

CONFERENCE PROCEEDINGS

МАТЕРІАЛИ КОНФЕРЕНЦІЇ

 **infoCom** *summer* **2017**

Summer InfoCom Advanced Solutions 2017

**IV МІЖНАРОДНА НАУКОВО-ПРАКТИЧНА КОНФЕРЕНЦІЯ
1-2 червня 2017 року**

ISBN 978-966-2344-54-7

Україна, Київ

Summer InfoCom Advanced Solutions 2017

**IV МІЖНАРОДНА НАУКОВО-ПРАКТИЧНА КОНФЕРЕНЦІЯ
1-2 червня 2017 року**

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ІНСТИТУТ МОДЕРНІЗАЦІЇ ЗМІСТУ ОСВІТИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»**

**SUMMER INFOCOM
ADVANCED SOLUTIONS
2017**

МАТЕРІАЛИ

IV МІЖНАРОДНОЇ НАУКОВО-ПРАКТИЧНОЇ КОНФЕРЕНЦІЇ

**CONFERENCE
PROCEEDINGS**

IV SCIENTIFIC AND PRACTICAL CONFERENCE

**КИЇВ, УКРАЇНА
1-2 ЧЕРВНЯ 2017 РОКУ**

УДК 004

Редакційна колегія:

Бідюк П.І., д.т.н., проф., ІПСА, КПІ ім. Ігоря Сікорського, Україна, Київ

Павлов О.А., д.т.н., проф., КПІ ім. Ігоря Сікорського, Україна, Київ

Теленик С.Ф., д.т.н., проф., КПІ ім. Ігоря Сікорського, Україна, Київ

Грیشин І.Ю., д.т.н., проф., Кубанський державний технологічний університет, Росія

Головний редактор:

Писаренко А.В., к.т.н., доц., КПІ ім. Ігоря Сікорського, Україна, Київ

Summer InfoCom 2017: Матеріали IV Міжнародної науково-практичної конференції, м. Київ, 1-2 червня 2017 р. – К.: Вид-во ТОВ «Інжиніринг», 2017. – 120 с. – Мови укр., рос., англ.

Конференція входить до Переліку міжнародних та всеукраїнських науково-практичних конференцій здобувачів вищої освіти та молодих учених у 2017 році (додаток до листа Міністерства освіти і науки України № 1/9-24 від 23 січня 2017 року).

Проведення конференції регламентоване наказом ректора КПІ ім. Ігоря Сікорського № 3-183 від 10 травня 2017 р.

Усі права застережено. Передруки та переклади дозволяються лише за згодою автора та редакції. За достовірність фактів, цитат, назв та іншої інформації несуть відповідальність автори.

Редакційна колегія дотримується прийнятих міжнародною спільнотою принципів публікаційної етики, відображених, зокрема, в рекомендаціях Комітету з етики наукових публікацій (Committee on Publication Ethics, COPE), а також враховує досвід авторитетних міжнародних видавництв. Щоб уникнути недобросовісної практики в публікаційній діяльності (плагиат, виклад недостовірних відомостей та ін.), з метою забезпечення високої якості наукових публікацій, визнання громадськістю отриманих автором наукових результатів, кожен член редакційної колегії, автор, рецензент, видавець, а також установи, які беруть участь в видавничому процесі, зобов'язані дотримуватися етичних стандартів, норм і правил та вживати всіх можливих заходів для запобігання їх порушень. Дотримання правил етики наукових публікацій усіма учасниками цього процесу сприяє забезпеченню прав авторів на інтелектуальну власність, підвищенню якості видання і виключення можливості неправомірного використання авторських матеріалів в інтересах окремих осіб.

ПРОГРАМА КОНФЕРЕНЦІЇ

ПРОГРАМА

Інформаційні системи та технології

Юрчина О. Бугай О.	Синхронізація у розподілених інформаційних системах
Глибовець А. М.	Розробка пошукового робота з можливістю гнучкої конфігурації
Гришин І. Ю. Тимиргалеева Р. Р. Миронов М. В.	Підхід к решению проблемы повышения информационной безопасности предприятия
Жабин В.И.	Функциональный контроль процессоров в системах на кристалле
Хмелюк В. С. Майер І. С. Озеракін М. Д. Тимошенко О. О.	Концепція сервісу програмного забезпечення
Подрубайло О.О. Лук'яненко Я.В.	Аналіз методів об'єднання таблиць у розподілених сховищах даних в оперативній пам'яті
Заміховський Л. М. Іванюк Н. І.	Параметрування програмних блоків модуля sm1281 для побудови системи діагностування гпа
Заміховський Л. М. Павлик В. В.	Використання дискримінантного аналізу для вибору діагностичної ознаки технічного стану газоперекачувальних агрегатів

Системи керування

Писаренко А. Тишенко Д.	Модель підсистеми діагностики транспортного засобу на основі байєсівської мережі
Юрчук Л. Ю.	Міжнародні нормативні засади людино-машинного інтерфейсу систем управління технологічними процесами
Николайчук М. Я.	Компоненти систем управління з розширенням реального часу

Технології програмування

Крак Ю.В. Коваль Ю.В.	Реалізація віртуальних змінних
Шишкін В. І. Суханюк М. В.	Причини отказа от Hibernate и как правильно использовать JDBC
Катин П. Ю.	Конструювання програмного забезпечення для вбудованих систем з використанням типових шаблонів

Оброблення інформації в складних системах

Самотий В.В. Павельчак А.Г. Дзелендзяк У.Ю.	Передача даних в автомобільній LIN-мережі на базі мікроконтролерів Cypress
Дорошенко А. Ю. Тітов Д. С.	Високопродуктивний алгоритм зіставлення патернів у часових рядах для задач аналізу якості потокових сервісів
Сергеев-Горчинский А.А.	Выбор меры различия образцового и отфильтрованного периодических сигналов при высоких уровнях флуктуационного шума

Інформаційні системи та технології

Погоржельський В. В. Полторак В. П.	Автентифікація RFID мітки на основі односторонніх перетворень
Mohammad Alhawawsha	Identification of the issues in the E-governance in Jordan and USA
Прокопович І. В. Духаніна М. О. Добровольська В. В. Дадерко О.І. Олех Г.С., Кошулян С.В.	Інформаційне моделювання процесів перенесення в гетерогенних середовищах
Савельєва О.С. Становська І.І., Гур'єв І.М. Малахова Д.О., Саух І.А.	Інформаційні технології управління проектною логістикою за допомогою віртуальних моделей
О.М. Моргаль Є.О. Покровський О.В. Савчук, І.О. Латаш	До питання оцінки якості web-сайту діяльності підприємства як моделі масового обслуговування
Троцький С. О. Вовк Є. А. Погорілий Ю. А. Коломійчук М. С.	Використання компонентного підходу при проектуванні сервісів для системи управління бізнес-процесами
О.Л. Становський П.С. Швець А.В. Торопенко О.Є. Науменко Абу Шена Осама	Інформаційна технологія проектування рівнонапружених деталей машин
Виноградов Ю. Н. Иванов В. Г.	Метод резервирования и восстановления данных в распределенных системах их хранения
Захаріудакіс Лефтеріс Олієвський А.А.	Метод строгої ідентифікації віддалених користувачів з використанням перетворень на полях Галуа
Сергієнко А. М. Молчанов О. А.	Мікроконтролер з системою команд, що розширюється

2 череня/June

Системи керування

Заміховський Л.М. Левицький І. Т.	Система керування механізмом видалення металевих включень в умовах виробництва керамічної цегли
Долина В. Г. Пріліпухов Є. В.	Проблеми управління групою автономних рухомих об'єктів у 3D просторі
В.А. Неволько	Синтез нечітких регуляторів на основі асоціативної пам'яті з апаратною реалізацією на ПЛІС

ЗМІСТ

<i>Тези конференції</i>	9
<i>Інформаційні системи та технології/ Information Systems and Technologies</i>	11
Юрчина Олексій, Бугай Олександр Синхронізація у розподілених інформаційних системах.....	13
Глибовець Андрій Миколайович Розробка пошукового робота з можливістю гнучкої конфігурації.....	16
Гришин Ігорь Юрьевич, Тимиргалеева Рена Ринатовна, Миронов Максим Вадимович Подход к решению проблемы повышения информационной безопасности предприятия.....	19
Жабин В.И. Функциональный контроль процессоров в системах на кристалле.....	23
Хмелюк Володимир Сергійович, Майер Ілля Сергійович, Озеракін Микита Дмитрович, Тимошенко Олександр Олександрович Концепція сервісу програмного забезпечення.....	25
Подрубайло О.О., Лук'яненко Я.В. Аналіз методів об'єднання таблиць у розподілених сховищах даних в оперативній пам'яті.....	28
Заміховський Леонід Михайлович, Іванюк Наталія Іванівна Параметрування програмних блоків модуля sm1281 для побудови системи діагностування ГПА.....	30
Заміховський Леонід Михайлович, Павлик Володимир Васильович Використання дискримінантного аналізу для вибору діагностичної ознаки технічного стану газоперекачувальних агрегатів	32
Погоржельський Валерій В'ячеславович, Полторак Вадим Петрович Автентифікація RFID мітки на основі односторонніх перетворень.....	34
Mohammad Alhawawsha Identification of the issues in the E-governance in Jordan and USA.....	37
Прокопович І. В., Духаніна М. О., Добровольська В. В., Дадерко О.І., Олех Г.С., Кошулян С.В. Інформаційне моделювання процесів перенесення в гетерогенних середовищах.....	41
Савельєва О.С., Становська І.І., Гур'єв І.М., Малахова Д.О., Саух І.А. Інформаційні технології управління проектною логістикою за допомогою віртуальних моделей.....	43
О.М. Моргаль, Є.О. Покровський, О.В. Савчук, І.О. Латаш До питання оцінки якості web-сайту діяльності підприємства як моделі масового обслуговування.....	45
Троцький Сергій Олександрович, Погорілий Юрій Анатолійович, Коломійчук Микита Сергійович, Вовк Євгеній Андрійович Використання компонентного підходу при проектуванні сервісів для системи управління бізнес-процесами.....	48
О.І. Становський, П.С. Швець, А.В. Торопенко, О.Є. Науменко, Абу Шена Осам Інформаційна технологія проектування рівнонапружених деталей машин.....	51
Виноградов Ю. Н., Иванов В. Г. Метод резервирования и восстановления данных в распределенных системах их хранения.....	53
Захаріюдакіс Лефтеріс, Олієвський А.А. Метод строгої ідентифікації віддалених користувачів з використанням перетворень на полях Галуа	56
Сергієнко Анатолій Михайлович, Молчанов Олексій Андрійович Мікроконтролер з системою команд, що розширюється.....	58
<i>Системи керування/ Control Systems</i>	61
Писаренко Андрій, Тищенко Дмитро Модель підсистеми діагностики транспортного засобу на основі байєсівської мережі.....	63
Юрчук Леонід Юрійович Міжнародні нормативні засади людино-машинного інтерфейсу систем управління технологічними процесами.....	66
Николайчук Микола Ярославович Компоненти систем управління з розширенням реального часу.....	69
Заміховський Леонід Михайлович, Левицький Іван Теодорович Система керування механізмом видалення металевих включень в умовах виробництва керамічної цегли.....	71
Долина Віктор Георгійович, Пріліпхув Євгеній Вадимович Проблеми управління групою автономних рухомих об'єктів у 3D просторі.....	73

В.А. Неволько	
Синтез нечітких регуляторів на основі асоціативної пам'яті з апаратною реалізацією на ПЛІС.....	76
<i>Технології програмування/ Programming Technologies</i>	79
Крак Ю.В., Коваль Ю.В.	
Реалізація віртуальних змінних.....	81
Шишкин Владислав Игоревич, Суханюк Марина Валентиновна	
Причины отказа от Hibernate и как правильно использовать JDBC.....	83
Катин Павло Юрійович	
Конструювання програмного забезпечення для вбудованих систем з використанням типових шаблонів.....	86
<i>Оброблення інформації в складних системах/ Information Processing in Complex Systems</i>	91
Самотий В.В., Павельчак А.Г., Дзелендзяк У.Ю.	
Передача даних в автомобільній LIN-мережі на базі мікроконтролерів Cypress.....	93
Дорошенко Анатолій Юхимович, Тітов Дмитро Сергійович	
Високопродуктивний алгоритм зіставлення патернів у часових рядах для задач аналізу якості потокових сервісів.....	96
Сергеев-Горчинский А.А.	
Выбор меры различия образцового и отфильтрованного периодических сигналов при высоких уровнях флуктуационного шума.....	100
<i>Abstracts</i>	105

ТЕЗИ КОНФЕРЕНЦІЇ

**ІНФОРМАЦІЙНІ СИСТЕМИ
ТА ТЕХНОЛОГІЇ**

**INFORMATION SYSTEMS AND
TECHNOLOGIES**

Синхронізація у розподілених інформаційних системах

Юрчина Олексій
студент
КПІ ім. Ігоря Сікорського
Україна, Київ

Бугай Олександр
студент
КПІ ім. Ігоря Сікорського
Україна, Київ

У статті здійснено аналіз особливостей синхронізації даних у розподілених інформаційних системах. Розглянуто розподілені інформаційні системи та окреслено їхні сутнісні характеристики. Автором здійснено дослідження прикладів синхронізації.

Ключові слова: синхронізація, синхронізація даних, розподілені інформаційні системи

ВСТУП

Задачі та виклики, що постають перед людьми у сучасному інформаційному суспільстві, характеризуються значною складністю, елементами невизначеності, великими потребами у матеріальних та інтелектуальних ресурсах. Для їх розв'язання необхідно об'єднати зусилля багатьох людей та залучати потужні засоби обчислювальної техніки. Така кооперація може призвести до виникнення неконтрольованих або хаотичних режимів роботи системи, критично пов'язаних із людським чинником. У зв'язку з цим актуальним є дослідження та реалізація алгоритмів та методів синхронізації файлової інформації.

Ні для кого не секрет, що створення програмного забезпечення – це складний процес, який, з одного боку, має багато загального з творчістю, а з іншого, – хоча і високоприбутковий, але й високозатратний бізнес. Жорстка конкуренція на ринку змушує розробників до пошуку більш ефективних методів роботи, шляхів створення програмних систем у більш короткі строки, з меншими витратами і кращою якістю. Складність програм постійно збільшується.

На сьогоднішній день практично всі провідні компанії - розробники технологій і програмних продуктів (IBM, Oracle, Borland) мають розвинені технології створення ПЗ, які створювалися як власними силами, так і за рахунок придбання продуктів і технологій, створених невеликими спеціалізованими компаніями.

В даний час одним з найбільш популярних носіїв інформації є флеш-диск. Його зручність підключення до комп'ютера, компактні розміри і, в той же час, досить велика ємність, дозволяють зберігати на ньому усю необхідну в повсякденному користуванні інформацію. Але, не дивлячись на всі плюси даного носія, повністю покладатися на нього не можна: компактні розміри несуть загрозу втрати диска, а технологія зберігання інформації (електронні затвори) не дозволяє зберігати інформацію протягом тривалого періоду часу. Тому, найкращим рішенням буде використання флеш-диска з регулярним створенням резервних копій даних. Це досить монотонне заняття, яке можна доручити програмі.

У зв'язку з цим актуальним є дослідження синхронізації у розподілених інформаційних системах.

ВИКЛАД ОСНОВНОГО МАТЕРІАЛУ

Розподілена система - сукупність незалежних комп'ютерів, яка представляється користувачеві єдиним комп'ютером. Приклади: мережа робочих станцій (вибір процесора для виконання програми, єдина файлова система), роботизований завод (роботи пов'язані з різними комп'ютерами, але діють як зовнішні пристрої єдиного комп'ютера, банк з безліччю філій, система резервування авіаквитків.

Чому створюються розподілені системи? У чому їх переваги перед централізованими ЕОМ?

1-а причина - економічна. Закон Гроша (Herb Grosh, 25 років тому) - швидкодія процесора пропорційна квадрату його вартості. З появою мікропроцесорів закон перестав діяти - за подвійну ціну можна отримати той же процесор з дещо більшою частотою.

2-а причина - можна досягти такої високої продуктивності шляхом об'єднання мікропроцесорів, яка недосяжна в централізованому комп'ютері.

3-я причина - природна розгалуженість (банк, підтримка спільної роботи групи користувачів).

4-а причина - надійність (вихід з ладу декількох вузлів незначно знизить продуктивність).

5-я причина - нарощуваність продуктивності. В майбутньому головною причиною буде наявність величезної кількості персональних комп'ютерів і необхідність спільної роботи без відчуття незручності від географічного і фізичного розподілу людей, даних і машин [2, с. 7].

Чому потрібно об'єднувати РС в мережі?

- Необхідність розділяти дані.
 - Перевага розділення дорогих периферійних пристроїв, унікальних інформаційних і програмних ресурсів.
 - Досягнення розвинених комунікацій між людьми.
- Електронна пошта в багатьох випадках зручніше листів, телефонів і факсів.
- Гнучкість використання різних ЕОМ, розподіл навантаження.
 - Спрощення поступової модернізації допомогою заміни комп'ютерів [1, с. 44].
- Недоліки розподілених систем:
- Проблеми (програми, мови, ОС).
 - Проблеми комунікаційної мережі (втрати інфор-

мації, перевантаження, розвиток і заміна).

- Секретність.

Зазвичай децентралізовані алгоритми мають наступні властивості:

- Інформація, що відноситься до справи, розподілена серед ЕОМ.
- Процеси приймають рішення на основі тільки локальної інформації.
- Не повинно бути єдиної критичної точки, вихід з ладу якої призводив би до краху алгоритму.
- Не існує загальних годин або іншого джерела точного глобального часу [5, с. 33].

Перші три пункти всі говорять про неприпустимість збору всієї інформації для прийняття рішення в одне місце. Забезпечення синхронізації без централізації вимагає підходів, відмінних від використовуваних в традиційних ОС. Останній пункт дуже важливий - в розподілених системах досягти згоди щодо часу зовсім непросто. Важливість наявності єдиного часу можна оцінити на прикладі програми make в ОС UNIX. Головні теоретичні проблеми - відсутність глобальних годин і неможливість зафіксувати глобальний стан (для аналізу ситуації - виявлення дедлоків, для організації проміжного запам'ятовування).

Явища синхронізації систем поширені у природі, техніці та суспільстві [3]. За певних умов вони можуть зумовлюватись простими фізичними чи хімічними законами або колективною взаємодією активних об'єктів (агентів) зі складною динамікою поведінки. Тут під активними розуміємо автономні об'єкти, здатні контролювати стани зовнішнього середовища, виробляти і реалізовувати керуючі сигнали.

Прикладами синхронізації у природі є ритмічне світіння рою комах-світлячків, сюрчання цвіркунів, рух косяків риб, політ зграї птахів тощо. Узгодження біологічних ритмів живих організмів із зовнішніми природними чинниками, формування серцевих ритмів та електричних ритмів мозку є проявами синхронізації у біології та медицині. Синхронізація є центральним механізмом обробки інформації нейронами у різних ділянках мозку та передавання інформації між цими областями.

Синхронізація потоків коду та даних необхідна для забезпечення роботи сучасних розподілених програмних засобів і сховищ даних з мережевою реалізацією, прикладом яких є мультиагентні системи [3].

У мультиагентних системах синхронізація необхідна для забезпечення узгодженої роботи їх складових частин, передавання повідомлень між агентами, забезпечення умов самоорганізації, коли розподілена система поводить ся як цілісний, штучно сформований організм. Під самоорганізацією розуміють цілеспрямований процес створення, відтворення, впорядкування або вдосконалення організації (структури та функцій) складної динамічної системи за рахунок внутрішніх факторів, без відповідного зовнішнього впливу [4, с. 25].

Процеси синхронізації в системах різної природи мають багато спільного, їх можна вивчити з використанням загальних математичних та обчислювальних інструментів. Електротехнічними моделями синхронізації системи

розподілених об'єктів, як правило, слугують мережі синхронізації осциляторів [5]. Для вивчення синхронізації використовують методи теорії керування, коливальних, фазової динаміки, відображень, нелінійних середовищ та мереж, хаосу, фракталів, клітинних автоматів тощо.

У теорії синхронізації виділяють дві основні частини:

- 1) класичну теорію синхронізації, що вивчає явища у зв'язаних періодичних авто коливних системах;
- 2) теорію хаотичної синхронізації, яка вивчає кооперативну поведінку хаотичних систем.

Серед систем хаотичної синхронізації виділяють три головних типи:

- 1) повна (або ідентична) синхронізація – стани зв'язаних об'єктів повністю збігаються;
- 2) узагальнена синхронізація – виходи об'єктів пов'язані через деяку функцію;
- 3) фазова синхронізація – встановлення деяких співвідношень між фазами об'єктів, що взаємодіють, результатом чого є збіг їхніх характерних частот або характерних часових масштабів.

Розрізняють такі види синхронізації об'єктів:

- 1) примусова синхронізація об'єктів за допомогою зовнішнього джерела сигналів, наприклад, генератора тактової частоти;
- 2) вільна просторово розподілена синхронізація об'єктів між собою.

Розглянемо основні принципи алгоритму проведення синхронізації даних клієнта по кроках:

1. В певний момент часу роботи системи клієнта до підсистеми синхронізації надходить запит на початок проведення сеансу синхронізації даних. Якщо пристрій користувача в цей момент має підключений доступ до мережі інтернет та є можливість зв'язатись з центральним сервером синхронізації то продовжуємо цей процес, якщо ні – повертається помилка про неможливість проведення синхронізації в даних момент.

2. Генеруємо новий ідентифікатор версії стану локальних даних клієнта.

3. Далі на клієнті відбувається формування пакету змінених даних. Якщо з моменту проведення останнього сеансу синхронізації жодні зміни не були внесені – список змінених рядків буде порожнім.

4. Після того як список змінених даних сформований він відправляється на центральний сервер разом з ідентифікатором версії даних останньої синхронізації та ідентифікатором поточної версії, який щойно згенеровано. Якщо синхронізація ще жодного разу не проводилась то версія останньої синхронізації буде порожньою.

5. Отримавши пакет сервіс синхронізації перевіряє чи є в ньому змінені рядки: якщо немає то клієнт не вносив змін в локальну БД, тому переходимо до кроку 10, якщо є – виконуємо процес далі.

6. Якщо версія останньої синхронізації виявилась актуальною (останньою версією на сервері) – отже клієнт працював з поточним станом даних серверу і його зміни точно не конфліктують зі змінами на сервері (Розв'язання конфліктів не потребується тому переходимо до кроку 8).

7. У тому разі якщо на попередньому кроці виявилось, що з моменту коли клієнт синхронізувався востаннє були внесені зміни до центрального сховища то перед збереженням його змін потрібно виконати дослідження на наявність конфліктів та розв'язати їх.

ВИСНОВКИ

Отже, явища синхронізації систем поширені у природі, техніці та суспільстві. За певних умов вони можуть зумовлюватись простими фізичними чи хімічними законами або колективною взаємодією активних об'єктів (агентів) зі складною динамікою поведінки. Тут під активними розуміємо автономні об'єкти, здатні контролювати стани зовнішнього середовища, виробляти і реалізовувати керуючі сигнали. Результати роботи доцільно використати для узгодження роботи складових частин мультиагентних систем різного призначення, передавання повідомлень між агентами, побудови протоколів комунікації, забезпечення умов самоорганізації мультиагентних систем.

ПЕРЕЛІК ПОСИЛАНЬ

1. Дейт К.Дж. Введение в системы баз данных, 6-е издание: Пер. с англ. / Дейт К.Дж. – К.: Издательский дом «Вильямс», 2000. – 848 с.
2. Третяк В.Ф. Проблеми розвитку програмного забезпечення середовища розподілених обчислень GRID / В.Ф. Третяк, Д.Ю. Голубничий, І.О. Золотарьова // Матеріали третьої наукової конференції Харківського університету Повітряних Сил мені Івана Кожедуба. – Х.: ХУПС, 2007. – С. 7.
3. Синхронизация [Електронний ресурс]. Режим доступу: http://ru.wikipedia.org/wiki/Синхронизация_%28информатика%29.
4. Weiss G. Multiagent Systems. A Modern Approach to Distributed Artificial Intelligence / G. Weiss, editor. – Springer Verlag, Berlin, 1996. – 643 pp.
5. Wooldridge M. An Introduction to Multiagent Systems / M. Wooldridge. – John Wiley & Sons, 2002. – 366 p. 3

*Рецензент: к.т.н., доц. каф. АУТС, КІП ім. Ігоря Сікорського,
О.А. Амонс*

Розробка пошукового робота з можливістю гнучкої конфігурації

Глибовець Андрій Миколайович
доцент

Національний університет «Києво-Могилянська академія»
Україна, Київ

Проектування та побудова пошукового робота - досить нетривіальна задача. Актуальність даної теми полягає в тому, що не зважаючи на важливість пошукових роботів для ефективного розвитку пошукових та інформаційних систем загалом, на сьогодні усе ж дана проблема не є повністю вирішеною, існує досить небагато відкритих реалізацій пошукових роботів, які б можна було просто, швидко та гнучко налаштувати для своїх цілей та використовувати у своїх проектах. В роботі було запропоновано один з підходів до побудови пошукового робота з можливістю конфігурації.

Ключові слова: інформаційно пошукові системи, пошуковий робот

ВСТУП

На сьогодні неможливо уявити жодну інформаційну систему без можливостей пошуку, аналізу та збереження інформації. Будь-яка пошукова система повинна вміти проводити індексацію та аналіз (наприклад, веб-сторінок) для того аби надавати релевантну інформацію у відповідь на пошукові запити користувача в майбутньому. Для цих цілей широко використовують пошукових роботів, які проводять аналіз та індексацію веб-сторінок з метою пришвидшення роботи системи.

АНАЛІЗ ПРОБЛЕМ РОЗРОБКИ ПОШУКОВОГО РОБОТА

Пошуковий робот - це програма, яка аналізує так індексує веб-сторінки для подальшого використання цього індексу пошуковою системою та побудови веб-кешу [1]. Робот починає свою роботу із заданими посиланнями на веб-сторінки, при проведенні аналізу кожної сторінки робот виокремлює посилання на ресурси та додає їх у свою внутрішню чергу для подальшої обробки. Варто додати, що деякі реалізації пошукових збирачів вміють аналізувати вміст сторінки і на основі аналізу, ефективно відшукувати лише необхідну інформацію. Кожну успішно проаналізовану сторінку робот повертає для подальшого її аналізу клієнтом. Робот продовжує аналіз сторінок поки черга на обробку не стане порожньою.

Першою проблемою, яка виникає при реалізації такого алгоритму роботи є проблема зберігання проаналізованих сторінок, оскільки клієнти можуть мати досить обмежені ресурси.

Наступною проблемою є проблема доступу до сервера, на якому зберігаються веб-сторінки. Так, наприклад, погано спроектований або налаштований робот може спричинити відмову в обслуговуванні інших клієнтів, якщо він буде занадто часто доступатися до веб-сторінок і у такий спосіб перенавантажить сервер.

Іноді може трапитися ситуація, що за посиланням вже немає вказаного ресурс або веб-сервер повертає помилку при обробці запиту. Із цих ситуацій випливають підвищені вимоги до надійності та стабільності у роботі пошукового

робота, він має продовжувати аналіз електронного ресурсу навіть за наявності вищезгаданих помилок та відмов в обслуговуванні, по-можливості, інформуючи при цьому свого клієнта.

Аналіз навіть невеликого електронного ресурсу потребує часу, тим більше це стосується великих веб-порталів. За таких обставин можна стикнутися з проблемою, коли під час аналізу робот має повертатися до вже проаналізованих сторінок із метою отримання актуальної інформації. Із ростом розміру порталу та складністю алгоритму робота ситуація стає ще більш складною.

Таким чином, можна зробити висновок, що розробка пошукового робота – досить складна та нетривіальна задача із великою кількістю проблем [2].

