

5. Metzler, D. Similarity measures for tracking information flow [Text] / D. Metzler, Y. Bernstein, W. B. Croft, A. Moffat, J. Zobel // Proceedings of the 14th ACM international conference on Information and knowledge management – CIKM '05, 2005. – P. 517–524. doi: 10.1145/1099554.1099695
6. Buttler, D. A Short Survey of Document Structure Similarity Algorithms [Text] / D. Buttler // The 5th International Conference on Internet Computing, 2004.
7. Ідентифікація бібліографічних описів [Електронний ресурс]. – 2015. – Режим доступу: https://uk.wikipedia.org/wiki/Ідентифікація_подібності_бібліографічних_описів
8. Макар, В. Інформаційно-аналітична система для автоматизації підготовки наукових звітів підрозділів Львівської політехніки [Текст]: матер. 6-ї наук.-прак. конф. / В. Макар, Р. Тушницький // Інноваційні комп'ютерні технології у вищій школі. – Львів, 2014. – С. 177–182.
9. Федасюк Д. В. Структура інформаційно-аналітичної системи обліку підготовки наукових кадрів університету [Текст] / Д. В. Федасюк, В. М. Макар, Р. Б. Тушницький // Вісник Національного університету “Львівська політехніка”. Серія “Інформатизація вищого навчального закладу”. – 2013. – № 775. – С. 99–103.
10. Кушнарєнко, Н. М. Наукова обробка документів [Текст] / Н. М. Кушнарєнко, Б. К. Удалова; 4-те вид., перероб. і доп. – К. : Знання, 2006. – 334 с.
11. Haase, P. A Bibster – A Semantics-Based Bibliographic Peer-to-Peer System [Text] / P. Haase, B. Schnizler, J. Broekstra, M. Ehrig, F. van Harmelen, M. Menken et. al. // Semantic Web and Peer-to-Peer, 2006. – P. 349–363. doi: 10.1007/3-540-28347-1_19

Розроблено інформаційну технологію, що базується на знаннях, яка вирішує задачу автоматичної генерації тестових запитань з групуванням їх відповідно до ієрархії понять предметної області. В рамках розробленої технології створено інструментальний програмний засіб. Розроблена технологія дозволить збільшити кількість навчальних тестів, звільнивши час викладача від рутинної роботи на користь її творчої складової, при цьому підвищить якісний рівень освіти

Ключові слова: електронне навчання, дистанційне навчання, навчальний контент, онтології, бази знань

Разработана базируемая на знаниях информационная технология, которая решает задачу автоматической генерации тестовых вопросов с группированием их в соответствии с иерархией понятий предметной области. В рамках разработанной технологии создано инструментальное программное средство. Разработанная технология позволит увеличить количество учебных тестов, освободив время преподавателя от рутинной работы в пользу её творческой составляющей, при этом повысит качественный уровень образования

Ключевые слова: электронное обучение, дистанционное образование, контент, онтология, базы знаний

УДК 004.853
DOI: 10.15587/1729-4061.2015.51334

РОЗРОБКА ГЕНЕРАТОРА ТЕСТІВ ДЛЯ “MOODLE” НА БАЗІ ОНТОЛОГІЇ

С. В. Сирота

Кандидат технічних наук, доцент*

E-mail: sergiy.syrot@gmail.com

В. О. Ліскін

Аспірант*

E-mail: lis-580@rambler.ru

*Кафедра прикладної математики

Національний технічний університет України

«Київський політехнічний інститут»

пр. Перемоги, 37, м. Київ, Україна, 03056

1. Вступ

Застосування інформаційних технологій у системі освіти дозволяє удосконалювати навчальний процес шляхом впровадження нових методів і підходів не тільки в навчання, а й в контролі знань.

Стрімкий розвиток елементної бази та інформаційних технологій ставить завдання безперервно вдосконалювати, і тримати навчальний контент «up to date». Викладач працює над своїми курсами, використовуючи Інтернет, і редагує матеріали в реальному часі. Завдяки цьому студенти мають можливість централізовано і оперативно отримувати оновлену інформацію.

