

БАЗИ ДАНИХ, БАЗИ ЗНАНЬ
ТА ІНЖЕНЕРІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

УДК 004.75: 681.3

М.Л.Бутенко

**ВИКОРИСТАННЯ
АГЕНТНО-ОРІЄНТОВАНОГО
ПІДХОДУ ПРИ
ПРОЕКТУВАННІ СИСТЕМ
ЕЛЕКТРОННОГО
ДОКУМЕНТООБІГУ**

**Восточноукраїнський
національний університет**

**Кафедра автоматизації і
комп'ютерно-інтегрованих
технологій**

**Науковий керівник –
Морозова Т.Ю., д.т.н.,
доцент**

Використання агентних технологій дозволяє домогтися підвищення ефективності автоматизації бізнес-процесів в різних областях. Автором запропоновано використання агентно-орієнтованого підходу для задач електронного документообігу. У роботі зроблений аналіз існуючих агентних і агентно-орієнтованих методологій для використання в системах електронного документообігу. Розглянуто особливості побудови системи електронного документообігу як агентно-орієнтовані системи. Описана практична програмна реалізація

Использование агентных технологий позволяет добиться повышения эффективности автоматизации бизнес-процессов в различных областях. Автором предложено использование агентно-ориентированного подхода для задач электронного документооборота. В работе сделан анализ существующих агентных и агентно-ориентированных методологий для использования в системах электронного документооборота. Рассмотрены особенности построения системы электронного документооборота как агентно-ориентированной системы. Описана практическая программная реализация

Using agent-based technology allows for more effective business process automation in various fields. The author suggested the use of agent-oriented approach to electronic document management tasks. Made an analysis of existing agent-based methodologies for use in electronic document management systems. The author described features of the construction of electronic document management system as agent-based system. Describe a practical software implementation

Ключові слова: електронний документообіг, системи електронного документообігу, агентна система, агентно-орієнтований підхід, веб-технології, пошук документів, класифікація документів, зберігання документів

Вступ

У сучасних соціально-економічних умовах розвитку нашої держави ефективно управління підприємством неможливе без використання розвинених систем електронного документообігу (СЕД) підприємства. У зв'язку з цим, саме ринок інформаційних технологій і програмних продуктів, що дозволяють прискорити створення і, найголовніше, полегшити подальший супровід документів підприємства, найбільш бурхливо розвивається в сфері інформаційних технологій.

Більшість сучасних СЕД мають можливість працювати через глобальну мережу Інтернет, деякі – виключно зі застосуванням веб-технологій. Основою СЕД, яка працює через Інтернет, є спільна база документів (можливо розподілена, тобто фізично вона може перебу-

вати на різних серверах). Поняття „база документів” передбачає не лише базу даних, яка містить документи, а й такі варіанти, як звичайний файловий сервер, реляційна база даних, нереляційні бази даних, електронне сховище документів (Microsoft Exchange або Lotus Domino) тощо. На веб-сервері знаходиться веб-програма, що взаємодіє з базою документів. Цілком можливо, що використовується багатоланкова архітектура і в цьому ланцюзі присутній сервер програм. Користувач у своєму браузері (або спеціальному тонкому клієнті) отримує інтерфейс системи електронного документообігу та здійснює операції з документами.

Наведемо приклади деяких подібних систем, доступних на нашому ринку:

1. EMC Documentum

[<http://russia.emc.com/domains/documentum/index.htm>].

2. EOS ДЕЛЮ [www.eos.ru] (з модулем ДЕЛЮ-Web) або EOS for SharePoint.

3. Directum [<http://www.directum.ru>].

4. DocsVision [<http://www.docsvision.com>].

5. PayDox [<http://www.paydox.ru>].

Найсуттєвішим недоліком сучасних систем електронного документообігу є закритий характер, обмеженість їхніх ресурсів наявними у системі функціями. Зазначена характеристика зумовлена, по-перше, корпоративною бізнес-моделлю сучасних СЕД. Будь-яка компанія бажає прив'язати до себе користувачів та отримувати дохід. По-друге, слід відзначити невідпрацьованість технічних аспектів, використання різних технологій програмування і т. п. Всі другорядні причини неможливості взаємодії і об'єднання СЕД цілком вирішувані. Сучасні СЕД – це не тільки засоби зберігання та пошуку електронних документів, це й інтелектуальне середовище для створення, перетворення і супроводу документів [1]. Для планування і прогнозування роботи СЕД виникає необхідність побудови математичних (або імітаційних) моделей [2]. Відповідно, актуальним постає завдання побудови високоефективної відкритої системи електронного документообігу.

