

УДК 004.942:378.14

Литвак Олена Геннадіївна

старший викладач кафедри інформаційних технологій
Донецький державний університет управління, м. Донецьк, Україна
alfff@yandex.ru

Лашина Марія

студентка 3-го курсу спеціальності «Маркетинг»
Донецький державний університет управління, м. Донецьк, Україна
l.merii.san@gmail.com

МЕТОДИКА ВИКЛАДАННЯ ОНТОЛОГІЧНОГО ІНЖИНИРІНГУ ДЛЯ СТУДЕНТІВ ГАЛУЗЕЙ «ЕКОНОМІКА» І «МЕНЕДЖМЕНТ»

Анотація. Актуальність матеріалу, що подається в статті, обумовлена існуванням проблеми повторного використання знань на підприємствах. Для успішного розв'язання цієї проблеми необхідно, щоб спеціалісти самих різних напрямів (часто не пов'язаних з інформаційними технологіями) володіли основами онтологічного інжинирінгу знань. Дисципліни, що містять основи інжинирінгу знань та інших інформаційних технологій, студентами нетехнічних напрямів сприймаються важко. У статті на основі експерименту, проведеного в Донецькому державному університеті управління, надається методика викладання онтологічного інжинирінгу знань для студентів галузей «Економіка» і «Менеджмент». Доведено, що розроблена методика значно підвищує рівень сприйняття матеріалу студентами, сприяє розвитку аналітичного і творчого мислення.

Ключові слова: бази знань; онтології; онтологічний інжинирінг; бази даних; реляційна модель; апарат логічного висновку.

1. ВСТУП

У 90-х роках 20-го сторіччя в великих корпораціях з'явилась нова концепція – управління знаннями (knowledge management). Появі цієї концепції сприяла проблема повторного використання інформації, накопиченої спеціалістами компаній. Інформація має не тільки зберігатися, а й бути доступною для перегляду, вивчення та використання всіма співробітниками. Для розв'язання цієї проблеми на допомогу приходять інформаційні системи на основі баз даних. Але цього недостатньо. У великих масивах інформації людина не має можливості для пошуку прихованих закономірностей, взаємозв'язків, породження нової інформації на основі тієї, що існує. Попри це, існує велика кількість інформації, актуальної для роботи підприємства, яку неможливо структурувати за допомогою реляційної моделі (баз даних). Наприклад, це знання з предметних галузей, якими володіють економісти, інженери-технологи, маркетингологи та інші спеціалісти. Для успішної роботи з такими видами знань необхідні інші засоби формалізації цих знань. Одним із найпопулярніших засобів формалізації знань нині є онтологічний інжинирінг (ontological engineering) [1]. Онтологія – це структурна специфікація предметної галузі, її формальне представлення, яке включає словник термінів предметної галузі й логічні вирази, які описують, як саме ці терміни співвідносяться один з одним. Онтологічний інжинирінг – це сам процес побудування такої формальної специфікації предметної галузі, який фактично є ядром управління знаннями.

Постановка проблеми. Управління знаннями не може відбуватися без залучення спеціалістів із предметних галузей – носіїв тих знань, які мають потрапити до системи. На жаль, економісти, менеджери, маркетингологи, інженери-технологи та інші спеціалісти

не завжди мають високий рівень підготовки щодо інформаційних технологій взагалі і щодо онтологічного інжинирінгу знань зокрема.

Теорія і практика онтологічного інжинирінгу для студентів нетехнічних спеціальностей викладається в курсі «Мовної моделі сучасного інформаційного простору». Її сприйняття навіть студентами спеціальності «Інформатика» пов'язано з багатьма складнощами. Важко сприймається, у першу чергу, теорія дескрипційної логіки, на якій базується онтологічний інжинирінг.

Метою статті є обґрунтувати доцільність використання методики викладання основ онтологічного інжинирінгу для студентів напрямів «Економіка» і «Менеджмент», яка швидко надає необхідні практичні навички й уникає заглиблення в складну теорію.

Аналіз публікацій. Для сприйняття курсу «Мовна модель сучасного інформаційного простору», попри елементарні навички роботи за комп'ютером, студентам слід мати уявлення про основи роботи реляційних баз даних. Специфіка викладання реляційної моделі даних для студентів галузі «Економіка» і «Менеджмент» викладено в [2]. Щодо онтологічного інжинирінгу, то спеціальної навчальної літератури існує небагато. Більша частина написана англійською мовою або використовує термінологію дескрипційної логіки [1, 3], яка дуже важко сприймається студентами і спеціалістами нетехнічних напрямів. Досить привабливою з точки зору сприйняття студентами є методичний посібник Д. І. Муромцева [4], але він використовує редактор онтологій Protege 3.1, оснований на фреймовій моделі. Ця гілка розробки програмного забезпечення на теперішній час припинена і визнана застарілою технологією, тому її використання не є доцільним у навчальному процесі.

2. РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

2.1. Аналіз основних проблем сприйняття основ онтологічного інжинирінгу

Під час перевірки знань студентів шляхом диференційованого заліку було виявлено наявність певної проблеми в розумінні дисципліни багатьма студентами. Етапи створення ієрархічного дерева класів в онтології і розподілення екземплярів за класами, як правило, складності собою не представляють. Студенти також швидко набувають елементарних навичок користування редактором онтологій. Але після отримання цих знань вони починають сприймати онтологію як різновид реляційної бази даних. Оскільки онтологічна модель теж дозволяє зберігати дані і складати запити до них, студентам не зрозуміло, навіщо потрібна ще одна модель, яка має ті самі можливості, що й реляційна

Отже, критеріями успішного вивчення курсу є володіння логікою моделювання знань, поняттям про логічний висновок і чітке бачення різниці між двома основними моделями збереження інформації – онтологічною і реляційною.

2.2. Методика викладання курсу «Мовна модель сучасного інформаційного простору» для студентів нетехнічних галузей

Оскільки однією з основних вимог сучасної вищої освіти є адаптація змісту курсів за вибором під основний напрям навчання, то, на перший погляд, вважається доцільним обирати для моделювання інформацію професійної спрямованості. Але практика показала, що студенти другого, третього курсів, по-перше, ще не володіють достатнім рівнем знань з основних предметів за своїм напрямом, по-друге, далеко не завжди посправжньому зацікавлені своїм основним напрямом. Тому онтологічні моделі, які вони

намагаються розробити, виходять сильно обмеженими, «бідними». Часто студенти взагалі не знають, яку ще інформацію можливо додати до онтології.

Тому було прийнято рішення провести експеримент зі зміни тактики навчання. У ході експерименту було запропоновано виконання такого завдання.

1. Студент самостійно за власним бажанням обирає тему для моделювання онтології. Тема не обов'язково повинна мати зв'язок з основним напрямом навчання, але студент має глибоко нею володіти, бути «експертом» з цієї теми. Оскільки багато студентів мають хобі, то вибір теми для більшості дається легко. І практика показує, що теми, обрані студентами, досить далекі від напряму навчання.

2. Студент розробляє «чорновий» варіант онтології виключно у письмовій формі. При цьому онтологія повинна містити:

- не менше трьох рівнів субкласів;
- не менше двох класів верхнього рівня;
- не менше двох об'єктних властивостей;
- не менше п'яти властивостей простих типів;
- не менше двох правил логічного висновку;

3. Онтологія представляється викладачеві на розгляд.

4. Якщо викладач не знаходить зауважень до онтології, то студенту дозволяється створювати онтологію в редакторі Protégé 4.3.

5. Студенту пропонується розробити реляційну модель для зберігання тих же самих даних, що містяться в онтології.

6. Пропонується зробити порівняльний аналіз реляційної бази даних і онтології. Процес порівняння й висновки студент повинен викласти у звіті.

Розглянемо процес виконання такого завдання на прикладі роботи студентки 3-го курсу напряму «Маркетинг». За тему роботи було обрано управління даними у сфері виробництва аніме (японська мультиплікація) і супутньої продукції. Отже, текстовий опис інформації, яка повинна бути врахована в інформаційних системах буде таким.

Існують три види продукції, які потенціально можуть бути пов'язані між собою. Це аніме, манга (японській різновид коміксів) та комп'ютерна гра. Існують компанії, які працюють у сфері виробництва цих трьох (або менше) типів продукції. Утім, є четвертий тип продукції, який пов'язаний з попередніми – це музика, яка пишеться для супроводження аніме або комп'ютерної гри.

Існують три типи компаній: видавництво (яке друкує мангу), студія (яка виробляє аніме) та розробник (який розробляє комп'ютерну гру). При цьому компанія може належати одночасно до декількох типів. Кожне аніме має одного продюсера. Кожна манга або музика має автора або декількох авторів.

Будь-яке аніме, манга чи комп'ютерна гра можуть бути зроблені на основі іншого твору (аніме, манги чи комп'ютерної гри). Наприклад, спочатку була розроблена комп'ютерна гра, потім на її основі було знято аніме, пізніше з'явилося його продовження (інше аніме), а через деякий час аніме було адаптовано під мангу. Або навпаки, спочатку було надруковано мангу, потім на її основі було знято аніме, а ще пізніше розроблено комп'ютерну гру. У сталій термінології попередній продукт називається пріквелом, а наступний – сіквелом.

Необхідно розробити й реалізувати у відповідних програмах дві моделі для збереження і повторного використання цих даних. Одна модель має бути основана на реляційній теорії, а друга – на онтологіях. Обидві моделі – і онтологічна, і реляційна – мають надавати можливість збереження всіх наведених вище даних, включаючи зв'язки між ними, додавати нові дані, робити добір даних за всіма можливими критеріями, як то: визначати компанію, продюсера та авторів за назвою продукту, визначати продукти за назвою компанії, визначати роботи авторів за іменем авторів, визначати типи

продуктів за їх назвою, визначати типи компанії за назвою компаній, визначати всі продукти, на основі яких було зроблено даний продукт (пріквели), визначати всі продукти, які було зроблено на основі даного продукту (сіквели).