На сьогоднішній день тестування є однією з найбільш широко використовуваних форм перевірки знань. Одним з найяскравіших прикладів є ЗНО для випускників середніх шкіл, обов'язкове для вступу у ВНЗ з 2008 року та ДПА в середній школі.

Актуальною задачею є підвищення якості контролю знань. Аналіз методики роботи з тестовими запитаннями показав, що у випадку невеликої кількості банку запитань тести доцільно використовувати лише для фінального контролю в режимі екзамену. Звідси випливають дві полярні задачі. З одного боку тестування має виконувати навчальну функцію і бути максимально незалежним від випадковості, а з другого – реально відображати картину знань.

Використання тестування в навчальному процесі дозволяє в найкоротший термін перевірити знання великих груп учнів, провести контроль досягнень освітніх результатів, скоротити час обробки даних. Разом з тим створення ефективних і верифікованих тестів є достатньо трудомістким процесом, який містить багато рутинної роботи.

Тому створення у студента цілісного уявлення про предметну область формування умінь та навичок і автоматизованих засобів його ефективного діагностування та розробка і впровадження інформаційної технології, яка дозволить збільшити кількість навчальних тестів звільнивши час викладача від рутинної роботи на користь її творчої складової, максимально задовільнить потребам викладача, при цьому підніме якісний рівень освіти.

2. Аналіз літературних даних та постановка проблеми

У літературі є різні визначення педагогічних тестів. Найбільш розгорнуте визначення можна знайти в [1], де тест визначається як сукупність питань і завдань, що пройшли попередню експериментальну перевірку і спеціальну процедуру для їх поліпшення, і мають достатні характеристики своєї ефективності.

На сьогоднішній день в розвитку комп'ютерних навчальних систем можна виділити три напрямки:

- розробка адаптивних систем [2];
- розробка тестових шаблонів і алгоритмів автоматичної генерації тестів на підставі формалізованої предметної області [3];
- оцінювання знань учнів з використанням природної мови [4].

Вихідними даними для формування тестових завдань з конкретної дисципліни є номенклатура типів тестових завдань закритого і відкритого типу [5].

В роботі [6] запропонована понятійно-тезова модель подання знань, на базі якої автори планують розробити систему автоматизованої генерації тестів для контролю знань. Проте в ній немає інформації про повноту такого подання інформації та про повноту тесту, який повинен представляти знання учня усього навчального курсу. Інший підхід до вирішення аналогічної задачі [7, 8] базується на використанні аналізу формальних понять Вілле-Гантера і спрямований на автоматичному формуванні понять, закономірностей і асоціацій в предметній області. У роботах [9, 10] розглядається поняття педагогічних тестів. Педагогічний тест – це система паралельних завдань рівномірно зростаючій важкості, які дозволяють виміряти рівень і оцінити ступінь підготовленості учня.

Сучасні онтологокеровані інформаційні системи призначені для концептуалізації онтологічних категорій та удосконалення ієрархічних структур сутностей на всіх рівнях [11].

Спільне використання людьми або програмними агентами загального розуміння структури інформації є однією з найбільш спільних цілей розробки онтологій [12]. Останнім часом в якості стандарту мови обміну онтологіями в Ontolingua використовують мову OWL, запропоновану у проекті Semantic Web [13]. Серед відомих інструментальних засобів

найбільшою популярністю користується платформа Protégé – безплатний, відкритий редактор онтологій і фреймворк для побудови баз знань. Має відкрити, легко розширювану архітектуру і підтримується широким співтовариством, що складається з розробників і вчених, урядових і корпоративних користувачів, які використовують його для вирішення завдань, пов'язаних зі знаннями, в різноманітних областях. Онтології, побудовані в Protégé, можуть бути експортовані у різні формати, включаючи OWL і XML.

Досвід використання Moodle [14] виявив, що застосування тестів з невеликим банком запитань (менше 100) доцільне лише з метою перевірки, а використання тестів в режимі навчання можливе при наявності певної кількості (не менше 20) запитань, що стосуються одного і того-ж факту з різними формулюваннями. Створення таких банків запитань, її верифікація є дуже трудомістким процесом.