Одним із нових підходів до вирішення зазначеного завдання пропонується використання агентно-орієнтованого підходу (АОП) для проектування СЕД. У такому випадку СЕД є агентно-орієнтованою системою (АОС). Мета даного дослідження – обґрунтувати можливість використання агентно-орієнтованого підходу при проектуванні систем електронного документообігу.

Методології побудови агентно-орієнтованих систем

Головним елементом АОС є інтелектуальні агенти. Інтелектуальний агент – це програмний модуль, здатний до оперативного аналізу даних, адаптації до умов, що змінюються, активного обміну інформацією з іншими агентами з метою досягнення поставлених користувачем завдань.

У науковій літературі представлено основні можливості АОС [3], напрямки дослідження та розвитку агентних технологій [4; 5; 7]. Однак для використання в СЕД підходять далеко не всі методології побудови АОС. Найбільш адекватною з точки зору автора є методологія і набір інструментальних засобів MAS DK групи інтелектуальних систем Санкт-Петербурзького

інституту інформатики та автоматизації РАН, очолюваної В. Городецьким [6]. Ця методологія описує роботу багатоагентних систем (БАС), а розглянута нами СЕД все ж АОС (більш гнучка до складу програмних компонентів). Однак на дану методологію варто звернути увагу.

У MAS DK створено конструкцію „Типовий агент” (Generic Agent), що є базою для подальшої спеціалізації класів і структур даних, клонування екземплярів агентів та індивідуального доопрацювання специфічних компонентів. Для управління функціонуванням багатоагентної програми на всіх етапах життєвого циклу в системі MAS DK реалізовано концепцію „Системного ядра”.

Особливості системи електронного документообігу як агентно-орієнтованої системи

Опишемо деякі особливості СЕД як АОС. Документ у такій системі є інтелектуальним агентом, з усіма його властивостями і поведінкою. Важливо, що документ здатний змінити свій зміст в реальному часі без участі користувача.

Кількість агентів в нашій АОС змінюється з часом. Кількість типів агентів також не є постійною величиною. Обрана модель однорангової мережі агентів, хоча користувач може задавати пріоритет виконання завдань.

Агенти в АОС взаємодіють не тільки з іншими агентами, а й з іншими програмними компонентами системи. Також в агентно-орієнтованій СЕД документи-агенти можуть використовувати для своїх цілей звичайні документи і компоненти системи.

Однією з першочергових цілей при розробці подібних систем є простота і гнучкість створення нових агентів. Тому першочерговим завданням є докладний опис і програмна реалізація „Типового агента”, який не буде змінюватися протягом усього життєвого циклу СЕД.

Активність агента в СЕД полягає у створенні одного або кількох нових документів, які в свою чергу можуть бути агентами. Такий підхід може призвести до зациклення СЕД, чого не можна припустити. Тому необхідні спеціальні інструменти, що запобігають подібну поведінку.

Автономність агента в СЕД полягає в можливості використання інтелектуальний документ поза середовищем СЕД. Однак при цьому не реалізуються (навмисно, за вимогами безпеки) засоби комунікації агента. У цьому випадку агент в список своїх першочергових цілей ста-

вить завдання „повернутися” у середовище СЕД.

Розглянемо структуру СЕД без використання і з використанням АОП. СЕД, побудована на основі Web-технологій, має наступну структуру:

1. Сервер баз даних – забезпечує зберігання даних СЕД, управляє даними за запитом сервера програм.

2. Сервер програм – підтримує прикладне програмне забезпечення, яке взаємодіє з браузерами АРМів користувачів і сервером баз даних.

3. Автоматизовані робочі місця користувачів СЕД (АРМ) з браузерами, що забезпечують зв'язок з сервером програм.

Основні модулі сервера програм СЕД:

1. Модуль системного адміністрування: аутентифікація користувачів в СЕД, журнал доступу користувачів, журнал звернень до електронних документів.

2. Модуль класифікації документів: одна або кілька систем класифікації документів.

3. Модуль реєстрації документів: реєстрація документів, електронна картотека, вибірки метаданих по пошуковим запитам.

4. Модуль контролю версій: забезпечення контролю версій документів та спільного використання.

