УДК.004.415

к.т.н., доц. **Огнєвий О.В.** (ХмНУ)
Барабаш С.О. (ХмНУ)
Ряба Л.О. (ВІКНУ)

МЕТОД СТРУКТУРНОГО СИНТЕЗУ WEB – ДОДАТКІВ

Метою даної статті є розробка підходу до автоматизованої компонентної збірки Web-додатків. Проведено аналітичний огляд сучасного стану підходів, методологій і методик проектування інформаційних систем. Показано, що задача забезпечення інтероперабельності інформаційних систем і їх компонентів є актуальною і в цьому напрямку працюють вчені в усьому світі. Було встановлено, що існуючі підходи до проектування інформаційних систем, еталонні та прикладні моделі відкритих інформаційних систем, а також існуючі комбінаторно-логічні методи структурного синтезу вимагають подальшого розвитку в аспектах, що стосуються компонентної збірки інформаційних систем і, зокрема, Web-додатків.

Виконаний аналіз процесу побудови стандартизованого профілю інформаційної системи. Виконано формалізацію семантичної моделі стандартизованого профілю ІС, а також на її основі формалізована модель бази знань онтологічного типу. Запропонована модель включає 3 шари: шар специфікацій, шар предметної області, шар інформаційної системи. Модель визначає загальний каркас, в рамках якого можуть бути побудовані прикладні семантичні моделі, які застосовуються безпосередньо на практиці. Запропоновано етапи інформаційних перетворень процесу компонентної збірки Web-додатків. Представлена загальна схема автоматизованої системи, яка включає підсистеми первинного аналізу та форматування даних, підсистеми управління сховищем даних про процеси і компонентах, а також підсистеми, які формують компонентний склад Web-додатки. Запропоновано вдосконалений метод компонентної збірки Web-додатків, що включає три етапи: 1) вибір процесів; 2) вибір компонент; 3) формування файлового складу Web-додатку. Для організації механізмів введення / виведення була використана агентна схема управління запитами до бази знань стандартизованого профілю ІС, на якій базується автоматизована система компонентної збірки Web-додатків. Крім того, запропонований алгоритм перетворення даних про процеси і компоненти Web-додатку із загальної моделі, яка використовується для опису предметної області на стороні сервера додатків в модель бази знань онтологічного типу. Реалізована система запитів до бази знань, що дозволяє формувати стандартизований профіль ІС. Запропонований каркас визначає базові концепти і ролі, які використовуються при описі.

Ключові слова: Web-додаток, семантична модель, компонентна збірка, автоматизована система.

Вступ. З метою покращення збірки web-додатків розробники намагаються знайти рішення, яке дозволить максимально ефективно використовувати існуючу компонентну базу, що збільшить швидкість збірки та полегшить редагування проекту на пізніх етапах його створення.

Однак кожний продукт має власний метод опису залежностей для подальшої компонентної збірки проекту.

Рішення підтримки компонентної збірки проектів потребує чіткого визначення, опису та категоризації компонентів і модулів за допомогою семантичних мереж [1]. Використання таких мереж дозволить провести чіткий зв'язок між поставленими задачами та оптимальними рішеннями. Головна ідея - це сприйняття кожного логічного процесу, як окрему задачу, а програму, як існуюче рішення. Дотримуючись такої схеми, можна прискорити пошук рішень для конкретних задач [2].

З врахуванням вищесказаного, розробка методу компонентної збірки Web-додатків є актуальним науково-технічним завданням, вирішенню якого присвячена дана магістерська робота.

Постановка задачі. Існуючі підходи до проектування інформаційних систем, еталонні та прикладні моделі відкритих інформаційних систем, а також існуючі комбінаторно-логічні методи структурного синтезу вимагають подальшого розвитку в аспектах, що стосуються компонентної збірки інформаційних систем і, зокрема, Web-додатків [3].

У результаті проведеного огляду доведено актуальність завдання компонентної збірки Web-додатків, яке потребує розбудови семантичних моделей, а також інформаційно-пошукових алгоритмів [4].

Для подальшої роботи була сформульована мета розробки семантичної моделі процесів і компонент Web-додатки і заснованих на ній інформаційно-пошукових алгоритмів компонентної збірки, що дозволяють знизити тимчасові витрати і скоротити число помилок на етапі структурного синтезу Web-додатків.