ПРОЕКТУВАННЯ ТА РОЗРОБКА ПОШУКОВОГО РОБОТА

При проектуванні та розробці було приділено багато уваги можливості простого налаштування та вдосконалення робота в майбутньому. Основні проблеми, які необхідно було вирішити при проектуванні - це мати можливість налаштовувати робота під конкретний веб-ресурс, не змінюючи при цьому вихідний код системи та зробити компоненти системи якомога менше зв'язаними, а отже більш зручними та придатними для подальшого вдосконалення.

Для розв'язання проблеми налаштування було прийнято рішення розробити власний XML формат, який би у зручний спосіб давав змогу налаштувати робота. Для полегшення процесу модифікації системи було вирішено додати системний сервіс, який би на основі будь-якого зручного для користувача формату файлу налаштувань (XML, JSON, YAML та інших) генерував би "зрозумілий" системі екземпляр класу налаштувань. При такому підході стає можливим легко замінювати один тип файлу налаштувань іншим. UML діаграму такого модуля представлено рисунком 1

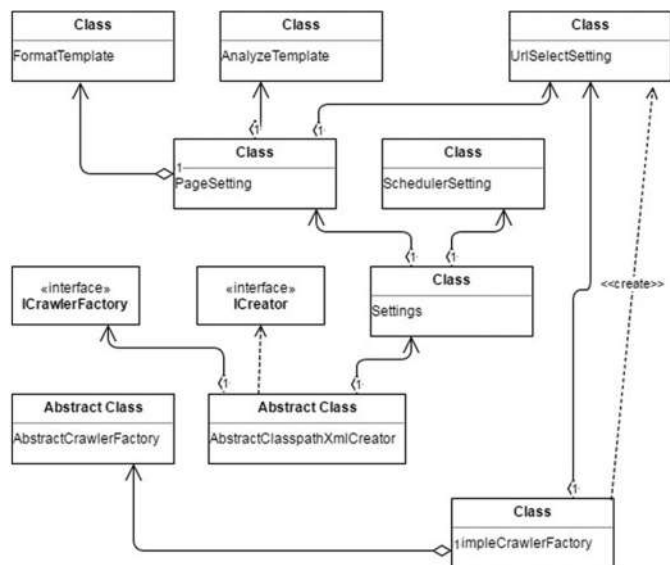


Рис. 1 UML діаграма модулю налаштувань

Основним класом у даній підсистемі є клас Settings, який містить основні налаштування системи: аналізу й форматування веб-сторінок і також налаштування планувальника задач, які представлені класами PageSetting та SchedulerSetting відповідно.

Налаштування планувальника задач повинні надавати змогу регулювати кількість потоків, що може використовувати робот для поточної переіндексації та обробки сторінок. Окрім того, для зменшення навантаження на веб-сервер, варто ввести можливість регульованої затримки між послідовними обробками веб-сторінок.

Фактичне створення екземплярів пошукового робота беруть на себе класи, що реалізують інтерфейс ICreator. Класи, що наслідують інтерфейс ICrawlerFactory також відповідають за створення екземпляру робота, але вже на основі готових екземплярів типу Settings.

Для того аби мати можливість підтримувати новий формат налаштувань достатньо реалізувати інтерфейс ICreator, при цьому можна використати існуючу реалізацію інтерфейсу ICrawlerFactory або замінити її на іншу.

Розробка формату налаштувань

XML досить часто використовують для написання файлів конфігурації через те, що його досить легко перевірити на коректність за допомогою XSD схем, до того ж XML досить близький до людської мови, а тому досить простий для розуміння людиною.

Якщо проаналізувати вищенаведені вимоги, які так чи інакше стосуються налаштувань пошукового робота, то можна зробити висновок, що конфігурація пошукового робота повинна мати такі можливості:

- Вказувати початковий список посилань на веб-сторінки
- Мати базові налаштування планувальника задач:
 - Можливість вказувати затримку між поточною обробкою сторінок
 - Можливість вказати кількість потоків, що їх може використовувати робот
- Мати можливість створювати шаблон сторінки:

- Вказувати унікальний ідентифікатор сторінки
- Вказувати налаштування аналізатора
- Вказувати налаштування форматування

Приклад налаштувань:

```
<?xml version=>1.0 encoding=>UTF-8 ?>
<crawler>
  <urls>
    <url>http://nz.ukma.edu.ua/index.php?option=com_content&
    amp;task=section&id=10&Itemid=47</url>
  </urls>
  <scheduler-params>
    <startup-delay>0</startup-delay>
    <index-delay>60000</index-delay>
    <threads>2</threads>
  </scheduler-params>
  <pages>
    <page id=>1>>
      <analyze-params>
        <analyze weight=>10>>body > table > tbody > tr >
        td > table:nth-child(5) > tbody > tr > td:nth-child(2) > table >
        tbody > tr:nth-child(3) > td > ul > li > a</analyze>
      </analyze-params>
      <url-params>
        <extract attr=>href>
          selector=>body > table > tbody > tr > td >
          table:nth-child(5) > tbody > tr > td:nth-child(2) > table > tbody
          > tr:nth-child(3) > td > ul > li > a>/>
        </url-params>
      </page>
    <page id=>2>>
      <analyze-params>
        <analyze weight=>70>>body > table > tbody > tr >
        td > table:nth-child(5) > tbody > tr > td:nth-child(2) > table >
        tbody > tr:nth-child(2) > td > form > table > tbody > tr > td >
        a</analyze>
      </analyze-params>
      <url-params>
        <extract attr=>href>
          selector=>body > table > tbody > tr > td >
          table:nth-child(5) > tbody > tr > td:nth-child(2) > table > tbody
          > tr:nth-child(2) > td > form > table > tbody > tr > td > a>/>
        </url-params>
      </page>
    </pages>
  </crawler>
```

Розроблений формат налаштувань дозволяє зручно та швидко створювати шаблони сторінок із метою подальшого аналізу. Так, можливо вказати унікальний ідентифікатор для кожного шаблону та мінімальне порогове значення важливості, при цьому кожен шаблон може містити необмежену кількість правил аналізу, форматування і видалення посилань зі сторінки (xml-елементи analyze та extract відповідно). Кожне таке правило є CSS-селектором, що дає змогу легко реалізувати роботу з DOM-структурою. Для валідації XML документу було створено XSD схему

ОПИС ВИКОРИСТАНИХ ЗАСОБІВ

Для розробки пошукового робота було вирішено використовувати мінімальну кількість сторонніх бібліотек задля зменшення залежностей від них й, відповідно за необхідності, полегшення процесу переходу на інші; також одним із критеріїв вибору інструментарію були простота у використанні, стабільність та поширеність.

Пошукового робота було розроблено із використанням б'єктно-орієнтованої мови програмування Java версії 1.8. У якості IDE було обрано IntelliJ Idea Community Edition 2016. Перелік використаних бібліотек включає в себе:

- Lombok версії 1.16.12, бібліотека для автоматичної генерації коду, допомагає усувати шаблонний код
- H2 database connector версії 1.3.170, з'єднання з базою даних H2
- JSoup версії 1.10.1, бібліотека для аналізу та модифікації DOM, побудови CSS селекторів
- Javax validation annotations версії 1.1.0, аотації для статичного аналізу коду засобами IntelliJ Idea.

ВИСНОВКИ

У ході виконання роботи було виявлено та проаналізовано основні проблеми, що виникають при розробці пошукових робіт: можливість створення великого наванта-

ження на аналізований веб-ресурс, обмеженість у ресурсах як аналізатора, так клієнта, високі вимоги до надійності та тривалість процесу індексації.

На основі проаналізованої інформації нами була спроектована UML діаграма класів, на основі якої вже було реалізовано власне сам пошуковий робот. У ході проектування були наведені аргументи на рахунок вибору того чи іншого підходу розробки. Наступним кроком було створення XML формат налаштувань який задовольняв висунутим вимогам та дозволяв швидко конфігурувати робота; для валідації XML файлу налаштувань була розроблена окрема XSD схема.

Для пришвидшення процесу розробки використовувались сторонні бібліотеки. Ключовим фактором при виборі того чи іншого інструменту були простота у використанні та надійність.

ПЕРЕЛІК ПОСИЛАНЬ

1. Manning C. Introduction to Information Retrieval / C. Manning, P. Raghavan, H. Schütze., 2008. – 482 с.
2. Глибовець А.М., Шабінський А.С. Один підхід до побудови інтелектуальної пошукової системи // Наукові записки НАУКМА. Комп'ютерні науки. —том 112. — 2010. - с. 26-30. (4 ст.)

Подход к решению проблемы повышения информационной безопасности предприятия

Гришин Игорь Юрьевич
профессор
Кубанский государственный
технологический университет
Россия, Краснодар

Тимиргалеева Рена Ринатовна
профессор
Кубанский государственный
технологический университет
Россия, Краснодар

Миронов Максим Вадимович
аспирант
Кубанский государственный
технологический университет
Россия, Краснодар

В работе проведён анализ основных угроз информационной безопасности предприятия, характерных для текущего времени. Показано, что важнейшим фактором, влияющим на этот показатель, является выбор метода аутентификации пользователя информационной системы. Предложено в качестве эффективного метода биометрической аутентификации применять аутентификацию по клавиатурному почерку пользователя.

Ключевые слова: информационная безопасность, угроза, аутентификация, динамический признак, пользователь, клавиатурный почерк.

АНАЛИЗ УГРОЗ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ ПРЕДПРИЯТИЯ

С каждым годом увеличивается число негативных воздействий на информацию и ИТ-инфраструктуру предприятий. Согласно ГОСТ Р 51275-2006 по отношению к объекту информации, факторы, воздействующие на безопасность защищаемой информации, подразделяются на внутренние и внешние, что подтверждается статистикой и аналитическими исследованиями ведущих компаний в области информационной безопасности. Так, по данным компании Info Watch [1], в 2016 году в 38% случаях утечки информации произошли из-за воздействия внешних атак на компьютерные системы предприятий, в 52% причиной утечек стали действия внутренних злоумышленников и сотрудников предприятий. По итогам 2016 года Россия заняла второе место в мире по числу известных случаев утечек. В исследуемый период было зарегистрировано 213 случаев утечки конфиденциальной информации из российских коммерческих и государственных организаций. При этом большее число инцидентов информационной безопасности было зафиксировано на предприятиях, принадлежащих следующим отраслям экономики: транспорт, интернет-сервисы и услуги, электронная коммерция и ретейл, медицина, финансовый и банковский сектор, государственный сектор.

Основным типом информации, в отношении которой были зафиксированы инциденты, по данным за 2016 год (рис. 1), стали персональные (85,6%) и платёжные данные (около 7%), коммерческая тайна (5,5%), государственная тайна (1,7%).

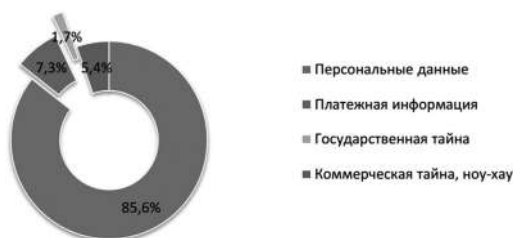


Рис. 1 – Распределение утечек по типам данных в 2016 году

Причиной большинства нарушений информационной безопасности данных стала утечка информации (около 83% случаев), возникшая в результате превышения прав доступа к информации и сервисам, несанкционированного доступа злоумышленника к данным, атаки на систему аутентификации и перебора паролей, перехвата информации или её разглашения сотрудниками (рис. 2).



Рис. 2 – Виды нарушений безопасности информации за 2016 год (по данным Info Watch)

По данным аналитики компании McAfee [2], среди сетевых атак со стороны злоумышленников в апреле 2017 года 22% приходится именно на системы аутентификации пользователей и парольные системы. Поэтому можно сделать вывод об актуальности и распространённости атак, связанных с попытками несанкционированного доступа и компрометации систем аутентификации пользователей и необходимости применения специализированных средств и механизмов защиты от них.

При выборе подобных систем и средств защиты необходимо учитывать область применения компьютерных систем, тип информации, который в них обрабатывается, и требования регулирующих органов к составу и функциям средств защиты, а также итоговому уровню защищённости компьютерной системы [3].

Информация, обрабатываемая в компьютерной системе, разделяется на общедоступную информацию и информацию ограниченного доступа. В свою очередь, информацию ограниченного доступа можно разделить на государственную тайну и конфиденциальную информацию.

Именно информация ограниченного доступа может быть подвержена угрозам несанкционированного доступа вследствие нарушения прав доступа.

Критерии безопасности, приведённые в «Оранжевой книге», позволяют сделать информационную систему надёжной, обеспечивая одновременную обработку информации разной степени доступности группой пользователей без нарушения прав доступа. Благодаря данным критериям можно ранжировать информационные системы по степени надёжности, то есть применять определённые требования для хранения и обработки к различного рода информации в ИС.

Независимо от того, является ли компьютерная система (КС) однопользовательской или многопользовательской, какую информацию обрабатывает и к какому классу защищённости принадлежит, стандарты ГОСТ Р 51241-2008, «Оранжевая книга», ISO/IEC 15408 регламентируют методы управления доступом субъектов, которые подразумевают собой процедуры обязательной аутентификации и идентификации пользователя как неотъемлемой составляющей безопасности КС.

Таким образом, можно сделать вывод, что система аутентификации и идентификации пользователей в компьютерной системе, вне зависимости от её типа, является необходимой и неотъемлемой частью системы защиты информации как в одной КС, так и для всей ИС организации.

СОДЕРЖАНИЕ ПРОЦЕДУРЫ АУТЕНТИФИКАЦИИ

Разграничение доступа является одним из наиболее важных компонентов в борьбе с угрозами, связанными с несанкционированным доступом и его последствиями. Смысл данной меры заключается в том, что субъекту доступа назначаются права на доступ к определённым объектам (так называемая матрица полномочий). Для того чтобы пользователь имел права на доступ к объектам, определённым в матрице полномочий, необходимо будет пройти процедуру авторизации, которой предшествует процедура аутентификации (проверка подлинности) для подтверждения того, что субъект является тем, за кого себя выдаёт.

Для корректной аутентификации необходимо, чтобы пользователь предъявил аутентификационную информацию некую уникальную информацию, которой должен обладать только он и никто иной.

Сам процесс аутентификации, на основании [4], можно представить как цепь последовательных процедур, происходящих в локальной или удалённой системе: однократной (регистрация нового пользователя), длительной по времени (хранение), часто повторяющихся (предъявление аутентификатора, протоколы обмена «клиент сервер», валидация, принятие решения «свой- чужой», передача заявки на авторизацию пользователя).

В соответствии с [5] гарантии аутентификации могут быть разделены на 4 уровня:

- Отсутствие требований конфиденциальности идентификационных данных субъекта;
- Некоторый уровень требований конфиденциальности идентификационных данных субъекта;

- Высокий уровень требований конфиденциальности идентификационных данных субъекта;
- Очень высокий уровень требований конфиденциальности идентификационных данных субъекта.

При оценке степени соответствия системы аутентификации перечисленным выше требованиям необходимо провести анализ не только технико-эксплуатационных характеристик системы и области ее применения, но и оценить потенциальные риски. Согласно [4], для систем аутентификации наиболее характерны следующие типы рисков:

- полный отказ системы аутентификации пользователей;
- отказ в регистрации пользователей;
- отказ в предоставлении доступа легальному пользователю;
- регистрация злоумышленника под видом легального пользователя;
- отказ в процедуре принятия решений о принадлежности пользователя к классу легальных;
- отказ протокола обмена данными;
- хищение и несанкционированное копирование злоумышленником идентификационных данных пользователя.

БИОМЕТРИЧЕСКАЯ АУТЕНТИФИКАЦИЯ

Исходя из оценки качества аутентификации [4, 5, 6], можно сделать вывод о том, что биометрическая аутентификация имеет наибольший обобщённый показатель и является одним из наиболее строгих методов аутентификации. Биометрическая аутентификация является перспективным направлением, что указывает на необходимость её исследования и совершенствования.

Общая схема функционирования биометрических методов аутентификации приведена на рис. 3.

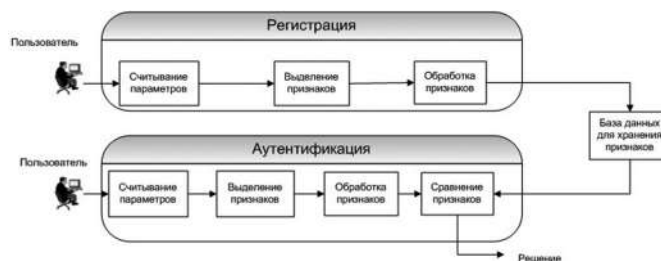


Рис. 3 – Обобщённая схема функционирования биометрических методов аутентификации

Одним из наиболее перспективных динамических биометрических методов аутентификации является распознавание по клавиатурному почерку [3, 4, 7, 8].

Как правило, для этого динамического метода идентификации человека используется его подпись (иногда написание кодового слова). Цифровой код идентификации формируется по динамическим характеристикам написания, то есть для идентификации строится свёртка, в которую входит информация о графических параметрах подписи, временных характеристик нанесения подписи и динамики набора кодового слова. Из оборудования требуется только

стандартная клавиатура. Основная характеристика, по которой строится свёртка для идентификации, – динамика набора кодового слова.

ПОДХОД К РАСЧЁТУ ЭФФЕКТИВНОСТИ РАЗРАБАТЫВАЕМОЙ МОДЕЛИ АУТЕНТИФИКАЦИИ ПОЛЬЗОВАТЕЛЯ НА ОСНОВЕ ПОВЕДЕНЧЕСКОЙ БИОМЕТРИИ

Основными характеристиками, оценивающими качество системы аутентификации, являются её стойкость и надёжность. Однако в данной работе предлагается для оценки качества разрабатываемой модели аутентификации пользователя на основе клавиатурного почерка использовать показатель эффективности, основанный на расчёте ошибок первого и второго рода. Это в первую очередь связано с тем, что при определении принадлежности пользователя к классу «свой - чужой» используется процедура классификации признаков. Как известно, при решении любой задачи классификации, вне зависимости от того, к какой группе относятся признаки - описательные или количественные, - возникает вероятность возникновения ошибок первого и второго рода, которые для задач аутентификации будут определяться следующим образом:

- принятие легального («своего») пользователя за незарегистрированного («чужого») - ошибки первого рода;
- принятие незарегистрированного в системе пользователя («чужого») за легального («своего») - ошибки второго рода, по данным [9, 10] считаются наиболее значимыми для систем аутентификации.

Таким образом, можно считать, что для биометрических систем аутентификации, основанных на распознавании образов, именно ошибки первого и второго рода будут определять качество функционирования системы. При этом на количество ошибок первого и второго рода оказывают непосредственное влияние выбранный алгоритм и метод аутентификации, способ принятия решений, объем обучающей выборки и количество статистических признаков субъекта в базе знаний, а также погрешность и метрика, которые используются при определении аффинности признаков. Для определения аффинности признаков могут использоваться следующие метрики [11, 12]:

- расстояние Хемминга;
- квадратичный индекс нечёткости;
- евклидово расстояние;
- коэффициент сходства Жаккара;
- коэффициент Коха;
- другие.

В соответствии с этим под эффективностью разрабатываемой модели аутентификации пользователя на основе клавиатурного почерка будет пониматься отношение показателей ошибок первого и второго рода при использовании биометрической аутентификации на основе клавиатурного почерка и при использовании других видов биометрической аутентификации.

ВЫВОДЫ

В настоящее время наибольшую опасность для информационной безопасности на предприятии несёт

угроза несанкционированного доступа, что позволяет злоумышленникам произвести незаконное проникновение в информационную систему и получить возможность к реализации угроз, связанных с НСД на предприятии.

Анализ существующих стандартов по защите информации показал, что для защиты от угроз, связанных с НСД, необходимо использовать подсистему аутентификации пользователей, которая является неотъемлемой частью системы защиты информации.

Исследованы различные методы аутентификации, и по результатам была выбрана группа методов аутентификации на основе биометрических характеристик человека. Определён подход к оценке эффективности данных методов.

Проанализированы методы биометрических характеристик человека, выбран один из них – динамика клавиатурного набора, который будет реализован и исследован в дальнейшей работе.

ПЕРЕЧЕНЬ ССЫЛОК

1. Глобальное исследование утечек конфиденциальной информации в 2016 году [Электронный ресурс] // Аналитический центр InfoWatch. – 2017. – Режим доступа до ресурсу: www.infowatch.ru/sites/default/files/report/analytics/russ/InfoWatch_Global_Report_2016_year.pdf?rel=1.
2. McAfee Labs Threats Report April 2017 [Электронный ресурс] // McAfee Labs. – 2017. – Режим доступа до ресурсу: <https://www.mcafee.com/ru/resources/reports/quarterly-threats-mar-2017.pdf>.
3. Гришин И. Ю. Анализ перспективных подходов к проектированию систем безопасности распределённых компьютерных сетей / Игорь Юрьевич Гришин. // Вестник Российского нового университета. Серия: Сложные системы: модели, анализ и управление. – 2015. – №2. – С. 36–40.
4. Гришин И. Ю. Аутентификации пользователей в компьютерной системе на основе поведенческой биометрии / И. Ю. Гришин, М. В. Миронов // Проблеми інформатики та моделювання. Тезиси шістнадцятої міжнародної науково-технічної конференції / И. Ю. Гришин, М. В. Миронов. – Харків: Національний технічний університет «Харківський політехнічний інститут», 2016. – С. 28.
5. Тимиргалеева Р. Р. Формирование бизнес-модели инновационного развития предприятия на основе сценарного подхода / Р. Р. Тимиргалеева, И. Ю. Гришин // Глобальные вызовы в экономике и развитие промышленности (INDUSTRY-2016) Труды научно-практической конференции с зарубежным участием / Р. Р. Тимиргалеева, И. Ю. Гришин. – Санкт-Петербург: Санкт-Петербургский политехнический университет, 2016. – С. 526–535.
6. Гришин И. Ю. Проблемы управления зенитными ракетными комплексами / И. Ю. Гришин, М. К. Можар, В. М. Решетник. // Наука и оборона. – 1994. – №3. – С. 27–32.
7. Ларина Р. Р. Метод динамического программирования и принцип максимума в задачах оптимизации маркетинг-логистических решений / Р. Р. Ларина, И. Ю. Гришин // Труды X Международной ФАМЭТ-2010 конференции /

Р. Р. Ларина, И. Ю. Гришин. – Красноярск: Красноярский государственный торгово-экономический институт, 2011. – С. 119–123.

8. Тимиргалеева Р. Р. Информационно-логистическое обеспечение процесса управления сложными организационно-экономическими системами / Р. Р. Тимиргалеева, И. Ю. Гришин. – Симферополь: ИТ «Ариал», 2013. – 248 с.

9. Гришин И. Ю. Модели предоставления, типы и основные проблемы информационной безопасности облачных вычислений / И. Ю. Гришин, А. М. Рябов, Р. А. Скидан // Стратегия качества в промышленности и образовании / И. Ю. Гришин, А. М. Рябов, Р. А. Скидан. – Варна-Днепропетровск: Варнский технический университет, 2016. – С. 534–537.

10. Тимиргалеева Р. Р. Обеспечение информационной безопасности и непрерывности бизнес-процессов при использовании мобильных технологий / Р. Р. Тимиргалеева, И. Ю. Гришин // Цифровая экономика и «Индустрия 4.0»: проблемы и перспективы. Труды научно-практиче-

ской конференции с международным участием / Р. Р. Тимиргалеева, И. Ю. Гришин., 2017. – С. 489–493.

11. Гришин И. Ю. Применение средств биометрической аутентификации для авторизации пользователя автоматизированного рабочего места / И. Ю. Гришин, М. Г. Ефимчик // Информатика, управління та штучний інтелект. Матеріали третьої Міжнародної науково-технічної конференції студентів, магістрів та аспірантів / И. Ю. Гришин, М. Г. Ефимчик. – Харків: Национальный технический университет «ХПИ», 2016. – С. 14.

12. Особенности применения биометрических методов для аутентификации обучаемого в системе дистанционного образования / Р. Р. Тимиргалеева, И. Ю. Гришин, М. В. Миронов, М. Г. Ефимчик // Филологические и социокультурные вопросы науки и образования Сборник материалов I Международной научно-практической конференции / Р. Р. Тимиргалеева, И. Ю. Гришин, М. В. Миронов, М. Г. Ефимчик. – Краснодар: КубГТУ, 2016. – С. 219–229.

Функциональный контроль процессоров в системах на кристалле

Жабин В.И.
д.т.н., проф.
КПИ им. Игоря Сикорского
Киев, Украина

Предложен метод обеспечения функционального контроля процессоров на программно-аппаратном уровне за счет проверки правильности прохождения участков программы. Метод обеспечивает малую задержку реализации вычислительных процессов и может быть использован для контроля хода произвольных программ.

Ключевые слова: система на кристалле, контроль хода программы, контроль процессора.

Важным требованием к вычислительным системам, работающим в контуре управления процессами в реальном времени, является высокая надежность и быстродействие. В ряде случаев для обеспечения требуемого времени обработки информации, которое обусловлено внешними факторами, необходимо использовать мультипроцессорные системы. Достижения в области интегральной технологии позволяют создавать параллельные системы CSoC (Configurable System on Chip – конфигурируемая система на кристалле), которые реализуются на ПЛИС с использованием интерактивных средств разработки. В общем случае микросхемы содержат вычислительные ядра, память и программируемую логику, что дает потенциальную возможность оптимально адаптировать аппаратные средства к конкретному применению.

Повышение эффективности методов и средств контроля вычислительных систем является одной из важнейших задач вычислительной техники [1]. Существует множество применений систем реального времени, при которых отказ оборудования может привести к катастрофическим последствиям. В этих случаях исправность оборудования должна контролироваться постоянно. Контроль, который осуществляется во время функционирования системы, принято называть функциональным контролем.

Методы контроля разделяют на программные, аппаратные и комбинированные [2]. Среди различных методов необходимо выделить такие, которые не могут применяться в качестве функциональных методов, поскольку они связаны с разрушением информации или частичным демонтажем системы. К таким методам можно отнести методы эмуляции, некоторые методы сигнатурного анализа и программного тестирования, приводящего к потере информации в системе [2, 3]. Конечно, процессор может выполнять в цикле управления тестовую программу, если на это имеется оперативный запас времени. Однако выполнение тестовых программ может привести к превышению допустимой длительности цикла управления.

Средства контроля процессоров можно разделить на контролирующие результаты вычислений и событий. Поскольку для систем управления быстрыми процессами первостепенным фактором является обеспечение допустимого времени обслуживания внешних событий, то целе-

сообразно использовать неразрушающие методы контроля, которые вносят минимальную задержку выполнения программы. К таким в меньшей степени относятся методы контроля результата, поскольку требуют повторного или параллельного счета, что связано с дополнительными затратами времени и ресурсов [3].

Методы контроля событий можно разделить по признаку предмета контроля на контроль временных и контроль логических событий. В первом случае могут оцениваться временные интервалы прохождения участков программы, интервалы между событиями и т.д. Для этого программа должна запускать и оценивать состояние, возможно, нескольких таймеров. В качестве логических событий может оцениваться количество выполненных команд, количество пройденных участков, число обращений к определенным ячейкам памяти и т.д.

Недостатком методов контроля событий, как правило, является их специализация под конкретную структуру процессора, а иногда и под конкретную программу, что существенно сужает область их применения [4, 5].

Таким образом, существует проблема создания метода контроля процессоров, который может быть использован для контроля хода произвольных программ и не требовать больших затрат времени на контроль.

Рассмотрим метод функционального контроля хода программы процессора по косвенным признакам, который вносит минимальную составляющую времени в цикл управления.

В состав каждого процессорного модуля вводится блок функционального контроля (рис. 1), который обеспечивает решение комплекса задач по определению исправности процессорного модуля. Современная технология конфигурируемых систем на кристалле (CSoC) позволяет вводить в состав микросхемы дополнительные устройства.