5. Маршрутизація і формальний контроль: завдання виконавцям і перевіряючим, попередження про терміни, повідомлення про невиконання завдань, звіти про стан справ керуючим.

6. Модуль довідників: електронні шаблони документів, електронні довідники, електронні класифікатори.

7. Модуль електронного архіву документів: підтримка електронного архіву, повнотекстовий пошук по архіву документів, переміщення документів до / з архіву.

8. Модуль перетворення документів: конвертація документів з одного формату в інші, в тому числі, з метою друку або розміщення в Інтернет.

9. Модуль обробки паперових документів: сканування паперових документів, ручне або автоматичне анування.

10. Модуль аналізу: формування стандартних звітів про роботу виконавців і керівників; вироблення прогнозів на основі статистичної обробки документообігу для підтримки управління документообігом.

Додатково можуть бути присутніми модулі шифрування та електронного підпису, формування контенту для веб-сайту, модулі динаміч-

ного формування документів на основі запитів до баз даних та інші.

Особливість СЕД, побудованої на основі АОП (і Web-технологій) в тому, що не можна виділити її чіткої структури. Кількість агентів в системі може змінюватися в залежності від завдання, що виконується в СЕД. Перше основна відмінність від вищенаведеної структури – відсутність „головною” бази даних системи. СУБД звичайно ж використовуються, але тільки самими агентами для виконання їхніх цілей. Друга відмінність – відсутність центрального сервера програм. Те, що раніше було модулями системи, зараз – автономні агенти-сервіси, об'єднані в однорангову мережу. Точкою входу в таку мережу є агент, який реалізує інтерфейс користувача, так як досить часто інші агенти його не мають. Друге призначення цього агента – управління системою, взаємодія з іншими агентами. Перелічимо основні типи агентів в структурі СЕД:

1. Інтерфейс користувача.

2. Список користувачів.

3. Аутентифікації користувачів – даний агент взаємодіє з агентом «Список користувачів» і визначає поточного користувача, запитуючи його логін / пароль.

4. Класифікатор документів – реалізація класифікатора документів. Можливо використовувати кілька подібних агентів для підтримки одночасно декількох систем класифікації документів.

5. Відображення списку документів для поточного класифікатора (може бути частиною агента інтерфейсу користувача).

6. Контроль доступу до документів – взаємодіє з попередніми агентами для визначення списку доступних для користувача документів.

7. Фіксування дій користувача.

8. Атрибути документів – підтримка необхідної кількості атрибутів документів для конкретної реалізації СЕД. Пошук по атрибутам може реалізуватися цим же або окремим агентом.

9. Зберігання документів – забезпечує завантаження і збереження документів зі сховища, взаємодіє з агентом контролю доступу.

10. Редагування та перегляд документів – взаємодіє з агентом зберігання документів (може забезпечувати друк та інші операції над вмістом документів).

11. Спільне використання – визначається політика спільного використання документів.

12. Пошук документів – взаємодіє зі зберіганням документів для індексації документів,

надає інтерфейс для повнотекстового пошуку документів.

13. Маршрутизація документів.

Програмна архітектура агентно-орієнтованої СЕД

При аналізі архітектури СЕД слід зробити припущення, що немає необхідності розглядати всі типи агентів, які можуть існувати в системі. У систему, в разі необхідності, будуть додані необхідні типи агентів.

Провівши аналіз існуючих багатоагентних систем (наприклад, MAS DK або JADE – jade.tilab.com), можна виділити кілька ключових особливостей, які вплинуть на архітектуру СЕД:

Обчислювальна модель агента є багатозадачною. У агента окремі завдання (або поведінки) можуть виконуватися одночасно. Кожні функціональна можливість і / або сервіс, що надається агентом, повинні бути реалізовані як одна поведінка або їх набір.

Агент може перебувати в одному із станів, згідно життєвому циклу реалізованої системи. Стан агента зазвичай визначається його статусом: ініційований, активний, зупинений, очікує, вилучений і т.і.

Присутній агент-менеджер системи. Цей агент необхідний для взаємодії всіх агентів, він забезпечує доступ агентів до загальних даних системи і управляє їх життєвим циклом.

Присутній агент-реєстратор. Даний агент забезпечує реєстрацію агентів та пошук агентів для виконання поставлених завдань. Також даний агент ініціює і спостерігає за роботою системи повідомлень.