Для реалізації задачі на основі реляційних баз даних розроблено проект бази даних, що показана на рис. 1. База даних знаходиться в третій нормальній формі [5], що означає можливість звертання до неї з будь-якими SQL-запитами.

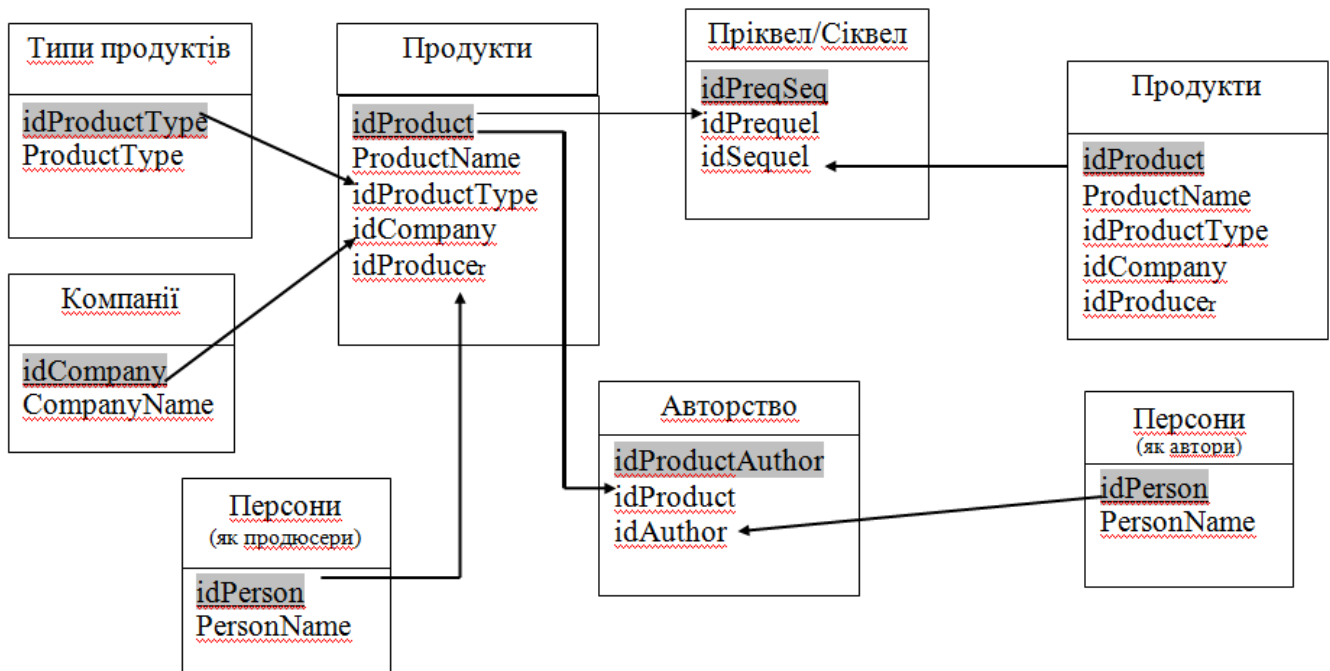


Рис. 1. Схема даних реляційної моделі системи

Для реалізації онтології було обрано редактор Protege 4.3. В основі онтологічної моделі лежать поняття класів, відношень, атрибутів та екземплярів [6]. Екземпляри є найнижчим рівнем онтології – рівнем фізичних об'єктів. У нашому випадку до екземплярів будуть відноситися конкретні компанії-виробники, їхні співробітники, конкретні зразки продукції, яку компанії виробляють. Класи – абстрактні групи екземплярів чи інших класів, які утворюються за певною ознакою. Наприклад, може існувати клас «Аніме», а його конкретним екземпляром буде стрічка «Сен і Тіхіро в полоні у духів». Може існувати клас «Компанія» і його конкретний екземпляр студія «Гіблі». Відношення віддзеркалюють зв'язки, що існують між класами й екземплярами класів. Наприклад, кожен екземпляр класу «Аніме» пов'язаний відношенням «вироблено» з екземпляром класу «Компанія». Якщо в базі знань зберігається ствердження <«Сен і Тіхіро в полоні у духів», «вироблено», «Гіблі»>, то це означає, що студія «Гіблі» є виробником стрічки «Сен і Тіхіро в полоні у духів». Атрибутами можуть виступати такі риси класів, які зберігаються не у вигляді екземплярів інших класів, а у вигляді простих типів – назви, дати народження, числові характеристики тощо. Наприклад, аніме може мати атрибут «довжина стрічки».

Дерево класів онтології виробництва аніме зображено на рис. 2.