Контроль осуществляется программно-аппаратным методом. Блоки устройства контроля выполняют следующие функции. Блок выделения строки команды определяет момент наличия на системной магистрали слова команды, постоянная часть которой (код операции, способ адресации операндов, формат команды) запоминается в регистре команды.

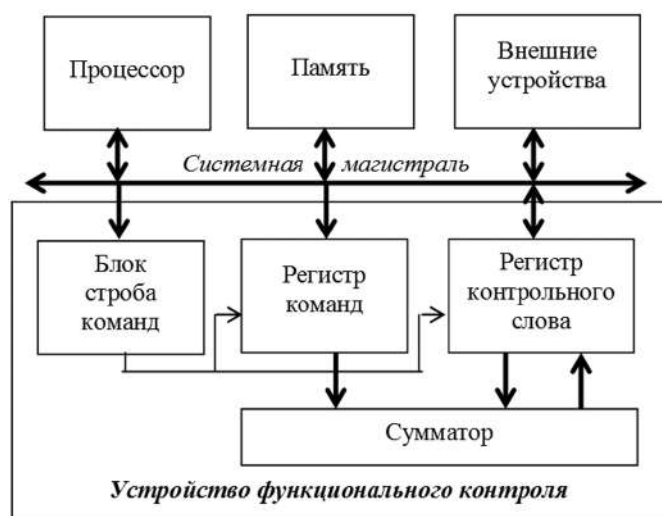


Рис. 1. Структура процессорного модуля

Выделенная часть слова команды прибавляется к контрольному слову с помощью сумматора и записывается в регистр контрольного слова. Контрольное слово может быть считано программой.

Любая программа может быть адаптирована к применяемому методу контроля. Это осуществляется автоматически на этапе компиляции программы. Как известно, любая программа может быть представлена совокупностью структур *следование*, *ветвление* и *повторение*. Если показать осуществимость адаптации каждой из перечисленной структур программы к методу контроля, то можно утверждать, что адаптация любой программы возможна.

Структура *следование* является линейным участком программы и поэтому возможность применения контроля на этом участке не вызывает особых затруднений. Вначале участка вставляется команда сброса регистра команд в устройстве контроля. В конце участка считывается содержимое этого регистра и сравнивается с непосредственным операндом (вычисленным компилятором). Несовпадение значений означает ошибку.

В случае структуры *ветвление* можно альтернативные ветви рассматривать как линейные участки. Можно также с помощью команд, не несущих функциональной нагрузки (в смысле реализуемого алгоритма), привести альтернативные ветви к одинаковому значению контрольного слова за счет коррекции соответствующего регистра. В этом случае участок становится квазилинейным (в дальнейшем просто линейным). Линейные участки могут объединяться.

Структуры *повторение* (циклы) можно рассматривать как один или несколько линейных участков внутри цикла. Таким образом, количество повторений тела цикла не будет нарушать ход контроля.

Для систем реального времени необходимо обеспечить эффективное обслуживание внешних прерываний, которые могут произойти во время выполнения любой программы, если прерывание разрешено. Поэтому, одним из возможных приемов является добавление в подпрограмму обработки прерывания команд сохранения значения контрольного слова, а затем выполнять подпрограмму по об-

щим правилом. Перед возвратом на основную программу необходимо восстановить старое значение контрольного слова. Таким образом, подпрограмма обработки прерывания также будет проконтролирована, а значения информативных параметров основной программы останутся неизменными после возвращения из подпрограммы обработки прерывания.

Эффективность средств встроенного функционального контроля определяется несколькими факторами:

- степень достоверности контроля функционирования процессора;
- затратами времени и ресурсов вычислительной системы на проведение функционального контроля.

Предлагаемый метод функционального контроля, безусловно, не может выявить все ошибки, возникающие в процессе выполнения программы. Поскольку практически при контроле проверяется последовательность выполнения и целостность команд, то ошибки, связанные с отказом ячеек памяти, каналов пересылки информации, несанкционированным обращением к разделам памяти контролируются. Это означает, что контролю подвергается память и устройство управления. Для контроля арифметических устройств необходимо вводить дополнительные тестовые команды, хотя ветвление программы по результату арифметической или логической операции косвенно проверяют арифметику и логику.

Если удастся сформировать достаточно длинный линейный участок, то доля специально включенных для обеспечения контроля команд относительно невелика. Действительно, при формировании квазилинейного участка при каждом ветвлении добавляется одна команда для коррекции контрольного слова. Если считать, что каждая альтернативная ветвь равновероятна, то число добавочных команд будет равно $\gamma/2$, где γ – частота команд ветвления в программах. Например, при $\gamma = 0,1-0,2$ число добавочных команд будет составлять 5-10%.

Таким образом, предлагаемый метод функционального контроля может быть применен для контроля хода любых программ, адаптированных к данному методу на этапе компиляции, и минимизирует непроизводительные затраты времени при решении задач реального времени.

ПЕРЕЛІК ПОСИЛАНЬ

1. Додонов А.Г. Введение в теорию живучести и вычислительных систем / А.Г.Додонов, Н.Г.Кузнецова, Е.С.Горбачик. – Киев: Наук. думка, 1990. – 184 с.
2. Каган Б.Н. Мкртумян И.Б. Основы эксплуатации ЭВМ / Б.Н.Каган, И.Б.Мкртумян. – М.: Энергоатомиздат, 1988. – 492 с.
3. Гуляев В.А. Организация систем диагностирования вычислительных машин / В.А.Гуляев. – Киев: Наук. думка, 1979. – 116 с.
4. Ушаков Г.Н. Аппаратный контроль и надежность спецЭВМ / Г.Н.Ушаков. – М.: Сов. радио, 1988. – 253 с.
5. Фараджев В.А. Комплекс аппаратных средств для отладки программ реального времени / В.А. Фараджев // УсиМ. – 1980. – №1. – С. 49-51.

Концепція сервісу програмного забезпечення

Хмелюк Володимир
Сергійович
ст. викладач
КПІ ім. Ігоря Сікорського
Україна, Київ

Майер Ілля Сергійович
студент
КПІ ім. Ігоря Сікорського
Україна, Київ

Озеракін Микита
Дмитрович
студент
КПІ ім. Ігоря Сікорського
Україна, Київ

Тимошенко Олександр
Олександрович
студент
КПІ ім. Ігоря Сікорського
Україна, Київ

Тези доповіді містять опис концепції програмного сервісу, як безперервного та безетапного еволюційного процесу надання програмних послуг, що заміняє стандартний життєвий цикл програмного забезпечення (планування, розробку, тестування, модифікацію, супровід) та підтримуючі процеси (оплату, інтеграцію, оренду, взаємодію клієнта та провайдера).

Ключові слова: програмне забезпечення, розробка, сервіс, провайдер, замовник

Завдяки планомірному розвитку та досягненням технічного прогресу багато промислових галузей переходять від продуктових політик до сервісних. Так, наприклад, замість того, щоб купити супутниковий телефон у певної компанії, вкласти з нею договір та оплачувати щомісячну абонплату (продуктовий підхід) – ви маєте змогу купити телефон будь-якого виробника на ваш розсуд, обрати компанію, що буде вам надавати послугу зв'язку (сервіс) і надалі купувати додаткові послуги у цієї компанії за необхідності, орієнтуючись лише на власні потреби та товщину гаманця. Або ж замість купувати собі електрогенератор, ви можете встановити собі лічильник використання електроенергії і платити за сервіс надання доступу до електричних мереж міста згідно обсягів використаної електроенергії та тарифу. Ваш автомобіль теж ймовірно обслуговується в сервісному відділі певного автосалону. Подібних прикладів існує безліч.

І, хоча сфера інформаційних технологій розвивається найбільш динамічно та зазвичай є форпостом запровадження різноманітних нововведень та новаторських рішень, з переходом від продуктового підходу до сервісного не все так гладко. Так в частині «хард» давно вже практикується надання сервісів оренди серверних потужностей, сервісів хмарних містилиць та хмарних обчислень, сервісів послуг інтернет-провайдерів, тощо. А от в частині «софт», тобто програмного забезпечення, – повна тиша. Ви маєте змогу лише купити програмний продукт, або замовити його розробку, що по суті одне й те ж. Правда деякі великі компанії розробники ПЗ, наприклад Microsoft, все ж намагаються запровадити ідею надання послуг використання ПЗ (той же Office 365), але такі виключення поодинокі і лише підтверджують сумне правило.

І, оскільки ще не існує типових сервісів програмного забезпечення¹, спробуємо з'ясувати, якими мають бути такі сервіси і яким вимогам вони мають задовольняти. Отож, основним видом сервісу програмного забезпечення

(СПЗ) має стати сервіс безперервної розробки ПЗ (що має замінити собою весь життєвий цикл ПЗ: планування, розробку, тестування, модифікацію, супровід, оплату, тощо). Бо, хоча деякі імениті компанії-розробники наголошують, що вони надають сервіс ПЗ, насправді вони надають лише можливість вибору однієї з продуктових політик використання вже існуючого продукту.

Спробуємо уявити, як би могла виглядати типова взаємодія замовника послуги ПЗ (клієнта сервісу) та розробника ПЗ (в даному контексті провайдера сервісу):

А) Я клієнт:

Я реєструюся на сайті сервісу ПЗ як Замовник. Я опишую функціонал, який я хочу отримувати від цільового ПЗ, орієнтовні терміни виходу на робочий режим ПЗ, допустиму вартість умовної години розробки та інші необов'язкові характеристики сервісу. Мою заявку зі статусом «Пошук виконавця» бачать всі зареєстровані розробники ПЗ (провайдері). Зацікавлені провайдері пропонують свої послуги (сервіс ПЗ). Обговорення деталей ведеться тут-же на сайті. В мене є можливість побачити контактні дані провайдерів та зв'язатися з ними безпосередньо для обговорення деталей та нюансів взаємодії. Після вивчення пропозицій я обираю провайдера та замовляю його послугу ПЗ, заключаючи з ним Договір та закриваючи цим свою заявку. З цього моменту мені надається сервіс ПЗ. Я оплачую обумовлену грошову суму на рахунок провайдера, якщо це передбачено (prepaid²). Далі в CS³ я створюю початкові вимоги до ПЗ. Провайдер оцінює мої вимоги в умовних трудовитратах. Я маю змогу деталізувати вимоги, змінити, відмінити, перевпорядкувати, тощо та запропонувати свою вартість реалізації кожної з вимог. Після змін у вимогах і я і провайдер можемо змінити власну запропоновану ціну реалізації вимог. Після певного ітеративного процесу узгодження вимог та їх вартості (так званий аукціон вимог) я підтверджую готовність обраних вимог до реалізації. Провайдер підтверджує факт запуску вимог у розробку.

¹ Під сервісом програмного забезпечення СПЗ надалі будемо розуміти повний комплекс робіт з підтримки життєвого циклу ПЗ в поєднанні з фінансовими аспектами розподілених в часі.

² Слово «prepaid» можна перекласти як «передплата». Якщо коротко, prepaid — це спосіб розрахунку з провайдером послуги. Традиційно існують два варіанти оплати надання послуг: кредитний і дебетний. Якщо користуєшся послугою, а потім здійснюєш оплату рахунків що надійшли — це кредитна система. Дебетна система початково передбачає наявність на вашому рахунку певної суми. Наприклад, авансову форму розрахунку можна віднести до дебетових систем: Ви кладете на рахунок певну суму і після цього її витрачаєте

³ CS – аббревіатура референсної програмної системи підтримки СПЗ CleanSlate, що розробляється за участі студентів та викладачів НТУУ «КПІ»

Після реалізації вимоги до ПЗ провайдер повідомляє мене про реалізацію вимоги в новій версії ПЗ, яка автоматично розгортається у мене підсистемою розгортання (згідно налаштувань розгортання та політик підтримки версійності ПЗ). Я підтверджую виконання вимоги в повному обсязі після перевірки функціонування ПЗ. В цей момент з мого рахунку списується оговорена грошова сума а вимога вважається закритою. В процесі моєї взаємодії з провайдером можуть виникати нові вимоги, вимоги можуть ієрархічно розбиватися на дрібніші, об'єднуватися, відмінюватися і т.д. згідно алгоритму життєвого циклу вимог. Одними з підвидів вимог будуть вимоги на виправлення помилок, покращення або зміну функціоналу, документування, створення навчальних матеріалів та інші. Надання сервісу провайдером припиняється за двосторонньою згодою, або в випадках передбачених укладеним Договором. Типовим варіантом є нескінченне надання послуги. Переваги, які я отримую від використання концепції програмного сервісу:

- можливість доступу до сервісу ПЗ за допомогою різних робочих обчислювальних машин (ноутбук, смартфон);
- механізм пошуку та конкурсного вибору провайдера сервісу ПЗ;
- середовище для спілкування з провайдером щодо замовленого сервісу ПЗ;
- мінімізацію часу та зусиль на створення технічного завдання на розробку ПЗ, методики перевірок та тестувань та іншої супутньої документації;
- можливість формування поточних версій деяких паперових документів в будь-який час (наприклад ТЗ, звіти, тощо);
- прозорість виконаних робіт провайдером згідно вимогам;
- можливість надгнучкої зміни вимог до ПЗ;
- систему встановлення пріоритетів реалізації вимог узгоджену з провайдером;
- можливість та середовище обліку помилок та відстеження їх виправлення;
- володіння та доступ до містилиця поточних напрацювань (вихідні коди, зкомпільовані модулі, документація, тощо);
- можливість використання професійного ревізора оцінок реалізації вимог до ПЗ;
- можливість повного або часткового залучення зацікавлених осіб в процес розробки зі сторони замовника згідно обраної рольової політики в межах надання сервісу ПЗ;
- спрощення розгортання та оновлення ПЗ;
- своєчасність та оперативність оплати робіт виконаних провайдером;
- відстеження та планування бюджету на сервіс ПЗ;
- можливість зміни провайдера в будь-який час без втрати існуючих напрацювань та історії змін вимог до ПЗ;

Б) Я провайдер:

Я реєструюся на сайті сервісу ПЗ як Провайдер. Я вказую профільні предметні області (в яких я маю певні напрацювання та/або конкурентні переваги), посилання на

інформацію по реалізованим мною проектам, кваліфікацію та склад команди розробки та супроводження ПЗ, орієнтовну вартість години надання СПЗ, тощо. Мій профіль провайдера разом з рейтингом, відгуками та іншими оцінками бачать як всі зареєстровані клієнти СПЗ так і інші розробники ПЗ (провайдери). Зацікавлені клієнти звертаються до мене з метою уточнення деталей надання СПЗ для можливої подальшої співпраці. Інші провайдери мають змогу звернутися до мене як до співвиконавця з метою доручити мені виконання певних робіт в межах їх власного СПЗ (тобто реалізується платформа пошуку виконавців робіт, але з урахуванням їх планової зайнятості та інших факторів). В мене є можливість побачити контактні дані клієнтів (якщо вони не вказали зворотного) та зв'язатися з ними безпосередньо для обговорення пропозиції по цільовому СПЗ. Після обговорення пропозицій я вкладаю договір на надання СПЗ з клієнтами або іншими провайдерами. Я розпочинаю уточнення початкових вимог замовника до ПЗ за допомогою механізму «аукціон задач» на базовому рівні (тобто до рівня достатнього для планування початкових робіт). Після підтвердження клієнтом починаю виконання робіт по реалізації вказаних вимог. Після виконання вимоги очікую оцінки клієнта на відповідність реалізації. В разі підтвердження клієнтом повного виконання вимоги проводиться фіксація в системі інформації про елемент оплати (фізичне перерахування коштів з таким обліком на пряму не зв'язане). Подальші вимоги до ПЗ з'являються в системі в процесі надання СПЗ. Вони можуть генеруватися замовником або розробником з обов'язковим затвердженням зацікавленими сторонами. Переваги, які я отримую від використання концепції програмного сервісу:

- можливість доступу до сервісу ПЗ за допомогою різних робочих обчислювальних машин (ноутбук, смартфон);
- механізм пошуку клієнтів або генеральних підрядників для сервісу ПЗ;
- середовище для спілкування з клієнтами щодо сервісу ПЗ, що надається;
- мінімізацію часу та зусиль на створення технічного завдання на розробку ПЗ, методики перевірок та тестувань та іншої супутньої документації;
- можливість формування поточних версій деяких паперових документів в будь-який час (наприклад звітність по виконанню робіт, часові графіки, іншу інформацію необхідну для подальшого планування та виконання робіт);
- можливість надгнучкої зміни вимог до ПЗ;
- систему встановлення пріоритетів реалізації вимог узгоджену з клієнтом;
- можливість та середовище обліку помилок та відстеження їх виправлення;
- можливість легкого залучення до виконання робіт інших провайдерів СПЗ в якості підрядників;
- спрощення розгортання та оновлення ПЗ;
- можливість передачі клієнта іншому провайдеру (за згоди клієнта) без втрати існуючих напрацювань та історії змін вимог до ПЗ;

- відстеження та часткове прогнозування фінансових надходжень;
- гарантування оплати виконаних робіт клієнтом незалежно від факту повної реалізації вимог у випадку зміни вимог клієнтом, або при припиненні робіт з ініціативи клієнта;
- гнучкі механізми прогнозування та відстеження часових термінів реалізації вимог, задач, робіт, тощо;
- мінімізація бюрократичних витрат;
- значне підвищення взаєморозуміння з клієнтом та максимальне уникнення конфліктних ситуацій;
- реалізація процедур та механізмів можливого вирішення конфліктів за допомогою залучення третьої сторони;
- середовище тісної інтеграції зовнішніх систем розробки та супроводження ПЗ клієнтів в спеціалізовані АРМ провайдера;

Впровадження концепції сервісу програмного забезпечення в життя передбачає проведення низки організаційних заходів, розробки нових та уточнення існуючих мето-

дологій розробки та супроводження ПЗ, а також створення програмного забезпечення для підтримки середовища функціонування бізнес-площадок надання СПЗ.

Першою референсною системою підтримки СПЗ є програмна система CleanSlate, що розробляється за участі студентів та викладачів НТУУ «КПІ» та буде детально розглянута в низці наступних тематичних публікацій.

ПЕРЕЛІК ПОСИЛАНЬ

1. Microsoft Office 365 [Електронний ресурс] – Режим доступу до ресурсу: <https://products.office.com/en-us/office-365-personal>.
2. Лармен К. Scaling Lean & Agile Development: Thinking and Organizational Tools for Large-Scale Scrum / Креіг Лармен., 2008. – 368 с. – (1-ше). – (978-0321480965).
3. Андерсон Д. Kanban: Successful Evolutionary Change for Your Technology Business / Девід Андерсон., 2010. – 278 с. – (978-0984521401)
4. Кніберг Г. Scrum and XP from the Trenches / Генрік Кніберг., 2007. – 142 с. – (2-ге). – (978-1-4303-2264-1).

Аналіз методів об'єднання таблиць у розподілених сховищах даних в оперативній пам'яті

Подрубайло О.О.

асистент

КПІ ім. Ігоря Сікорського
Україна, Київ

Лук'яненко Я.В.

студент

КПІ ім. Ігоря Сікорського
Україна, Київ

Доповідь присвячена аналізу методів об'єднання розподілених таблиць з метою їх застосування в розподілених сховищах даних в оперативній пам'яті. Визначено фактори, що відрізняють об'єднання в розподілених сховищах даних в оперативній пам'яті від аналогічної операції в сховищах іншого класу. Розглянуті алгоритми порівнювались за об'ємом пересилок та часом виконання. Розроблено програмне середовище для моделювання операцій об'єднання в таких системах.

Ключові слова: розподілене сховище в оперативній пам'яті, розподілене об'єднання, об'єднання хешуванням, фільтри Блума, шляхове об'єднання, модифіковане шляхове об'єднання

На сьогодні все більшої популярності набувають розподілені сховища даних в оперативній пам'яті (In-memory data grid, IMDG). Зберігаючи інформацію безпосередньо в основній пам'яті комп'ютерів, такі системи надають суттєву перевагу користувачам, оскільки забезпечують високу швидкість доступу до даних. Через особливості фізичної організації оперативної пам'яті, найвищої швидкодії сховища в оперативній пам'яті досягають при читанні інформації по асоціативному ключу. Однак для повноцінного застосування систем класу In-memory data grid виключно читання за ключем замало. Сценарії використання існуючих сховищ даних передбачають також об'єднання таблиць. Існуючі реалізації IMDG не мають такої функціональності, або вона реалізована дуже неефективно, що призводить до суттєвого падіння швидкості доступу до даних у багатьох сценаріях. При цьому операція об'єднання виконується на стороні клієнта, що знижує швидкодію системи до швидкодії мережевого з'єднання.

Ця доповідь присвячена аналізу методів об'єднання розподілених таблиць з метою оцінки їх застосовності у розподілених сховищах даних в оперативній пам'яті. Дослідження зосереджено на внутрішньому симетричному об'єднанні за ознакою рівності ключа в розподілених таблицях.

IMDG являє собою набір розподілених асоціативних масивів (таблиць, регіонів), розміщених на певній множині серверів у мережі. Дані в таблицях представлені в об'єктно-орієнтованій формі. Зазвичай, для забезпечення надійності та відмовостійкості системи, застосовується реплікація даних на декілька вузлів кластеру. Тобто можна виділити наступні фактори, важливі для розподіленого об'єднання:

- Оскільки ключем об'єднання не є первинний ключ таблиці, відсутня можливість аналітично визначити множину вузлів системи, що зберігають певний ключ об'єднання.

- Записи у вигляді об'єктів не мають фіксовану довжину. Маючи довільний рівень вкладеності, розмір різних записів в одній таблиці може суттєво відрізнитися.

- Швидкість передачі даних мережею на декілька порядків менша, ніж швидкість читання даних з оперативної пам'яті. Тобто ефективнішим буде алгоритм, що зменшить кількість пересилок, навіть шляхом більшої кількості операцій читання\запису з основної пам'яті.

- Реплікація даних означає, що множини даних, які зберігають різні вузли сховища, мають ненульовий перетин. Це потенційно надає можливість обирати вузли, що прийматимуть участь в об'єднанні, і, таким чином, мінімізувати об'єм пересилок.

Виходячи з вищевказаного, аналіз методів об'єднання відбувався за наступними критеріями:

- Об'єм пересилок даних (байт)
- Найбільший час виконання алгоритму на окремому вузлі (мс)
- Складність реалізації алгоритму в існуючих розподілених сховищах (якісна характеристика)

Аналіз виконувався на основі експериментальних досліджень. Для виконання експериментів була розроблена програмна модель розподіленого сховища даних в оперативній пам'яті, що дозволяє оцінити швидкодію різних методів розподіленого об'єднання. Алгоритми реалізовані у вигляді окремих класів. Система легко розширюється шляхом додавання нових класів з реалізаціями алгоритмів об'єднання.

Для моделювання були виділені наступні параметри:

- Відношення об'ємів даних у таблицях, що приймають участь в об'єднанні. (У кількостях записів)
- Закон розподілення розмірів записів у таблицях. (константний, лінійний, Гауса, Ерланга). Різний розмір запису зумовлений кількістю рекурсивних вкладень в об'єкт-запис.

- Пропускна здатність мережі (біт\с)
- Затримка мережі (мс)

- Кількість вузлів
- Ступінь реплікації
-

Приклад роботи програми моделювання представлений на рис. 1

Рис. 1 Результат роботи системи моделювання

В даній роботі аналізувалися наступні алгоритми розподіленого об'єднання:

1. Hash join

Хеш-з'єднання виконується в дві стадії: компоновка та проба. Під час компоновання здійснюється читання всіх рядків першого вхідного потоку (часто, його називають лівим потоком або потоком компоновання), хешування рядків по ключам з'єднання еквівалентності і створення в оперативній пам'яті хеш-таблиці. [1] Під час проби здійснюється читання всіх рядків другого вхідного потоку (часто його називають правим потоком або пробним потоком), хешування рядків цього потоку по тим же ключам з'єднання еквівалентності, а потім здійснюється перегляд або пошук відповідників рядків в хеш-таблиці. Так як хеш-функції можуть бути схильні до колізій (два різних значення ключа мають однаковий хеш), доводиться перевіряти кожне потенційне відповідність, що необхідно для гарантії того, що в цьому випадку дійсно схожість зумовлена з'єднанням, а не колізією.

2. Bloom join

Цей алгоритм традиційно використовується для ефективного обміну даними між вузлами кластера.[2]

Ймовірнісна структура даних, котра називається фільтр Блума, дозволяє перевірити належність елемента множині. При цьому існує можливість отримати хибнопозитивні спрацьовування (елемента в множині немає, але структура даних повідомляє, що він є), але не псевдонегативні. Фільтр Блума може використовувати будь-який обсяг пам'яті, заздалегідь заданий користувачем, причому чим він більший, тим менша ймовірність помилкового спрацьовування.

3. Track join

Метод шляхового об'єднання (track-join), мінімізує кількість пересилань кортежів по мережі. [3] Основна ідея цього методу полягає у визначенні мети відправки кожного конкретного кортежу. Метод шляхового об'єднання істотно знижує використання мережевих ресурсів у порівнянні з іншими відомими методами виконання об'єднань розподілених таблиць. Однак він орієнтований на використання в традиційних базах даних, де зберігаються кортежі строго типізовані і фактичний обсяг збережених даних в різних рядках однієї таблиці відрізняється несуттєво.

4. Modified track join

Модифікований алгоритм шляхового об'єднання приймає до уваги розмір кожного запису у сховищі [4], підвищуючи ефективність об'єднання в умовах, коли розміри записів в одній таблиці сильно відрізняються один від одного.

ПЕРЕЛІК ПОСИЛАНЬ

1. C. Balkesen et al. Multicore, main-memory joins: Sort vs hash revisited. // PVLDB, 7(1) – Sept. 2013 – p.85-96.
2. M.-C. Albutiu, A. Kemper, and T. Neumann. Massively parallel sort-merge joins in main memory multi-core database systems. // PVLDB, 5(10) – 2012 – p.1064-1075.
3. O. Polychroniou, R. Sen and K. Ross. Track join: distributed joins with minimal network traffic. // SIGMOD Conference – 2014 – p. 1483-1494.
4. Бузовский О.В., Подрубайло А.А. Методы и алгоритмы объединения таблиц для распределенных хранилищ данных в оперативной памяти. // Вісник НТУУ «КПІ». Інформатика, управління та обчислювальна техніка: Зб. наук. пр. – К.: Век+, – 2014. – № 60. С.74-83

Рецензент: д.т.н., проф. каф. ОТ, КПІ ім. Ігоря Сікорського, О.В. Бузовський

Параметрування програмних блоків модуля sm1281 для побудови системи діагностування ГПА

Заміховський Леонід Михайлович

завідувач кафедри «Інформаційно-телекомунікаційних технологій та систем»

Івано-Франківський національний технічний університет нафти і газу
Україна, Івано-Франківськ

Іванюк Наталія Іванівна

асистент кафедри «Інформаційно-телекомунікаційних технологій та систем»

Івано-Франківський національний технічний університет нафти і газу
Україна, Івано-Франківськ

Розглядається процедура параметрування програмних блоків технологічного модуля стосовно вимірювання параметрів вібрації SM1281 для побудови системи діагностування газоперекачувальних агрегатів на базі програмованого логічного контролера Simatic S7-1200 та середовища проектування TIA Portal V13SP1 (Totally Integration Automation) «Siemens».

Ключові слова: Параметрування, програмний блок, діагностика, ГПА, віброшвидкість, налаштування, компіляція.

Сьогодні для визначення технічного стану лопатевого апарату газоперекачувальних агрегатів (ГПА) в процесі експлуатації найбільшого розповсюдження знайшли методи віброакустичної діагностики. Технічна реалізація вказаних методів базується на використанні як спеціально розроблених (нестандартних) технічних засобів збору і обробки віброакустичної інформації, так і стандартних технічних засобів - в основному для проведення експериментальних досліджень вібраційного стану ГПА та дослідження нових методів діагностування технічного стану його лопатевого апарату.