Сервер обміну повідомленнями одна з основних частин програмної архітектури. Сервіс заснований на асинхронній передачі повідомлень. Кожен агент має чергу вхідних повідомлень, куди містяться всі спрямовані агенту повідомлення. У той момент, коли повідомлення поміщається в чергу вхідних повідомлень, агент оповіщається про це. Існують багатоагентні системи, побудовані за принципом передачі повідомлень р2р. Але даний підхід не підходить для СЕД з наступних причин: немає механізмів забезпечення безпеки, асинхронна система повідомлень забезпечує більшу готовність і надійність (хоч і меншу швидкість). Тому взаємодія агентів у нашому випадку проходить тільки за допомогою сервера обміну повідомленнями. Єдине виключення – агенти, що мають інтерфейс користувача.

Ще одним елементом системи буде диспетчер зберігання даних. Цей елемент необхідний з причини великої різноманітності типів даних СЕД (файли-документи, медіа-файли, електронна пошта тощо).

На рис. 1 показана програмна архітектура агентно-орієнтованої СЕД.

Таким чином, ядром системи є: менеджер системи, реєстратор агентів, диспетчер зберігання даних і сервер обміну повідомленнями і даними. Всі інші елементи системи, головним чином агенти, підключаються (видаляються) протягом життєвого циклу системи.

Точна архітектура системи (при реалізації СЕД) описується в документі „Структура СЕД”. На практиці, документ „Структура СЕД” – це масив записів бази даних або набір xml-файлів. При ініціалізації СЕД менеджер системи отримує доступ до цього документа і, надалі, проводить пошук і ініціалізацію зазначених у ньому програмних агентів. Для забезпечення надійності та відмовостійкості роботи системи для виконання одного завдання повинно бути зареєстровано кілька агентів, неважливо, одного або різних типів. У разі недоступності агента або його відмови від виконання покладених на нього зобов'язань буде задіяний наступний за пріоритетом агент. Подібна процедура проводиться для кожної нової сесії роботи користувача. Надалі менеджер системи читає необхідні системні документи і, використовуючи необхідних агентів, створює робочий простір для користувача.

Сервер обміну повідомленнями, або сервер черг повідомлень, надає агентам сервіс для відправки та отримання повідомлень. Черги є проміжне місце зберігання повідомлень, як би спеціалізовану базу даних з усіма механізмами, що гарантують збереження: журнал, відновлення після збоїв, транзакційна обробка. Повідомлення, призначене для іншого агента, повинно бути доставлено в чергу призначення. При цьому гарантується, що повідомлення не буде втрачено або отримано двічі. Агент-адресат звертається до цільової черги і отримує доступ до повідомлення. Як сервер обміну повідомленнями можуть використовуватися власні розробки, але рекомендується дотримуватися стандартів, таких як Java Message Service або Advanced Message Queuing Protocol (AMQP). Наприклад, стандарт AMQP реалізований в сервері черг повідомлень RabbitMQ (www.rabbitmq.com).

