

ПОБУДОВА РЕКОМЕНДАЦІЙНИХ СИСТЕМ НА ОСНОВІ ОНТОЛОГІЙ

У цій статті наведено опис основних традиційних типів рекомендаційних систем, а також рекомендаційних систем, побудованих на основі онтологій. Проведено огляд способів використання, технологій та засобів реалізації онтологій.

Також у статті описано функціонал, структуру та реалізацію рекомендаційної навчальної системи на основі онтологій у вигляді плагіну до системи Protege.

Ключові слова: рекомендаційна система, рекомендації, онтологія, дескриптивна логіка, OWL, редактор онтологій, Protégé, плагін.

Вступ

Рекомендаційні системи створюють рекомендації автоматично для кожного користувача на основі попередніх його пошуків, оцінок товарів, покупок та іншої активності на сайті.

Проте далеко не завжди традиційні типи рекомендаційних систем можуть надати користувачеві якісні рекомендації. Одними з основних причин є проблема холодного старту, масштабованості, розрідженості даних та відсутність відповідної кількості рейтингів для товарів, які не часто купують. Також здебільшого у випадках пропонованих системою елементів зі складнішою предметною галуззю користувачам необхідно явно задати критерії їхнього пошуку. Саме для задоволення таких потреб системи застосовують рекомендаційні системи на основі онтологій, які дають змогу вирішити низку цих та інших проблем.

Метою цієї статті є визначити основні переваги рекомендаційних систем, побудованих на основі онтологій, у порівнянні з традиційними типами рекомендаційних систем, описати технології та засоби реалізації онтологій, а також запропонувати реалізацію рекомендаційної навчальної системи на основі онтологій у вигляді плагіну до системи Protege.

Рекомендаційні системи та їх типи

Рекомендаційна система – це система, яка використовує наявну інформацію про потребу користувача та певний алгоритм фільтрування, рекомендує користувачу набір альтернатив, які вона вважає найбільш корисними для задоволення цієї потреби [4].

Основними підходами роботи більшої кількості рекомендаційних систем є контентна фільтрація (content-based filtering) та колаборативна фільтрація (collaborative filtering). Також існують гібридні рекомендаційні системи [1; 8].

Іншим типом рекомендаційних систем є рекомендаційні системи, які базуються на онтологічній моделі представлення знань. Розглядають системи на базі обмежень (constraint-based), на базі доказів (case-based) [3], а також системи, які використовують семантичні технології дескриптивної логіки (semantic reasoning-capabilities).

Поняття онтологій, засоби реалізацій, способи застосування

У загальному вигляді онтології визначаються як спільно використовувані формальні концепції конкретних предметних галузей, інформацією з яких можуть обмінюватися люди і додатки. Вони дозволяють концептуалізувати певну предметну галузь фіксуванням її сутностей і зв'язків між ними [5].

Для створення онтологій використовують рекомендовану W3C формальну мову для визначення і подання структурованих веб-онтологій OWL, що є розширенням RDF/RDFS. Формальною основою мови опису OWL-онтологій є дескриптивні логіки (description logics), що забезпечують можливість логічного виведення [9; 6].

OWL має декілька діалектів: OWL Lite, OWL DL, OWL full та інші.

Розробники онтологій, що використовують OWL, мають визначити, який з діалектів краще підходить до їхніх завдань, це залежить від ступеня виразності потрібних користувачам констукцій.

У роботі було використано діалект OWL DL, який базується на SHIQ Description Logic [9], що надає можливість використовувати так звані «reasoning algorithms», які добре розроблені в рамках дескриптивної логіки.

Реалізація таких алгоритмів забезпечує можливість автоматичного виведення знань з онтології, що не задані явно, а випливають із наявних в онтології знань. Ключовими можливостями тут є здатність автоматично виводити класову ієрархію, а також здатність перевіряти несуперечливість опису класів (логічної узгодженості онтології).