В [1] розглядається структура системи вібраційного діагностування лопатевого апарату газоперекачувальних агрегатів, основу якої складає вібраційний модуль SM1281 з комплектом акселерометрів (промисловий стандарт для акселерометрів ІЕРЕ – Integrated Electronics Piezo-Electric), який є новітньою розробкою концерну Siemens.

З метою практичної реалізації вказаної системи розглянемо процедуру параметрування програмних блоків модуля SM1281.

Для отримання від комплекту акселерометрів, які встановлені в найбільш віброактивних місцях на корпусах основних агрегатів ГПА, віброакустичної інформації та подальшої її обробки у програму, що розробляється, необхідно додати програмний блок «SM1281_Module» [2]. Налаштування блоку «SM1281_Module» показані на рис. 1 та 2.

Основними налаштуваннями функціонального блоку «SM1281_Module» є параметри HW_Submodule, що здійснюють присвоєння відповідного підмодуля SM1281 у конфігурацію PLC S7-1200, «IPAddress» – параметр, що використовується для встановлення IP-адреси модуля (у шістнадцятковому форматі), а також параметр OpMode, завдяки якому вибирається режим роботи модуля SM 1281 [3]. Можливе задання наступних значень:

0: STOP: Система готова

1: RUN: Моніторинг

2: RUN: Моніторинг деактивовано

3: RUN: Вимірювання

4: RUN: Діагностика системи

Значення > 4 інтерпретуються як 0.

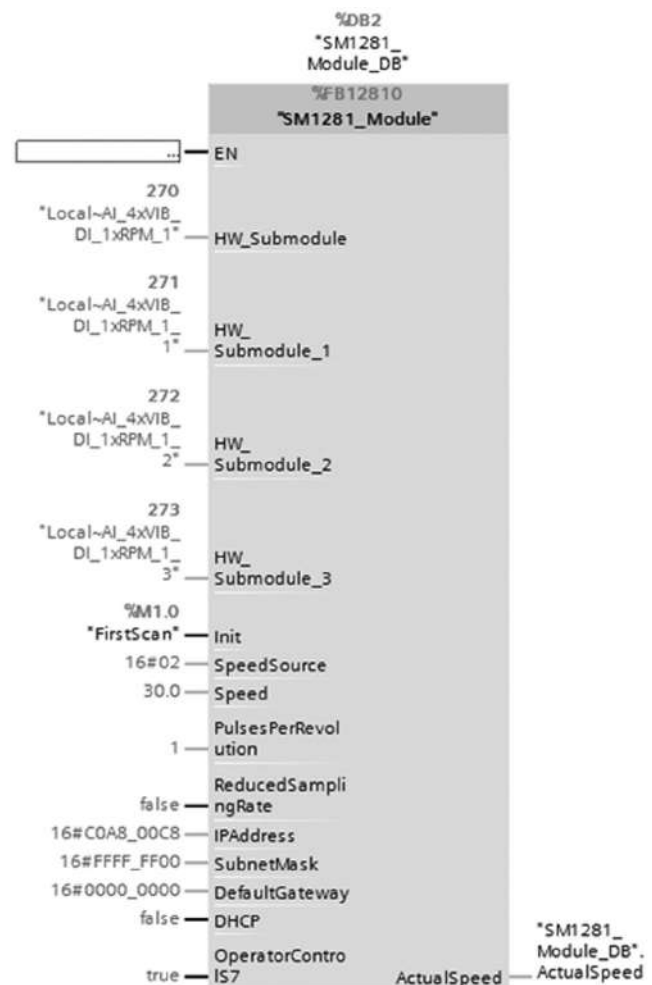


Рис. 1 - Налаштування блоку «SM1281_Module»

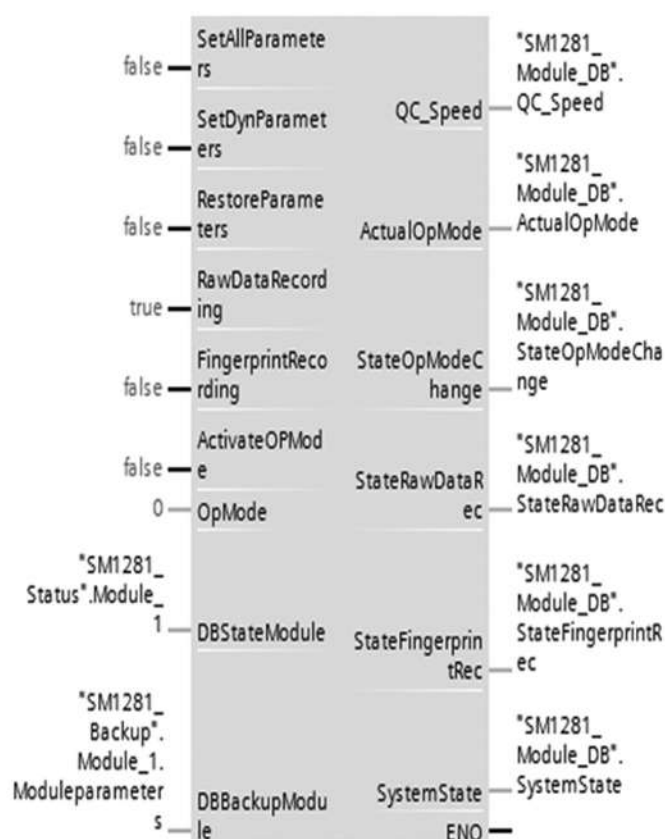


Рис. 2 - Налаштування блоку «SM1281 Module» (продовження)

Після правильного встановлення усіх налаштувань блоку «SM1281 Module», у проект додається функціональний блок «SM1281_Channel». Налаштування блоку «SM1281_Channel» зображені на рис. 3.

Для присвоєння функціональному блоку номера каналу вимірювання (від 1 до 4), використовується параметр ChannelNR, за допомогою параметра Enable встановлюється дозвіл на включення і відключення каналу. Вихідними параметрами блоку «SM1281_Channel» є значення віброшвидкості (vRMS), а також середньоквадратичне значення прискорення вібрації (aRMS) в м/с².

Після налаштувань виконується компіляція проекту та завантаження його у контролер [3]. Для доступу до веб-сервера модуля SM1281, у функціональному блоці «SM1281_Module» слід встановити значення параметра OperatorControlS7 - false (рис. 4).

Після завантаження проекту в контролер, відбувається встановлення зв'язку з веб-сервером контролера S7-1200 і з веб-сервером модуля SM1281.

ВИСНОВОК

Запонована процедура параметрування програмних блоків технологічного модуля для вимірювання параметрів вібрації SM1281 дозволяє оперативного включити в роботу систему діагностування ГПА з метою подальшої обробки отриманої віброакустичної інформації.

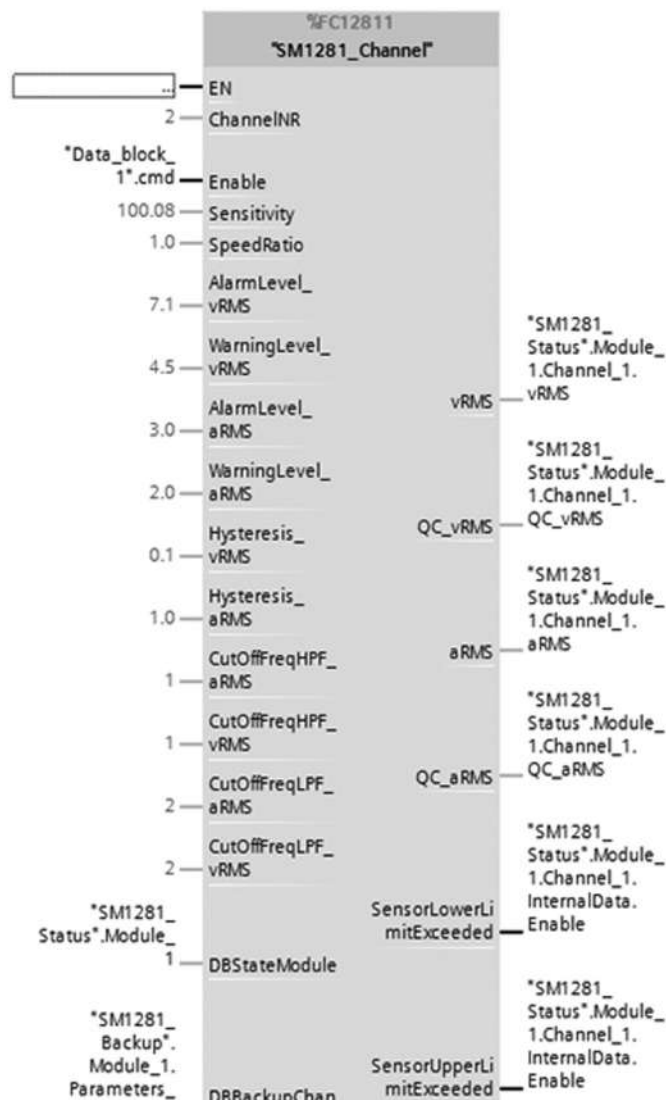


Рис. 3 - Налаштування блоку «SM1281_Channel»



Рис. 4 – Встановлення дозволу для доступу до веб-сервера модуля SM1281

ПЕРЕЛІК ПОСИЛАНЬ

1. Заміховський Л.М. Система діагностування лопатевого апарату газоперекачувальних агрегатів [Текст] / Л.М. Заміховський, Н.І. Іванюк// Інтеллект. продукт вечних і винахід. Прикарпаття – 2016: Щорічний каталог найвагом. винах., корисн. моделей, пром. зразків і рац. пропозицій Галицьких кмітливців. – Івано-Франківськ, 2016.– С. 116-120.
2. Condition Monitoring Systems. Operating Instructions (A5E36912951-AB) - Siemens AG, 06/2016. - 186 p.
3. Step 7 Professional V14 – Siemens AG, 2016 – 14418 p.
4. Visualizing trends and raw data in X-Tools. SIPLUS CMS1200 SM 1281 - Siemens AG, 10/2016. - 11 p.

Використання дискримінантного аналізу для вибору діагностичної ознаки технічного стану газоперекачувальних агрегатів

Заміховський Леонід Михайлович

завідувач кафедри «Інформаційно-телекомунікаційних технологій та систем»

Івано-Франківський національний технічний університет нафти і газу
Україна, Івано-Франківськ

Павлик Володимир Васильович

начальник служби КВП і А Богородчанського ЛВУМГ
УМГ «Прикарпаттрансгаз»
Україна, Богородчани

Запропонована процедура застосування дискримінантного аналізу для визначення технічного стану газоперекачувального агрегату (ГПА) з використанням результатів спостережень за зміною його технологічних параметрів в процесі експлуатації для різних технічних станів ГПА.

Показано, що за вибраною діагностичною ознакою можна не лише оцінити поточний технічний стан ГПА, але й прогнозувати його зміну наперед заданий період експлуатації.

Ключові слова: дискримінантний аналіз, ідентифікатор, кореляція, діагностична ознака, технологічні параметри, технічний стан.

Забезпечення ефективної роботи газотранспортної системи України залежить, в першу чергу, від надійної роботи газоперекачувальних агрегатів (ГПА), які є її основними складовими. Надійна робота ГПА обумовлюється його поточним технічним станом, який потрібно контролювати в режимі реального часу, що вимагає використання ефективних методів діагностування ГПА.

Сьогодні відома значна кількість методів параметричної і віброакустичної діагностики ГПА, а також методів діагностування, створених на базі сучасних інформаційних технологій – штучних нейронних мереж, генетичних алгоритмів і ін. В той же час відсутні методи діагностування, які б нашли широке практичне застосування.

В роботі розглядаються особливості використання дискримінантного (лінійного) аналізу для визначення діагностичної ознаки технічного стану з використанням результатів спостережень за зміною його технологічних параметрів в процесі експлуатації для різних технічних станів ГПА.

Для аналізу використовувалася інформація по ГПА-3 ГТК 25і фірми «Нуово Пінйоне», який знаходиться в експлуатації на КС-Богородчани Богородчанського ЛВУМГ УМГ «Прикарпаттрансгаз». Зокрема, аналізувалися вибірки технологічних параметрів роботи ГПА перед виводом його в ремонт, після ремонту, а також після тривалого періоду експлуатації, що фактично відповідало трьом різним технічним станам ГПА.

В роботі обґрунтовується вибір 8 з 16 технологічних параметрів роботи ГПА, які в подальшому підлягали дискримінантному аналізу.

Дискримінантний аналіз це один з методів багатовимірного статистичного аналізу, який дозволяє на основі спостережень над об'єктами з відомих кількох груп знайти правило, за яким новий об'єкт слід відносити до тієї чи іншої групи. При цьому мінімізується ймовірність неправильної класифікації. В лінійному дискримінантному ана-

лізі використовуються лінійні (дискримінантні) функції багатьох змінних (лінійна комбінація характеристик об'єкта плюс деяке число). Для кожної групи будується своя функція з умови найкращого розрізнення об'єктів різних груп. Новий об'єкт класифікується до тієї групи, в якій він має найбільше значення відповідної дискримінантної функції.

Методи дискримінантного аналізу реалізовано в багатьох програмних продуктах. В роботі використовувались можливості програмного середовища R [1] та системи STATISTICA [2].

Розглядається процедура дискримінантного аналізу часових реалізацій технологічних параметрів (даних) записаних через кожні 30 с. для трьох технічних станів ГПА: до ремонту (ідентифікатор “b”=“befor”), після ремонту (ідентифікатор “a”=“after”) та після напрацювання (ідентифікатор “w”=“work”).

Щоб позбутися кореляції між сусідніми даними здійснювали їх випадковий відбір, застосовуючи можливості стандартних засобів R [1]. Було відібрано по 340 даних з кожного набору відповідно розглянутих ситуацій. Аналіз отриманих графіків автокореляційних функцій показав, що часові ряди даних містять лише білий шум, що вказує на відсутність кореляції між ними. Це дозволяє розглядати відібрані дані як звичайні багатовимірні вибірки.

Основою для дослідження діагностичної ознаки стану ГПА була об'єднана вибірка з побудованих трьох вибірок (“b”-“a”-“w”).

Для вибору діагностичної ознаки технічного стану ГПА та її перевірки було проведено поділ вибірки на дві частини (парні та непарні номери). За першою підвибіркою з допомогою дискримінантного аналізу будуємо процедуру визначення стану, а елементи другої підвибірки перевіряємо на належність до наявних станів (“b”-“a”-“w”). Для цього використовуємо покроковий дискримінантний

аналіз реалізований в STATISTICA [2].

Отриманими результатами є (табл. 1):

- набір характеристик (Variable), за якими можна найбільш достовірно визначити стан ГПА;
 - коефіцієнти дискримінантних функцій (Classification Functions) для кожного із станів
- Наступна таблиця містить згадані результати.

Таблиця 1. Результати дискримінантного аналізу

Variable	Classification Functions; grouping: group (R_sample.sta) Include condition: Ceiling(v0/2)-v0/2=0		
	a p=,33333	b p=,33333	w p=,33333
Тиск на виході	210	534	471
Рівень масла в баці	354	382	395
Оберти ТВТ	32	29	29
Температура на вихлопі	-4	-6	-4
Тиск осового копресора	270	255	265
Оберти ТНТ	-3	-4	-4
Антипомпажний параметр	118	117	119
Тиск гідравлічного масла	158	158	160
Constant	-169874	-180456	-184631

Далі розглядався варіант, коли жодної попередньої інформації про стан ГПА немає. Це означає, що апріорні ймовірності перебування ГПА в станах “a”, “b” чи “w” однакові і дорівнюють одній третій (p=,33333).

Методика визначення діагностичної ознаки стану ГПА (одержана в результаті дискримінантного аналізу) наступна:

1. Визначаємо значення дискримінантних функцій досліджуваного ГПА для кожного стану. Це лінійні комбінації значень характеристик (із стовпчика Variable) з коефіцієнтами, наведеними в стовпчиках “a”, “b” чи “w” плюс вільний член з рядка Constant.

2. Стан ГПА визначається за найбільшим значенням дискримінантних функцій для нього.

Застосовуючи описану методику до елементів другої частини вивірки, одержали результати відображені в табл. 2

Таблиця 2. Результати визначення діагностичної ознаки стану ГПА

Group	ClassificationMatrix (R_sample.sta) Rows: Observedclassifications Columns: Predictedclassifications Excludecondition: Ceiling(v0/2)-v0/2=0			
	Percent Correct	a p=,33333	b p=,33333	w p=,33333
a	100,0000	170	0	0
b	100,0000	0	170	0
w	100,0000	0	0	170
Total	100,0000	170	170	170

Рядки “a”, “b”, “w” табл. 2 відображають спостережувані (реальні) стани ГПА, а такі ж стовпчики – прогнозовані

(одержані за пропонованою методикою). На перетині цих рядків та стовпчиків наведені кількості випадків.

Засоби R [1] дозволяють спроектувати три групи даних – діагностичних ознак у вигляді «хмарок» на площину і відобразити їх на графіку. Найкращий напрям проектування вибирається з допомогою того ж дискримінантного аналізу.

На рис. 1 зображена така проекція діагностичних ознак. Кожне значення діагностичної ознаки позначено буквою (“a”, “b”, “w”), що відповідає реальному стану ГПА. LD1 та LD2 – означають координати на площині проекції.

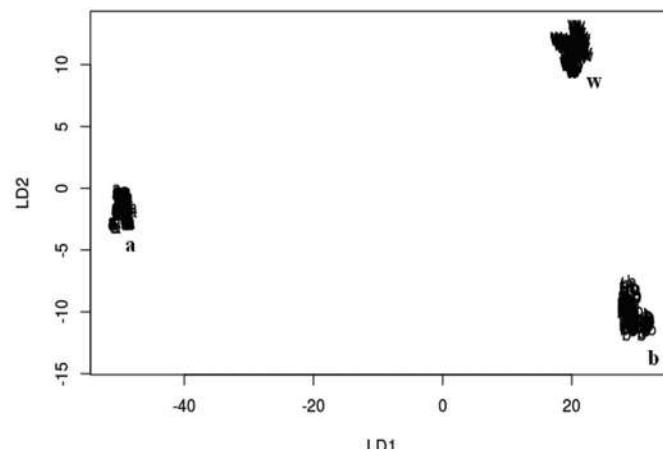


Рис. 1 – Проекція значень діагностичних ознак на площину станів ГПА

Як видно з рис. 1 всі три значення діагностичних ознак чітко розрізняють три стани ГПА.

Найкращий стан ГПА (a) після проведення ремонтних робіт знаходиться на початку площини -найменше значення діагностичної ознаки, незадовільний стан (перед проведенням ремонтних робіт) (b) відповідає найбільшому значенню діагностичної ознаки і, нарешті, після тривалого напрацювання стан ГПА наближається до незадовільного (w).

ВИСНОВОК.

Запонована процедура вибору діагностичної ознаки стану ГПА з використанням дискримінантного аналізу його поточних технологічних параметрів дозволяє чітко розпізнавати технічний стан ГПА, а маючи значення діагностичної ознаки, що відповідає його незадовільному (критичному) стану, прогнозувати його зміну наперед заданий період часу.

ПЕРЕЛІК ПОСИЛАНЬ

1. R Core Team (2016). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL <https://www.R-project.org/>.
2. StatSoft, Inc. (2011). STATISTICA (data analysis software system), version 10. www.statsoft.com.

Автентифікація RFID мітки на основі односторонніх перетворень

Погоржельський Валерій В'ячеславович
студент
КПІ ім. Ігоря Сікорського
Україна, Київ

Полторак Вадим Петрович
к.т.н., доцент
КПІ ім. Ігоря Сікорського
Україна, Київ

Розглядається підхід до побудови системи автентифікації для пристроїв з обмеженими ресурсами і пам'яттю, таких як RFID мітки, брелки, смарт-карти і т. п. Стійкість протоколу заснована на стійкості використаних крипто примітивів, а також теоретичній можливості стійкого об'єднання логіки при проектуванні схем пристрою зчитувача.

Ключові слова – мітка, RFID, симетричні алгоритми, системи автентифікації.

ВСТУП

Відомим стандартом і досить популярним способом реалізації операцій ідентифікації являються так звані паролі системи [1, 2], в яких кожному учаснику системи відповідає пара (ім'я, пароль). Для того щоб здійснити автентифікацію в таких системах користувач повинен назватись (повідомити своє ім'я) та назвати правильний пароль, який відповідає даному імені. Значним недоліком даного підходу є те, що термінал повинен знати про користувачів всю секретну інформацію. Таким чином, потрібно побудувати таку систему, в якій термінал здатний перевірити наявність у користувача, який видає себе за U , секрету s_u , не знаючи цього секрету.

У одній з робіт [3] про ідентифікацію користувачів в RFID системах було запропоновано ідею використання в якості відкритого ключа P_U користувача U безпосередньо дані про ідентичність цього користувача (або значення, яке можна обчислити виходячи з цих даних). Крім того, що такий підхід є досить зручним, головною перевагою його є те що зберігається унікальність відкритого ключа. Такий метод отримав назву «заснованого на ідентичності» (ID-based). Варто відзначити, що даний тип криптографії особливо зручний в задачі автентифікації, оскільки в будь-якому протоколі автентифікації користувач спочатку повинен повідомити термінал «за кого він себе видає»; в ID-based це дозволяє одночасно передати відкритий ключ.

У даній роботі пропонується система автентифікації, яка може застосовуватись в RFID системах, в яких є обмежені обчислювальні ресурси.

Схема автентифікації основана на ізолюванні запиту

Проаналізуємо підхід до організації автентифікації між RFID-міткою та зчитувачем, заснований на так званій ізоляції запиту.

Для початку розглянемо мітку, яка належить користувачу U . Далі не будемо розрізняти користувача і його мітку і будемо називати їх «мітка U ». Припустимо що ми розглядаємо ID-based криптографію і відкритий ключ користувача обчислюється із даних про ідентичність даного користувача. У зв'язку з цим будемо вважати, що $P_U = U$.

Секретний ключ s_U обчислюється центром автентифікації за допомогою секретного перетворення E . Фактично

це означає, що при обчисленні E використовується секретний аргумент. Будемо вважати, що секретний центр A обчислює секретний ключ користувача U , як $S_U = E_{s_A}(U)$, де s_A – секретний ключ центра A . Структура мітки зображена на рисунку 1.

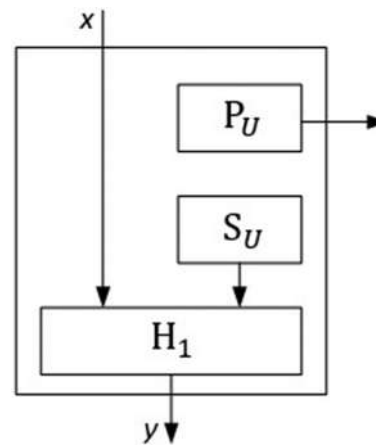


Рис. 1 – Структура мітки U

Тут H_1 – (несекретне) одностороннє перетворення. Мітка здатна повертати свій відкритий ключ P_U , а також обчислювати функцію S_U для деякого вхідного значення x .

Функціональна структура зчитувача зображена на рисунку 2. Однією із основних умов запропонованої системи являється те, що пристрій, який реалізує дану функціональність і надається центром автентифікації всім зчитувачам, становить для них «чорний ящик». Вона приймає на вхід два параметра x_1 та x_2 і обчислює y_1 та y_2 , але не має змоги бути (за допустимий час) розкладеним на складові H_1 , H_2 та E_{s_A} , або значення еквівалентні їм. На практиці це може бути реалізованим на рівні інтегральної мікросхеми. Таким чином, на рисунку 2 показано саме функціональність блоку на стороні терміналу, а не дійсна архітектура.

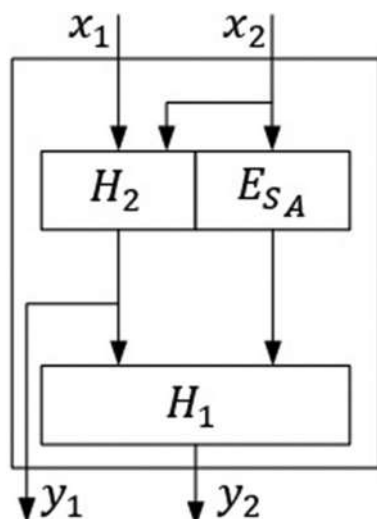


Рисунок 2 – Структура зчитувача

Блоки H_1 і H_2 – відкрите, але одностороннє перетворення, а E_{S_A} – секретне перетворення центра автентифікації (див. рис. 2). Внаслідок передбачуваної неможливості реверс інженерингу перетворення E_{S_A} , а також всі проміжні значення «всередині» зображеного блоку залишаються недоступними ззовні. При цьому сам пристрій, як певний пристрій, який має два входи та два виходи, може розглядатись як загальнодоступний і слугує аналогом відкритого ключа центру автентифікації А.

Тепер розглянемо здійснення процесу автентифікації між зчитувачем та міткою.

1. На першому кроці протоколу карта називає себе і таким чином передає терміналу свій відкритий ключ $U = P_U$.

2. Зчитувач подає на вхід свого пристрою $x_1 = R$ і $x_2 = P_U$, де R – випадкове число, отримуючи на виході y_1 та y_2 . Значення y_1 передається мітці.

3. Мітка подає на вхід свого пристрою отримане значення $x = y_1$ та обчислює відповідний y , котрий передає зчитувачу.

4. Зчитувач перевіряє $y_2 = y$ і у випадку рівності вважає ідентифікацію успішною. В іншому випадку автентифікація мітки відміняється.

Розглянемо описаний протокол [1, 2] більш детально. Зчитувач задає питання (випадкове число), мітка в свою чергу повинна правильно на нього відповісти, використовуючи свій секретний ключ.

У даному випадку за критерій взято спільне значення мітки і зчитувача. Проте в якості оклику використовується не саме випадкове число, а його образ, обчислювальний за допомогою односторонньої функції H_2 . Далі приведено значення, яке обраховується міткою.

$$y = H_1(y_2, S_U),$$

а значення, обраховане зчитувачем:

$$y_2 = H_1(H_2(R), E_{S_A}(U)) = H_1(y_1, S_U) = y$$

Таким чином, при правильному функціонуванні та зберігаючи в мітці значення S_U , повинно виконуватись $y_2 = y$ що й перевіряє зчитувач.

Варто відзначити, що основними елементами схеми мітки і зчитувача на рис. 1 та 2 є секретна однонаправлена

функція E_{S_A} та дві відкриті однонаправлені функції H_1 і H_2 (при цьому немає необхідності спеціально публікувати H_1 та H_2). Також потрібно відмітити, що вимоги однонаправленості функцій H_1 та H_2 задовольняються використанням (безключових) криптографічних хеш-функцій, а вимоги секретної односторонньої функції можуть бути реалізовані за допомогою симетричного блочного шифру, або ключових хеш-функцій (наприклад коду автентифікації повідомлення MAC) [1, 2].

АНАЛІЗ ПРОТОКОЛУ

Розглянемо властивості отриманого протоколу.

Автентифікація користувача. По побудові даний протокол являється основаним на автентифікації користувача, що є бажаною властивістю, сформульованою вище.

Швидка і компактна реалізація. Протокол використовує криптопримітиви, які легко реалізуються: хеш-функції та симетричний блочний шифр, який більш раціональніше використовує ресурси у порівнянні з теоретико-числовими протоколами автентифікації. Для багатьох практичних систем цей вигравш може досягати тисячі одиниць.