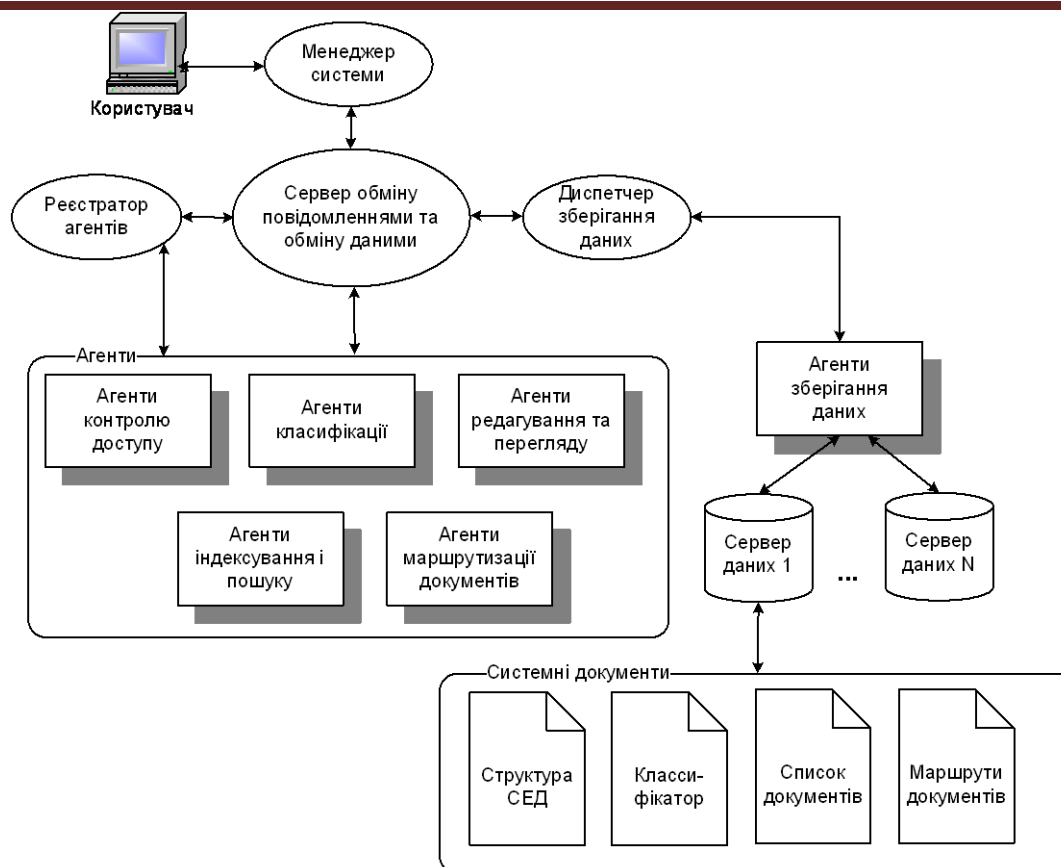


Рис. 1. Програмна архітектура агентно-орієнтованої СЕД

Коротко розглянемо роботу деяких агентів.

Агенти контролю доступу – реалізують завдання аутентифікації користувачів і регламентують доступ користувачів до документів або інших агентів (тим самим дозволяють або забороняють використання користувачем функцій СЕД). Найімовірніше наявність декількох агентів, але не всі з них будуть використовуватися одночасно. Наприклад, тільки для аутентифікації, це можуть бути:

- проста аутентифікація на основі документа „Список користувачів”;
- аутентифікація на основі сервера LDAP;
- аутентифікація на основі протоколу OpenID;
- аутентифікація на основі протоколу OAuth і т. і.

Агенти класифікації – може застосовуватися проста ієрархічна класифікація документів. Інші варіанти (і, отже агенти): ієрархічна класифікація з урахуванням прав доступу, фасетна класифікація, кілька ієрархічних класифікаторів, експертна система класифікації (з підтвердженням користувачем) та інші.

Агенти редагування та перегляду, агенти зберігання – в залежності від прав користувача і типу документа будуть використані різні агенти. Наведемо приклад. Користувач редагує документ, для цього лінія поведінки агента редагування приблизно наступна:

- отримання повідомлення від менеджера системи;
- звернення до агента контролю доступу (для забезпечення додаткової безпеки) ;
- звернення до агента зберігання даних (читання документа);
- запуск стандартної програми редагування документа (наприклад, Microsoft Word);
- завершення редагування користувачем;
- звернення до агента зберігання даних (запис документа);
- звернення до менеджера системи – передача необхідних повідомлень.

Агент індексування «підписаний» на повідомлення від агента редагування, у підсумку, змінений документ буде заново проіндексований.

Найголовніше, що побудова СЕД на основі АОП – реальне завдання, що показують експериментальні розробки. Хоча більш детальна

проробка агентно-орієнтованого підходу для використання в сучасних реаліях електронного документообігу – завдання подальших досліджень.

Практична реалізація

Розглянуті методології та особливості проектування СЕД використовувалися на практиці

в розроблених автором системах електронного документообігу DocsDale (рис. 2) і QDocuments (рис. 3).

Для програмної реалізації вибрана технологія СОМ, що дозволяє для написання агентів вибрати майже будь-яку мову програмування (з урахуванням початкового обмеження, що розробка ведеться для ОС Windows).