Для здійснення запитів до класифікованої за допомогою використання OWL DL онтології існують різні мови запитів, як-от: OWL-QL, SQWRL, SPARQL Query, DL Query.

Завдяки додатковій семантичній підтримці інформації, онтології широко використовуються в різних напрямках, як-от: семантичний ВЕБ, створення загальних словників для вчених у певній предметній галузі, надання рекомендацій користувачам рекомендаційних систем та інше [3; 2].

Опис функціоналу та реалізації рекомендаційної навчальної системи на основі онтологій

Призначення системи – надання студентіві рекомендацій щодо вибіркового дисциплін на основі явно заданих ним обмежень. До уваги було взято лише предмети факультету інформатики НаУКМА.

Зважаючи на те, що редактор онтологій Protege має відкриту архітектуру, використовує OSGI framework як інфраструктуру плагінів, яка дозволяє програмістам інтегрувати плагіни, які можуть відображатися у вигляді окремих вкладок, специфічних компонентів користувацького інтерфейсу (віджетів), або виконувати будь-які інші завдання по поточній моделі [9], було вирішено реалізувати рекомендаційну систему у вигляді WorkspaceTab-плагіну системи Protégé 5.1.0 з використанням мови програмування Java.

Для отримання рекомендацій користувач має вказати такі обмеження: галузь науки, до якої входить предмет, спеціальність та напрям дослідження, який студент хоче вивчати, тип предмета, семестр, у якому користувач хоче слухати цей предмет, та кількість кредитів, яку повинен мати предмет. Обов'язковим обмеженням для пошуку є галузь науки, решту обмежень користувач може вказувати за бажанням.

Рекомендаційна система здійснює пошук рекомендацій, які задовольняють усі обмеження, надані користувачем, за допомогою здійснення Conjunctive query до відповідної онтології предметів. У разі існування рекомендацій система видає користувачеві список відповідних рекомендованих предметів із можливістю перегляду деталей кожного з них, а отже, надання пояснення виданого результату.

Система надає користувачу такий опис предмета: Назва, Кафедра, Тип предмета, Семестр, Загальна к-сть годин, К-сть кредитів, К-сть годин на тиждень, Список напрямів дослідження, які вивчає цей предмет, Базові предмети, які необхідно прослухати перед початком вивчення цього предмета, якщо такі є, та Предмети-продовження цього курсу, якщо такі є.

Якщо система не знаходить предметів, що задовольняють усі обмеження користувача, вона здійснює спроби зменшити кількість обмежень (здійснює query relaxation), доки не знайде хоч один предмет, який не порушуватиме жодне з обмежень. Перед показом результату система повідомляє користувача про обмеження, які їй довелося відкинути для знаходження рекомендацій, і перепитує, чи такий результат влаштовуватиме користувача. У разі згоди система видає користувачеві відповідні результати. При цьому користувач має змогу переглянути свої початкові обмеження та змінити їх, якщо рекомендації не цілком задовольняють його потреби.

У процесі реалізації цієї рекомендаційної системи було створено дві онтології за допомогою редактора онтологій і фреймворку для побудови інтелектуальних систем Protege 5.1.0: <<http://www.semanticweb.org/solomka/ontologies/specialitiesPasports>> та <<http://www.semanticweb.org/solomka/ontologies/subjects>>.

<<http://www.semanticweb.org/solomka/ontologies/specialitiesPasports>> – онтологія всіх паспортів спеціальностей, які входять до таких галузей науки: технічні науки (інформатика), фізико-математичні науки (математика, інформатика і кібернетика) згідно з Наказом Вищої атестаційної комісії України про Перелік спеціальностей, за якими проводяться захист дисертацій на здобуття наукових ступенів кандидата наук і доктора наук, присудження наукових ступенів і присвоєння вчених звань.

Вона задає ієрархію класів відповідних галузей науки, спеціальностей, що входять до цих галузей науки, та напрямів дослідження цих спеціальностей (рис. 1).