Основна частина. Web-додаток можна сміливо віднести до інформаційної системи (ІС). Тому представлену ІС можна розглядати, як сукупність компонентів та зв'язків між ними. Важливим етапом на шляху до розробки системи компонентної збірки web-додатків є структурний синтез [5].

Базова модель автоматизованої системи не акцентує уваги на характеристиках відкритості, тому вона була доповнена узагальненою структурою інформаційної системи (рис. 1).

Рис. 1. Загальна структура відкритої автоматизованої інформаційної системи

Загальна структура відкритої автоматизованої системи представлена у вигляді інтерфейсів API, внутрішніх і міжрівневих протоколів та служб, які згруповані за двома внутрішніми рівнями системи і одному зовнішньому.

Будь-який стандартизований профіль має перелік характеристик, в тому числі і профіль, який визначає інформаційну систему [6].

Щоб сформувати стандартизований профіль інформаційної системи необхідно:

- список сервісів, які виступають в якості точки об'єднання для двох інших сервісів,
- повний профіль процесу, який визначений усіми своїми інтерфейсами,
- підмножина інтерфейсів, які функціонують на одному або більше рівні сумісності.

В одній предметній області може спокійно функціонувати декілька альтернативних інформаційних систем, які направленні на вирішення однакових задач, але такі системи мають різне архітектурне рішення і компонентну складову [7]. Системи, що автоматизують подібні процеси можуть мати різні характеристики програмно-апаратної складової і в кінцевому результаті відрізнятися в структурному і функціональному аспектах. Процеси і компоненти інформаційних систем можуть добре формалізуватися.

Актуальність використання семантичних мереж у нашому проекті дуже велика і визначається тим, що такі мережі дозволяють формалізувати предметні області, як сукупність процесів, що автоматизуються за допомогою компонентів [4]. Взявши за основу семантичну

модель, можна побудувати інформаційне середовище для зберігання і редагування інформації про процеси і компоненти інформаційної системи.

У нашій роботі використовується модель, для профілю інформаційної системи (на основі онтологічного каркасу) [8, 9]. Базовий шар каркасу буде формуватися за рахунок класифікаторів і специфікацій. На верхньому рівні каркасу буде знаходитися концепт Document, який став основою для подальших наслідувань специфікацій і стандартів. Все документи поділені на два рівні: рівень еталонних документів і рівень прикладних документів.

Перший рівень налічує декілька концептів: Classifier, ModelClassifier, InteroperabilityModelClassifier і Reference. Така модель може розширюватися на будь-якому рівні розробки, за допомогою додавання у її склад класифікаторів кожної групи, також можна додавати і самі групи, які не будуть вносити серйозних змін до структури.

На другому рівні з'являється група концептів, яка виконує формалізацію прикладних документів. Для визначення множини всіх документів використовується концепт AppDocument, його наслідує InterfaceDocument, який виділяє лише документи, що мають регламентовану поведінку для кожного інтерфейсу. Кожний документ має модульну структуру, яка задається за допомогою концепту InterfaceDocumentPart, кожний модуль може нести в собі певну кількість логічних груп параметрів ParameterSet (кожна група має власні параметри Parameters). Якщо частини документів починають класифікуватися хоча б за однією ознакою, то вони стають ClassifiedDocumentPart. Параметри, які регламентуються в нормативній документації, мають певну область значень, що задається доменом. Такий домен описується за допомгою концепту DomainMode. Виконуючи опис домену задається базовий системний тип SystemDataType, який буде визначати параметри і обмеження (Restriction). Один параметр може мати декілька визначених екземплярів DomainMode. Це необхідно для формування стандартизованих профілів ІС. Існує концепт (граничний) InterfaceProfile, який використовується для формування вибірки параметрів нормативних документів і режиму їх використання. Даний концепт є основою для інших концептів, які моделюють роботу функцій входу-виходу і способи взаємодії із системою. В залежності від системи, інтерфейси визначаються форматами і структурами даних, які у них передаються.