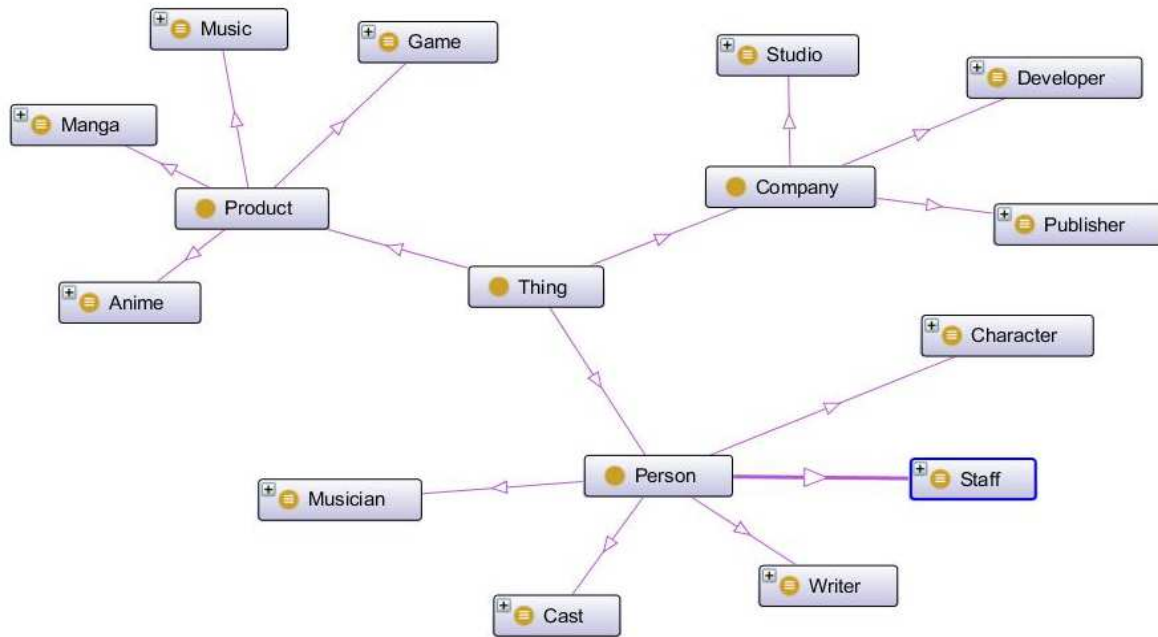


Рис. 2. Дерево класів онтології

Онтологія складається з трьох основних класів – Product (продукція), Company (компанія), Person (персона). Кожен клас розділяється на субкласи. Оскільки продукт може бути аніме, мангою, грою або музикою, то клас Product складається з чотирьох неперетинних субкласів:

$$\begin{aligned}
 Product &= Anime \cup Manga \cup Music \cup Game; \\
 Anime \cap Manga &= \emptyset; \\
 Anime \cap Music &= \emptyset; \\
 Anime \cap Game &= \emptyset; \\
 Manga \cap Music &= \emptyset; \\
 Manga \cap Game &= \emptyset; \\
 Music \cap Game &= \emptyset;
 \end{aligned}$$

Клас Person складається із субкласів Musician (музиканти), Cast (актори), Writer (письменники) та Staff (продюсери). Але ці класи можуть бути перетинними, тому що одна людина може виступати як у ролі актора, так і в ролі музиканта, наприклад.

Клас Company теж складається з трьох субкласів, які можуть перетинатися – Studio (студія), Developer (розробник) та Publisher (видавництво).

Між класами Developer і Game існує відношення developed і зворотне – isDevelopedBy. Між класами Studio й Anime – відношення made і зворотне – isMadeBy. Між класами Writer і Manga – відношення wrote і зворотне isWrittenBy. Між класами Musician і Music – відношення sang та зворотне isSungBy (рис. 3).

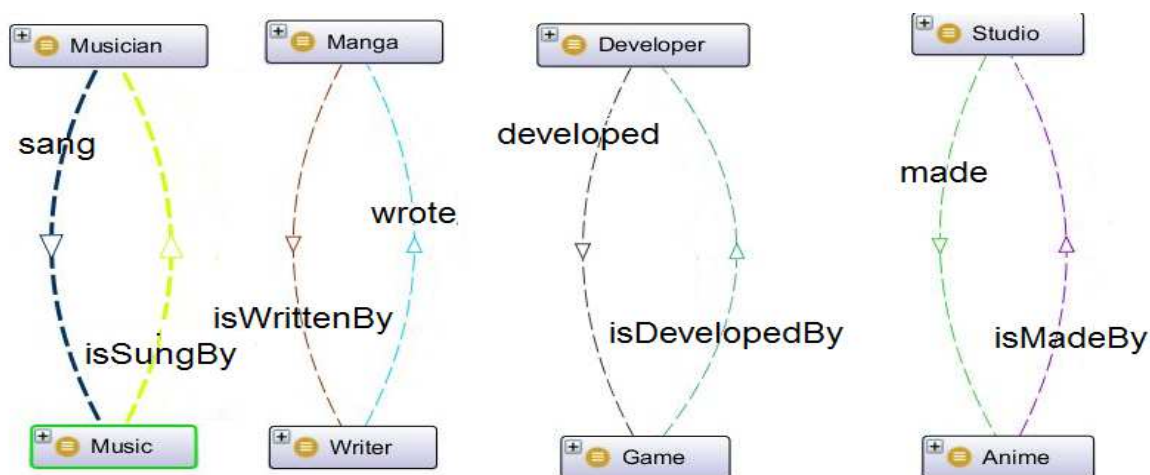


Рис. 3. Відношення між класами

Додамо чотири аксіоми еквівалентності:

- ✓ *Writer: wrote some Manga*; (Письменник – це той, хто написав будь-яку мангу);
- ✓ *Studio: made some Anime*; (Студія – це те, що зробило будь-яке аніме);
- ✓ *Developer: developed some Game*; (Розробник – це те, що розробило будь-яку гру);
- ✓ *Musician: sang some Music*; (Музикант – це той, хто написав будь-яку музику).

Між екземплярами об'єднання класів Anime, Manga, Game встановимо відношення *hasSequel* (має продовження) і зворотне йому *hasPrequel* (має приквел). Обидва відношення створимо як транзитивні. Тобто, якщо a_1, a_2, a_3 є екземплярами класу Anime, та відомо що $\langle a_1, hasSequel, a_2 \rangle$ і $\langle a_2, hasSequel, a_3 \rangle$ (тобто a_1 має продовження a_2 і a_2 має продовження a_3), то звідси однозначно прямує, що $\langle a_1, hasSequel, a_3 \rangle$ і $\langle a_3, hasPrequel, a_1 \rangle$.

Слід пам'ятати, що на цьому в проектуванні онтології не обов'язково ставити крапку. У процесі роботи можуть з'являтися додаткові класи та інші відношення. Розглянемо, як розв'язуються поставлені задачі в обох моделях. За нашим припущенням аніме має лише одного продюсера. У реляційній моделі це віддзеркалено зв'язком типу «один до багатьох» між таблицями «Персони» і «Продукти» за ключем «idPerson». В онтологічній моделі між класами Staff й Anime слід ввести відношення *isProducerOf* і зворотне *hasAsProducer*. Відношення *hasProducer* слід оголосити функціональним (це забезпечить наявність лише одного продюсера), тоді відношення *hasAsProducer* стає зворотньофункціональним. Якщо раптом виникне необхідність змінити початкові умови і поставити одному аніме у відповідність більше ніж одного продюсера, то в реляційній моделі буде необхідним перепроєктування бази даних для створення зв'язку типу «багато-до-багатьох» зі складною операцією перенесення даних у нову таблицю.

Ситуація, коли в онтологію потрапляє два ствердження $\langle p001, isProducerOf, a001 \rangle$ та $\langle p002, isProducerOf, a001 \rangle$ на відміну від реляційної бази зовсім не є неможливою. У цьому випадку онтологія здійснить логічний висновок на основі зворотної функціональності відношення *isProducerOf*, що $p001$ та $p002$ є однією й тією ж персоною! Якщо при цьому в онтології є явне ствердження, що екземпляри

p001 та p002 є нееквівалентними, то апарат логічного висновку повідомить про наявність протиріччя.

У цьому випадку протиріччя знімається видаленням ствердження про функціональність і зворотну функціональність вказаних вище відношень. Це проста операція, яка робиться натисненням двох прапорців на інтерфейсі Protege.

Припустимо, що дані необхідно поповнити новим аніме під назвою «Самурай Чамплу», продюсером якої був Ватанабе Сінтіро. У реляційній моделі додаємо запис щодо Ватанабе Сінтіро в таблицю «Персона», додаємо запис щодо аніме «Самурай Чамплу» до таблиці «Продукти», вказуємо код продюсера у відповідному полі.

Процедура розв'язування задачі в онтологічній моделі показана на рис. 4. Достатньо зробити два анонімних екземпляри (відмічено цифрою 1), не вказуючи явно, до якого класу вони відносяться. Далі явно вказати, що вони пов'язані властивістю isProducerOf (цифра 2). Під цифрою 3 на рис. 4 показано результат, який автоматично дає апарат логічного висновку онтології – онтологія сама здійснила висновок, що Ватанабе Сінтіро відноситься до класу Staff, а «Самурай Чамплу» до класу Anime.