3. Ціль та задачі дослідження

Метою даної роботи є розробка інформаційної технології, яка вирішує задачу автоматичної генерації тестових запитань з групуванням їх відповідно до ієрархії понять предметної області.

Для цього необхідно вирішити наступні задачі:

- дослідити існуючі методи представлення онтологій в OWL та доступні інструментальні засоби створення тестів для Moodle;
- розробити правила або рекомендації щодо побудови онтологій предметної області доступними інструментальними засобами з тим, щоб створена онтологія була придатна для подальшої обробки;
- виділити цільові типи тестових запитань, які придатні для автоматичної обробки, розробити алгоритми їх генерації;
- реалізувати програмні засоби автоматизованого генерування верифікованих запитань із розподілом їх за категоріями відповідно до тестованих понять, та дослідити показники їх роботи.

4. Технологія генерації тестових запитань з використанням онтологій

4.1. Особливості існуючих програмних засобів для роботи з онтологіями та організації навчання

Методика автоматичного генерування тестових запитань з використанням онтологій певного поняття та поєднанням її з мініонтологією тестового запитання була запропонована в [14].

В основі мови OWL лежить представлення предметної області в моделі концептів «об'єкт – властивість». OWL придатний для опису не лише веб-сторінок, але і будь-яких об'єктів дійсності. Кожному елементу опису в цій мові (у тому числі властивостям, що зв'язують об'єкти) ставиться у відповідність URI (англ. Uniform Resource Identifier) – уніфікований ідентифікатор ресурсу.

Moodle – це система управління навчанням (система управління курсом навчання, віртуальне навчальне

середовище), що представляє собою веб-додаток, який викладачі можуть використовувати для створення ефективних навчальних сайтів [15].

Будь-який тест в Moodle створюється на основі Банка питань. Які створюються автором курсу вбудованими засобами Moodle або можуть бути імпортовані в різноманітних форматах. Найбільш зручним форматом для імпорту є GIFT оскільки фактично є текстовим файлом з певною розміткою і спеціальними знаками. Підтримує питання різних, які можуть бути поєднані в одному файлі. Формат також підтримує назви питань, коментарі до варіантів відповідей, зворотний зв'язок і процентне оцінювання.

При створенні тесту в Moodle автору потрібно визначити чи буде тест мати які-небудь обмеження за часом, скільки питань буде відображатися на одній сторінці, випадковий або невипадковий порядок цих питань, скільки спроб буде надано кожному тестованому, які методи оцінювання будуть застосовуватися та інші параметри. При запуску тесту забезпечується випадковий порядок слідування відповідей і дистракторів.

4. 2. Онтологія предметної області

Як згадувалося раніше, основними будівельними блоками онтології OWL в системі Protege 4 є класи. Класи інтерпретуються як множина індивідів (чи об'єктів). Властивості в OWL представляють відношення. Існує два основні типи властивостей: властивості об'єктів і властивості типів даних. Властивостями об'єкту є відношення між двома індивідами. Властивості об'єктів зв'язують індивідів. OWL також має властивості анотації, які можуть використовуватися для додавання метаданих для класів, окремих індивідів і властивостей об'єктів/типів даних [16]. Для досягнення мети пропонується розглядати поняття як набір ланцюжків сутність – зв'язок – опис, якому передеує нотація:

Concept=Notatio: Essentia–Ligamentum–Descriptio (лат.)

або

$$C: \langle N, E, L, D \rangle, \tag{1}$$

де N – нотація ($E=\{E_i\}$ – кортеж сутностей предметної області, L – зв'язки, $D=\{D_i\}$ – кортеж властивостей).

У зв'язку з тим, що в різних роботах українські терміни: «поняття», «контекст», «сутність» мають різні значення ми вживатимемо латинські терміни, які відповідають структурним частинам наведених далі прикладів.

Приклад концептів «відеокарта» та моделі кольору з курсу «Ком'ютерна графіка» наведено на рис. 1.

