таких як OWLIM. Той же FE також підтримує висновок SPARQL Notation (SPIN). З іншого боку, SPIN можуть бути використані для визначення обмежень цілісності, що призначені для виділення недоступних даних у момент редагування.

Swoor є відкритим вихідним кодом, веб-редактором OWL онтологій та браузером, що містить перевірку OWL і пропонує синтаксис представлення різних OWL-переглядачів (абстрактний синтаксис, N3 і т.д.). Він використовує міркування (RDFS-like і Pallet) підтримує (OWL Inference Engine), а також забезпечує множинне онтологічне середовище, в якому сутності і зв'язки з різних онтологій можна порівнювати, редагувати і об'єднувати безкоштовно. Різні онтології можна порівнювати за описом їх визначень на основі логіки, пов'язаної з використанням екземплярів та їх властивостей. Навігація може бути простою і легкою завдяки можливості гіперпосилань в інтерфейсі Swoor. Цей програмний засіб дає змогу не дотримуватись методології побудови онтологій, оскільки користувачі можуть повторно використати зовнішні онтологічні дані або імпортувати всю зовнішню онтологію, проте не можуть зробити частковий імпорт OWL, хоча можна шукати концепції по декількох онтологіях. Swoor застосовує алгоритми пошуку онтологій, які поєднують у собі ключові слова з DL на основі конструкцій, щоб знайти пов'язані з ним концепції в онтологіях. Цей пошук здійснюється за всіма онтологіями, що зберігаються в базі знань Swoor [2, с. 167].

Виходячи зі змісту зазначених вище умінь інженера-педагога в галузі комп'ютерних технологій та проаналізувавши застосування онтологій комп'ютерних систем у різних сферах, а також програмних засобів розробки цих онтологій, виділимо три основні критерії вибору програмного забезпечення для професійної діяльності у тій чи іншій сфері.

1. Архітектура програмного забезпечення та розвиток інструментів (таблиця 1) – містить інформацію про необхідні платформи для використання інструменту. Важливою є така інформація, як архітектура за замовчуванням (автономна робота, тип роботи клієнт/сервер, багатоланковість додатків), розширюваність, зберігання онтологій (бази даних, файли ASCII і т.д.) і управління резервним копіюванням – всі ці інструменти пристосовані до Java-платформ.

2. Функціональна сумісність (таблиця 2) – містить інформацію про інструменти і



взаємодію з іншими мовами та інструментами розробки онтологій, переклад з деяких мов онтологій. Це ще одна важлива особливість інтеграції онтологій в додатках та вибору програмних засобів створення онтологій у професійній діяльності майбут-

нього інженера-педагога комп'ютерного профілю. Більшість із цих інструментів підтримує імпорт та експорт із багатьох мов у різні формати.

3. Зручність і простота (таблиця 3) – включає в себе роботу з графічними редак-

Таблиця 1

Архітектура та розвиток інструментів

Характеристика	Apollo	OntoStudio	Protégé	Swoop	TopBraid
Архітектура	Автономна	клієнт/сервер	Автономна, клієнт/сервер	Web-бази, клієнт/сервер	Автономна, базується plug-in
Розширюваність	Додаткові модулі (Plug-ins)	Додаткові модулі (Plug-ins)	Додаткові модулі (Plug-ins)	Технологія Via, Додаткові модулі (Plug-ins)	Додаткові модулі (Plug-ins)
Зберігання онтологій	У файлі	У СУБД (системи управління базами даних)	У файлі та СУБД (системи управління базами даних)	Як HTML-модель	У СУБД (системи управління базами даних)
Управління резервним копіюванням	Немає	Немає	Немає	Немає	Немає

Таблиця 2

Функціональна сумісність програмних засобів

Характеристика	Apollo	OntoStudio	Protégé	Swoop	TopBraid
Взаємодія з іншими засобами розробки онтологій	Немає	OntoAnnotate, OntoBroker, OntoMat, Semantic and Miner	PROMPT, OKBC, JESS, FaCT and Jena	Немає	Sesame, Jena and Allegro Graph
Імпорт у мови	Apollo Meta Language	XML(S), OWL, RDF(S), UML діаграми, бази даних схеми (Oracle, MS-SQL, DB2, MySQL), Outlook, файлові системи Метадані і віддалений OntoBroker	XML(S), RDF(S), OWL, (RDF, UML, XML) backend, Excel, BioPortal and DataMaster	OWL, XML, RDF та текстові формати	RDFa, WOL, RDF(s), XHTML, Microdata та RDFa, Data sources, SPIN, News Feed, RDF Files into a new TDB, Email and Excel
Експорт у мови	OCML та CLOS	XML(S), OWL, RDF(S), UML та OXML	XML(S), RDF(S), OWL, Clips, SWRL-IQ, Instance Selection, Meta-Analysis, OWLDoc, Queries and (RDF, UML, XML)	RDF (S), OIL та DAML	Merge / Convert RDF Graphs, RDF(S), WOL

