

## ФАБРИКА ОНТОЛОГО-КЕРОВАНИХ ІНФОРМАЦІЙНО-ПОШУКОВИХ СИСТЕМ

*Глибовець Микола, Шабінський Антон*

Національний університет «Києво-Могилянська академія»

### Анотація

*У роботі запропоновано використання методології компонентно-орієнтованої програмної інженерії при розробці онтолого-керованих інформаційно-пошукових систем. У якості інструментального засобу наведено реалізацію фабрики програмного забезпечення, яка орієнтована на дослідження та розробку онтолого-керованих інформаційно-пошукових систем.*

### Abstract

*This paper proposes application of component-based software engineering methodologies to ontology-driven information retrieval systems development. Implementation of software factory of ontology-driven information retrieval systems is described.*

### Вступ

Проблема становлення інформаційного пошуку як інженерної дисципліни була озвучена Норбертом Фюром на щорічній конференції SIGIR у 2012 році. Як зазначив Н. Фюр, для вирішення проблеми необхідно досягти збалансованого поєднання розвитку теоретичної бази, формулювання гіпотез та експериментальних досліджень. [1] В рамках реалізації даної стратегії нами досліджується проблема інженерії автономних онтолого-керованих інформаційно-пошукових систем (ОКПС). Використання онтологій є перспективним методом інтелектуалізації інформаційного пошуку.

### Постановка проблеми

Проблематика розробки методів та засобів інженерії онтолого-керованих інформаційно-пошукових систем (ОКПС) розглядається за двома напрямками:

- 1) дослідження та розробка моделей інформаційного пошуку та інформаційно-пошукової інфраструктури;
- 2) використання методів програмної інженерії у процесі проектування та розробки ОКПС.

Наше завдання полягає у розробці інструментальних засобів інженерії ОКПС. За першим напрямком нами були запропоновані відповідні моделі та методи [2, 3, 4], які підлягають інтеграції у процеси інженерії кінцевих систем.

### Пропонований підхід

Для швидкого прототипування, розробки та експериментальних досліджень ОКПС перспективним підходом є використання фабрик програмного забезпечення (ПЗ). Фабрика програмного забезпечення (software factory) – це методологія компонентно-орієнтованої програмної інженерії [5, р. 608], модель організації процесу розробки програмних продуктів, та інструментальні засоби, що реалізують принципи конвеєрного виробництва у індустрії програмного забезпечення. [6] Фабрика як інструментальний засіб є середовищем розробки (integrated development environment, IDE) разом із набором програмних активів (software assets), яке попередньо налаштоване на деякий збірковий (assembly) процес розробки сімейства програмних продуктів. Такі сімейства програмних продуктів називають лініями програмного забезпечення. За визначенням Інституту програмної інженерії Університету Карнегі-Меллон, лінія програмного забезпечення – це «набір програмних систем, що поділяють спільний керований набір рис, які

задовольняють вимогам деякого сегменту ринку або деякого завдання і є розробленими зі спільного набору програмних активів заздалегідь визначеним способом».

Фабрика ОКІПС орієнтована на розробку сімейств ОКІПС як ліній програмних продуктів. Архітектура фабрики визначає взаємодію підсистем та компонентів ОКІПС: початкова обробка документів, індексація, ранжування, інтерпретація інформаційної потреби (запитів користувача), представлення результатів. Ключовою складовою інформаційно-пошукової системи є модель інформаційному пошуку. У нашому випадку це покращена тематична векторна модель eTVSM із спеціальною тематично-сутнісною онтологією [3], проте процеси розробки ОКІПС є абстрактними відносно моделей чи їх конкретних реалізацій. У фабриці ОКІПС модель інформаційного пошуку є лише компонентом, який разом з іншими компонентами збирається у кінцеву систему. Компонент моделі eTVSM є повністю незалежним від решти інфраструктури фабрики і придатний до повторного використання у будь-яких системах.

Ядром нашої реалізації фабрики ОКІПС є розроблений нами програмний фреймворк Falco.NET, побудований на платформі Microsoft .NET і мові C#. Інфраструктура фабрики базується на принципах інверсії контролю через ін'єкції залежностей та трансформації XML-моделей за принципами модельно-керованої архітектури. [7, 8] Компонентно-орієнтована парадигма у фабриці реалізована фреймворком NReco 2.0 [9]. Основною базовою реалізацією, що постачається разом із фабрикою, є реалізація моделі інформаційного пошуку eTVSM [10] у вигляді повторно вживаного компоненту NEtvsm. Для цієї базової реалізації моделі інформаційного пошуку постачаються також базові реалізації інших компонентів: початкової обробки текстових документів, індексації, інтерпретування інформаційної потреби згідно із [4], а також компонент для видобування онтологій з ймовірнісних тематичних моделей Noesis.TM.

Цілісний комплекс фабрики ОКІПС представлений такими інструментальними засобами:

- інтегроване середовище розробки Visual Studio 2013;
- репозиторій пакетів NuGet [11];
- інфраструктурний фреймворк NReco;
- основний інформаційно-пошуковий фреймворк Falco.NET;
- базова реалізація моделі інформаційного пошуку NEtvsm;
- інструментарій для обробки онтологій та базова реалізація компонента видобування онтологій для NEtvsm – компонент Noesis.TM.