Вимоги до трафіку. Зчитувач передає мітці значення y_1 являється результатом H_2 . Мітка передає зчитувачу ідентифікатор U і значення y , яке являється виходом H_1 . Якщо допустити, що в якості однонаправленої хеш-функції використовується ГОСТ Р 34.11-94 з 256-бітними виходами, то зчитувач передає мітці 256 біт і мітка передає 256 біт та додатково розмір U . Якщо рахувати, що розмір U не перевищує 256-512 біт, отримаємо загальний трафік не більше 1024 біти. Це відповідає лише середньому розміру одного числа в теоретико-числовому протоколі автентифікації [1, 4], при цьому в протоколі кількість подібних переданих чисел оцінюється приблизно як $2t$, де t – кількість раундів, зазвичай обирається декількох порядків.

Універсальність мітки. Представлена на рис. 1 мітка являється універсальною у тому сенсі, що при зміні користувача, або навіть алгоритму перетворення E центру автентифікації немає необхідності проектувати нову мітку – достатньо перезаписати значення, які в ній зберігаються U та S_U (звичайно, апаратне забезпечення зчитувача при цьому все ж таки потрібно буде проектувати спочатку).

Стійкість. Розглянемо необхідність використання одностороннього перетворення H_2 . У випадку його відсутності, можна описати протокол наступним чином.

1. Мітка називає себе та передає зчитувачу свій відкритий ключ $U = P_U$.

2. Зчитувач відповідає мітці випадковим числом R , одночасно обчислюючи при цьому $y_2 = H_1(R, E_{S_A}(U))$.

3. Мітка подає на вхід свого пристрою отримане значення і обчислює відповідний y , який передає зчитувачу.

4. Зчитувач перевіряє $y_2 = y$ та у випадку рівності вважає автентифікацію успішною. В іншому разі автентифікація мітки відміняється.

Не важко побачити, що даний протокол буде працювати тільки з точки зору «легальних» користувачів, у тому сенсі, що якщо всі учасники чесні, то виконується умова $y = y_2$.

Проте згадаємо, що ми вважали пристрій зчитувача доступним усім учасникам, як своєрідний відкритий ключ центру автентифікації A . У цьому випадку ніщо не заважає зловмиснику, який не володіє міткою U , але який бажає видати себе за U , скористатись точно таким же пристроєм, що є в зчитувача, подати на його вхід відомі йому R і U та обчислити f . Для запобігання такого типу атак, випадкова відповідь зчитувача забороняється за допомогою односторонньої функції.

Стійкість даної системи залежить від стійкості функцій хешування (необхідна стійкість до обчислення другого прообразу [3]), а також шифрування.

ВИСНОВОК

У роботі розглянута схема автентифікації, яка використовує в якості базових блоків ефективні в реалізації і не вимогливі до ресурсів симетричні алгоритми. Даний тип алгоритмів використовується в пристроях з обмеженими ресурсами, в яких важко використовувати традиційну

теоретико-числову криптографію з відкритим ключем. Запропоноване рішення дозволяє пришвидшити роботу мітки та дає змогу спростити алгоритм автентифікації, що в подальшому зменшує вартість самої мітки.

ПЕРЕЛІК ПОСИЛАНЬ

1. Шнайер Б. Прикладная криптография. Протоколы, алгоритмы, исходные тексты на языке Си. — М.: Триумф, 2002. — 816 с.
2. Menezes A. J., van Oorschot P. C., Vanstone S. A. Handbook of Applied Cryptography. — CRC Press, 1996. — 780 p.
3. Shamir A. Identity-Based Cryptosystems and Signature Schemes. Advances in Cryptology // Advances in Cryptology — CRYPTO 84: Lecture Notes in Computer Science. 1984. Vol. 196. N 7. P. 47–53.
4. Fiat A., Shamir A. How to Prove Yourself: Practical Solutions to Identification and Signature Problems // Advances in Cryptology — CRYPTO 86: Lecture Notes in Computer Science. 1987. Vol. 263. P. 186–194.

Identification of the issues in the E-governance in Jordan and USA

Mohammad Alhawawsha
Taras Shevchnko National University of Kyiv
Ukraine, Kyiv

The application of the “electronic government” is one of the most trending and simplest forms of ICT (“Information and Communication Technology”) application used for governmental services. In spite of the significant advantage and benefits obtained from the e-government, various nations have failed to fully utilize the advantages and successfully implement the system. This paper evaluated the critical factors that impact and determines the failure and success of the e-government implementation project. The article focuses on the investigation of the e-government implementation case in Jordan. The various technical challenges and issues have been identified that has finally lead to the inefficient development and application of the e-government system. The evaluation of the strategies used by the Jordan e-government project team has helped in identification of the technical issues like privacy, security, integrity, reliability, interoperability, cloud management, and content management system. Based on the identified issues, an enhanced model for e-government has been proposed with data authentication, encryption and many other techniques for overcoming the technical issues.

1. INTRODUCTION

In the field of Information technology, the development of the e-government is considered as a new area concerned with the utilization if the ICT for delivering the governmental services to the citizens. The e-government system comprises various technologies including mobile computing, the internet and wide area network for providing real-time services and solution to the citizens. Bakunzibake, Grönlund, and Klein, (2016) showed that the United Nations described e-government as the application and utilization of the “Information and Communication Technology” for provisioning the public services and information to the public. The implementation of the e-government helps in including various advanced concepts like citizen participation, accountability, and transparency in evaluating the performance of the e-government.

According to Solinthone, and Rummyantseva, (2016), in the recent days, Jordon is recognized as one of the developing countries that took the initiatives of including the Governmental payment and transactions through an electronic medium. Weerakkody et al., (2015) illustrated that the development of the e-government has failed to live up to the expectation and potential. This paper evaluates the issues with the implementation of the Jordan e-government system and provides relevant solutions to them.

2. IDENTIFICATION OF THE ISSUES

IN THE E-GOVERNANCE IN JORDAN AND USA

2.1. E-government strategies used by Jordan

The development of the e-government strategy aims at linking the governmental procedure of Jordan with national priorities. The development and implementation of the e-government in Jordan have been developed for dedicating the governmental services to the citizen across the location and society. The e-government strategy helps in crystallizing the projects, governmental services to the citizens while raising and controlling the performance and operations through the e-government services. The developed e-government system at Jordon aimed at increasing the effort of government initia-

tive within the time span of three years between 2014 to 2016

2.2. Issues

The implementation and development of the e-government platforms require large capital, technical infrastructure for training, hardware implementation and software. Choi et al., (2016) showed that in the developing countries the implementation of the e-government needs to be ensured with a financial option in the long term. The Jordanian e-government system was based on cloud technology for providing complete associability and security to the information shared through the system.

Privacy and Security: Abdulkareem, (2015) cited that ensuring privacy and security in the cloud platform is one of the crucial challenged during the development of the e-government system. Data integrity, data availability and data confidentiality are the major criteria for privacy and security. Joseph, (2015) illustrated enabling and ensuring the security in the cloud government helps in establishing trust and reliability of the citizen for using the system. Therefore, the implementation of the e-government system is often considered as a development of cloud application for ensuring the security requirements of the transaction and processes involved in the e-government. The e-government network often suffered from the various threats and virus attacks that volatile the requirements of security within the e-government system. One of the significant and most common attacks on the e-governments system is DoS (Denial of Service). Bwalya, Du Plessis, and Rensleigh, (2014) showed other significant attacks in the e-government system are U2L, U2R, and probe. In addition to that, the data and information relevant to the citizen of Jordan are controlled and stored by the third party rather than the government itself.

Integrity: According to Wirtz et al., (2015), maintaining the consistency and accuracy of the data and information stored and communicated within the system forms significant issues in maintaining the integrity of data. Chatfield, and AlAnazi, (2015) illustrated that integrity is a vital criterion for data storage and databases. The data integrity requires implementation

of significant processes for ensuring the data reliability and accuracy throughout the lifetime of the data.

Reliability: Rahman et al., (2014) cited that maintaining the reliability of the data in the cloud based E-government system is forms a major issue during development. Ensuring the operational infrastructure is essential for ensuring the data accessibility in unexpected and catastrophic situations.

Interoperability issues: Various services providers collaborate together for ensuring the proper services being provided to the customer. In the e-government system, the governed needs to include the banking services, and various public services procedure for ensuring proper functionality and operations to the citizen.

Managing Cloud: The developed e-government system needs to provide seamless availability and accessibility of the services and information to the citizen. The functions and operations used in the e-government need to be regularly changed or updated for the change in the requirement of the citizen and the government itself. The updating of the entire process and functions within the system without the involvement of the government by the third party is a risky task while ensuring no data loss from the system.

Content Management System: The e-government system includes various operations and functional modules for providing all the required governmental procedure and functionalities to the citizen. The inappropriate deployment of the “content management system” for maintaining the different modules within the e-government system. Snead, and Wright, (2014) have shown that not all the citizens are experts in e-payments. Therefore, the designer needs to ensure the applicator of appropriate style and design for enhancing the visually and constant structure throughout all the pages in the system.

2.3. Solutions

An enhanced model has been proposed in this assignment for reducing and eliminating the potential issues identified in the previous section. The proposed model has been developed with “system feedback process” that ensured the proper functionality and operations conducted through the developed e-government system. The proposed model of e-government while eliminating the technical difficulties observed in the Jordan E-government project has been proposed in the section below:

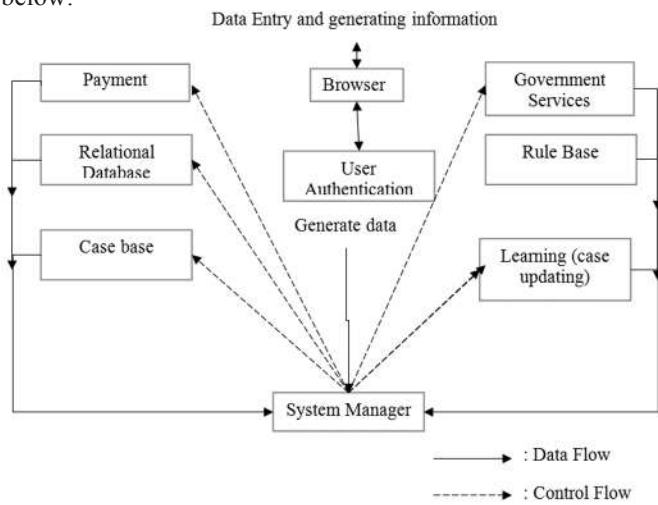


Figure 1: Proposed e-government model (Source: Created by Author)

The database of the proposed system needs to be robust for storing the argument details, profile data, and banking details. The database will be developed over the cloud system for ensuring the accurate and faster queries of the system. The proposed system will run on SQL Server 2008 for retrieving and storing the sensitive information. The developed e-government includes particular comparison algorithm for efficient searching of the elements and services through the entire e-government system.

The time complexity of the proposed model will be the following:

$$\begin{aligned}
 T(n) &= n + n + n(n+1) + n(n+1) + n(n+1) \\
 &= 2n + n^2 + n + n^2 + n + n^2 + n \\
 &= 5n + 3n^2
 \end{aligned}$$

Therefore the complexity of the proposed model is $O(n^2)$.

According to Zhang, Xu, and Xiao, (2014), the application of the appropriate search algorithm in the e-governmental system helps in enhancing the performance of the developed system. In the proposed system, Lexical Method will be used as the required search algorithm. The matching system of the lexical similarity has been defined in WordNet and performed in synset (Ziemba, & Papaj, 2013). The e-government system will include a lexical dictionary for all the available governmental services provided to the citizen. The searching of a particular word in the system will begin with comparison with all the word available in the system. The pseudo code for the lexical similarity has been described as follows:

Variable: “A” word

Input: search text

Output: list array of words

Read input text

A ← gets word from text

Intarray[] as array

For each A to end of text

If “read character” = “NULL”, “which is space” and special character “etc > ? / *”

Then

Consider as end of word

Else continue

End

In addition to that, the SO values of the words compared to the system are derived and calculated using the PMI (Point wise Mutual Information) methods (Elbahnasawy, 2014). The PMI calculated for the proposed model can be described as:

$$PMI(word1, word2) = \log_2 \left(\frac{p(word1 \text{ and } word2)}{p(word1)p(word2)} \right)$$

The application of the above stated methods will be utilized for comparing the two words. According to proposed formula, the probability of the getting the positive results of comparing the two words is true if the value provides value if base-2 log (Wilson, 2014). The application of the appropriate comparison algorithm helps in ensuring the reliability of searches and data applicability in the e-government system.

In addition to that, in the proposed system, Three-tier techniques will be used for developing the page building system in the e-government system. The application of the Three-tier techniques helps in strengthen and enhancing the security of the services and database used in the developed e-government model (Chairunisa, Malik, & Rahman, 2016). The three tier techniques sub-divided the entire system into three major level or tier namely, intermediate tier, application tier and database tier. The database tier in the system has been used for storing and extracting the raw data from the database to the developed system. Further, the application tier enhanced the interface between the user and the client (Khamallag, Kamala, & Tassabehji, 2016). Therefore, during modifications or updating of the systems or application, the other module that is application and the database will not be affected by the other.

For ensuring interaction between the citizen and e-government system, the service providers needs to implement the “Hypertext Transfer Protocol” (HTTP) for ensuring secure communication and data exchange through the “Secure Socket Layer protocol” (SSL) over the internet (Dwivedi et al., 2015). The network of the e-government system has been coded with “Extensible Markup Language” (XML) for executing the SQL statement and query.

The application of the SQL injection helps in increasing the application confidence and prevents the e-government from security attack (Venkatesh, Sykes, & Venkatraman, 2014). In addition to that, the system will include the data encryption procedure for preventing the unauthorized users for reading sensitive information over the e-government system. The e-government system has its inbuilt key generation algorithm for generating the public key and symmetric key for securing the access control over the e-government.

In the section below, the posed ULM class diagram for the posed enhances e-government system has been provided:

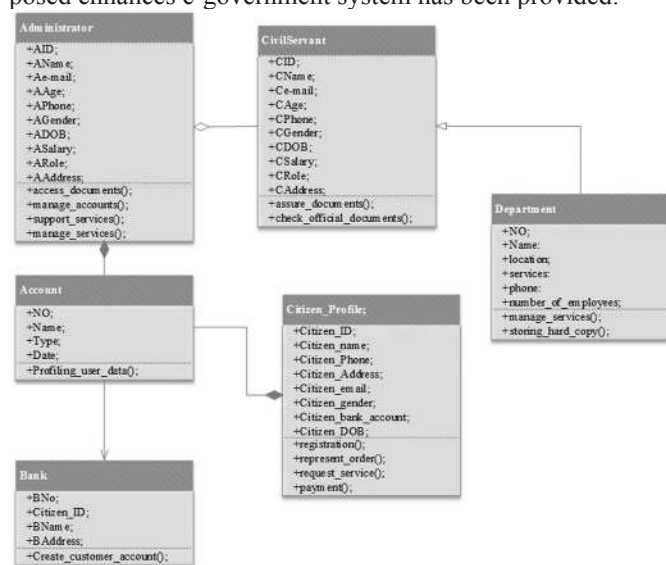


Figure 2: Proposed ULM Class Diagram for enhanced e-Government System (Source: Created by Author)

3. CONCLUSION

The application and integration of the e-government services help in mobilizing the service and official procedure. The effective application of the e-government service allows

the country to increase the efficiency of the governmental procedure. Based on the technical issues identified, the proposed system has been developed for enhancing the security concern and reliability of data. The proposed model of the e-government system has allowed in maintaining the security and reliability of data through the connected database and application developed. The application of security measures like SQL injection, three-tier models, and encryption models helped in ensuring the security and protecting the system from security threats. Furthermore, the developed model also helped in ensuring the data wastage and reducing the cost and time for implementation.

REFERENCE

1. Abdulkareem, A. K. (2015). Challenges of E-Government Implementation in the Nigerian Public Service. *Journal of Creative Writing (ISSN 2410-6259)*, 1(4), 45-56.
2. Bakunzibake, P., Grönlund, Å., & Klein, G. O. (2016). E-Government Implementation in Developing Countries: Enterprise Content Management in Rwanda. In *15th IFIP Electronic Government (EGOV)/8th Electronic Participation (ePart) Conference, Univ Minho, Guimaraes, Portugal, September 5-8, 2016* (pp. 251-259). IOS Press.
3. Bwalya, K. J., Du Plessis, T., & Rensleigh, C. (2014). E-government implementation in Zambia—prospects. *Transforming Government: People, Process, and Policy*, 8(1), 101-130.
4. Chairunisa, F., Malik, M. T., & Rahman, M. A. (2016). Contextual-Participative e-Government Implementation Model: A Case Study of the Office of Demographic and Citizen Data Records of Makassar. *Mediterranean Journal of Social Sciences*, 7(1 S1), 345.
5. Chatfield, A. T., & AlAnazi, J. (2015). Collaborative governance matters to e-government interoperability: An analysis of citizen-centric integrated interoperable e-government implementation in Saudi Arabia. *International Journal of Public Administration in the Digital Age (IJPADA)*, 2(3), 24-44.
6. Choi, H., Park, M. J., Rho, J. J., & Zo, H. (2016). Rethinking the assessment of e-government implementation in developing countries from the perspective of the design–reality gap: Applications in the Indonesian e-procurement system. *Telecommunications Policy*, 40(7), 644-660.
7. Dwivedi, Y. K., Wastell, D., Laumer, S., Henriksen, H. Z., Myers, M. D., Bunker, D., ... & Srivastava, S. C. (2015). Research on information systems failures and successes: Status update and future directions. *Information Systems Frontiers*, 17(1), 143-157.
8. Elbahnasawy, N. G. (2014). E-government, internet adoption, and corruption: an empirical investigation. *World Development*, 57, 114-126.
9. Irani, Z. (2014). E-government Implementation Factors: A Conceptual Framework. *Journal of Modern Accounting and Auditing*, 10(2), 241.
10. Joseph, S. R. (2015). Success factors influencing e-government implementation. *International Journal of Marketing and Technology*, 5(1), 127.

11. Khamallag, M. M., Kamala, M. A., & Tassabehji, R. (2016). Digital Government Implementation in Chaotic Environment-Case Study of Libya.
12. Rahman, S., Rashid, N., Yadlapalli, A., & Li (Eva) Yiqun. (2014). Determining Factors of E-Government Implementation: a Multi-criteria Decision-Making Approach. In *PACIS* (p. 302).
13. Snead, J. T., & Wright, E. (2014). E-government research in the United States. *Government Information Quarterly*, 31(1), 129-136.
14. Solinthone, P., & Romyantseva, T. (2016). E-Government Implementation. In *MATEC Web of Conferences* (Vol. 79, p. 01066). EDP Sciences.
15. Venkatesh, V., Sykes, T. A., & Venkatraman, S. (2014). Understanding e-Government portal use in rural India: role of demographic and personality characteristics. *Information Systems Journal*, 24(3), 249-269.
16. Weerakkody, V., Irani, Z., Lee, H., Osman, I., & Hindi, N. (2015). E-government implementation: A bird's eye view of issues relating to costs, opportunities, benefits and risks. *Information systems frontiers*, 17(4), 889-915.
17. Wilson, S. C. (2014). e-Government legislation: Implementation issues for programs for low-income people. *Government Information Quarterly*, 31(1), 42-49.
18. Wirtz, B. W., Weyerer, J. C., Thomas, M. J., & Möller, A. (2015). E-government implementation: Theoretical aspects and empirical evidence. *Public Organization Review*, 1-20.
19. Zhang, H., Xu, X., & Xiao, J. (2014). Diffusion of e-government: A literature review and directions for future directions. *Government Information Quarterly*, 31(4), 631-636.
20. Ziemba, E., & Papaj, T. (2013, June). A Pragmatic approach to the e-government maturity in Poland-implementation and usage of SEKAP. In *European Conference on e-Government* (p. 560). Academic Conferences International Limited.

Reviewer: associated prof., National University "Kyiv-Mohyla Academy",
Andrii Glybovets.

Інформаційне моделювання процесів перенесення в гетерогенних середовищах

Прокопович І. В.
д.т.н., доцент
Одеський національний
політехнічний університет
Україна, Одеса

Духаніна М. О.
ст. викладач
Одеський національний
політехнічний університет
Україна, Одеса

Добровольська В. В.
аспірант
Одеський національний
політехнічний університет
Україна, Одеса

Дадерко О.І.
аспірант
Одеський національний
політехнічний університет
Україна, Одеса

Олех Г.С.
аспірант
Одеський національний
політехнічний університет
Україна, Одеса

Кошулян С.В.
аспірант
Одеський національний
політехнічний університет
Україна, Одеса

Здійснено дослідження провідності у зернистому середовищі для процесу переносу теплоти. Запропоновано інформаційні моделі перенесення в гетерогенних середовищах у вигляді двох взаємопроникних полімерних сіток. Теплопровідність середовища визначали за інтегральною характеристикою структур першого та другого порядку його моделі.

Ключові слова: гетерогенні матеріали, провідність, інтегральна характеристика.

Значна частина сучасних машинобудівних деталей виробляється з гетерогенних матеріалів (системи, які складаються з малих однорідних областей (компонентів), розграничених поверхнями розподілу), деякі властивості яких (наприклад, характеристики проникнення) не можуть бути із відповідною точністю обчислені як інтегральна характеристика матеріалів, які входять до композиту в цілому [1 – 2]. Тому проблема пошуку більш точних моделей такого обчислення є вельми актуальною задачею.

Характерною особливістю зернистих систем є безперервний контакт між зернами, або, іншими словами, існування при будь-яких значеннях пористості нескінченного кластера. Зернисті матеріали можна розділити на два класи: монодисперсні, зернисті матеріали з близькими за розміром (ізомерними) частками і полідисперсні зернисті системи, розміри зерен яких відрізняються більш, ніж на порядок.

Дослідження процесу перенесення в зернистих матеріалах проводиться давно і запропоновано різні моделі і методи розрахунку коефіцієнта провідності. Найбільш повний опис структури таких матеріалів наведено в [3 – 4], де запропоновано дві схеми розрахунку теплопровідності зернистої системи.

У цих схемах зернисті структури розглядаються у вигляді двох взаємопроникних полімерних сіток: каркасу, утвореного хаотичною, але відносно щільною системою, зерна якої постійно контактують (структура першого порядку) і просторової мережі більш великих порожнин, що пронизують каркас (структура другого порядку).

Коефіцієнти переносу для гетерогенних систем визначають в два етапи: на першому вивчають розподілення поля в компонентах системи з урахуванням граничних умов, на другому – знаходять на основі суперпозицій полей компонентів середні поля по об'єму всієї системи. Для

i -ї області (компонента i) потік $j_i(r)$ пов'язаний з потенціалом $\varphi_i(r)$ рівнянням.

$$j_i(r) = -\Lambda_i \nabla \varphi_i(r), \quad (1)$$

де $r = xi + yi + zk$ – радіус-вектор точки в об'ємі V .

Середні по об'єму системи V фізичні поля визначаються залежностями

$$\langle j \rangle = \frac{1}{V} \int_V j_i dV; \quad \langle \nabla \varphi \rangle = \frac{1}{V} \int_V \nabla \varphi_i dV; \quad \langle j \rangle = -\Lambda \langle j \rangle. \quad (2)$$

Таким чином, ефективний коефіцієнт узагальної провідності є функцією провідності компонентів Λ_i і їх концентрацій m_i , тобто

$$\Lambda = f(\Lambda_i, m_i). \quad (3)$$

Визначення вигляду цієї функціональної залежності і являється задачею узагальної теорії провідності.

Рішення системи рівнянь (1) – (3) для гетерогенної системи пов'язано з великими математичними труднощами і можливе для найпростіших структур. Тому звертаються до моделювання структури гетерогенної системи, а також до різноманітних математичних наближень. Більш ніж на століття вивчення фізичних властивостей гетерогенних систем було запропоновано велику різноманітність моделей і рівнянь для визначення Λ . При цьому для отримання нового результату використовувались різні методи рішення, які повторюють попередні або отриманий результат не має переваг в порівнянні з попередніми роботами.

В роботі здійснено дослідження провідності у зернистому середовищі для процесу переносу теплоти. Для цього процес перенесення розділяли на два етапи: спочатку встановлювали ефективну теплопровідність каркасу λ_k , а потім – теплопровідність всієї системи λ . Відношення теплопровідності системи до теплопровідності каркасу для взаємопроникних полімерних компонентів визначали за формулою:

$$\lambda/\lambda_K = c_K^2 + (1 - c_K)^2 v_K + 2v_K c_K (1 - c_K) / (v_K c_K + 1 - c_K), \quad (4)$$

де $v_K = \lambda_{22} / \lambda_K$; $c_K = \Delta / L = c(m_{22})$ та визначається з графіку.

Тут λ_{22} – провідність крупних пор, пронизують каркас, а пористість структури другого порядку:

$$m_2 = \frac{m_2 - m_{2K}}{1 - m_{2K}}, \quad (5)$$

де m_2 – об'ємна концентрація пор в зернистій системі; m_{2K} – пористість каркаса.

При розрахунках теплопровідності всієї системи був використаний метод елементарної комірки; невідомою величиною в (4) є теплопровідність каркаса λ_K . Для її розрахунку зручно використовувати метод усередненого елемента.

Як відомо, в порах існує молекулярне і променисте перенесення тепла. Тому провідність крупних пор є їх сумою

$$\lambda_{22} = \lambda_{22M} + \lambda_{22L}, \quad (6)$$

величина якої залежить не тільки від фізичних властивостей газу, але і від геометричних і фізичних параметрів самої пори.

Молекулярна складова може бути обчислена за формулою, в якій невизначеним є розмір великих пустот δ_2 . У першому наближенні його можна прийняти рівним відстані між брусами у каркасі. Наближена оцінка δ_2 визначається як:

$$\delta_2 = 3d(c_K^{-1} - 1), \quad (7)$$

де d – середній діаметр зерна.

На основі формули (7) отримуємо молекулярну складову теплопровідності?

$$\lambda_{22M} = \lambda_f \left[1 + \frac{Bc_K}{3Hd(1 - c_K)} \right]^{-1}. \quad (8)$$

Для оцінки променистої складової теплопровідності λ_{22L} у великих наскрізних порах весь шар зернистого матеріалу розглядали як напівпрозоре ізотропне середовище. Невизначеним параметром тут є коефіцієнт ослаблення випромінювання β .

Для зернистих матеріалів $d > 0,1$ мм за рахунок поглинання і розсіювання випромінювання на поверхнях коефіцієнт ослаблення випромінювання β становить:

$$\beta = \frac{4}{3d} c_K^2 (1 - c_K) (2 - \epsilon) = \beta(m_2).$$

Приймаючи $(16/3)n_{i0}\sigma \approx 0,3$, отримуємо значення [1]:

$$\lambda_{22L} \approx 0,3 \left(\frac{T}{100} \right)^3 \frac{Yd}{c_K^2 (1 - c_K) (2 - \epsilon)}, \quad (9)$$

де $Y = f(\tau, \epsilon)$; $\tau = \beta l_{сш} = \tau(m_2)$.

В цій формулі $l_{сш}$ – товщина шару засипки.

Зокрема, для зернистих систем з пористістю $m_2 < 0,95$ $Y \approx 1$.

Друга схема розрахунку провідності зернистої системи запропонована М. А. Єремєєвим і цілком базується на формулах, отриманих з моделі усередненого елемента.

Розрахунки за обома схемами призводять до приблизно однакових чисельних результатів.

В ряді випадків формули можуть бути істотно спрощені. Так, наприклад, для вільної засипки з розміром зерен $d > 0,1$ мм при тиску газу-наповнювача понад $1,33 \cdot 10^4$ Па можна знехтувати впливом мікрошорсткостей і вважати зерна гладкими ($h_0 = 0$) і жорсткими ($y_1 = y_2 = 0$). Тоді вираз для провідності Λ приймає вигляд

$$\frac{\Lambda}{\Lambda_1} y_4^2 \approx \left[\frac{D}{y_3^2} + \frac{2v_f}{1 - v_f} \left(D - 1 - W \ln \frac{W - D}{W - 1} \right)^{-1} \right]^{-1} + v_{2сш} E. \quad (10)$$

При розрахунках може виявитися корисною емпірична залежність між середнім діаметром частинок і пористістю зернистої системи у стані вільної засипки.