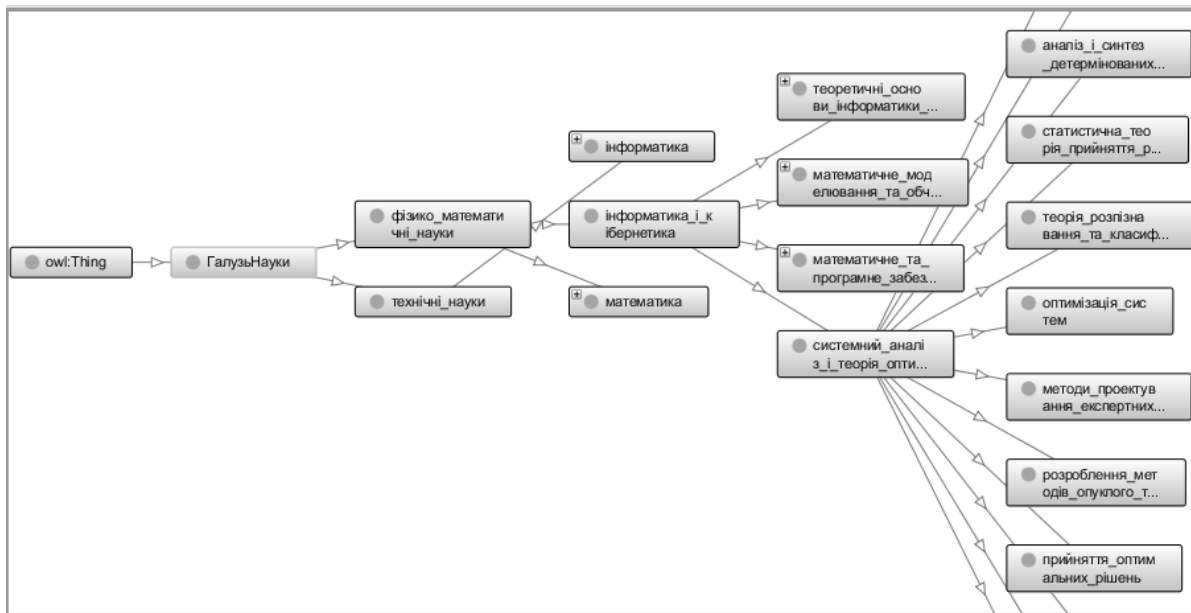


Рис. 1. Діаграма класів specialtiesPasports онтології (фрагмент)

<<http://www.semanticweb.org/solomka/ontologies/subjects>> – онтологія предметів, які викладають на факультеті інформатики НаУКМА. Іменовані класи кожного предмета визначаються за допомогою задання аксіом анонімних суперкласів, що фактично визначають такі характеристики предмета: напрями дослідження, які вивчає цей предмет (їх може бути декілька), загальна кількість годин, кількість годин на тиждень, кількість кредитів, кафедра, до якої належить предмет, семестр викладання предмета та тип предмета.

В онтологію <<http://www.semanticweb.org/solomka/ontologies/subjects>> було імпортовано онтологію <<http://www.semanticweb.org/solomka/ontologies/specialitiesPasports>> для визначення зв'язків між предметами та напрямками дослідження, які вони вивчають.

Логіка застосування зумовила таку архітектуру проекту (рис. 2):

ua.com.yaremko.system.view – пакет, який містить класи, відповідальні за створення графічного інтерфейсу плагіну: створення нового ViewComponent у системі Protege, який буде розташований у WorkspaceTab;

ua.com.yaremko.system.view.panel – пакет, що містить панелі, що відображаються на Protege View;

ua.com.yaremko.system.tab – пакет, що містить клас SubjectsRecommenderTab.java, відповідальний за створення WorkspaceTab – нового табу в системі Protégé.

src/main/resources – містить xml-файли, що забезпечують створення відповідних компонентів плагіну у системі Protégé та їх правильне розташування в системі.