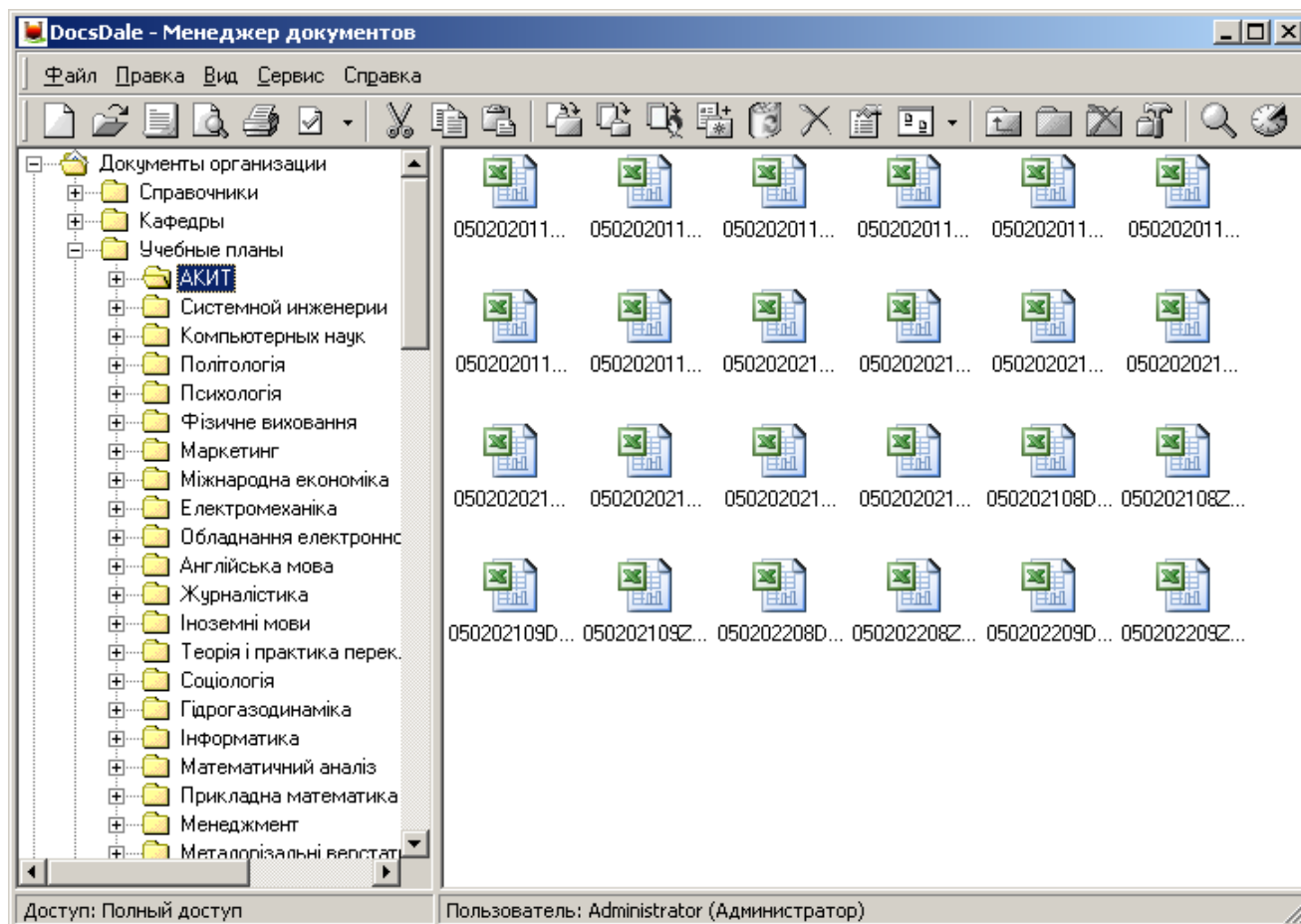


Рис. 2. Система електронного документообігу DocsDale

Розробка системи DocsDale ведеться з 2004 року, система орієнтована на автоматизацію діяльності університету у сфері управління навчальним процесом. У системі DocsDale агентно-орієнтований підхід використовується не повною мірою, хоча програмні модулі системи і являють собою інтелектуальних агентів, але слабо реалізована підсистема комунікації їх між собою.

Розробка системи QDocuments ведеться з 2010 року, система від початку була задумана як агентно-орієнтована система. Система є експериментальною і ще не використовується на

практиці тому, що система QDocuments не схожа на типові СЕД. Наприклад, самі документи (точніше типи документів), підсистеми аутентифікації, контролю доступу та класифікації документів реалізовані у вигляді агентів. Той функціонал, що в типовій СЕД є частиною ядра системи, в QDocuments реалізований у вигляді агентів. Головна програма системи – „QDocuments Manager” є агентом, що реалізує інтерфейс користувача і систему повідомлень. Активно застосовуються концепції динамічних документів і алгебри документів.