Як було зазначено, в OWL кожному елементу опису (у тому числі властивостям, що зв'язують об'єкти) ставиться у відповідність URI, тобто і об'єкти і властивості є в машинному представленні даними одного типу. Звідси випливає правило, якого слід дотримуватись при побудові онтології предметної області для подальшої обробки генератором тестів:

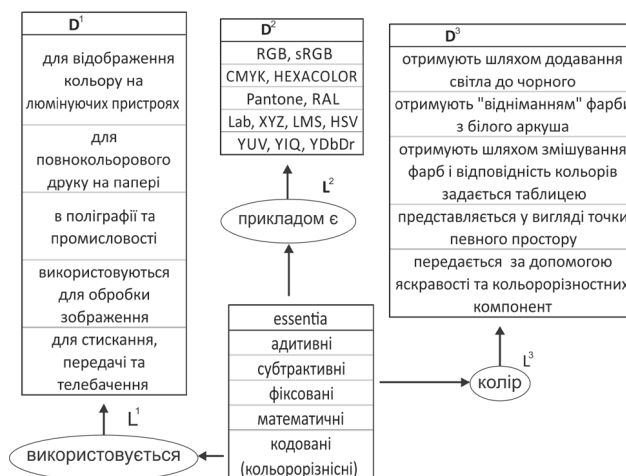
– Ієрархія властивостей (*rdf:Property*) повинна відповідати зв'язці L (*Ligamentum*).

Concept: «Відеокарта», Notatio: «Функціональний блок»

E	L	D
графічний процесор	виконує функцію	розраховує вміст відеопам'яті
контролер монітора		забезпечує відображення вмісту відеопам'яті на моніторі
відеопам'ять (V-RAM)		зберігає зображення кожного кадру
розширення BIOS		зберігає налаштування та проводить діагностику при включенні
контролер внутрішньої шини		забезпечує обмін даними з центральним процесором

а

C= «Моделі кольору», N= «Відомі моделі представлення кольору»



б

Рис. 1. Приклад концептів: а – Концепт «Відеокарта»; б – Концепт «Моделі кольору»

4. 3. Онтологія запитань

Згідно термінології, що склалася тест (quiz) розуміється як набір з певної кількості тестових запитань. Відрізняють запитання в закритій та відкритій формі. Для автоматичного оцінювання зручніше використовувати запитання в закритій формі. Загальна структура запитання в закритій формі представлена на рис. 2. Серед варіантів відповідей відрізняють правильні – *відповіді* і неправильні – *дистрактори*.



Рис. 2. Структура запитання в закритій формі

Розрізняють наступні типи запитань в закритій формі:

1. *На відповідність (Matching) «М.»* – задається певна кількість сутностей зв'язок і така ж кількість описів задача полягає в тому щоб правильно співвіднести сутність з описом.

2. *Множинний вибір (Multiple choice):*

– *Багато відповідей (Multiple-answer) «МА»* – задається одна сутність, зв'язок, певна кількість описів. Задача полягає в тому щоб позначити описи, які відповідають сутності.

– Одна відповідь (*Single answer*) «SA» – задається одна сутність, певна кількість описів. Задача полягає в тому щоб вибрати тільки одну правильну відповідь.

3. Так/Hi (*True/False*) «TF»– дається ланцюжок сутність – зв'язок – опис треба відповісти істинна вона або хибна.

В більшості випадків кожне запитання відповідає певному поняттю предметної області.

Нехай на вході генератора тестів є концепт у вигляді $C: \langle N, E, L, D \rangle$, для прикладу рис. 2, а це буде:

- C = «Відеокарта»,
- N = «Функціональний блок»,
- L = «виконує функцію»,
- E_1 = «графічний процесор»,
- D_1 = «розраховує вміст відеопам'яті»,
- E_2 = «контролер монітора»,
- D_2 = «забезпечує відображення ...»,
- E_3 = «відеопам'ять (V-RAM)»,
- D_3 = «зберігає зображення кож...»,
- E_4 = «розширення BIOS»,
- D_4 = «зберігає налаштування та...»,
- E_5 = «контролер внутрішньої шини»,
- D_5 = «забезпечує обмін даними...».