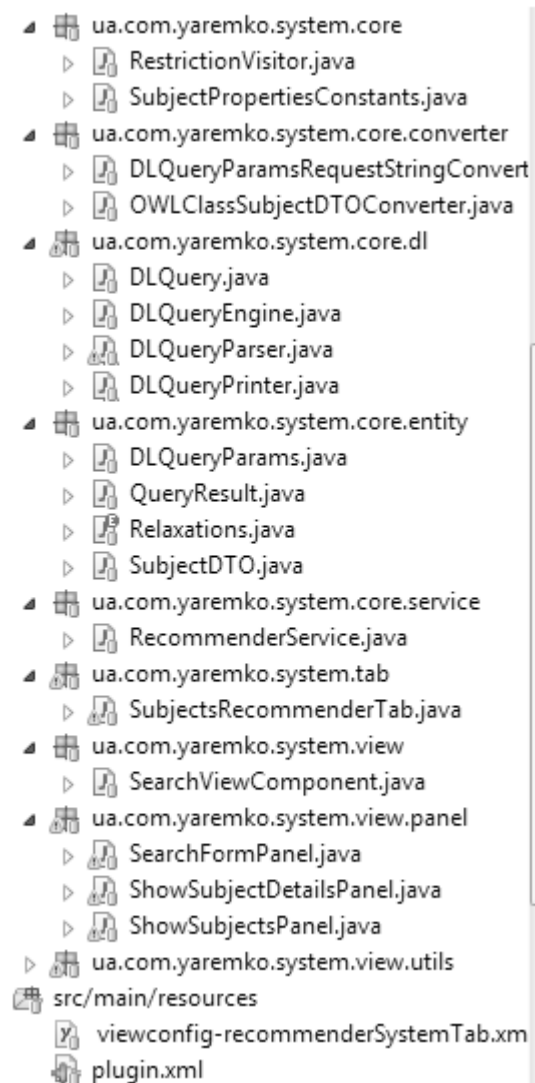


Рис. 2. Структура рекомендаційної навчальної системи

plugin.xml – файл, що містить визначення способу, яким плагін буде розширювати можливості системи Protege.

ua.com.yaremko.system.core – пакет, який містить бізнес-логіку застосування.

Пакет ua.com.yaremko.system.core.service містить клас RecommenderService.java. Це основний сервіс застосування, відповідальний за здійснення процесу знаходження відповідних рекомендацій користувачу та подання їх у зручній для відображення формі.

Метод processQuery(DLQueryParams dlQueryParams) сервісу на вхід отримує об'єкт, що містить параметри-критерії пошуку, задані користувачем. Він здійснює перетворення цього об'єкта в запит мови запитів до онтологій DLQuery, що базується на Manchester OWL синтаксисі. Після цього здійснюється DLQuery – запит до активної, проаналізованої мислителем онтології у системі Protege, що повертає набір OWL-класів, що відповідають обмеженням запиту.

Якщо результат пустий, то здійснюється відкидання обмежень, заданих користувачем, доки хоч один предмет не задовольнить запит (рис. 3).

Обмеження відкидаються в такому порядку: Тип предмета; Напрямок дослідження; Напрямок дослідження і Тип предмета; Напрямок дослідження Тип предмета і Спеціальність; Напрямок дослідження Тип предмета Спеціальність і Семестр.

```
public QueryResult processQuery(DLQueryParams dlQueryParams) {
    QueryResult queryResult = new QueryResult();

    String dlQueryRequest = DLQueryParamsRequestStringConverter.fromDLQueryParamsToRequest(dlQueryParams);
    Set<OWLClass> recommSubjects = dlQuery.getSubClassesSet(dlQueryRequest, true);

    if (recommSubjects.size() == 1 && recommSubjects
        .contains(owlDataFactor.getOWLClass(IRI.create(SubjectPropertiesConstants.NOTHING_CLASS_IRI)))) {
        queryResult = processRestrictionsRelaxation(dlQueryParams, recommSubjects);
    } else {
        queryResult.setOwlClasses(recommSubjects);
    }
    return queryResult;
}
```