Процес компонентної збірки відбувається за допомогою виконання трьох основних стадій, які пов'язані з рівнем представлення: процесна (формування графа, який визначає функціональні особливості додатку), компонентна (стадія формування компонентів з їх залежностями, які реалізують процеси логічного рівня), структурна (формує план, по якому потрібні компоненти будуть імпортуватися і розміщатися у структурі проекту).

Розроблений вдосконалений метод компонентної збірки Web-додатків (рис. 2), відрізняється тим, що дозволяє відновлювати набори процесів і компонентів по їх частковому опису.

Він включає три етапи: 1) вибір процесів; 2) вибір компонентів; 3) формування файлового складу Web-додатку.

На першому кроці (вибір процесів) виконується перевірка типу запиту.

Якщо вхідні дані представляють собою ідентифікатор класу *Cl* - Web-додатку, то формується множина процесів, де *hasInsertedProcess* визначається продукційними правилами: p_i U *pj:* (p_i *hasWebAppClass Cl)* U (p_t *hasInsertedProcess* p_i).

На кожному кроці ітерації формується множина рекомендованих наборів процесів, які уточнюються. Введено параметр максимально можливого набору рекомендованих процесів *n*, який обмежує список альтернативних варіантів. В результаті визначається множина s^p семантичної мережі *S*, що містить опис цільових процесів з ієрархією вкладеності і потоками даних.