ПЕРЕЛІК ПОСИЛАНЬ

1. Bhatia, Sangeeta; John V Frangioni; Robert M Hoffman; A John Iafrate; Kornelia Polyak (10 July 2012). "The challenges posed by cancer heterogeneity". *Nature Biotechnology*. 30: 604–610. doi:10.1038/nbt.2294.
2. Общий курс процессов и аппаратов химической технологии: Учебник: в 2 кн./ В. Г. Айнштейн, М. К. Захаров, Г. А. Носов и др.; Под ред. В. Г. Айнштейна. — М.: Логос; Высшая школа, 2003. — Кн. 2. — 872 с.
3. Дульнев Г.Н, Новиков В.В. Процессы переноса в неоднородных средах. Санкт-Петербург: Энергоатомиздат, 1991.
4. Олейник О.А., Иосифьян Г.А., Шамаев А.С. Математические задачи теории сильно неоднородных упругих сред МГУ, 1990 312 p. Russian djvu, 3155 KB 10.1 KB/p. 300dpi OCR

Інформаційні технології управління проектною логістикою за допомогою віртуальних моделей

Савельєва О.С.

д.т.н., проф.

Одеський національний політехнічний університет
Україна, Одеса

Становська І.І.

к.т.н., доц.

Одеський національний політехнічний університет
Україна, Одеса

Гур'єв І.М.

аспірант

Одеський національний
політехнічний університет
Україна, Одеса

Малахова Д.О.

аспірант

Одеський національний
політехнічний університет
Україна, Одеса

Саух І.А.

аспірант

Одеський національний
політехнічний університет
Україна, Одеса

Запропоновано дискретну адаптивну віртуальну просторову модель надзвичайної логістики, яка дозволить постійно під час виконання проекту, незалежно від того, відбуваються деякі ризикові події або ні, відтворювати поточний стан наявних в елементах та на складах ресурсів. Це забезпечить можливість здійснювати ефективне управління проектом.

Ключові слова: *проектні ризики, просторова модель, логістика.*

Управління проектом, наприклад, створення великих розповсюджених об'єктів, є складним стратегічним процесом, який підлягає ретельному структурному плануванню для забезпечення цілей проекту завдяки ефективному виконанню всіх передбачених проектом робіт в задані строки. Таке планування передбачає і очікування різноманітних подій, які в проектному управлінні називаються ризиковими.

План проекту в цілому передбачає деякі дії з компенсації прогнозованих ризикових подій, для чого створює запаси коштів, матеріалів, часу, виходячи з ймовірності настання кожної такої події та прогнозних витрат на таку компенсацію. Однак, проектна діяльність, як правило, розвивається не за первинним планом. Крім безлічі планових ризикових подій, під впливом турбулентного навколишнього середовища можуть відбуватися і абсолютно стохастичні незаплановані ризики, найбільш небезпечні саме своєю несподіваністю [1]. При цьому живучість проекту (ймовірність його виконання у визначені терміни, при заданих ресурсах і заданій якості) знижується, аж до повної зупинки його виконання.

У зв'язку з цим менеджмент проекту повинен реагувати на будь-які ризикові події швидко, найчастіше, в форс-мажорних обставинах, які не залишають достатньо часу для ретельного аналізу і прийняття ефективних управлінських рішень щодо усунення наслідків таких подій, особливо, незапланованих. Найважливішою складовою такого реагування є вірогідний перерозподіл ресурсів проекту (фінансів, енергії, матеріалів, виконавців, інформації) між його елементами – проектна логістика [2], яка для планових ризикових подій є плановою, а для незапланованих – надзвичайною.

Таким чином, реагування на ризики як планових так і надзвичайних етапів проектів містить елементи, неможливі без логістики: закупівля, доставка та ін. передбачені відповідними стандартами [3]. На жаль, ці стандарти не дають інформацію про те, як вибудувати таку логістику, щоб вона при мінімальних допустимих витратах у межах проектного бюджету та часу забезпечила виконання проекту. Оскільки ресурси і час завжди обмежені, а при авральному перерозподілі внаслідок незапланованих ризикових подій можуть бути зачеплені інтереси ще й інших, не тільки постраждалих від ризикових подій, функціональних областей проекту (а, також, і його учасників), завдання неминуче зводиться до багатоцільовий багатовимірної оптимізації з великою кількістю обмежень [1, 4].

Швидко і точно аналітичне рішення подібних задач оптимізації в умовах проектної діяльності неможливе через відсутність на момент ризикової події багатьох початкових даних і адекватних математичних моделей, а існуючий когнітивний підхід, в якому пошук кращих рішень здійснюють на шляху досягнення критеріальної подібності між параметрами проектного та термодинамічного процесів, обмежений відсутністю адекватної формалізації проектного середовища, в якому ці процеси протікають [4].

Дійсно, в одне логістичне проектне рівняння необхідно звести і фінанси, і матеріали, і енергію, і терміни, і документи, і багато іншого. Причому керувати цим, за аналогією з рівняннями переносу, доводиться залежною змінною в просторі-часу незалежних змінних, яких з урахуванням функціональних областей проектної діяльності, як мінімум, одинадцять [2].

Для забезпечення можливості здійснення ефективного управління проектом за рахунок вчасного реагування на

ризиків та компенсації їх наслідків в усіх функціональних областях запропоновано впровадити в управління ризиками проекту дискретну адаптивну віртуальну просторову модель надзвичайної логістики. Ця модель дозволить постійно під час виконання проекту, незалежно від того, відбуваються деякі ризикові події або ні, відтворювати поточний стан наявних в елементах та на складах ресурсів. Загальна схема підтримки проектних рішень, яка базується на віртуальній просторовій моделі, наведена на рис. 1.

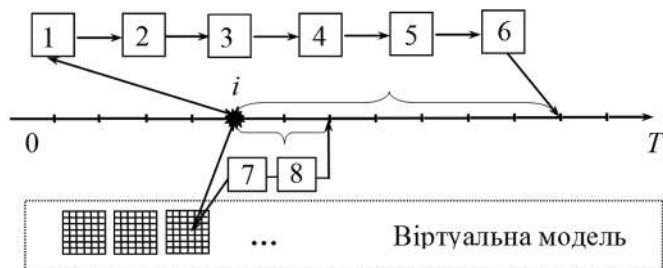


Рис. – Схема методу попередньої «гарячої» передислокації дискретного простору-часу

На рис. позначено: 1 – збирання інформації з усіх елементів проектної діяльності про наявність необхідного ресурсу, 2 – визначення елементів, з яких можна «позичити» необхідну кількість ресурсу, 3 – визначення шляху доставки ресурсу, 4 – визначення способу (технології) доставки ресурсу, 5, 7 – визначення плану надзвичайного переміщення ресурсу, 6, 8 – здійснення надзвичайного переміщення ресурсу до елемента, де відбулася ризикова подія.

В її основі дискретна адаптивна віртуальна модель простору управління проектами, яка від ітерації до ітерації накопичує інформацію про зміну поточного стану розподілу ресурсів по елементах проектної діяльності та на складах.

Згідно з рисунком, плановий життєвий цикл проекту $0 \dots T$, який складається з окремих ітерацій, на i -й ітерації переривається незапланованою ризиковою подією в елементі R , що потребує негайного втручання менеджера для ліквідації наслідків останньої. Хай до такої ліквідації необхідно залучити додаткову до планової – надзвичайну ло-

гістику, суть якої полягає в аварійному перекиданні різних ресурсів (матеріалів, коштів, енергії, обладнання, тощо) від одних до інших елементів (передбачається, що цей ресурс там є в достатній кількості), незалежних складів або інших джерел.

Розрахунок параметрів надзвичайної логістики потребує від менеджера отримання та переробки значного об'єму інформації, що он-лайн з плановим виконанням проекту може розтягнутися на велику кількість ітерацій проектної діяльності, аж до зупинки останньої. Якщо ж з самого початку проекту створюється та підтримується на кожній ітерації віртуальна модель, то її наявність може значно зменшити час прийняття рішень (див. рис.), оскільки уся необхідна інформація про поточний стан розподілу ресурсів по елементах та джерелах міститься в такій моделі.

ПЕРЕЛІК ПОСИЛАНЬ

1. Віртуальна передислокація дискретного простору-часу в задачах планування проектної логістики / Савельєва О. С. та ін. // Вісник національного технічного університету «ХПІ». Серія «Механіко-технологічні системи та комплекси», 2016. № 49 (1221). С. 56 – 62.
2. Чернов С. К. Учет рисков и неопределенностей в организационных проектах // Управление проектами та розвиток виробництва, 2006. № 1 (17). С. 41 – 44.
3. Формализация пространства управления проектами / Савельева О.С. и др. // Нові рішення в сучасних технологіях: зб. наук. праць. Харків: НТУ «ХПІ», 2016. Вип. 42 (1214). С. 154 – 159.
4. Руководство к Своду знаний по управлению проектами (Руководство PMBOK®). 5-е изд. USA/США: Project Management Institute, 2013. 586 с.
5. Савельева, О. С. Разработка термодинамической критериальной поддержки когнитивных моделей переноса в управлении проектами и программами / О.С. Савельева, И.И. Становская, А.В. Торопенко, И.Н. Щедров, Е.И. Березовская // Восточно-европейский журнал передовых технологий. – Харьков, 2015. – № 6/3 (78). – С. 53 – 59.

До питання оцінки якості web-сайту діяльності підприємства як моделі масового обслуговування

О.М. Моргалъ
ст. викладач
КПІ ім. Ігоря Сікорського
Київ, Україна

Є.О. Покровський
доцент
КПІ ім. Ігоря Сікорського
Київ, Україна

О.В. Савчук
доцент
КПІ ім. Ігоря Сікорського
Київ, Україна

І.О. Латаш
студент
КПІ ім. Ігоря Сікорського
Київ, Україна

У статті пропонується метод підходу до проектування сайту через аналіз структури його в середовищі MATLAB і приведені результати його застосування при проектуванні Web-системи автоматизації діяльності підприємства

Ключові слова: веб-сайт, веб-сервіс, автоматизація діяльності підприємства.

Зі збільшенням числа користувачів інтернету на мережу збільшується й навантаження. Якщо відкинути технічні проблеми, пов'язані з роботою web-сервера, в більшості випадків збої при завантаженні сторінки відбуваються в разі перевантаження web-сайту. Система не може обробити велику кількість запитів в одиницю часу через її обмежену пропускну здатність. Організація сайту не може бути розрахована на опрацювання надто великої кількості запитів від користувачів без відмов в системі.

Перераховані вище недоліки треба усувати під час проектування системи для забезпечення її ефективної та стабільної роботи.

ПОСТАНОВКА ЗАДАЧІ

Розробити Web- сайт системи комп'ютеризованої автоматизації діяльності підприємства, що пропонує деякий набір продуктів або послуг.

Забезпечити ефективну та стабільну роботу Web-системи завдяки застосуванню сучасного Web-сервісу.

ОПИС СТРУКТУРИ ДІЯЛЬНОСТІ ПІДПРИЄМСТВА ЯК СИСТЕМИ МАСОВОГО ОБСЛУГОВУВАННЯ

Для даної роботи може бути обрана модель сервера з N потоками обробки заявок (рис. 1).

Розглянемо алгоритм обробки заявок даними сервером. Заявка, що прийшла на сервер, починає опитувати накопичувачі (буфери) потоків, починаючи з першого, і зберігається в першому вільному накопичувачі. Якщо всі накопичувачі зайняті, заявка відкидається, таким чином, виникає функціональна відмова. Під час роботи потоки П1-

ПN постійно опитують накопичувачі, і, якщо у відповідному накопичувачі міститься заявка, вона витягується з накопичувача і починається її обробка. Кожен потік обробки заявок з певною частотою і на деякий проміжок часу може перериватися через апаратні/ програмні відмови/ збої. У даній ситуації, якщо в момент відмови/збою сервер обробляв деяку заявку, він змушений відкинути її. Якщо ж заявка успішно оброблена, SMTP-сервер відправляє одержувачу

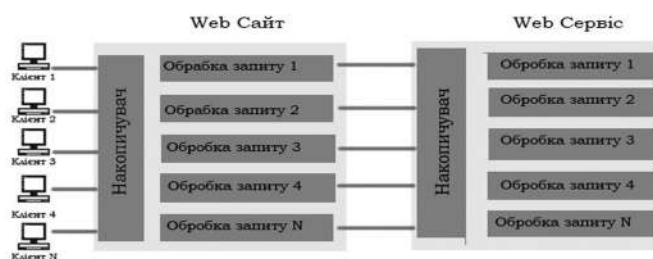


Рис. 1 Структура моделі Web-системи

У середовищі MATLAB була синтезована модель SMTP-сервера (рис.1), яка дає можливість моделювати обробку повідомлень сервером з N

(наприклад, чотирма) каналами передачі, при цьому можна конфігурувати різні структури та параметри генератора заявок і сервера. Зокрема, є можливість вибору частоти генерації запитів, середньої тривалості виконання запиту («складності» запитів), частоти відмов каналів обробки запитів, а також часу їх відновлення і загального часу. Якщо надходить заявка, а всі накопичувачі зайняті, то має місце функціональна відмова.

Дана структурна схема описує систему взаємодії клієнта з Web-сайтом і з Web-сервісом. Перша фаза, яка містить блоки H1, K11 - K1N, відповідає за процес моделювання Web-сайту, а друга фаза, яка містить блоки H2, K21 - K2N, відповідає за процес моделювання на Web-сервісі. Джерело запитів являє собою сукупність запитів від користувачів Інтернету на доступ до певної Web-сторінки. Запит від клієнта, потрапляючи в систему, в першу чергу обробляється одним з каналів Web-сайту - K11 - K1N, або стає в чергу в накопичувач H1 до тих пір, поки не звільниться один з каналів. Далі, оброблений запит звертається до Web-сервісу для отримання додаткової інформації. Потрапляючи на Web-сервіс, запит, аналогічно, обробляється одним з каналів K21 - K2N, або потрапляє в накопичувач H2. Після обробки на Web-сервісі, інформація передається клієнту через Web-сервер у вигляді html-сторінки.

Опис вимоги дозволяє в загальних рисах окреслити функціональну схему проектованої моделі, а можливість

використання ієрархічних блоків (підсистем) в Simulink - почати складання цієї схеми. Ключовим моментом в понятті системи масового обслуговування є обслуговування і на відміну від реального світу «обслуговування» моделі - це просто затримка заявки на певний час.

ЗАГАЛЬНИЙ ОПИС ФУНКЦІОНАЛЬНОСТІ ВЕБ-СИСТЕМИ ПІДПРИЄМСТВА

В основі побудови системи знаходиться динамічна об'єктна модель, в якій всі дані зберігаються в реляційній базі даних, моделюючи її відповідно до вимог і побажань клієнта без додаткового втручання розробника в структуру.

Web-система автоматизації діяльності підприємства має шаблонну структуру. Основні можливості, якими володіє система:

- динамічне формування структури сайту, що дозволяє змінювати її у відповідності до поточних потреб і побажань підприємства;
- динамічна зміна наповнення - оновлення інформації в режимі реального часу, причому зміст може бути найрізноманітнішим, як простий текст, так і таблиця, списки, посилання;
- підключення таких додаткових розділів, як новини і каталог продукції або послуг;
- підключення додаткових механізмів, таких як підписка і розсилка новин підписчикам;
- розміщення коротких рекламних текстовок – так званих банерів, для представлення користувачу короткого опису деякої важливої інформації (проведення акції, останніх новин, реклами продукту тощо).

Функціонально Web-система автоматизації діяльності підприємства складається з 2-х частин: адміністративної і клієнтської. В адміністративній частині задаються і редагуються настройки відображення інформації, а клієнтська частина – це власне сайт, в якому в заданому порядку і відповідно до настройок відображається введена інформація.

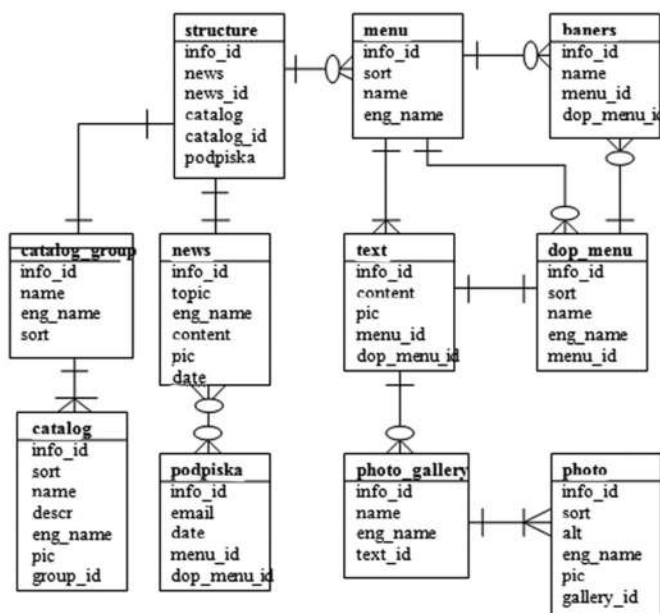


Рис. 2 E-R діаграма Web-сервіса

АЛГОРИТМ ФОРМУВАННЯ ІНФОРМАЦІЙНОГО НАПОВНЕННЯ СТОРІНКИ

Спочатку в адміністративній частині формується структура, потім підключаються додаткові розділи і механізми. Далі користувач вводить інформацію, яка буде відображатися на сайті.

В клієнтській частині спочатку підвантажується вибраний шаблон, потім формуються пункти меню і виводиться перша сторінка.

Формування підменю здійснюється при виборі певного пункту меню. Якщо підменю є, тоді формується його склад і завантажується та сторінка, яка належить до першого пункту підменю. Якщо підменю в розділі немає, тоді завантажується інформація, котра відноситься до даного пункту меню.

Сформовані динамічні сторінки кешуються. Час зберігання кеша – доба, проте, якщо відбулися зміни в якійсь з таблиць в перебігу останніх діб, тоді буде завантажуватися нова інформація. Але при прогляданні клієнтської частини також слід урахувувати особливості кешування браузерів, які використовуються для проглядання Web-сторінок. Наприклад, Opera добре кешує проглянуті сторінки і часто для відображення нової інформації треба сторінку відновити.

Керівництво по роботі з адміністративною частиною сайту складається з форм „Додавання” “Сортування”, “Редагування”, „Пошуку” і „Видалення”.

Для формування структури і зовнішнього вигляду сайту першочергово треба визначити, які пункти меню і спеціальні розділи повинні бути присутні. Таке розділення необхідне, оскільки принцип обробки інформації, що міститься в даних розділах істотно відрізняється. В цілому, процедура формування структури і зовнішнього вигляду така:

- 1) динамічне формування пунктів меню,
- 2) підключення додаткових розділів і механізмів.

Динамічне формування пунктів меню

Для додавання або редагування кількості і порядку дотримання пунктів меню є набір відповідних форм.

В нашому випадку передбачена можливість додавання підменю, тобто меню 2-го рівня. Для цього треба зайти в відповідний пункт меню і заповнити всі поля. У випадковому списку „пункт меню” відображається перелік всіх введених пунктів меню першого рівня.

ПІДКЛЮЧЕННЯ ДОДАТКОВИХ РОЗДІЛІВ І МЕХАНІЗМІВ

Серед основних розділів можна виділити: новинний розділ та каталог, а серед додаткових механізмів наступні: підписка на новини і розсилка новин, пошук по сайту тощо.

КЕРІВНИЦТВО ПО РОБОТІ З КЛІЄНТСЬКОЮ ЧАСТИНОЮ

Власне призначена для користувача частина Web-системи підприємства являє собою відображення заданої структури і наповнення в адміністративній частині.

Шаблон представлення наповнення „Новин” трохи відрізняється від стандартного, але всі принципи оформлення зберігаються.

Для проглядання архіву новин достатньо натиснути на посилання з відповідним діапазоном і всі новини за цей період відобразяться на цій же сторінці перед архівом.

Для „Каталогу” передбачене формування підрозділу з переліку груп, а на сторінці відображається перелік одиниць продукції з фото (якщо воно є) з коротким описом (150 символів).

Детальну інформацію можна проглянути у вікні, що відкривається, кликнувши на посилання „Розсилання новин” здійснюється автоматично кожного дня зранку, оскільки тоді є можливість поправити неправильно введену інформацію.

ОЦІНКА ЯКОСТІ WEB-СЕРВІСУ ДІЯЛЬНОСТІ ПІДПРИЄМСТВА

Серед категорій якості перше місце займає теорія надійності, ядром якої є показники безвідмовності. Тому для оцінки якості нашої Web-системи прийняті залежності ефективності і надійності від часу. При моделюванні у звичайному режимі при заданих умовах моделювання [1], у випадку абсолютно надійної системи (модель системи масового обслуговування без відмов) у разі одноканальної системи імовірність відмови складає 0,46. Це є дуже низький показник ефективності. Для отримання високої ефективності та безвідмовності роботи системи треба або збільшити число каналів обслуговування, або збільшити ємність накопичувачів. При числі каналів 4 отримані результати, відображені на рисунках 3 та 4.

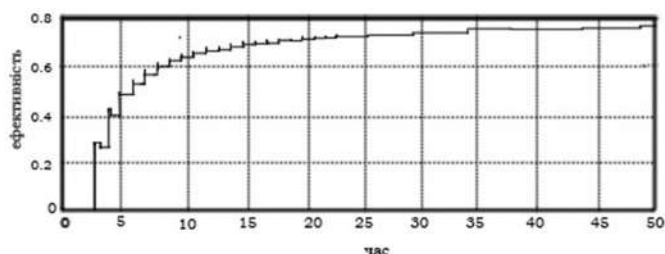


Рис.3 Залежність ефективності WEB-сервісу від часу (години)

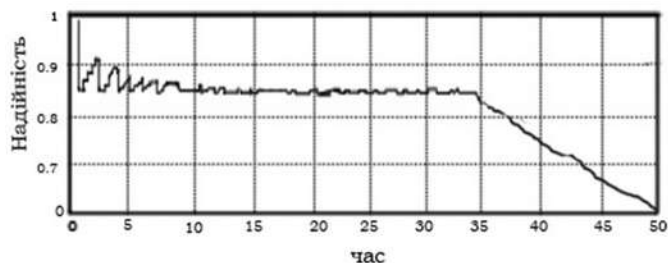


Рис.4 Залежність надійності WEB-сервісу від часу (години)

ПЕРСПЕКТИВИ РОЗВИТКУ ТА ВИСНОВКИ

На теперішньому етапі реалізації проекту існує ряд недоліків, що обмежують функціональність розробленої Web-системи. Наступні напрями можуть бути розроблені в подальшому:

1. Розмежування прав доступу до системи для адміністратора та користувачів-редакторів.
2. Додаткові механізми: форум, голосування, закриті розділи. Механізм закритих розділів, інформація яких доступна тільки обмеженому числу користувачів, характерний для вузькоспеціалізованих сайтів великих корпорацій, а форуми і голосування використовуються на великих корпоративних сайтах.
3. Публікація і ротація інформації. Задання періоду ротації і остаточної публікації інформації доцільно на великих новосних порталах, проте така функція також може бути корисна і в Web-системах великих корпорацій.
4. Система статистики відвідин. Вона потрібна для об'єктивного контролю відвідин клієнтської частини. Проте можна не розробляти власну систему статистики, а використовувати вже існуючі, представляючи дуже великий спектр різноманітної інформації про відвідувачів, наприклад, статистику середовища MATLAB.

ПЕРЕЛІК ПОСИЛАНЬ

1. Савчук О.В. Моделирование мультиагентного smtp сервера в среде matlab/ Покровский Е.В., Савчук О.В., Кривенко К.С. // Интеллектуальный анализ информации IAI-2012 / 36. праць. – К.: Просвіта, 2012. – С.311-317.
2. Репнікова Н.Б. Теорія автоматичного керування: класика і сучасність: підручник. – К.: НТУУ «КПІ», 2011. – 328 с.

Використання компонентного підходу при проектуванні сервісів для системи управління бізнес-процесами

Троцький Сергій
Олександрович
аспірант
КПІ ім. Ігоря Сікорського
Київ, Україна

Погорілий Юрій
Анатолійович
студент
КПІ ім. Ігоря Сікорського
Київ, Україна

Коломійчук Микита
Сергійович
студент
КПІ ім. Ігоря Сікорського
Київ, Україна

Вовк Євгеній
Андрійович
студент
КПІ ім. Ігоря Сікорського
Київ, Україна

В роботі розглянуто можливості використання сервіс-орієнтованої архітектури для побудови систем виконання бізнес-процесів. Також представлено загальну математичну модель для такої системи і показано переваги сервіс-орієнтованої архітектури на прикладі бізнес-процесу описаного мовою BPEL.

Для автоматизації бізнес-процесів використовують спеціальні інформаційні системи виконання бізнес-процесів. Використання сервіс-орієнтованої архітектури в таких системах дозволяє швидко адаптуватися до мінливих умов бізнес-середовища. В свою чергу, автоматичний пошук та композиція існуючих Web-сервісів може значно полегшити впровадження та підтримку цих систем.

ЗАГАЛЬНА МАТЕМАТИЧНА МОДЕЛЬ КОМПОНЕНТО-БАЗОВАНОЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ

Задача (Task) – це деяка одиниця роботи (unit of work), що має бути зроблена на протязі деякого часу, та яка має на меті обробку вхідних даних за допомогою певного алгоритму.

Компонент в SOA архітектурі – це програмний модуль що виконує одну або більше задач. Компоненти комбінуються в сервіси. Сервіси виконують певні процеси.

Архітектура програмного забезпечення інформаційної системи, яка побудована на базі компоненто – базового підходу може бути визначена за допомогою множин $\{C, P, R, S, K, T, Q\}$

$\{C\}$ - це множина компонентів системи;

$\{P\}$ - це множина бізнес процесів;

$\{R\}$ - це множина взаємозалежностей між компонентами системи;

$\{S\}$ - це множина сервісів системи;

$\{K\}$ - це множина взаємозалежностей між сервісами системи;

$\{T\}$ - це множина задач;

$\{Q\}$ - це множина взаємозалежностей між задачами.

Видимі ззовні властивості визначають вхідні і вихідні параметри компонентів. Якість будь-якого програмного забезпечення визначається його архітектурою, а це означає, що архітектура інформаційної системи є важливим і вирішальним елементом в виборі підходу до розробки системи.

ФОРМУЛЮВАННЯ ПРОБЛЕМИ ТА МЕТОДОЛОГІЯ

Необхідно розробити математичну модель яка б дозволила за допомогою аналітичних засобів показати, як найоптимальніше згрупувати задачі в сервіси за допомогою критерію оптимальності.

У загальному випадку бізнес-процес – це сукупність взаємопов'язаних заходів і завдань, спрямованих на створення певного продукту або послуги для споживачів. Бізнес-процеси виконуються всередині кожної організації, незалежно від того, формалізовані вони чи ні. Популярна сьогодні система умовних позначень (нотація) BPMN визначає скінченний набір графічних елементів для описання бізнес-процесу:

- Об'єкти потоку управління: події, дії та логічні оператори;
- З'єднувальні об'єкти: потік управління, потік повідомлень та асоціації;
- Ролі: пули і доріжки;
- Артефакти: дані, групи і текстові анотації.

Найменшою елементарною дією в бізнес-процесі є завдання (task). З точки зору автоматизації бізнес-процесів, саме завдання є головним цільовим об'єктом. Інші елементи пов'язані з розв'язанням задач оркестрування, координування, та ін. Для цих задач успішно застосовують мови виконання бізнес-процес, наприклад, BPEL.