Очевидно що, отримано тестове запитання типу **M**. з точністю до форматування. Приклад результату наведено в Розділі 5 (рис. 4, а).

З питання типу **M** побудувати питання типу **SA** можна наступним чином: кожній сутності E_i з кортежу **E** зіставляється кортеж **D** при цьому D_i буде правильною відповіддю, а решта $D_{j \neq i}$ виконуватимуть роль дистракторів. Більше того, в коментарі до $D_{j \neq i}$ записуються відповідні E_j з певним префіксом, наприклад «Невірно! Це ». Приклад результату наведено на рис. 4, б.

Для того щоб отримати питання типу **MA**, є два наступних способи.

Перший полягає в тому що зв'язка **L** замінюється на її заперечення $\neg L$ тоді D_i стане дистрактором, а решта $D_{j \neq i}$ відповідами. Заміна **L** на $\neg L$ виконується користувачем вручну.

Другий спосіб використовує сутності, які мають більше одного зв'язку (рис. 1, б), тоді сутності E_i з кортежу **E** зіставляється ланцюжок $\langle L^k, D^k \rangle$, де індекс **k** означає зв'язок і відповідний йому кортеж **D** при цьому D^k_i буде правильною відповіддю, а решта $D^k_{j \neq i}$ виконуватимуть роль дистракторів.

Питання типу **TF** генеруються наступним чином: будуються ланцюжки з сутностей E_i з кортежу **E** зі зв'язками **L** та їм зіставляється елементи кортежу **D**, при цьому елемент D_i дасть істинне твердження, а решта $D_{j \neq i}$ хибні.

5. Результати дослідження, програмний засіб для генерації тестових запитань

Програмний засіб для генерації тестових запитань має простий та зрозумілий інтерфейс, екранна форма якого представлена на рис. 3. На вхід генератора тестових питань подається онтологія в форматі OWL.

Програма аналізує інструкції *rdf:Property* і пропонує користувачеві знайдені концепти у вигляді (1) (рис. 3), кожний концепт на окремій вкладці.

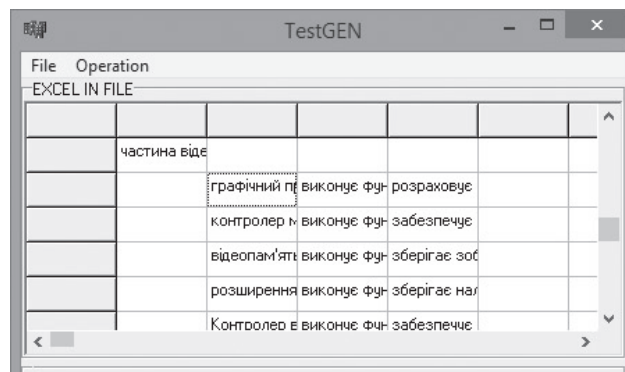


Рис. 3. Екранна форма генератора тестів

Також в даному програмному засобі передбачена можливість ручного введення даних для генерації тестів.

Користувач може відредагувати *ligamento* та задає параметри бажаних запитань:

- тип запитання;
- кількість варіантів відповіді;
- метод оцінювання (зі штрафними балами або без).

Відповідно до заданих параметрів для кожного концепту генеруються запитання, які зберігаються в окремий файл. При цьому розподіл позитивних і негативних оцінок здійснюється автоматично відповідно до кількості відповідей і дистракторів в запитанні. На виході користувач отримує файл зі згенерованими тестовими завданнями у форматі GIFT (рис. 4) В прикладі (рис. 4, б) «%-10 %» це штрафні бали які нараховуються за неправильну відповідь.

```
// question: 0 name: Switch category to $module$
/Default for Пробний тест $CATEGORY: $module$/Default for Пробний тест//
question: 25 name: Відеокарта::Функціональний блок::
[html]Необхідно побудувати вірні твердження\{
=графічний процесор -> розраховує вміст відеопам'яті
=контролер монітора -> забезпечує відображення вмісту відеопам'яті на моніторі
=відеопам'ять (V-RAM) -> зберігає зображення кожного кадру
...}
```