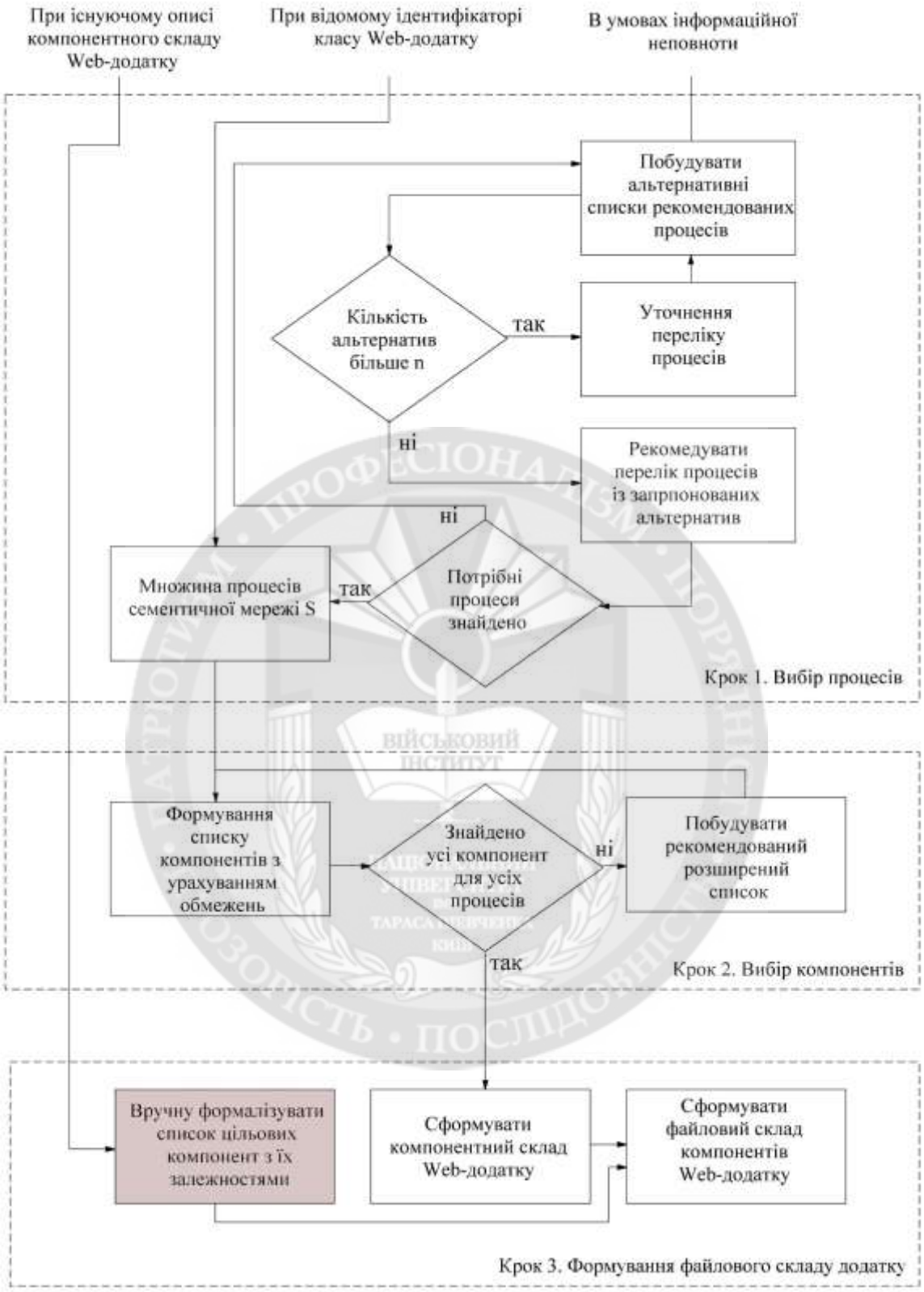

Рис. 2. Вдосконалений метод компонентної збірки Web-додатків

На другому кроці (вибір компонент) s^p розбивається на непересічні підмножини, для елементів яких визначено потоки передачі даних. На кожній ітерації формується список компонент з урахуванням обмежень: операційна система, платформа розробки, вартість. Список обмежень може бути розширено. якщо на поточному кроці список інтероперабельних компонентів не побудований, то формується рекомендаційні розширення обмежень і

ініціалізується наступна ітерація. Якщо список інтероперабельних компонентів був побудований, то виконується перехід до третього кроку, на якому відбувається формування файлового складу Web-додатку.

На етапі розробки архітектури системи онтологічна модель була спрощена. Замість опису кожного параметра, домену, режиму і т.д. пропонується описувати відповідні розділи документів і з ними пов'язувати множину ознак відповідно до системи класифікаторів. Сховище містить: електронні документи стандартів, специфікацій, стандартизованих профілів ресурсів і процесів, файли з описом властивостей електронних документів, файли з описом ресурсів і процесів, файли з описом класифікаторів.

У сховищі кожен електронний документ однозначно визначається Інтернет-адресою і може розташовуватися на будь-якому сервері в мережі. Класифікатори використовуються для групування електронних документів і їх частин за різними ознаками.

Для організації механізмів введення/виведення пропонується застосувати мультиагентну [10, 11] схему управління запитами до бази знань стандартизованого профілю. У кожної транзакції беруть участь 4 групи агентів:

1. Агенти призначеного для користувача інтерфейсу (ClientUI).
2. Агенти планування підготовки і розподілу завдань (TaskPlanTranslator).
3. Агенти, відповідальні за виконання запиту (TaskPlanAgent).
4. Агенти, що витягають дані з бази знань (за спеціалізацією).

Безпосередньо витяганням даних займаються агенти групи BPPerformerAgent. На етапі створення і ініціалізації клієнтської системи формуються і налаштовуються агенти (1 і більше для кожної групи), що відповідають за прийом запитів від користувачів і висновок підсумкових списків.

Агенти, що відповідають за прийом і попередню підготовку клієнтського запиту прослуховують спеціальний порт і переходять в стан «Попередня підготовка запиту», якщо в системній черзі є хоча б один запит. Далі, запит форматується відповідно до вимог агентів групи TaskPlanTranslator і передається їм на обробку. Слухач клієнтських запитів повертається в стан «Очікування запиту».

Агент групи TaskPlanTranslator, який перебував в стані «Очікування підготовленого запиту», після його отримання проходить ряд станів, в рамках яких виконується:

1. Формалізація завдання для відповідального виконавця (TaskPlanAgent).
2. Класифікуються агенти, вилучаючі дані з бази знань.
3. Створення відповідального виконавця - агента групи TaskPlanAgent і передача йому завдання. Завдання включає дескриптор клієнтської системи, від якої надійшла попередня формулювання.
4. Формується глобально-унікальний дескриптор завдання.
5. Передача дескриптора завдання агенту групи виведення підсумкового списку на стороні клієнта.