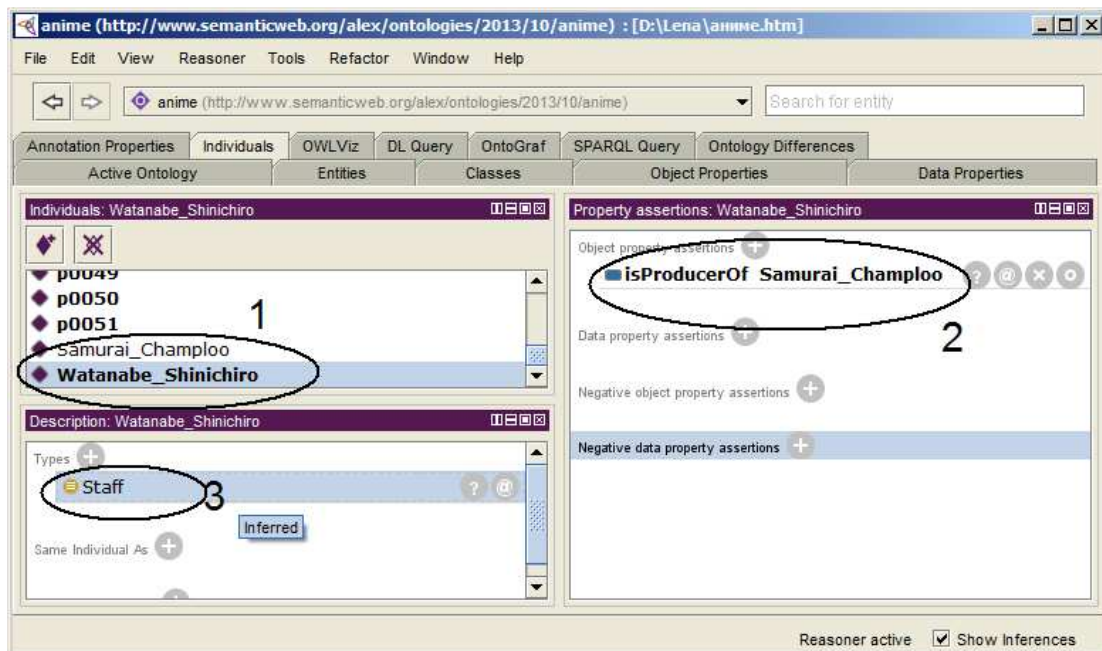
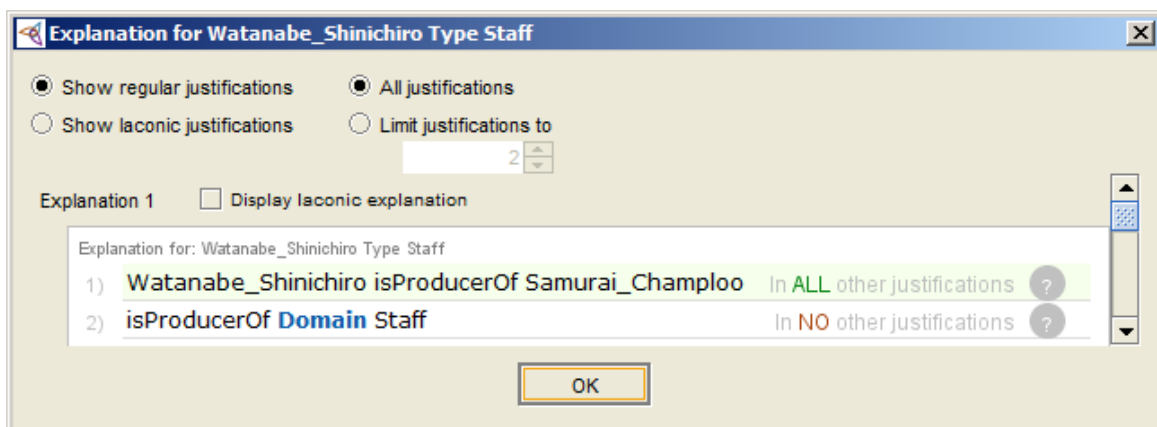


Рис. 4. Додавання екземплярів

Роботу апарату логічного висновку показано на рис. 5.



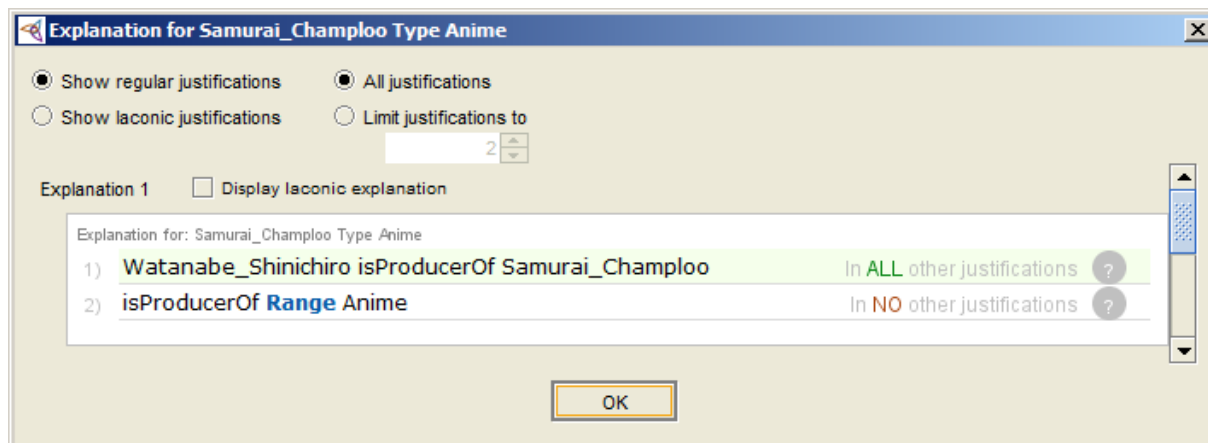


Рис. 5. Пояснення апарату логічного висновку щодо зроблених висновків

Якщо Ватанабе Сінтіро є продюсером Самурай Чамплу, а властивість «бути продюсером» діє саме між класами Staff та Anime, то звідси однозначно випливає, що