а

```
// question: 23 name: Відеокарта 1 SA
::Відеокарта::[html]<pr>
Функціональний блок</pr><pr> контролер монітора виконує функцію </pr>\{
~%-10% розраховує вміст відеопам'яті #Невірно!\n
= контролер монітора #Вірно,\n\n
~%-10% зберігає зображення кожного кадру #Невірно! Це відеокарта (V-RAM) \n\n
...}
```

б

Рис. 4. Результати роботи генератора тестових запитань: а – питання типу **M**; б – питання типу **SA**

Беручи до уваги те, що індивіди в OWL є екземплярами класів та підкласів, згенеровані файли розташовуються за ієрархією вкладених папок файлової системи відповідно до ієрархії класів у вхідному OWL файлі.

6. Обговорення результатів дослідження, кількісні та якісні показники роботи генератора запитань

Аналіз отриманих результатів показав, що для кожного концепта в залежності від кількості заданих *Essentia (m)* та *Descriptio (n)* *Ligamento (l)* можна отримати:

- одне запитання типу **M**;
- $m \times l$ запитань типу **SA**;
- $m \times l$ запитань типу **MA**;
- $m \times l \times n$ запитань типу **TF**;
- $m \times C_{l \times n}^k$ запитань типу **MA** з k варіантами відповідей або $m \times 2^{l \times n - 1}$ запитань типу **MA** з довільною кількістю варіантів відповідей.

Якщо прийняти за нормальну кількість варіантів відповіді 5 і покласти $m=n=5$, то за одним концептом який має щонайменше 3 види зв'язків, буде згенеровано 3109 запитань! До позитивних результатів слід віднести те, що згенеровані тестові запитання не містять помилок виходячи з того, що кожному *Essentia* однозначно співставлено *Ligamentum* та *Descriptio* і згенерований тест не потрібно перевіряти на помилки.

Звичайно, слід зазначити, що дана технологія може бути застосована лише для предметних областей, поняття яких представлені у вигляді (1). Це накладає певні обмеження на онтології, які можуть бути коректно використані в якості вхідних файлів. Технологія вирішує тільки питання кількості тестових запитань, залишаючи відповідальність за якість тестів авторів курсу.

В будь-якому випадку, дану техніку можна застосувати до банків існуючих запитань на відповідність (**M**) для того щоб почати використовувати тести, які були призначені лише для фінального контролю, з метою навчання.

7. Висновки

На основі дослідження існуючих методів представлення онтологій, аналізу форматів і технологій, які використовують існуючі програмні засоби Protégé і MOODLE, були розроблені правила та рекомендації щодо побудови онтології предметної області. А саме те, що ієрархія властивостей в представленні онтології повинна відповідати зв'язці тестового запитання. Виділено клас онтологій, що відповідають даній вимозі, які можуть бути використані для подальшої обробки запропонованими методами.

Виділено чотири цільові типи тестових запитань в закритій формі, які придатні для автоматичної обробки, побудовано їх інфологічну модель, і розроблено алгоритми їх генерації, в основі яких полягає перебір і комбінація текстових рядків.

Створено програмний засіб автоматизованого генерування верифікованих запитань із розподілом їх за категоріями відповідно до тестованих понять, виявлено що згенеровані тестові запитання є верифікованими та їх кількість достатня для використання тестування в навчальних цілях.

Таким чином розроблена технологія дозволить збільшити кількість навчальних тестів звільнивши час викладача від рутинної роботи на користь її творчої складової, при цьому підвищить якісний рівень освіти.