На рис. 3 показана блок-схема алгоритму перетворення даних із загальної GENERIC-моделі в Web-онтології, представлені в форматі OWL-DL. При роботі алгоритму виділимо наступні основні кроки:

1. Ініціалізація основних контейнерів даних: *classTree*, який містить плоске дерево назв ключових ресурсів (в подальшому, класів і індивідів); *individualsHas*, який містить зазначені списки карт, в яких записані конкретні властивості ресурсу (літерали); model-фінальна структура JENA; ініціалізація тимчасових карт ресурсів.
2. Послідовний розбір структури *classTree* для створення, з ключових ресурсів, класів в моделі JENA. Заповнення тимчасових карт ресурсів для спрощення навігації по плоскому дереву, і подальшого склеювання індивідуальних ресурсів з класами.
3. Створення властивостей предикатів на основі дерева. створюються предикати виду *hasX*, де *X*-назва ключового ресурсу (вже класу). всі класи послідовно перев'язуються предикатами.

4. Парсинг *individualsHas* для створення ресурсів індивідів в моделі JENA, організація плоского дерева індивідів з урахуванням предикатів їх класів.

5. Зв'язування всіх індивідів через предикати.

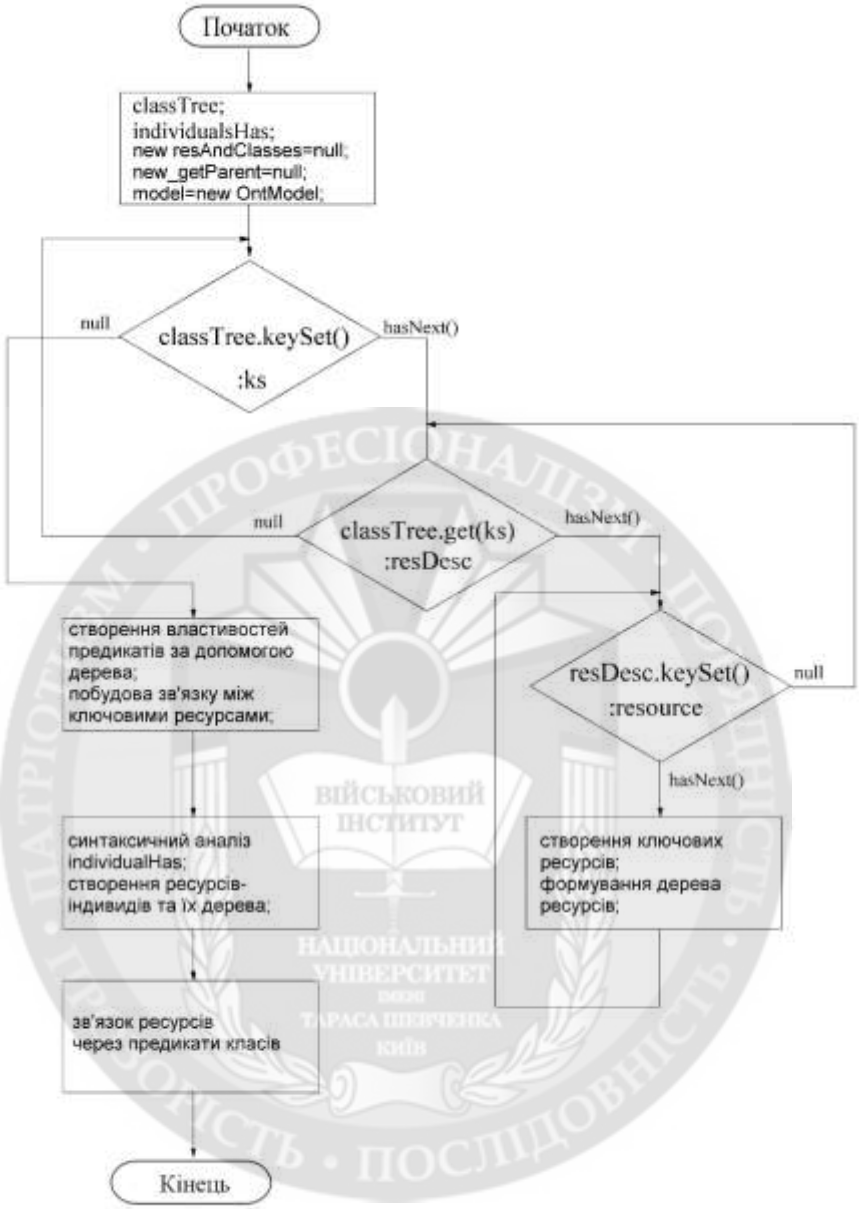

Рис. 3. Алгоритм перетворення даних із загальної GENERIC-моделі в Web-онтології

При побудові стандартизованого профілю автоматизованої системи компонентної збірки Web-додатків будемо базуватися на еталонній моделі OSE/RM [12].