Фреймворк NReco доступний як пакет у репозиторії NuGet. Фреймворк Falco.NET, компонент NEtvsm та компоненти проекту Noesis також розроблені у вигляді пакетів репозиторію. Фреймворк Falco.NET інкапсулює усі залежності та конфігураційні файли для інфраструктури: при встановленні пакету Falco.NET автоматично встановлюється решта пакетів, що містять конфігурацію та необхідні базові реалізації.

Основний сценарій використання фабрики передбачає суто збірковий процес розробки. Для виробництва різних ОКІПС як лінії ПЗ достатньо замінити ті чи інші компоненти, завантаживши їх до проекту та змінивши конфігурацію. Оскільки конфігурація у NReco здійснюється через XML-файли, то можна заздалегідь підготувати різні конфігураційні файли. Якщо всі необхідні компоненти постачаються через репозиторій NuGet, то конфігурування фабрики повністю автоматизується і полягає лише у заміні конфігураційних файлів, а всі потрібні залежності середовище розробки Visual Studio розв'язує автоматично. Програмування фабрики не вимагається, хоча і є можливим за потреби. Програмування необхідне лише для розробки нових компонентів.

Використання фабрики спрощує процес дослідження та розробки ОКІПС. По-перше, спрощується розробка предметно-орієнтованих пошукових систем та робота із різними форматами документів. Наприклад, одну систему можна швидко адаптувати під роботу із відеофайлами замість текстових документів. Більше того, використання базової

реалізації NExtsm дозволяє розробляти багатформатні ОКПС, які можуть одночасно працювати із різними форматами – наприклад, порівнювати за змістом відеофільми та тексти. Реалізуючи компонент для розпізнавання графічних зображень хімічних сполук можна створити предметно-орієнтовану систему пошуку статей з хімії. З іншого боку, можна досліджувати особливості індексації певних типів документів, маючи кілька реалізацій компонента індексації та підставляючи ці реалізації в одну систему. Аналогічно, можна досліджувати використання різних типів онтологій, із різним ступенем виразності тощо.

### Висновки

Таким чином, нами розроблено інструментальний засіб – фабрику ПЗ для побудови ОКПС. Фабрика реалізована згідно з теоретичними моделями та методами, розробленими у рамках дослідження проблеми інженерії ОКПС. Фабрика передбачає напрацювання бази компонентів, зокрема через репозиторій NuGet, які зрештою можна використовувати у різноманітних конфігураціях для створення сімейств ОКПС.

### Список використаних джерел:

1. Fuhr N. Information Retrieval as Engineering Science / Norbert Fuhr // SIGIR Forum / Association for Computing Machinery. – New York, 2012. – Vol. 46. – № 2. – P. 19–28.
2. Глибовец Н. Н. Применение онтологий и методов текстового анализа при создании интеллектуальных поисковых систем / Николай Глибовец, Андрей Глибовец, Антон Шабинский // Проблемы управления и информатики / Институт кибернетики им. В.М.Глушкова. – Киев, 2011. – № 6. – С. 96–102.
3. Шабінський А. Змішана тематично-сутнісна онтологія у покращеній тематичній векторній моделі / Антон Шабінський // Матеріали дев'ятої міжнародної науково-практичної конференції УкрПРОГ'2014. (Київ, 20 – 22 травня 2014). – Київ, 2014. – С. 182–187.
4. Глибовець М. Онтологічно-орієнтована інтерпретація інформаційної потреби у покращеній тематичній векторній моделі / Микола Глибовець, Антон Шабінський // Матеріали міжнародної конференції «Інтелектуальні системи прийняття рішень та проблеми обчислювального інтелекту ISDMCI 2014» (Залізний Порт, Україна, 28 – 31 травня 2014). – Херсон, 2014. – С. 53–55.
5. Vliet H. Software Engineering: Principles and Practice / Hans van Vliet // New York: Wiley, 2007. – 740 pages.
6. Aaen I. The Software Factory: Contributions and Illusions / Ivan Aaen, Peter Bøttcher, Lars Mathiassen // Twentieth Information Systems Research Seminar. – Oslo, 1996.
7. Glibovets N. Simplified infrastructure for the transformation of XML models / Nikolai Glibovets, Vitaliy Fedorchenko // Cybernetics and Systems Analysis / Springer. – New York, 2014. – Vol. 46. – № 1. – P. 93–97.
8. Федорченко В. Каркас для підтримки модель-орієнтованої розробки на основі спрощеної інфраструктури трансформації XML-моделей / Віталій Федорченко // Наукові записки НаУКМА / Вид-во «Кієво-Могилянська академія». – Київ, 2008. – Т. 86. – С. 61–65.
9. nReco: Innovative model-driven approach for web applications [Онлайновий]. Доступ: <http://www.nreco.site.com>. [Дата звернення: 25 червня 2014].
10. Kuroppa D. Modelle zur Repräsentation natürlichsprachlicher Dokumente / Dominik Kuroppa // Berlin: Logos Verlag, 2003. – 264 pages.
11. NuGet Gallery [Онлайновий]. Доступ: <https://www.nuget.org>. [Дата звернення: 25 червня 2014].