Отже, нехай бізнес процес, який необхідно автоматизувати, описаний мовою BPEL. Мова BPEL, використовуючи мову XML за основу, підтримує стек технологій розробки та надання Web-сервісів, в тому числі SOAP, WSDL, UDDI, WS-Reliable Messaging, WS-Addressing, WS-Coordination, WS-Transaction. Процес, описаний мовою BPEL, складається з кроків, які називаються “завданнями”. Мова BPEL підтримує примітивні, а також структуровані завдання. Структуровані завдання, або підпроцеси, є невід'ємною складовою частиною мови BPEL. Вони дозволяють будувати бізнес-процес, використовуючи рівні абстракції. Таким чином, при правильній побудові, на кожному рівні бізнес-процес залишається простим і зрозумілим. Структуровані завдання повністю складаються з примітивних завдань. Примітивні завдання є основними конструкціями

і використовується для загальних задач, однією з яких є виклик інших Web-сервісів. Таке завдання в XML представлено бізнес-процесу позначається тегом <invoke>. Тобто, для виявлення системних завдань як у складі процесу, так і у складі структурованих завдань достатньо віднайти такі теги у тексті XML.

Проблема SOA є надзвичайно складною і важливість початкового підходу до встановлення належної абстрактної методології, відображеної щодо реальної проблеми, алгоритму належної кластеризації не може бути недооцінена.

Загальна структура бізнес-процесу складається з декількох етапів, кожен з яких може бути розбитий на більш фундаментальні одиниці, які ми називаємо Задача (Task), які можуть бути використані різними процесами. Послідовність виконуваних дій (W) може бути зображена, як об'єднання різних бізнес процесів (P):

$$W = \bigcup_{i=1}^N P_i \quad (1)$$

Кожна задача є незалежною одна від одної:

$$t_j \cap t_i = 0 \quad (2)$$

для будь-яких i та j .

Сам процес це множина задач, які інкапсулюються в компонентах.

Процес можливо описати наступним чином:

$$P = t_1 t_2 t_3 t_4 \dots t_k \dots t_n \quad (3)$$

Таким чином можливо побудувати матрицю взаємодії процесів та задач

$$\begin{pmatrix} P_1 & 1/0 & 1/0 & 1/0 & 1/0 & 1/0 & 1/0 \\ P_2 & 1/0 & 1/0 & 1/0 & 1/0 & 1/0 & 1/0 \\ P_3 & 1/0 & 1/0 & 1/0 & 1/0 & 1/0 & 1/0 \\ P_4 & 1/0 & 1/0 & 1/0 & 1/0 & 1/0 & 1/0 \\ P_5 & 1/0 & 1/0 & 1/0 & 1/0 & 1/0 & 1/0 \\ P_n & 1/0 & 1/0 & 1/0 & 1/0 & 1/0 & 1/0 \end{pmatrix} \quad (4)$$

де 1 відповідає така конфігурація коли процес включає дану задачу, а 0 відповідає конфігурація коли дана задача не включається до складу даного процесу[5].

Бізнес-процеси виконуються всередині кожної організації, незалежно від того, формалізовані вони чи ні. Популярна сьогодні система умовних позначень (нотація) BPMN визначає скінченний набір графічних елементів для описання бізнес-процесу [1]:

- Об'єкти потоку управління: події, дії та логічні оператори;
- З'єднувальні об'єкти: потік управління, потік повідомлень та асоціації;
- Ролі: пули і доріжки;
- Артефакти: дані, групи і текстові анотації.

На Рис. 1 у нотації BPMN 2.0 зображений спрощений узагальнений бізнес-процес. Він складеться з двох учасників: деякого програмного забезпечення (Customer system) та оператора (Operator), що взаємодіє з ним. У наведеному процесі дії, що автоматично виконуються системою (Auto-task) межують з діями, що потребують втручання людини (User task). При цьому, логіка виконання системних дій є невід'ємною частиною програмного забезпечення, що відповідає за виконання бізнес-процесу.

Процес, побудований таким чином, може добре виконувати покладені на нього функції та має як переваги, насамперед, простоту та повний контроль над системою, так і недоліки, серед яких найбільш суттєвими є статичність процесу, висока складність розробки, складність підтримки та модифікації, складність повторного використання логіки виконання системних дій.

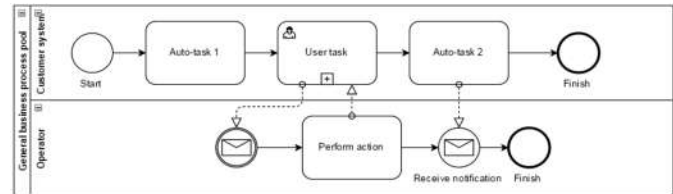


Рис. 1. Приклад бізнес-процесу

Частково усунути недоліки описані вище можна за рахунок впровадження сервіс-орієнтованої архітектури. Сервіс-орієнтована архітектура (Service-Oriented Architecture, SOA) становить собою стиль створення архітектури ІТ, спрямований на перетворення бізнесу в ряд пов'язаних сервісів – стандартних бізнес-задач, які можна при необхідності викликати через мережу [2]. Це може бути Інтернет, локальна мережа або географічно розподілена мережа, яка об'єднує різні технології і поєднує сервіси так чином, ніби вони встановлені на одній локальній машині. Для виконання певного бізнес-завдання можна об'єднувати множину таких сервісів. Це дозволяє компаніям швидко адаптуватися до змін умов і вимог ринку.

Визначимо Web-сервіси як автономні, модульні програми, призначені для реалізації бізнес-процесів. Побудова і надання Web-сервісів спираються на ряд галузевих стандартів: WSDL (для описання сервісів), UDDI (для інформування та публікації); SOAP (для обміну повідомленнями) [3]. Ці специфікації не залежить від платформи і мови, завдяки чому користувачі можуть пов'язувати різні компоненти з різних організаційних структур. Розглянемо сучасний підхід, при якому існуючий бізнес процес буде виконуватися за допомогою інформаційної системи управління бізнес-процесами. А сам бізнес-процес, за виключенням логічних операцій та ручних дій, має підтримуватися існуючими Web-сервісами, задовольняючи основоположні принципи SOA.

Модифікований бізнес-процес, що задовольняє умови описаного підходу та характеризується такими перевагами:

- озділення логіки управління бізнес-процесу та бізнес-логіки, що додає гнучкості системи та зменшує вартість підтримки;
- можливість розробки функціоналу бізнес-логіки зручними для кожного окремого випадку способами;
- можливість використання Web-сервісів третіх сторін, що зменшує час на розробку системи;
- можливість повторного використання Web-сервісів у різних частинах та процесах системи.

В даному процесі логіка виконання системних дій ви-

несена у Web-сервіси, що можуть бути реалізовані будь-якою мовою програмування та за допомогою будь-яких технологій. В основному потоці при цьому лишаються користувацькі дії та дії виклику Web-сервісів. Таким чином, ми можемо повністю перекласти виконання основного потоку на один з серверів виконання бізнес-процесів. Промисловим стандартом у даній галузі є Oracle BPEL Process Manager Server [4], що є сервером виконання бізнес-процесів, описаних мовою виконання бізнес-процесів BPEL (Business Process Execution Language). Головним недоліком, описаного вище сучасного і досить розповсюдженого підходу є необхідність ручного пошуку Web-сервісів та їх жорстка прив'язка до кожної дії бізнес-процесу.

При цьому, за будь-яких змін у структурі Web-сервісу або виходу його з ладу – вся система буда непрацездатною. Для уникнення такої ситуації і надання більшої незалежності системам управління бізнес-процесів підприємства пропонується побудувати окрему систему, яка візьме на себе задачу пошуку та композиції Web-сервісів для підтримки будь-якого бізнес-процесу та зведе до мінімуму необхідність ручного втручання в бізнес-логіку процесу.

ВИСНОВОК

В даній роботі було розглянуто інформацій системи виконання бізнес процесів, які застосовують для їх автоматизації. запропоновано математичну модель яка дозволяє

оптимально згрупувати задачі в сервіси за допомогою певного критерію. Використання сервіс-орієнтованого підходу в проектуванні та побудові таких системах дозволяє швидко адаптуватися до мінливих умов бізнес-середовища. В свою чергу, автоматичний пошук та композиція існуючих Web-сервісів може значно полегшує впровадження та підтримку цих систем.

ПЕРЕЛІК ПОСИЛАНЬ

1. Information technology – Object Management Group Business Process Model and Notation, ISO/IEC 19510, 2013;
2. Мохаммед И. Мабрук. Краткие основы SOA // IBM Developer Works [Електронний ресурс]. –2010.–Режим доступу до ресурсу: <http://www.ibm.com/developerworks/ru/edu/ws-soa-ibmcertified/index.html>
3. Web Services Architecture –W3C Working Group Note 11. –The World Wide Web Consortium (W3C) [Електронний ресурс]. –2004. –Режим доступу до ресурсу: <http://www.w3.org/TR/ws-arch/>.
4. ORACLE Data Sheet –ORACLE BPEL Process Manager – Режим доступу: <http://www.oracle.com/technetwork/middleware/bpel/overview/ds-bpel-11gr1-1-134826.pdf>;
5. McIlraith S. Semantic Web services / McIlraith S., Son T.C., Zeng H. // IEEE Intelligent Systems. Special Issue on the Semantic Web. –2001. –Vol. 16. –N. 2–P. 46-53.

Рецензент: д.т.н., проф., зав. каф. АУТС, КПІ ім. Ігоря Сікорського, С.Ф. Теленик

Інформаційна технологія проектування рівнонапружених деталей машин

О.Л. Становський

д.т.н., зав. каф.

Одеський національний політехнічний університет
Україна, Одеса

П.С. Швець

к.т.н., доцент

Одеський національний політехнічний університет
Україна, Одеса

А.В. Торопенко

к.т.н., доцент

Одеський національний
політехнічний університет
Україна, Одеса

О.Є. Науменко

ст. викладач

Одеський національний
політехнічний університет
Україна, Одеса

Абу Шена Осама

аспірант

Одеський національний
політехнічний університет
Україна, Одеса

Запропоновано метод проектування рівнонапружених вузлів циліндричних резервуарів, що містять плоскі круглі пластини змінної товщини, форма діаметральних перерізів яких моделюється рівнянням Гаусса. Для рішення рівняння вигину цих пластин використовуються вироджені гіпергеометричні функції Куммера та Уїттекера.

Ключові слова: рівнонапружені деталі, пластини змінної товщини, САПР

Сьогодні *практично немає* продукції або послуги, яка б не містила або не використовувала інформаційні технології (ІТ) – основи створення високих *технологій*.

Необхідність впровадження ІТ для розвитку машинобудування пояснюється вимогами до скорочення термінів проектування і підготовки виробництва для випуску нових і модернізуються виробів, витрат на проектування і виробництво, вартості довготривалого післяпродажного обслуговування. Крім того, ІТ необхідні для перебудови (реінжиніринга) підприємств відповідно до сучасних вимог підвищення якості і конкурентоспроможності виробів, відновлення старих ринків збуту і виходу на нові ринки.

На етапах життєвого циклу рівнонапружених деталей машин широко застосовуються наступні ІТ:

- **«механічні» САПР**, що забезпечують моделювання окремих деталей, вузлів, пристроїв, комплексний опис компонентів проєктованих пристроїв, моделювання механічних полів навантажень, напружень, тривимірних структур і т.д. Тут виділяють «легкі» (з меншою кількістю функцій і більш дешеві), «середні» і «важкі» САПР (з розширеними можливостями і більш дорогі).

- **спеціалізовані інформаційні технології** і системи, наприклад, CASE (Computer-Aided Software / System Engineering) – технології, SCADA (Supervisor Control And Data Acquisition) системи, системи моделювання і аналізу електронних схем і т.д.

- технології класу **MRPII** (Manufacturing Resource Planning) і **ERP** (Enterprise Resource Planning), що забезпечують рішення широкого спектру задач планування ресурсів і управління діяльністю підприємств. Останніми роками, які характеризуються підвищеною конкуренцією, інтенсивно розвиваються **CRM** (Customer Relationship Management) системи як набір додатків або у вигляді надбудови над **ERP**. В CRM-системах акцент робиться на взаємостосунки «компанія—клієнт і, перш за все, утримання

старих клієнтів за рахунок обліку їх індивідуальних потреб і особливостей.

- Розширяється застосування *технології XML* (eXtensible Markup Language), яка охоплює найважливіші задачі обміну даними між системами різних виробників, обміном документів між підприємствами.

Без використання ІТ неможливе функціонування багатьох сучасних систем, наприклад САПР, АСУ, CALS (Continuous Acquisition and Lifecycle Support), логістики і т.д.

Грамотне використання ІТ дозволяє витягувати максимум користі зі всієї *інформації*, яка є на підприємстві, і завдяки цьому робити більш точні прогнози, уникаючи можливих помилок при ухваленні управлінських і проєктних рішень в умовах невизначеності і ризику. Жорстка конкурентна боротьба робить підприємства у край чутливими до щонайменших прорахунків в управлінні, переваги мають підприємства, які використовують сучасні *інформаційні технології*.

Так, при розробці та конструюванні нових деталей машин проєктувальник повинен завжди прагнути до максимальної ефективності проєкту. Однією з таких складових є вимога забезпечення рівнонапруженості деталей, що забезпечує максимальне значення відношення «опір / маса» і, відповідно, мінімальну матеріаломісткість виробу в цілому.

На жаль, домогтися повної рівності напруг у всіх точках деталі неможливо навіть для статичної задачі. Це пояснюється різним впливом навантаження на окремі елементи деталей складної форми, неоднорідністю їх матеріалу та іншими конструктивними і технологічними особливостями. І будь-яке намагання спроектувати рівно напружену деталь є лише спробою вирішити проблему за рахунок варіювання однією з основних конструктивних характеристик, наприклад, товщиною елемента конструкції. Напри-

клад, посудини, що працюють під тиском, мають корпус, який складається з оболонок.

В більшості випадків поля напружень, які в них виникають, суттєво неоднорідні. Отже, практично неможлива мінімізація маси корпусних конструкцій без використання в них елементів змінної товщини [1].

За умовами навантаження таких об'єктів найбільш напруженим ділянкою є місця стику (найчастіше, зварювання) обичайки і днища [2]. Розрахунки на міцність показують, що саме в цьому місці товщина днища повинна бути найбільшою, що призводить до вельми нетехнологічних рішень: виготовляти днища посудин у вигляді круглих пластин із змінною від центру до краю товщиною.

Природно припустити, що конструкцію такого виробу одержують в результаті складних розрахунків за рівняннями опору матеріалів, які є неоднорідними диференціальними рівняннями другого ступеня [3]. Рішення таких рівнянь для конкретних об'єктів записують у вигляді суми загального і частинного рішень, тобто воно складається з двох лінійно незалежних функцій [4]. Однак, такі підходи не забезпечують оптимізацію конструкцій посудин, оскільки одержувані при цьому деталі і вузли однакового напруження ще не гарантують одночасного досягнення мінімальної маси майбутнього об'єкта [5]. У той же час, існує математичний апарат гіпергеометричних функцій, за допомогою якого подібна проблема може бути вирішена.

Для одночасного досягнення рівнонапруженості і мінімальної маси складного корпусного виробу запропоновано метод, який враховує обидві вимоги та базується на розрахунку пластин різної товщини.

Розглядали в якості прикладу корпусні деталі типу «кругла жорстко защемлена пластина». Зміну товщини для таких деталей в давальному напрямі в загальному випадку записують як функцію Гауса [1, 3]:

$$\delta(r) = \delta_0 \exp(-n^2/6R^2), \quad (1)$$

де: δ_0 – товщина пластини в центрі при $r = 0$.

Параметр n в рівнянні (1) визначає інтенсивність зміни товщини круглої пластини в радіальному напрямку. В коллому напрямку товщина залишається постійною, тобто форма пластини передбачається осесиметричною.

Форми поверхонь з параметром $n < 0$ можна рекомендувати для круглих пластин, що згинаються поперечним навантаженням p при жорсткому защемленні їх контуру, коли на контурі діє максимальний вигинальний момент. При шарнірному закріпленні контуру максимальний вигинальний момент виникає в центрі пластини, і кращою стає форма з максимальною товщиною в центрі, коли $n < 0$.

При оптимізації форми діаметрального перерізу круглої пластинки прагнули до мінімізації її маси, яка визначається обсягом використаного матеріалу.

Формула, що визначає зміну товщини такої пластини в радіальному напрямку, отримана з (1) у вигляді:

$$\delta_r(x) = \frac{V_0}{\pi R^2} \cdot \frac{n}{6[1 - \exp(-n/6)]} \exp\left(-\frac{nx^2}{6}\right). \quad (2)$$

Диференціальне рівняння осесиметричного вигину такої пластини при рівномірно розподіленому навантаженні (тиску) p щодо кута повороту нормалі до серединної поверхні φ має другий порядок [5]:

$$\frac{d^2\varphi}{dx^2} + \left(\frac{1}{x} - nx\right) \frac{d\varphi}{dx} - \left(\frac{1}{x^2} + \mu n\right) \varphi = -\bar{p}x \exp\left(\frac{nx^2}{2}\right). \quad (3)$$

Загальне рішення однорідного рівняння (3) (при рівності нулю правої частині) задане статичним рядом, що при практичних розрахунках представляє певні обчислювальні труднощі. Для застосування в САПР рівнонапружених деталей рішення цього однорідного рівняння представлено за допомогою вироджених гіпергеометричних функцій Уїттекера $M_{k,\gamma}(z)$, $W_{k,\gamma}(z)$ [5]

Запропонована модель у вигляді експоненційної функції Гауса дозволяє визначати оптимальну (рівнонапруженість, мінімальну масу) форму суцільний круглої пластини з довільним закріпленням по зовнішньому контуру. Модель дозволяє відображати таке закріплення в проміжку від абсолютно вільного (шарнірного обпирання) до абсолютно жорсткого (защемленого).

Результатом дослідження є підтвердження ефективності використання методу оптимізації форми круглої пластини змінної товщини, що полягає в переході від фіксованої товщини пластинки в її центрі до її фіксованому обсягу. Результат також доповнює модель вигину круглої пластини змінної товщини у вигляді експоненційної функції Гауса, що враховує залежності товщини в центрі пластинки від її обсягу, що використовується в рамках запропонованого методу.

ПЕРЕЛІК ПОСИЛАНЬ

1. Optimization of uniformly stressed structures of cylindrical tanks in CAD / O. Saveleva, Yu. Khomyak, I. Stanovska, A. Toropenko, E. Naumenko // Eastern-European Journal of Enterprise Technologies. 2016. № 6/7 (84). p. 10 – 16.
2. ДНАОП 0.00–1.07–9,4. Правила устройства и безопасной эксплуатации сосудов, работающих под давлением / Киев: Комитет по надзору за охраной труда Украины, 1994. – 55 с..
3. Математическое моделирование профиля равного сопротивления / В. В. Новиков, В. Г. Максимов, С. А. Баллан, О. Е. Гончарова // Оптимизация в материаловедении. – Одесса: АстроПринт, 1999. – С. 151.
4. Abramovitz, M. Handbook of mathematical functions with formulas, graphs and mathematical tables // Edited by M. Abramovitz, I. A. Stegun – Washington, 1972. – 832 p..
5. Хомяк Ю. М., Тшигам Г. Ж. Розв'язок задачі вигину круглої пластини змінної товщини з використанням функцій Уїттекера / Моделирование в прикладных научных исследованиях. Одесса: ОНПУ, 2015. Вып. 23. С. 79 – 80.

Метод резервирования и восстановления данных в распределенных системах их хранения

Виноградов Ю. Н.

ст. преп.

КПИ им. Игоря Сикорского
Украина, Киев

Иванов В. Г.

студент

КПИ им. Игоря Сикорского
Украина, Киев

В работе предложен способ резервирования и восстановления данных, хранящихся на разнесенных удаленных носителях при постоянной или временной потере доступа к любым n из них. Высокая эффективность предложенного способа достигается за счет использования теоретически минимального числа резервных носителей, равного количеству носителей, доступ к которым утрачен, а также предельно простых вычислительных процедур, обеспечивающих высокую скорость реконструкции утраченных данных. Приведен теоретическое обоснование, изложена методика формирования резервных кодов и их использования для восстановления информации.

Ключевые слова: распределенные системы удаленного хранения информации, восстановление данных, восстанавливающие коды, доступ к удаленно хранящимся данным, надежность удаленного хранения данных.

Быстрый прогресс средств телекоммуникаций и средств передачи данных привели к расширяющему использованию распределенных систем обработки и хранения информации. Такие системы позволяют гибко реализовать прогрессивные концепции обмена и концентрации вычислительных ресурсов и информации для решения прикладных задач.

Для обеспечения высокой надежности хранения информации в распределенных системах необходимы эффективные средства резервирования и восстановления данных при утрате доступа к одному или нескольким узлам хранения.

Таким образом, научная задача разработки эффективных методов резервирования и восстановления информации в распределенных системах ее хранения является важной и актуальной.

Целью исследований является разработка метода и средств восстановления данных в распределенных системах, который бы обеспечивал решение этой задачи при потере доступа к относительно большому числу узлов хранения информации с использованием минимального числа резервных узлов.

Рассматривается следующая модель: доступная абонентам системы информация хранится на m распределенных узлах. Условно считается, что объем блока информации, хранящейся на каждом из узлов одинаков. Блоки информации, хранящиеся на каждом из m узлов обозначены как B_1, B_2, \dots, B_m . Каждый из блоков содержит n фрагментов. То есть, каждый j -тый ($j=1, \dots, m$) блок B_j может быть представлен как набор фрагментов $B_j = \{a_{j,1}, a_{j,2}, \dots, a_{j,n}\}$.

Предлагается эффективный способ резервирования хранящейся в распределенной системе информации, позволяющий восстанавливать данные при потере доступа к любым их q узлам хранения с использованием q дополнительных (резервных) узлов, то есть способ, позволяющий достичь теоретического минимума избыточности резерви-

рования.

Способ основан на специфическом представлении информации в распределенной системе в виде матрицы A , m строк которой соответствуют узлам хранения, n столбцов - одноименным фрагментам блоков данных, хранящихся на узлах:

$$A = \begin{pmatrix} a_{1,1} & a_{1,2} & \dots & a_{1,n} \\ a_{2,1} & a_{2,2} & \dots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m,1} & a_{m,2} & \dots & a_{m,n} \end{pmatrix}$$

Коды резервирования, используемые для восстановления данных, доступ к которым утрачен, формируются как XOR-суммы элементов столбцов матрицы A , а также расположенных под разными углами восходящих и нисходящих диагоналей матрицы A . В настоящей работе предлагаемый способ конкретизирован для случая восстановления данных с пяти ($q=5$) узлов хранения с использованием пяти резервных узлов, информация которых образует 5 резервных блоков R_1, R_2, \dots, R_5 .

Фрагменты первого резервного блока $R_1 = \{r_{1,1}, r_{1,2}, \dots, r_{1,n}\}$ формируются как XOR-суммы элементов столбцов матрицы A :

$$\forall j = 1, \dots, n : r_{1,j} = \bigoplus_{i=1}^m a_{i,j} \quad (1)$$

Фрагменты второго резервного блока $R_2 = \{r_{2,1}, r_{2,2}, \dots, r_{2,n+m-1}\}$ формируются как XOR-суммы элементов диагоналей под углом 45° матрицы A :

$$\begin{aligned} \forall 1 \leq i < m : r_{2,i} &= \bigoplus_{j=1}^i a_{j,i-j+1}, \\ \forall m \leq i \leq n : r_{2,i} &= \bigoplus_{j=1}^m a_{j,i-j+1}, \\ \forall n < i < m+n : r_{2,i} &= \bigoplus_{j=1}^{n+m-i} a_{i-m+j,m-j+1} \end{aligned} \quad (2)$$

Фрагменты третьего резервного блока $R_3 = \{r_{3,1}, r_{3,2}, \dots, r_{3,n+m-1}\}$ формируются как XOR-суммы элементов диагоналей под углом -45° матрицы A :

$$\begin{aligned} \forall 1 \leq i < m : r_{3,i} &= \bigoplus_{j=1}^i a_{m-j+1, i-j+1}, \\ \forall m \leq i \leq n : r_{3,i} &= \bigoplus_{j=1}^m a_{m-j+1, i-j+1}, \\ \forall n < i < m+n : r_{3,i} &= \bigoplus_{j=1}^{n+m-i} a_{n+m-i-j+1, n-j+1} \end{aligned} \quad (3)$$

В формулах (3) обозначение $\alpha \text{ div } \beta$ соответствует операции целочисленного деления (то есть с отбрасыванием остатка) α на β .

Фрагменты данных четвертого резервного узла $R_4 = \{r_{4,1}, r_{4,2}, \dots, r_{4,n+2 \cdot (m-1)}\}$ формируются как XOR-суммы элементов диагоналей под углом 22.5° матрицы A :

$$\begin{aligned} \forall 1 \leq i < 2 \cdot (m-1) : r_{4,i} &= \bigoplus_{j=0}^{(i-1) \text{ div } 2} a_{j+1, i-2 \cdot j}, \\ \forall 2 \cdot (m-1) \leq i \leq n : r_{4,i} &= \bigoplus_{j=0}^{m-1} a_{j+1, i-2 \cdot j}, \\ \forall n < i \leq n+2 \cdot (m-1) : r_{4,i} &= \bigoplus_{j=(i-n-1) \text{ div } 2+1}^{m-1} a_{j+1, i-2 \cdot j} \end{aligned} \quad (4)$$

Фрагменты данных четвертого резервного узла $R_5 = \{r_{5,1}, r_{5,2}, \dots, r_{5,n+2 \cdot (m-1)}\}$ формируются как XOR-суммы элементов диагоналей под углом -22.5° матрицы A .

Предположим, что в результате действия определенных причин был утрачен доступ к пяти узлам распределенной системы хранения информации. Предположим, что порядковые номера узлов, к которым утрачен доступ: v, w, u, y, z , причем: $v < w < u < y < z$. Соответственно, задача состоит в восстановлении блоков информации, хранящейся на узлах, доступ к которым утрачен: $B_v = \{a_{v,1}, a_{v,2}, \dots, a_{v,n}\}$, $B_w = \{a_{w,1}, a_{w,2}, \dots, a_{w,n}\}$, $B_y = \{a_{y,1}, a_{y,2}, \dots, a_{y,n}\}$, $B_u = \{a_{u,1}, a_{u,2}, \dots, a_{u,n}\}$ и $B_z = \{a_{z,1}, a_{z,2}, \dots, a_{z,n}\}$.

Предлагаемый способ предполагает восстановление указанных блоков в следующем порядке.

1. Вначале восстанавливаются, с использованием данных четвертого резервного носителя, первые $w-v$ фрагментов $a_{v,1}, a_{v,2}, \dots, a_{v,w-v}$ блока B_v . Эти фрагменты могут быть восстановлены, так как каждый из них принадлежит одной из диагоналей, расположенных под углом 22.5° и никакие другие элементы блоков B_v, B_w, B_y, B_u и B_z не лежат на этих диагоналях.

2. Восстанавливаются, с использованием данных пятого резервного носителя, первые $z-u$ фрагментов $a_{z,1}, a_{z,2}, \dots, a_{z,z-u}$ блока B_z . Эти фрагменты могут быть восстановлены, так как каждый из них принадлежит одной из диагоналей, расположенных под углом -22.5° и никакие другие элементы блоков B_v, B_w, B_y, B_u и B_z не лежат на этих диагоналях.

3. Организуется цикл по переменной j , значение которой последовательно меняется от 1 до n . В цикле выполняются следующие действия:

3.1. Если $j < n-w+v$, то с использованием фрагмента $r_{4,j+w+v-2}$ четвертого резервного носителя восстанавливается фрагмент $a_{v,j+w-v}$ лежащий на $(j+w+v-2)$ -той диагонали, проходящей в матрице под углом 22.5° .

3.2. Если $j < n-z+u$, то с использованием фрагмента $r_{5,j+2 \cdot m-u-z}$ четвертого резервного носителя восстанавливается фрагмент $a_{z,j+z-u}$ лежащий на $(j+2 \cdot m-z-u)$ -той диагонали, проходящей в матрице под углом -22.5° .