Висновки. У статті розглянуто автоматизовану компонентну збірку Web-додатків. Кінцевими продуктами таких збірок є додатки, які відносяться до інформаційних систем (ІС), тому їм приділенно достатньо уваги. До компонентів ІС застосовується структурний синтез. Було розглянуто способи представлення такого синтезу та ряд математично-логічних моделей, які орієнтовані на вирішення задач компонентної збірки [4]. Результати проведеного дослідження зводяться до наступних пунктів:

1. Спочатку було проведено аналітичний огляд уже існуючих готових рішень для Web-проектів. Після визначення недоліків і постановки задачі їх вирішення, Web-додаток почав роглядатися, з точки зору інформаційної системи. Було показано, що існуючі підходи до

проектування інформаційних систем і моделі відкритих інформаційних систем потребують розвитку у аспекті компонентної збірки. На основі огляду було виконано обгрунтування актуальності поставленої задачі і сформована ціль - побудова алгоритму і моделі для компонентної збірки Webß-додатків.

Запропоновано вдосконалити метод компонентної збірки Web-додатків, який налічує 3 етапи: вибір процесів, вибір компонентів та формування файлового складу Web-додатку. Для організації механізмів введення / виведення була використана агентна схема управління запитами до бази знань стандартизованого профілю ІС, на якій базується автоматизована система компонентної збірки Web-додатків. Крім того, запропонований алгоритм перетворення даних про процеси і компонентах Web-додатку із загальної моделі (GENERIC), яка використовується для опису предметної області на стороні сервера додатків в модель бази знань онтологічного типу.

Побудований стандартизований профіль автоматизованої системи компонентної збірки Web-додатків. Всі включені специфікації класифіковані згідно еталонної моделі OSE / RM.

Практична реалізація роботи виконана у вигляді тестової системи, яка базується на трьох рівнях: рівень ядра, рівень базових операцій та функціональний рівень. Виконана перевірка системи запитів до мережі. Платформа забезпечую інтероперабельність із клієнтськими системами і може виконувати розширення. Взаємодія із усією сукупністю файлів виконується ніби ми працюємо із однією БД.

ЛІТЕРАТУРА:

1. Быстрая разработка программ. Принципы, примеры, практика. /Роберт Мартин, Джеймс Ньюкирк, Роберт Косс – Изд-во: Диалектика-Вильямс, 2004. – 752 с.

2. Ханссон Д. Х. Гибкая разработка веб-приложений в среде Rails. / Д. Х. Ханссон, Д. Томас – Санкт-Петербур - Питер, 2008. – 720с.

3. Салмре Иво. Конечный автомат для пользовательского интерфейса. Программирование мобильных устройств на платформе .NET Compact Framework. / Иво Салмре – Изд-во: Издательский дом "Вильямс", 2006. – 736с.

4. Аристов А.В. Разработка и исследование алгоритмов компонентной сборки Web-приложений на основе семантических сетей: дис. д-ра техн. наук: 05.13.11, / Аристов Алексей Владиславович. – К., 2016. – 125 с.

5. Гибкая разработка программ на Java и C++. Принципы, паттерны и методики. /Роберт С. Мартин, Джеймс Ньюкирк, Роберт Косс – Изд-во: Диалектика-Вильямс, 2016. – 704 с.

6. Муляр І.В. Аналіз підходів до структурної збірки web-додатків / І.В. Муляр, В.М Лоза, С.Б. Войнарович // Збірник наукових праць Військового інституту Київського національного університету імені Тараса Шевченка. – К.: ВІКНУ, 2017. – Вип. № 56. – С. 132- 138

7. Муляр І.В., Предметно-орієнтований погляд на розробку програмного забезпечення / І.В. Муляр, В.О.Браун, В.К. Шваб, Я.М. Проценко, Р.М. Глушко // Збірник наукових праць Військового інституту Київського національного університету імені Тараса Шевченка. – К.: ВІКНУ, 2017. – Вип. № 57. – С. 123- 133

8. Eric Evans, Domain-Driven Design Reference: Definitions and Pattern Summaries / Eric Evans – 1 st Edition - Dog Ear Publishing, 2014, 88 с.