Рис. 3. Код методу processQuery(DLQueryParams dlQueryParams) класу RecommenderService.java

Пакет ua.com.yaremko.system.core.dl містить класи, відповідальні за здійснення DLQuery запиту та повернення його результату у вигляді Set<OWLClass> owlClasses.

DLQuery.java – клас, відповідальний за надання методів для здійснення запиту;

DLQueryPrinter.java – відповідальний за повернення описів класів-результатів (OWLClass) у короткій формі (лише назва класу).

DLQueryParser.java – клас, відповідальний за перетворення рядка-запиту у OWLClass Expression.

DLQueryEngine.java – основний клас, що безпосередньо здійснює DLQuery – запит до активної, проаналізованої мислителем онтології у системі Protege.

Пакет ua.com.yaremko.system.core.entity містить основні DTO-об'єкти, необхідні для передачі даних між компонентами застосунку.

Пакет ua.com.yaremko.system.core.converter містить класи-конвертери:

DLQueryParamsRequestStringConverter.java – клас, відповідальний за перетворення DTO-об'єкта DLQueryParams, що містить усі обмеження користувача, у рядок, що задовольняє Manchester OWL – синтаксис мови запитів до онтологій DLQuery.

OWLClassSubjectDTOConverter.java – клас, що відповідає за перетворення OWLClass-об'єкта в SubjectDTO – об'єкт, що містить усі необхідні для відображення користувачеві дані про предмет у зручному форматі. Цей клас використовує у своїй роботі RestrictionVisitor.java, що є реалізацією патерна Visitor. Його робота полягає у «відвідуванні» всіх existential restrictions (some) і hasValue restrictions (value) кожного іменованого класу предмета, які задають його анонімні суперкласи, та збирає відповідно всі DataProperty, ObjectProperty та їхні значення.

Пакет ua.com.yaremko.system.core складається з класу RestrictionVisitor.java та класу

SubjectPropertiesConstants.java, що містить використовувані в програмі константи.

Для реалізації заданого функціоналу було використано OWL API (4.2.6) – Java – інтерфейс із відкритим кодом і реалізацією для W3C OWL, та Protégé-OWL API (5.0.0) – Java – бібліотека для мови веб-онтологій (OWL) і RDF (S) з відкритим вихідним кодом.

У процесі реалізації рекомендаційної системи вибірково дисциплін для студентів за явно заданими критеріями пошуку було виявлено проблему відсутності навчальних планів цих

дисциплін з уніфікованою чіткою структурою, які можна було б використати для встановлення міждисциплінарних зв'язків між предметами для ефективного здійснення рекомендацій, використовуючи можливості дескриптивної логіки. Для вирішення цієї проблеми було взято паспорт спеціальностей згідно з Наказом Вищої атестаційної комісії України про Перелік спеціальностей, за якими проводяться захист дисертацій на здобуття наукових ступенів кандидата наук і доктора наук, присудження наукових ступенів і присвоєння вчених звань для встановлення цих зв'язків. Подальшими кроками розвитку системи є встановлення міждисциплінарних зв'язків на основі навчальних планів цих дис-

інформацію про Базові предмети, які рекомендовано прослухати перед початком вивчення цього предмета, якщо такі є, та Предмети-продовження цього курсу, якщо такі є.

Оскільки обов'язковим обмеженням є лише галузь науки, до якої входить предмет, обравши лише цей критерій пошуку, або обравши лише галузь знань та спеціальність, за допомогою ієрархічної структури онтології, що надає додаткову семантичну підтримку, система надасть рекомендації предметів, що вивчають напрями дослідження заданої спеціальності, або список предметів, що вивчають напрями дослідження всіх спеціальностей обраної галузі науки в разі задання лишень обмеження на галузь науки.