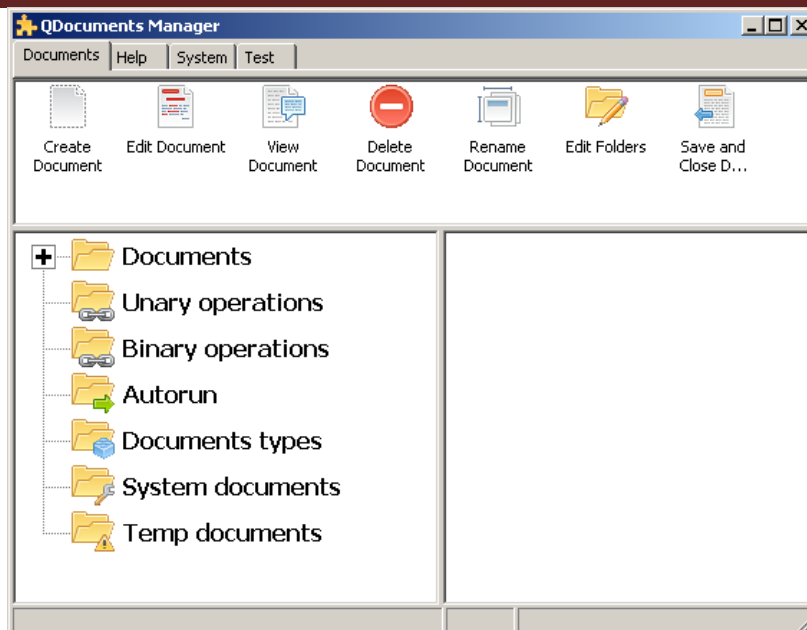


Рис. 3. Експериментальна СЕД QDocuments

Висновки

Розглянуто підхід до проектування СЕД з використанням агентно-орієнтованого підходу. Запропонована методологія проектування СЕД дозволить створювати гнучкі інтелектуальні системи автоматизації документообігу. Результати роботи можуть бути використані при проектуванні інших інформаційних систем, а також при навчанні студентів інформаційним технологіям та програмної інженерії.

Список використаних джерел

1. Хмелюк В. С. Засоби автоматизації генерації електронних документів в системах організаційного управління / В. С. Хмелюк, О. А. Амонс // Проблеми програмування. – 2008. – № 2 – 3. – С. 641 – 649.
2. Задорожна Н. Т. Кероване проектування документообігу в управлінських інформаційних системах / Н. Т. Задорожна, К. М. Лаврішева // Проблеми програмування. – 2006. – № 4. – С. 37 – 48.

3. Wooldridge M. Agent-Based Software Engineering // IEE Proc. Software Eng. – 1997. – Vol. 144, № 01. – P. 26 – 37.

4. Городецкий В.И. Многоагентные системы (обзор) / В. И. Городецкий, М. С. Грушинский, А. В. Хабалов // Новости искусственного интеллекта. – 1998. – № 2. – С. 64 – 116.

5. Плєскач В. Л. Агентні технології / В. Л. Плєскач, Ю. В. Рогушина. – К. : КНТЕУ, 2005. – 337 с.

6. Городецкий В. И. MAS DK: инструментарий для разработки многоагентных систем и примеры приложений / В. И. Городецкий, О. В. Карсаев, И. В. Хотенко, А. В. Хабалов // Искусственный интеллект в XXI веке (ICAI 2001) : труды Международного конгресса (3 – 8 сентября 2001 г., г. Москва). – М. : Физматлит, 2001. – С. 249 – 262.

7. Гриценко В. И. Модель мультиагентной системы для е-бизнеса и технология ее программной реализации / В. И. Гриценко, А. Я. Гладун, Ю. Д. Журавлев, М. В. Несен // Проблеми програмування. – 2004. – № 2,3. – С. 510 – 519.

Відомості про авторів



Бутенко Максим Леонідович – старший викладач Східноукраїнського національного університету імені Володимира Даля, наукові інтереси – інформаційні системи, системи електронного документообігу

E-mail: mlbutenko@gmail.com



Морозова Тетяна Юрїївна - доцент кафедри інформатики Східноукраїнського національного університету імені Володимира Даля, кандидат технічних наук, наукові інтереси – методи та практика навчання

E-mail: tmorozova@i.ua