$Watanabe_Shinichiro \in Staff$ та $Samurai_Champloo \in Anime$

Найбільшу цікавість представляє виявлення транзитивних залежностей. Відношення між сіквелами і пріквелами в реляційній моделі показані в таблиці «Пріквел/Сіквел», де кожний запис є парою – код пріквела і код відповідного сіквела. Для того щоб визначити всі пріквели (або всі сіквели) обраного продукту, потужності мови SQL недостатньо. Необхідно розв'язувати проблему реалізацією циклічного алгоритму (наприклад, збереженої процедури). Водночас цей алгоритм буде здійснювати пошук транзитивних відношень лише для конкретного випадку.

В онтологічній моделі достатньо написати короткий запит:

isPrequelOf value Samurai_Champloo – показати всі пріквели для екземпляру *Samurai_Champloo*. На основі транзитивності відношення *isPrequelOf* запит покаже всі пріквели, які є як у прямій, так і в транзитивних залежностях довільної довжини з *Samurai_Champloo*. Наразі пошук транзитивних відношень виконується однаково для будь-якого відношення, що визначено як транзитивне.

3. ВИСНОВКИ

Наведена схема виконання індивідуального завдання дає студентам розуміння про те, що онтологічна і реляційна моделі не є рівними альтернативами, які можливо використовувати одну замість іншої. У процесі виконання роботи студенти чітко бачать, що відміною баз даних й онтологічних баз знань є так зване припущення про замкнутість/відкритість світу [7]. У реляційній моделі діє припущення про замкнутість світу, тобто істинними вважаються лише ті факти, які вказані в явній формі. Усі інші факти вважаються хибними. Тому бази даних не підходять для систем, де необхідно виявляти приховані зв'язки між даними, – усі зв'язки мають бути вказані явно. Але добір збережених даних за багатьма критеріями здійснюється з високою швидкістю.

Онтологічні бази знань основані на припущенні про відкритість світу. Це означає, що, якщо факт не вказано явно, то він не вважається ані істинним, ані хибним. На цьому ствердженні працює апарат логічного висновку, який, як ми бачили в процесі порівняльного аналізу, може знаходити зв'язки між даними, які явно вказано

не було. Це і є основною перевагою онтологій. Але доведено, що час, потрібний на логічний висновок, зростає за експоненціальним законом [8], тобто логічний висновок потребує багато часу.

Отже, студенти приходять до правильного висновку, що в сучасних інформаційних системах для різних потреб мають поєднуватись обидві моделі – як онтологічна, так і реляційна. Реалізувати це можливо встановленням відображення між реляційною й онтологічною моделями, збереженням екземплярів онтології в реляційній базі або просто реплікацією даних між онтологією і реляційною базою.

Після того, як студенти отримують досвід структурування тих знань, якими вони добре володіють, їм стає набагато легше використовувати отримані навички для структурування знань професійного спрямування на подальших етапах вивчення дисципліни. У цілому групи, які навчались за експериментальною робочою програмою, проявляли набагато вищий рівень зацікавленості в процесі вивчення курсу і більш високі результати під час модульного контролю.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Gruninger M. An Ontology Framework / M. Gruninger, L. Obrst // *Ontology Summit – NIST*. – Gaithersburg, 2007. – P. 191–200.
2. Космінська О. М. Специфіка методики викладання баз даних для студентів економічних і управлінських спеціальностей / Космінська О. М., Литвак О. Г. [Електронний ресурс] // *Інформаційні технології і засоби навчання*. – 2012. – № 4 (30). – Режим доступу до журналу : <http://journal.iitta.gov.ua/index.php/itlt/article/view/704#.UwHZLM6XpPg>.
3. Mizoguchi R. Tutorial on ontological engineering / Mizoguchi R // *New Generation Computing, OhmSha & Springer*, 2004. – Vol. 22. – № 2. – P. 198–220.
4. Муромцев Д. И. Онтологический инжиниринг знаний в системе Protégé / Д. И. Муромцев. – СПб : ГУ ИТМО, 2007. – 62 с.
5. Ульман Д. Д. Введение в системы баз данных / Д. Ульман, Д. Уидом, – Лори, 2006. – 379 с.
6. Тузовский А. Ф. Системы управления знаниями (методы и технологии) / А. Тузовский, С. Чириков, В. Ямпольский. – Томск : НТЛ, 2005. – 258 с.
7. Cadoli M. The complexity of propositional closed world reasoning and circumscription / Cadoli M., Lenzerini M. // *Journal of Computer and System Sciences*. – 1994. – P. 255–310.
8. Bechhofer S. Web Ontology Language Reference W3C Recommendation 10 February 2004 [Електронний ресурс] : офіційні рекомендації W3C щодо мови веб онтологій / Bechhofer S., Hendler J., Harmelen F. – Режим доступу : URL: <http://www.w3.org/TR/owl-ref/#Sublanguage-def>.