The screenshot shows a web application interface for a 'Subjects Recommendation System'. It features a table of recommended subjects, a search filter dropdown, and a detailed view of a selected subject.

№	Назва предмету	Кафедра
1	Паралельні_і_розподіленні_обчислення	кафедра_мережних_технологій
2	Тестування_підтримка_якості_програмного_забезпечення	кафедра_мультимедійних_систем
3	Технології_прикладного_програмування	кафедра_інформатики

Деталі предмету

Назва	Кафедра	Тип предмету	Семестр	Загальна к-сть годин	К-сть кредитів	К-сть годин на тиждень
Тестування_підтримка_якості_програмного_заб_х_систем	кафедра_мультимедійних_систем	нормативний	8	162.0	4.5	4.0

Вивчає: розробка_моделей_оцінювання_якості_інф_систем, методи_та_засоби_оцінювання_якості_та_оптимізації_програми, діагностування_та_забезпечення_надійності_АСК, методи_та_засоби_забезпечення_ефективності_надійності_комп_систем

Базові предмети: Розробка_програмного_забезпечення

Предмети-продовження:

Search filters: інформатика_і_кібернетика, математичне_та_програмне_забезпечення_обчислювальних_машин_..., стандартизація_інформаційних_систем, вільногоВибору

Buttons: Шукати, Reasoner active, Show Inferences

Рис. 4. Результат роботи системи: надання рекомендацій на основі заданих обмежень користувача

циплін після приведення їх до певної уніфікованої структури, а також використання бази даних студентів з інформацією про прослухані раніше дисципліни та оцінки, отримані за них, для покращення якості наданих студентові рекомендацій.

На рис. 4 наведено приклад надання рекомендацій з підбору курсів вільного вибору. Користувач задає обмеження, які рекомендації, надані системою, повинні задовольняти. У наведеному нижче прикладі система надає список рекомендацій на основі таких обмежень користувача: галузь науки, спеціальність, напрям дослідження, тип предмета, семестр та кількість кредитів, яку повинен мати предмет. Також приклад демонструє вигляд деталей опису одного з предметів, обраних зі списку рекомендацій. Окрім повного опису курсу: Назва, Кафедра, Тип предмета, Семестр, Загальна к-сть годин, К-сть кредитів, К-сть годин на тиждень, Список напрямів дослідження, які вивчає цей предмет, – система надає

Висновки

Отже, рекомендаційна система – це система, яка скеровує користувача в персоніфікованому напрямі до цікавих або корисних йому елементів у значній масі можливих варіантів.

Для вирішення проблем, з якими стикнулися колаборативні рекомендаційні системи та системи на основі вмісту, а також для надання рекомендацій у разі складної предметної галузі та необхідності явного задання критеріїв пошуку використовують семантичні рекомендаційні системи, що підвищує ефективність наданих користувачеві рекомендацій.

Провівши аналіз можливостей формальної мови OWL, можна стверджувати, що її використання в процесі створення онтологій є доречним та ефективним завдяки наданню додаткової семантичної підтримки та можливості здійснення логічного виведення.

Також у статті зроблено детальний опис функціонування, структури та реалізації системи надання студентів рекомендацій щодо вибірових дисциплін на основі явно заданих ним

обмежень із використанням онтологій. Обґрунтовано обраний тип та формат рекомендаційної системи. Зазначено основні проблеми її реалізації та перспективи подальшого розвитку.