Література

1. Рапопорт, И. А. Тесты в обучении иностранным языкам в средней школе. [Текст] / И. А. Рапопорт, Р. Сельг, И. Соттер. – Таллин: Валгус, 1987. – 350 с.
2. Žitko, V. Dynamic test generation over ontology-based knowledge representation in Expert Systems with Applications [Text] / S. Stankov, M. Rosić, A. Grubišić // Expert Systems with Applications. – 2008. – Vol. 36, Issue 4. – P. 8185–8196. doi: 10.1016/j.eswa.2008.10.028 – Available at: http://www.academia.edu/3187332/dynamic_test_generation_over_ontology-based_knowledge_representation_in_authoring_shell
3. Durlach, P. J. Adaptive Technologies for Training and Education [Text] / P. J. Durlach, A. M. Lesgold. – Cambridge : Cambridge University Press, 2012. – 380 p. doi: 10.1017/cbo9781139049580
4. Gutl, Ch. Enhanced Approach of Automatic Creation of Test Items to foster Modern Learning Setting [Text] / Ch. Gutl, K. Lankmayr, J. Weinhofer, M. Hofler // Electronic Journal of e-Learning. – 2011. – Vol. 9, Issue 1. – P. 23–38.
5. Уэно, Х. Представление и использование знаний [Текст] / Х. Уэно, М. Исудзука. – М. : Мир, 1989. – 220 с.
6. Титенко, С. В. Семантична модель знань для цілей організації контролю знань у навчальній системі [Текст]: сб. тр. VI міжк. конф. / С. В. Титенко, О. О. Гагарін // Интеллектуальный анализ информации. – Киев, 2006. – С. 298–307.
7. Таран, Т. А. Обучение понятиям в интеллектуальных обучающих системах на основе формального концептуального анализа [Текст] / Т. А. Таран, С. В. Сирота // Искусственный интеллект. – 2000. – № 3. – С. 340–347.
8. Таран, Т. А. Методика извлечения знаний при построении интеллектуальных обучающих систем [Текст]: сб. тр. VI Между. конф. / Т. А. Таран, С. Н. Копычко, С. В. Сирота, Н. А. Гулякина // Интеллектуальный анализ информации. – Киев, 2006. – С. 282–287.
9. Чельшкова, М. Б. Теория и практика конструирования педагогических тестов [Текст]: учеб. пос. / М. Б. Чельшкова. – М. : Логос, 2002. – 432 с.
10. Нейман, Ю. М. Введение в теорию моделирования и параметризации педагогических тестов [Текст] / Ю. М. Нейман, В. А. Хлебников. – М., 2000. – 168 с.
11. Палагин, А. В. Онтологические методы и средства обработки предметных знаний [Текст]: монография / А. В. Палагин, С. Л. Кривый, Н. Г. Петренко. – Луганск: изд-во ВНУ им. В. Даля, 2012. – 324 с.
12. Gruber, T. R. The role of common ontology in achieving sharable, reusable knowledge bases [Text] / T. R. Gruber; J. A. Allen, R. Fikes, E. Sandewell (Eds.). – Principles of Knowledge Representation and Reasoning – Proceedings of the Second International Conference, 1991. – P. 601–602.
13. Сирота, С. В. Огляд сучасних онтологізованих інформаційних систем та сервісів і перспективи їх застосування в електронній освіті [Текст] / С. В. Сирота, В. О. Ліскін // Технологічний аудит та резерви виробництва. – 2015. – Т. 5, № 6 (25). – С. 58–60. doi: 10.15587/2312-8372.2015.51234
14. Сирота, С. В. Використання системи “Moodle” для викладання дисципліни “Алгоритми і структури даних” – досвід і проблеми [Текст] / С. В. Сирота, В. О. Ліскін // ScienceRise. – 2015. – Т. 9, № 2 (14). – С. 30–35. doi: 10.15587/2313-8416.2015.50610
15. Офіційний сайт Moodle [Електронний ресурс]. – Режим доступу: <https://moodle.org/>
16. Noy, N. F. Ontology Development 101: A Guide to Creating Your First Ontology [Text] / N. F. Noy, D. L. McGuinness. – Stanford Knowledge Systems Laboratory Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report SMI-2001-0880, 2001. – Available at: http://protege.stanford.edu/publications/ontology_development/ontology101.html