Матеріал надійшов до редакції 17.02.2014 р.

МЕТОДИКА ПРЕПОДАВАННЯ ОНТОЛОГІЧЕСКОГО ІНЖИНИРИНГА ДЛЯ СТУДЕНТІВ ОТРАСЛЕЙ «ЕКОНОМІКА» І «МЕНЕДЖМЕНТ»

Литвак Елена Геннадиевна

старший преподаватель кафедры информационных технологий
Донецкий государственный университет управления, г. Донецк, Украина
alttt@yandex.ru

Лашина Мария

студентка 3-го курса специальности «Маркетинг»
Донецкий государственный университет управления, г. Донецк, Украина
I.merii.san@gmail.com

Аннотация. Актуальность материала статьи обусловлена существованием проблемы повторного использования информации на предприятиях. Для успешного решения этой проблемы необходимо, чтобы специалисты разных направлений (часто не связанных с информационными технологиями напрямую) владели основами онтологического

инженеринга знаний. Дисциплины, содержащие основы онтологического инженеринга знаний и другие информационные технологии, студентами нетехнических направлений воспринимаются с трудом. В статье на основе эксперимента, проведенного в Донецком государственном университете управления, приводится методика преподавания онтологического инженеринга знаний для студентов направлений «Экономика» и «Менеджмент». Доказано, что разработанная методика значительно повышает уровень восприятия материала студентами, способствует развитию аналитического и творческого мышления.

Ключевые слова: базы знаний; онтологии; онтологический инженеринг; базы данных; реляционная модель; аппарат логического вывода.

TEACHING METHODS OF ONTOLOGICAL ENGINEERING FOR STUDENTS OF "BUSINESS" AND "MANAGEMENT" DISCIPLINE

Olena G. Lytvak

senior lecturer of the Department of information technologies
Donetsk State University of Management, Donetsk, Ukraine
alttt@yandex.ru

Mariia Lashyna

third year student of "Marketing" speciality
Donetsk State University of Management, Donetsk, Ukraine
I.merii.san@gmail.com

Abstract. Actuality of the material presented in the paper is conditioned by existence of problem of the repeated use of information at enterprises. It is necessary for the successful decision of this problem, that the specialists of different directions (often unconnected with information technologies straightly) should know bases of ontological engineering of knowledge. Disciplines, containing the bases ontological engineering of knowledge and other information technologies, are difficult for the students of nontechnical directions. In the article on the basis of the experiment conducted in the Donetsk State University it is presented a teaching method of ontological engineering of knowledge for the students of «Economic» and «Management» disciplines. It is well-proven that the method considerably promotes the level of perception of material by students, promotes the development of analytical and creative thought.

Keywords: bases of knowledge; ontology; ontological engineering; databases; relational model; inference.

REFERENCES (TRANSLATED AND TRANSLITERATED)

1. Gruninger M. An Ontology Framework / M. Gruninger, L. Obrst // *Ontology Summit – NIST*. – Gaithersburg, 2007. – P. 191–200 (in English).
2. Kosminska O. M. Specifics of the technique of teaching of databases for students of economic and management specialties [online] / Kosminska O. M., Lytvak O. G. // *Informational Technologies and Learning Tools*. – 2012. – № 4 (30). – Available from : <http://journal.iitta.gov.ua/index.php/itlt/article/view/704#.UwHZLM6XpPg> (in Ukrainian).
3. Mizoguchi R. Tutorial on ontological engineering / Mizoguchi R. // *New Generation Computing, OhmSha & Springer*. – 2004. – Vol. 22. – № 2. – P. 198–220.
4. Muromcev D. I. *Ontological engineering knowledge in the system Protégé*. – SPb : SPb GU ITMO, 2007. – 62 p (in Russian).
5. Ullman J. *A first course to databases system* / Ullman J., Widom J. – Lori, 2006. – 379 p.
6. Tuzavsky A. F. *Knowledge management systems (methods and techniques)* / A. Tuzavsky, S. Chirkov, V. Yampolsky. – Tomsk : NTL, 2005. – 258 p. (in Russian).
7. Cadoli M. The complexity of propositional closed world reasoning and circumscription / Cadoli M., Lenzerini M. // *Journal of Computer and System Sciences*. – 1994. – P. 255–310 (in English).
8. Bechhofer S. *Web Ontology Language Reference W3C Recommendation 10 February 2004* [online] : official W3C Recommendation Web Ontology Language / Bechhofer S., Hendler J., Harmelen F. // Available from : <http://www.w3.org/TR/owl-ref/#Sublanguage-def> (in English).