Список літератури

1. Глибовець А. М. Агенти для рекомендацій у колаборативних середовищах / А. М. Глибовець, С. С. Гороховський, А. А. Піка // Наукові праці МДУ ім. Петра Могили. Комп'ютерні технології. – 2010. – Вип. 121, т. 34. – С. 142–151.
2. Глибовець М. М. Розробка агентної рекомендаційної системи віртуального університету засобами JADE / М. М. Глибовець, О. В. Конюшенко // Проблеми програмування. – 2015. – № 1. – С. 104–115.
3. Codina V. A Recommendation System for the Semantic WEB [Electronic resource] / Victor Codina and Luigi Ceccaroni // Proceedings of the International Symposium on Distributed Computing and Artificial Intelligence (pp. 45–52). September 7–10, 2010. – Valencia, Spain : Springer. – Mode of access: <http://www.lsi.upc.edu/~vcodina/papers/dcai10>. – Title from the screen.
4. Ekstrand M. D. Collaborative Filtering Recommender Systems [Electronic resource] / Michael D. Ekstrand, John T. Riedl, Joseph A. Konstan. – Mode of access: <http://files.grouplens.org/papers/FnT%20CF%20Recsys%20Survey.pdf>. – Title from the screen.
5. Noy N. F. Ontology Development 101: A Guide to Creating Your First Ontology [Electronic resource] / Natalya F. Noy and Deborah L. McGuinness // Stanford Knowledge Systems Laboratory Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report SMI-2001-0880, March 2001. – Mode of access: http://protege.stanford.edu/publications/ontology_development/ontology101.html. – Title from the screen.
6. OWL Web Ontology Language [Electronic resource]. – Mode of access: <https://www.w3.org/TR/owl-guide/>
7. Protégé – OWL API Programmer's Guide [Electronic resource]. – Mode of access: https://protegewiki.stanford.edu/wiki/ProtegeOWL_API_Programmers_Guide
8. Vozalis E. Analysis of Recommender Systems [Electronic resource] / Emmanouil Vozalis, Konstantinos G. Margaritis // 9th Panhellenic Conference in Informatics. Thessaloniki, 21–23 November 2003. – Mode of access: <https://pdfs.semanticscholar.org/6543/1c3a27f01e70bb5e4b1e68fc6a91249f6114.pdf>. – Title from the screen.
9. The description logic handbook: Theory, implementation, and applications / ed. by Franz Baader, Deborah L. McGuinness, Daniele Nardi, Peter F. Patel-Schneider. – New York, NY : Cambridge University Press, 2003. – 505 p.

O. Zhezherun, S. Yaremko

CONSTRUCTION OF THE RECOMMENDER STUDYING SYSTEM BASED ON ONTOLOGIES

Traditional types of recommender systems – content-based filtering and collaborative filtering – have a number of problems: excessive specialization of domain-based recommendations, a cold start, scalability, data sparsity, and the unusual user problem. Ontology-based recommender systems allow to avoid some of them due to the semantic reasoning – capabilities that are provided by the description of logic reasoners that conduct a semantic analysis of ontologies in order to improve the quality and truthfulness of recommendations. This is a special type of recommender systems based on the clear knowledge of the elements intended for recommendations, user needs, and recommendation criteria.

The formal basis of the OWL – ontologies description language – are description logics that provide the possibility of logical reasoning.

Taking into consideration the fact that the Protégé ontology editor uses the OSGI framework as a plugin infrastructure, which allows programmers to integrate plugins that can be displayed as separate tabs, it was decided to implement the system of giving student recommendations on choosing of selective disciplines of the NaUKMA informatics department on the basis of explicitly prescribed student restrictions in the form of WorkspaceTab – Plug-in Protégé 5.1.0 using the Java programming language. Since the search for recommendations should be based on the clearly defined user constraints, the system is implemented as a constraint-based recommender system based on ontologies.

The functionality, structure, implementation, the main problems that were encountered during the system implementation, and the prospects for its further development are identified in the article.

Keywords: Recommender system, recommendations, studying system, ontology, descriptive logic, OWL, ontology editor, Protégé, plugin.

Матеріал надійшов 19.10.2017