9. Scott Millett, Patterns, Principles, and Practices of Domain-Driven Design / Scott Millett – 1st Edition – Wrox, 2015, 792 с.

10. Кузнецов М. PHP. Практика создания Web-сайтов. / М. Кузнецов, И.Симдянов – БХВ-Петербург - Москва, 2012. – 47 с.

11. Митчелл Скотт 5 проектов Web-сайтов от фотоальбома до магазина. / Скотт Митчелл – М.: НТ Пресс – Москва, 2013. – 224 с.

12. Фрейен Бен HTML5 и CSS3.Разработка сайтов для любых браузеров и устройств. / Бен Фрейен – Питер - Москва, 2014. – 304 с.

REFERENCES:

1. Martin, Robert, Nyukirk, Dzheyms and Koss, Robert (2004),” Byistraya razrabotka programm. Printsipyi, primeryi, praktika” [Rapid development of programs. Principles, examples, practice], Izd-vo: Dialektika-Vilyams, 752 p.

2. Hansson, D. X. and Tomas, D. (2008),” Gibkaya razrabotka veb-prilozheniy v srede Rails” [Flexible development of web applications in the Rails environment], Sankt-Peterbur - Piter, 720p.

3. Salmre, Ivo (2006),” Konechnyiy avtomat dlya polzovatelskogo interfeysa. Programmirovanie mobilnyih ustroystv na platforme .NET Compact Framework” [A state machine for the user interface. Programming mobile devices on the .NET Compact Framework], Izdatelskiy dom "Vilyams", 736p.

4. Aristov A.V. (2015)” Razrabotka i issledovanie algoritmov komponentnoj sborki Web-prilozhenij na osnove semanticheskih setej” [Development and research of algorithms for component assembly of Web-applications on the basis of semantic networks], Kiev, 125 p.

5 Martin, Robert, Nyukirk, Dzheyms and Koss, Robert (2016),” Gibkaya razrabotka programm na Java i C. Printsipyi, patternyi i metodiki.” [Flexible development of Java and C ++ programs. Principles, Patterns and Techniques], Izd-vo: Dialektika-Vilyams, 704p.

6. Muliar I.V., Loza V.M., and Vojnarovich S.B. (2017),” Analiz pidkhodiv do strukturnoji zbirky web-dodatkiv” [Analysis of approaches to the structural build of web-applications], Zbirnyk naukovyh prac' Vijs'kovogo instytutu Kyi'vs'kogo nacional'nogo universytetu imeni Tarasa Shevchenka, No. 56, pp. 132-138.

7. Muliar I.V., Braun V.O., Shvab V.K., Procenko Ya.M. and Glushko R.M. (2017),” Predmetno-oriyentovanij poglyad na rozrobku programnogo zabezpechennya” [Analysis of approaches to the structural build of web-applications], Zbirnyk naukovyh prac' Vijs'kovogo instytutu Kyi'vs'kogo nacional'nogo universytetu imeni Tarasa Shevchenka, No. 57, pp. 123-133.

8. Eric Evans (2014), Domain-Driven Design Reference: Definitions and Pattern Summaries,1 st Edition, Dog Ear Publishing

9. Scott Millett (2015), Patterns, Principles, and Practices of Domain-Driven Design– 1st Edition – Wrox, 792 c.

10. Kuznetsov, M. and Simdyanov, I. (2012),” PHP. Praktika sozdaniya Web-saytov” [PHP. The practice of creating Web sites], BHV-Peterburg - Moskva, 347p.

11. Mitchell, Skott (2013), ”5 proektov Web-saytov ot fotoalboma do magazina” [5 projects of Web sites from photo album to shop], M.: NT Press - Moskva, 224p.

12. Freyen, Ben (2014),” HTML5 i CSS3.Razrabotka saytov dlya lyubyih brauzerov i ustroystv” [HTML5 and CSS3. Development of sites for all browsers and devices], Piter - Moskva, 304p.