Таблиця 3

Зручність і простота програмних засобів

Характеристика	Apollo	OntoStudio	Protégé	Swoop	TopBraid
Графічна система	-	+	+	+	-
Багатокористувацький інтерфейс	-	+	+	+	+
Бібліотеки онтологій	+	+	+	-	+
Функція допомоги в інтерфейсі	+	+	+	-	+



торами, спільну роботу декількох користувачів і необхідність надання багаторазового використання бібліотек онтологій. Для більшості користувачів Protйгй забезпечує дружній і простий у використанні графічний інтерфейс. Крім того, в Protйгй і OntoStudio використовують макет інтерфейсу і візуалізації онтології, що значно полегшує створення онтології майбутнім інженером-педагогом. Protйгй і OntoStudio дають змогу графічно переглянути таксономію. Допомога системи також має важливе значення для користувачів і повинна бути легкодоступною і простою у використанні. Система допомоги Apollo, Protйгй, OntoStudio і TopBraid Composer (FE) складається з довідки та керівництва користувача. Swoor не забезпечують функцію допомоги в інтерфейсі. Співпраця має важливе значення у процесі побудови як простих, так і складних онтологій, тому Protйгй, OntoStudio і TopBraid Composer (FE) дозволяє спільну побудову онтологій. Swoor дає можливість користувачам писати і ділитися анотацією на будь-яку онтологічну сутність.

Висновки з проведеного дослідження. Таким чином, різноманітність інструментів для побудови та об'єднання онтологій робить складним їх безпосереднє порівняння. Фактично, коли потрібно вирішити питання, який інструмент є найбільш зручним для створення онтології майбутнім інженером-педагогом комп'ютерного профілю, все буде залежати від конкретного завдання і напряму застосування онтології. Також слід зазначити, що інструменти загалом схожі за своєю функціональністю, але мають відмінності за внутрішнім способом побудови бази знань. Вибір платформи, що недостатньо підтримує відкриті стандарти представлення онтологій (незважаючи на відкритість самої платформи), може завадити подальшому розвитку й інтеграції семантичних додатків, розроблених на

її основі. Вибір найзручнішого інструменту залежить насамперед від цілей розробника і розроблюваної онтології, тому варто звертати увагу на такі критерії вибору програмного забезпечення, як архітектура та розвиток інструментів, функціональна сумісність програмних засобів, зручність і простота програмних засобів.

ЛІТЕРАТУРА:

1. Коваленко О.Е. Концепція професійно-педагогічної підготовки студентів інженерно-педагогічних спеціальностей / О.Е. Коваленко, Н.О. Брюханова, О.О. Мельниченко // Проблеми інженерно-педагогічної освіти. – Харків, 2005. – Вип. 10. – С. 7–20.
2. Лапшин В.А. Онтологии в компьютерных системах / В.А. Лапшин. – М.: Научный мир, 2010. – 222 с.
3. Литвин В.В. Интеллектуальные системы / В.В. Литвин, В.В. Пасічник, Ю.В. Яцишин. – Львів: Новий Світ – 2000, 2013. – 406 с.
4. Овдей О.М. Обзор инструментов инженерии онтологий / О.М. Овдей, Г.Ю. Проскудина // Журнал ЭБ. – 2004. – № 4. [Електронний ресурс] – Режим доступу: <http://www.elbib.ru/index.phtml?page=elbib/rus/journal/2004/part4/op>
5. Стрижак О.Є. Методика створення онтологічного інтерфейсу у середовищі WEB-порталу / О.Є. Стрижак, М.А. Попова, К.В. Ляшук // Радіоелектронні і комп'ютерні системи. – 2014. – № 2. – С. 78–84.
6. Фіногєєва Т.Є. Формування особистості майбутніх інженерів-педагогів у процесі самостійної роботи з педагогічних дисциплін / Т.Є. Фіногєєва // Теорія і практика управління соціальними системами. – 2014. – № 1. – С. 121–131.
7. Nirenburg S. Ontological Semantics / S. Nirenburg, V. Raskin – Cambridge: MA, 2004. – 240 p.
8. Noy N. Ontology Development 101: A Guide to Creating Your First Ontology. // Stanford Knowledge Systems Laboratory Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report SMI – 2001 – 0880, March 2001 / N. Noy, D. L. McGuinness. [Електронний ресурс] – Режим доступу: http://protege.stanford.edu/publications/ontology_development/ontology101.pdf – Назва з екрана.