к.т.н., доц. Огневой О.В., Барабаш С.О., Рябая Л.А.

МЕТОД СТРУКТУРНОГО СИНТЕЗА WEB - ПРИЛОЖЕНИЙ

Целью данной статьи является разработка подхода к автоматизированной компонентной сборки Web-приложений. Проведен аналитический обзор современного состояния подходов, методологий и методик проектирования информационных систем. Показано, что задача обеспечения интероперабельности информационных систем и их компонентов является актуальной и в этом направлении работают ученые во всем мире. Было установлено, что существующие подходы к проектированию информационных систем, эталонные и прикладные модели открытых информационных систем, а также существующие комбинаторно-логические методы структурного синтеза требуют дальнейшего развития в аспектах, касающихся компонентной сборки информационных систем и, в частности, Web-приложений. Выполненный анализ процесса построения стандартизированного профиля информационной системы. Выполнена формализация семантической модели стандартизированного профиля ИС, а также на ее основе формализованная модель базы знаний онтологического типа. Предложенная модель включает 3 слоя: слой спецификаций, слой предметной области, слой информационной системы. Модель определяет общий каркас, в рамках которого могут быть построены прикладные семантические модели, применяемые непосредственно на практике. Предложено этапы информационных преобразований процесса компонентной сборки Web-приложений. Представлена общая схема автоматизированной системы компонентной сборкой, которая включает подсистемы первичного анализа и форматирования данных, подсистемы управления хранилищем данных о процессах и компонентах, а также подсистемы, которые формируют

компонентный состав Web-приложения. Предложено усовершенствованный метод компонентной сборки Web-приложений, включает три этапа: 1) выбор процессов; 2) выбор компонент; 3) формирование файлового состав Web-приложения. Для организации механизмов ввода / вывода была использована Агентная схема управления запросами к базе знаний стандартизированного профиля ИС, на которой базируется автоматизированная система компонентной сборки Web-приложений. Кроме того, предложенный алгоритм преобразования данных о процессах и компонентах Web-приложения с общей модели, используемой для описания предметной области на стороне сервера приложений в модель базы знаний онтологического типа. Реализована система запросов к базе знаний, позволяет формировать стандартизированный профиль ИС. Предложенный каркас определяет базовые концепты и роли, которые используются при описании.

Ключевые слова: Web-приложение, семантическая модель, компонентная сборка, автоматизированная система.

Ph.D. Ohnievyi O.V., Barabash S.O., Riaba L.O.

THE METHOD OF STRUCTURAL SYNTHESIS WEB - APPLICATIONS

The purpose of this article is to develop an approach to the automated component construction of Web-based applications. An analytical review of the current state of approaches, methodologies and design techniques of information systems was conducted. It is shown that the task of ensuring interoperability of information systems and their components is relevant and scientists in the whole world work in this direction. It was found out that existing approaches to the design of information systems, reference and applied models of open information systems, as well as existing ones combinatorial-logical methods of structural synthesis - require further development in aspects relating to the component construction of information systems and, in particular, Web-applications. An analysis of the process of constructing a standardized profile of information systems is carried out. The formalization of the semantic model of the standardized IC profile is performed, and also on its basis, has been formalized a model of knowledge base ontological type. Proposed model includes 3 layers: a layer of specifications, a layer of a domain, a layer of information systems. The model defines a general frame within which applied semantic models may be constructed. Those are applied directly in practice. It is introduced stages of information transforms of the process of a component construction of Web-applications. We present general scheme of the automated system, which includes the subsystems of the primary analysis and data formatting, subsystem that manages manages data of processes storage and components, as well subsystems that form part of the Web application. There is an advanced method of component assembly of Web-applications that includes three stages: 1) processes pick; 2) components pick; 3) the formation of the file composition of the Web-application. To organize the mechanisms of input / output we used an agent scheme of query management to the knowledge base of the standardized IC profile on which automated component assembly system for Web applications is based. Also we offer an algorithm for converting data about processes and components of a Web application from a general model that is is used to describe the domain domain to the server side of the application in the base model knowledge of ontological type. A system of queries to the knowledge base that allows you to formulate standardized IP profile is implemented. The proposed framework defines main concepts and roles that areused in the description.

Keywords: Web-application, semantic model, component assembly, automatedsystem.