3.3. С использованием фрагмента $r_{2,j+w-1}$ второго резервного носителя восстанавливается фрагмент $a_{w,j}$ лежащий на $(j+w-1)$ -той диагонали, проходящей в матрице под углом 45° .

3.4. С использованием фрагмента $r_{3,j+u}$ третьего резервного носителя восстанавливается фрагмент $a_{u,j}$ лежащий на $(j+u)$ -той диагонали, проходящей в матрице под углом -45° .

3.5. Поскольку все j -тые фрагменты блоков B_v, B_w, B_u и B_z уже восстановлены, с использованием фрагмента $r_{1,j}$ первого резервного носителя восстанавливается фрагмент $a_{y,j}$ лежащий в j -том столбце матрицы.

Таким образом, описанный способ позволяет последовательно восстанавливать данные, хранящиеся на любых 5-ти узлах, к которым утрачен доступ. Предложенный способ может быть использован для восстановления данных с большего числа узлов хранения информации путем формирования и использования резервных кодов, формируемых как суммы элементов матрицы, лежащих на диагоналях под углом $11.25^\circ, -11.25^\circ, 5.625^\circ, -5.625^\circ$ и так далее.

Из приведенного описания очевидно, что на первом резервном узле хранится n фрагментов, на втором и третьем - по $n+m-1$, на четвертом и пятом - $n+2 \cdot (m-2)$, на пятом и шестом по $n+3 \cdot (m-1)$ и так далее. На практике, число m узлов хранения информации распределенной системы составляет десятки и сотни, в то время, как количество фрагментов - n данных на каждом из узлов составляет миллионы, то есть $n \gg m$. Поэтому, можно говорить о том, что на каждом резервном узле хранится примерно n фрагментов, то есть столько же, как и на любом из основных.

Из сказанного следует, что для восстановления данных, хранящихся на любых k узлах системы предложенный способ использует только k резервных узлов, что составляет теоретический минимум избыточности.

При потере доступа к резервным узлам снижается способность восстанавливать данные из основных узлов хранения. Так при потере доступа к q из k резервных узлов хранения можно восстановить данных только $k-q$ основных узлов.

Таким образом, по критериям отсутствия ограничений на количество узлов хранения, доступ к которым утрачен и числу используемых резервных узлов хранения предложенный метод превосходит все известные методы решения рассматриваемой задачи.

Доказано, что предложенный метод позволяет решать задачу восстановления данных при потере доступа к любым узлам хранения информации распределенной системы при использовании минимального числа резервных узлов хранения.

Предложенный метод предельно прост и легко может быть реализован программно на любых вычислительных платформах, а также аппаратно с использованием технологий FPGA.

ПЕРЕЧЕНЬ ССЫЛОК

1. Blaum M., Hafner J.I., Hetzler S. Partial MDS Codes and Their Application to RAID Type of Architectures / IEEE Transaction on Information Theory.- Vol.59.- No.7.- 2013.- PP. 4510-4519.
2. W Wesley Peterson and E.J. Weldon Jr. Error-Correcting Codes.- New York: MIT Press, 1984.
3. Abdel-Ghaffar K.A.S., Weber J.H. Parity-check matrices separating erasures from errors. / IEEE Transaction on Information Theory.- Vol.59.- No.6.- 2013.- PP. 3332-3346.
4. Alexandros G.Dimakis, Vinod Prabhakaran, and Kannan Ramchandran. Decentralized Erasure Codes for Distributed Networked Storage. - Berkeley: University of California.-2006. - 176 P.
5. P. Corbett , B. English, A. Goel, T. Gracanac, S. Kleiman, J. Leong, and S. Sankar. Row-diagonal parity for double disk failure. In *Proceedings of the Third USENIX Conference on File and Storage Technologies*, - 2004 - PP.1-14.

*Рецензент: к.т.н., доц. каф. ОТ, КПІ ім. Ігоря Сікорського,
О.П. Марковський*

Метод строгої ідентифікації віддалених користувачів з використанням перетворень на полях Галуа

Захаріудакіс Лефтеріс
старший викладач
навчальний центр ім. Макарія, Кіпр

Олієвський А.А.
студент
КПІ ім. Ігоря Сікорського

Пропонується метод прискорення строгої ідентифікації віддалених користувачів. Запропонований метод реалізує для ідентифікації криптографічно строго концепцію “нульових знань”. Розроблений метод базується на використанні незворотних перетворень на полях Галуа. Детально розроблені процедури ресстрації та циклів ідентифікації. Показано, що запропонована технологія дозволяє прискорити процес ідентифікації при програмній та апаратній реалізації.

Ключові слова: схеми ідентифікації, ідентифікація на основі концепції нульових знань, незворотні перетворення на полях Галуа.

Розвиток хмарних технологій дозволяє надати широкому колу користувачів доступ значного обсягу інформаційних, обчислювальних та програмних ресурсів. Разом з тим, розвиток хмарних технологій породжує ряд проблем, пов'язаних з захистом даних та розподіленням прав доступу до них. В світлі цього, ключового значення набуває проблема ідентифікації віддалених користувачів.

Комерціалізація віддаленого надання ресурсів вимагає суттєвого підвищення надійності ідентифікації. З іншого боку, збільшення кількості користувачів диктує необхідність підвищення швидкості ідентифікації. Вказані чинники зумовлюють актуальність розробки нових методів ідентифікації.

Високий рівень ефективності ідентифікації віддалених абонентів досягається як результат певного компромісу між рівнем захищеності від несанкціонованого доступу та швидкістю ідентифікації.

Теоретично доведено [1], що найбільший рівень надійності ідентифікації захищеності досягається в рамках теоретичної концепції “нульових знань”. Концепція базується на виконанні наступних положень:

- 1) користувач має спеціальний механізм генерації паролів, що змінюються при кожному сеансі ідентифікації;
- 2) система має певний механізм перевірки легітимності сеансового паролю, причому цей механізм не дозволяє системі самостійно генерувати пароль.

Концепція нульових знань передбачає використання теоретично незворотних криптографічних перетворень. Це означає, що існує алгоритм перетворення в прямому напрямку, але принципово неможливим є аналітичне віднаходження алгоритму зворотного перетворення. В існуючих схемах ідентифікації на основі концепції нульових знань для реалізації такого перетворення використовуються задачі теорії чисел, що мають аналітичного розв'язку, зокрема відома задача дискретного логарифмування. На практиці найбільшого поширення набули методи FESIS [2], Guillou-Quisquater [3] та Schnorr [1].

Ці методи реалізації вказаної концепції мають за математичну основу аналітично нерозв'язну задачу дис-

кретного логарифмування. Відповідно, базовими обчислювальними операціями для відомих методів виступають мультиплікативні операції модулярної арифметики, зокрема модулярне експоненціювання над числами, розрядність яких значно перевищує розрядність процесора.

Мета досліджень полягає в створенні ефективного методу ідентифікації віддалених абонентів в рамках концепції нульових знань, що орієнтований на використання алгебри полів Галуа, експоненціювання в якій виконується швидше, ніж в традиційній алгебрі.

Для досягнення поставленої мети досліджено циклічні властивості операції експоненціювання на полях Галуа, утворюючий поліном $M(x)$ яких являє собою поліноміальний добуток двох простих поліномів $d(x)$ та $g(x)$ з різними степенями v та u , відповідно.

Аналогічно тому, як у традиційній алгебрі базовою операцією криптографічних механізмів є модулярне експоненціювання $A^E \bmod M$, в алгебрі скінчених полів Галуа для криптографічних застосувань в якості базової застосовується експоненціювання $A|_E \bmod M$ полях [4]. Ця операція включає в себе дві обчислювальні процедури: поліноміальне множення, або множення без переносів (Multiplication Without Carry - MWC), яке далі позначено символом \otimes , і редукція (позначається як \bmod) – віднаходження залишку при поліноміальному діленні результату множення на утворюючий поліном M поля.

Встановлено та теоретично доведено, що якщо утворюючий поліном поля Галуа $M(x)$ являє собою добуток двох простих поліномів $d(x)$ і $q(x)$: $M = d \otimes q$, то для будь-якого k такого, що $0 < k < h$, де $h=2^u$ справедливо: $(k \otimes d)^h \bmod M = k \otimes d$. Наприклад, якщо $d(x) = x^3 + x + 1$, $v=3$ і $q(x) = x^5 + x^4 + x^3 + x + 1$, $u = 5$, $h = 2^5 = 32$, то $M(x) = d(x) \otimes q(x) = x^8 + x^7 + x^4 + x^2 + 1$. Якщо вибрати довільне k таке, що $0 < k \leq h$, наприклад, $k = 10$, то $k \otimes d = 78$. Тоді $(k \otimes d)^{h+1} \bmod M = (10 \otimes 11)^{32} \bmod 405 = 78$.

Базуючись на встановленій властивості розроблено метод ідентифікації віддалених користувачів, який реалізує концепцію “нульових знань” – теоретично строгої іденти-

фікації. Метод включає в себе дві базові процедури: реєстрації та безпосередньо ідентифікації.

Процедура реєстрації полягає в виконанні такої послідовності дій:

1) Користувач довільним чином вибирає пару простих поліномів $d(x)$ та $q(x)$ з різними степенями: $d(x) = x^v + d_{v-1} \cdot x^{v-1} + \dots + d_1 \cdot x + d_0$ та $q(x) = x^u + q_{u-1} \cdot x^{u-1} + \dots + q_1 \cdot x + q_0$, де $d_0, d_1, \dots, d_{v-1} \in \{0, 1\}$, $q_0, q_1, \dots, q_{u-1} \in \{0, 1\}$.

2) Користувач формує поліном $M(x)$ у вигляді поліноміального добутку вибраних двох простих поліномів $d(x)$ та $q(x)$: $M(x) = d(x) \otimes q(x)$.

3) Користувач вибирає випадкове число $k \in \{1, \dots, 2^w - 1\}$ і виконує обчислення f як експоненті на полі Галуа з утворюючим поліномом $M(x)$: $f = g^k \text{ rem } M$.

4) Користувач шифрує коди M та f відкритим ключем системи та надсилає їх системі. Остання зберігає прийняті коди M та f , котрі є відкритим ключем ідентифікації користувача.

Запропонована процедура сеансу ідентифікації передбачає виконання наступної послідовності дій:

1) Система формує випадкове число r та надсилає його користувачеві.

2) Користувач приймає код r від системи і знаходить таке число n , для якого виконується наступна умова: $r \cdot n \text{ mod } (2^w - 1) = k$.

3) Користувач формує сеансовий пароль у вигляді $p = g^n \text{ rem } M$; отриманий таким чином сеансовий пароль p надсилається системі.

4) Система, отримавши від користувача сеансовий пароль p обчислює $r = p^n \text{ rem } M$. Отриманий результат порівнюється з фіксованим для даного користувача кодом f : якщо ці коди співпадають, тобто $p=f$, то цикл ідентифікації вважається успішним.

Як і для будь-якого механізму криптографічного захисту даних, ефективність запропонованого методу ідентифікації віддалених користувачів оцінюється за двом базовими критеріями: рівнем захисту та часовими характеристиками реалізації.

Для зловмисника задача відтворення коректного паролю зводиться до задачі розкладання відомого поліному $M(x)$ на два простих множника $d(x)$ та $q(x)$ з різними степенями. Вирішення такої задачі шляхом перебору для реальних степенів поліномів виходить далеко на межі технічних можливостей сучасних комп'ютерних систем.

Основна перевага запропонованого способу ідентифікації віддалених абонентів в рамках концепції "нульових знань" полягає в тому, що використання експоненціювання в полях Галуа на відміну від модулярного експоненціювання дозволяє значно прискорити час виконання програм та спростити апаратну реалізацію. В роботі [2] показано, що експоненціювання в полях Галуа виконується на 2-3 порядки швидше у порівнянні з модулярним експоненціюванням, яке є базовою для існуючих методів Guillou-Quisquater [3] та Schnorr [1] реалізації ідентифікації в рамках концепції "нульових знань". Важливою перевагою запропонованого методу є те, що на відміну відомих методів, і, насамперед, методу FESIS, він використовує лише один цикл передачі даних.

ПЕРЕЛІК ПОСИЛАНЬ

1. Schneier B. Applied Cryptography. Protocols, Algorithms and Source codes in C. / B. Schneier - Ed. John Wiley, 1996 - 758 p.
2. Feige U. Zero knowledge proofs of identity / U. Feige., A. Fiat., A. Shamir // Journal of Cryptology.- 1988.- Vol.1, №2.- P.77-94.
3. Bengio S. Secure implementation of identification system / S. Bengio, G. Brassard, Y.G. Desmedt C. Goutier, J.J. Quisquater // Journal of Cryptology.-1991.- Vol.4, №3. - P.186-192.
4. Николайчук Я.М. Коды поля Галуа: теория та застосування. – Тернопіль: ТзОВ "Тернограф", - 2012.- 576 с.

Рецензент: к.т.н., доц. каф. ОТ, КПІ ім. Ігоря Сікорського, О.П. Марковський

Мікроконтролер з системою команд, що розширюється

Сергієнко Анатолій Михайлович
провідний інженер
КПІ ім. Ігоря Сікорського
Україна

Молчанов Олексій Андрійович
аспірант
КПІ ім. Ігоря Сікорського
Україна

Запропонована архітектура восьмирозрядного стекового процесора SM8, який призначений для реалізації у програмованих логічних інтегральних схемах (ПЛІС). Мікроконтролер з цією архітектурою має невеликі апаратні витрати, скорочений об'єм програмного забезпечення а також можливість додавати користувачем до сотні нових команд до його системи команд.

Ключові слова: стековий процесор, Forth, ПЛІС, VHDL.

ВСТУП

При розробці системи на кристалі програмованої логічної інтегральної схеми (ПЛІС) часто виникає потреба організувати обмін даними через такі інтерфейси, як I2C, SPI, Ethernet та інші. Для цього часто використовують RISC-процесори, які керовані операційною системою з відповідним стеком протоколів обміну. Але у багатьох випадках обмін даними виконується за нескладним протоколом і з порівняно невеликою швидкістю, як наприклад, через інтерфейс I2C. При цьому раціональніше використати ядро мікроконтролера, яке має як невеликі апаратні витрати, так і прості процедури програмування та відлагодження. Крім того, такий мікроконтролер може замінити собою керуючий автомат.

Але для реалізації у ПЛІС важливо мати мінімізований об'єм матзабезпечення цього мікроконтролера, оскільки об'єм блоків пам'яті, вбудованих в ПЛІС, є суттєво обмеженим. При цьому бажано мати такий мікроконтролер, систему команд якого програміст власноруч може скоректувати під потреби проекту, для покращення зручності програмування, а також для мінімізації довжини програм.

МІКРОКОНТРОЛЕРИ ДЛЯ РЕАЛІЗАЦІЇ У ПЛІС

Найпоширенішою восьмирозрядною архітектурою мікроконтролера вважається архітектура i8051, яка реалізована у багатьох мікроконтролерних ядрах для ПЛІС. У роботі [1] досліджено особливості проектування його клона FS8051 для ПЛІС. Але навіть з відкиданням периферійних пристроїв та скороченням системи команд апаратні витрати залишаються доволі великими (см. Табл. 2).

Фірмою Xilinx для вказаних потреб пропонується поширене ядро мікроконтролера PicoBlaze. Воно описане на рівні логічних таблиць (LUT) та тригерів ПЛІС цієї фірми, як наприклад, ядро мікроконтролера KCPSM6 [2]. Тому воно має мінімальні апаратні витрати та доволі високу тактову частоту (см. Табл. 2). Хоча цей мікроконтролер обробляє восьмирозрядні дані, він керується 18-розрядними командами, кожна з яких виконується за два такти. Отже, для програмування мікроконтролерів цієї архітектури необхідні програми відносно великого об'єму. Так само, як і для архітектури i8051, для цієї архітектури неможливе додавання нових команд.

Система команд стекового процесора відрізняється тим, що операнди мають неявну адресацію, тому що вони, як правило, розміщуються у небагатьох фіксованих регістрах стекової пам'яті. Тому такі команди мають короткий формат команд. Оскільки ці команди підтримують алгоритми, що активно використовують стекову адресацію, програми, які складені для такого процесора мають мінімізовану довжину [3].

Різними авторами розроблено декілька проектів стекових процесорів, які реалізовано в ПЛІС і які є доступними для відтворення [4-6]. Усі вони мають 16-розрядні команди та обробляють 16-розрядні дані. В роботі [6] показано, що стековий процесор має приблизно у 2,5 рази меншу довжину програми, ніж програма для процесора Xilinx MicroBlaze при реалізації протоколів обміну даними через інтерфейси. Крім того, усі стекові процесори дають змогу збільшити систему команд на кілька команд користувача. При цьому слід внести відповідні зміни до опису процесора на рівні регістрових передач, що часто є незручним для користувача процесора.

Отже, архітектура стекового процесора дає змогу мінімізувати об'єм вбудованого матзабезпечення та одержати мінімізовані апаратні витрати за рахунок реалізації спрощених команд. Крім того, для такої архітектури нескладно розробляти компілятори, тому що, як правило, система команд є підмножиною команд мови Forth. Відомо, що ця мова є зручною для реалізації як граматичного розбору рядків, так і для інтерпретації операторів мов високого рівня. Програма на мові асемблера стекового процесора має такий самий синтаксис, як програма на Forth [3]. Тому доцільно розробити таку архітектуру стекового процесора, яка дає не тільки мінімізовані апаратні витрати, але й спрощену реалізацію команд користувача.

МІКРОКОНТРОЛЕР SM8

Структура розробленого мікроконтролера SM8 показана на рисунку. Цей процесор має поширену двохстекову архітектуру. Він складається з лічильника команд PC, блоку пам'яті даних Data RAM, блоку пам'яті програм Program ROM, регістра команди IR, шифратора команд користувача CDC, стека адрес повернення RStack глибиною 33,

стека даних DStack глибиною 34, арифметико-логічний пристрій ALU та периферійні регістри R0,..., Rn, n < 32. Регістри T, N, P – це крайні регістри стеку Dstack і призначені для зберігання операндів та результату ALU. Program ROM об’ємом до 7936 байт разом з Data RAM об’ємом до 256 байт мають один адресний простір.

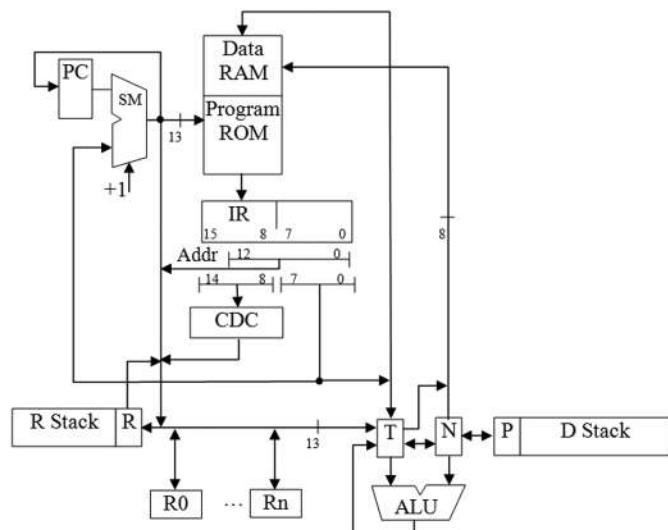


Рис. 1 Структура мікроконтролера SM8

У таблиці 1 представлена система команд мікроконтролера SM8. Усі команди, крім команд CALL, LIT та IF, є однобайтовими. Команди переходу та вводу константи виконуються за два такти, а решта команд – за один. Команди виконуються у конвеєрному режимі. Типові програми на Forth мають багато викликів підпрограм, завдяки чому вони є короткими. За рахунок цього, а також через часте використання команд LIT та IF, середня тривалість виконання однієї команди в даній архітектурі складає півтора такти.

Команди користувача реалізуються наступним чином. Після запису такої команди у регістр IR її код перекодується в CDC у адресу підпрограми. Ця підпрограма викликається з бібліотеки підпрограм команд користувача і виконує усі дії, які необхідні для реалізації даної команди. Перехід до наступної команди програми виконується за командою RET. У цій підпрограмі можна також читати та обробляти поля операндів, які слідує за байтом коду операції. При цьому належним чином коректується адреса повернення у регістрі R.

Таблиця 1. Система команд мікроконтролера SM8

Ім'я	Формат	Дія
CALL	001 Addr	PC+1 -> R, PC = Addr, виклик підпрограми
INR	010 n	Rn -> T, прийом даного
OUTR	011 n	Rn = T, вивід даного
NOP	0000 0000	Немає операції
LIT	0000 0001 B	B -> T, ввід константи
IF	0000 0010 D	PC = PC + D при T = 0, інакше PC = PC + 1
DUP	0000 0100	N = T->
OVER	0000 0101	P = T->
SWAP	0000 1001	X = T, T = N, N = X
@	0000 1010	T = RAM[T], читання пам'яті,
!	0000 1011	DRAM[T] = N, запис в пам'ять, T->
R>	0000 1100	R -> T
>R	0000 1110	T -> R
RET	0000 1101	R -> PC, повернення з підпрограми

DROP	0000 1111	T->, опорожнення стеку
NOT	0001 0000	T = not T
OR	0001 0001	T = T or N
AND	0001 0010	T = T and N
XOR	0001 0011	T = T xor N
ADD	0001 1000	T = T + N
INC	0001 1001	T = T + 1
SUB	0001 1010	T = T - N
DEC	0001 1011	T = T - 1
	Ixxx xxxx	Команда користувача

Команди користувача можуть зберігатись як в Program ROM, так і в Data RAM. Таким чином, в мікроконтролер можна записувати певний динамічний сценарій обробки даних, який записано командами користувача або виконувати розбір рядка, який набраний з таких команд. Наприклад, це може бути рядок з команд десяткового калькулятора.

У таблиці 2 представлені результати синтезу мікроконтролера SM8 у ПЛІС Xilinx Spartan-6 при встановленні параметрів оптимізації за апаратними витратами. Також для порівняння представлені параметри перелічених вище мікроконтролерів, які були синтезовані у таких самих умовах. Аналіз таблиці показує, що мікроконтролер SM8 має найнижчі апаратні витрати у числі LUT і конфігурованих логічних блоків (CLB) та найбільшу продуктивність у мільйонах команд за секунду (MIPS) серед стекових процесорів. Це роз'яснюється тим, що скорочення даних, що обробляються, до восьми розрядів, дало змогу зменшити як апаратні витрати, так і затримку в ALU.

Таблиця 2. ПАРАМЕТРИ ЯДЕР МІКРОКОНТРОЛЕРІВ ПРИ ЇХ КОНФІГУРУВАННІ У ПЛІС XILINX SPARTAN-6

Мікро-контролер	Розрядність команд	Апаратні витрати		Макс. тактова частота, МГц	Продуктивність, MIPS
		LUT	CLB		
FS8051 [1]	8, 16, 24	1293	470	89	30
KCPSM6 [2]	18	87	26	140	70
MSL16 [4]	16	235	61	100	67
b16-small [5]	16	280	73	100	50
J1 [6]	16	342	93	106	70
SM8	8, 16	181	50	140	94

ВИСНОВКИ

Запропонований мікроконтролер SM8 має невеликі апаратні витрати при високій продуктивності та скороченому об'ємі програм і призначений для реалізації нескладних алгоритмів керування, наприклад, для керування обміном даних через інтерфейс I2C. Проект мікроконтролера описано мовою VHDL і він може бути реалізований у ПЛІС будь-якої серії. Програміст має можливість додавати власні команди до його системи команд без зміни його опису. Заплановано, крім асемблера, розробити для цього мікроконтролера компілятор з мови високого рівня для спрощення проектування пристроїв, які реалізують протоколи обміну через такі інтерфейси, як I2C, SPI, Ethernet.

ПЕРЕЛІК ПОСИЛАНЬ

1. Сергієнко А.М., Лепеха В.Л. Деякі особливості проектування мікроконтролерів для СНК // Вісник НТУУ «КПІ», сер. Інформатика, управління та обчислювальна техніка. – Т. 50. – 2009. – С. 70-73.

2. Chapman, K. PicoBlaze for Spartan-6, Virtex-6, and 7-Series (KCPSM6). – Xilinx, Inc. – 2012. —118 p.
3. Koopman P. Stack computers: the new wave. — CA: Ellis Horwood, Mountain View Press. – 1989. – 234 p.
4. Leong P. H. W., Tsang P.K., Lee T.K. A FPGA Based Forth Microprocessor // Proc. of the IEEE Symposium on Field-Programmable Custom Computing Machines (FCCM). – USA, Napa Valley, California. – 1998. — P.1-2.
5. Paysan B. b16-small—Less is More // EuroForth 2004 Proceedings, Jul. 9, 2006, pp. 1-8.
6. Bowman J., Garage W. J1: a small Forth CPU Core for FPGAs // Proc. EuroForth'2010, January, 2010. – P. 1–4.

СИСТЕМИ КЕРУВАННЯ

CONTROL SYSTEMS

Модель підсистеми діагностики транспортного засобу на основі байєсівської мережі

Писаренко Андрій
к.т.н., доцент
КПІ ім. Ігоря Сікорського
Україна, Київ

Тищенко Дмитро
магістрант
КПІ ім. Ігоря Сікорського
Україна, Київ

На основі розробленого інструментарію для пакету MATLAB/Simulink запропоновано динамічну модель діагностики стану транспортного засобу. В основі діагностичної підсистеми лежить байєсівська мережа. Розроблена модель дозволяє виявляти причини несправності транспортного засобу в залежності від показань давачів, що встановлені на ключові вузли автомобіля.

Діагностика відіграє важливе значення у багатьох галузях людської діяльності. Вчасна та точна діагностика у медицині дозволяє рятувати життя. Діагностика технічних засобів допомагає уникнути тяжких наслідків при виході з ладу важливих вузлів та механізмів. Сучасна діагностика широко використовує можливості інтелектуальних систем, наприклад таких як експертні системи чи штучні нейронні мережі [1]. Наприклад, Мусін – експертна система для діагностики інфекційних хвороб крові та рекомендації антибіотиків [2]. Нейронні мережі використовуються для боротьби з раком, зокрема, з раком молочної залози.

Сучасний світ неможливо уявити без транспорту. Необхідність використання транспортних засобів пов'язана зі значною небезпекою, яка супроводжує людину. За даними Всесвітньої організації охорони здоров'я смерть унаслідок ДТП входить в десятку причин, що її викликають. Розвиток електроніки та мікропроцесорної техніки призвів до впровадження моніторингу на транспорті – постійного контролю ключових параметрів транспортних засобів. Впровадження моніторингу переводить організацію транспортних процесів на більш високий якісний рівень, дозволяє збільшити ефективність транспортних засобів та зробити їх використання більш безпечним.

Розрізняють моніторинг стану та моніторинг параметрів, принциповою відмінністю яких є наявність інтерпретатора вимірюваних параметрів у термінах стану. Зазвичай для цього використовуються експертні системи підтримки ухвалення рішень про стан об'єкта та подальше керування.

Проте для прийняття рішень можна використовувати не тільки експертні системи. Байєсівські мережі, як один з інструментів інтелектуального аналізу даних, також з успіхом використовують для діагностики та прийняття рішень.

Байєсівська мережа – ймовірнісна модель, представлена у формі спрямованого ациклічного графа, вершинами якого є змінні досліджуваного процесу [3]. Така графічна модель для представлення ймовірнісних взаємозв'язків на множині змінних, поєднує у собі теорію ймовірності та теорію графів. Байєсівська мережа є ефективним інструментом для вирішення проблем, пов'язаних з невизначеностями та опису складних процесів.

Ефективне застосування байєсівських мереж тісно пов'язано з можливістю їх комп'ютерного моделювання. На теперішній час створено ряд програмних засобів, що дозволяють працювати з байєсівськими мережами.

Аналіз існуючих програмних пакетів, враховуючі їх функціонал та наявність безкоштовної ліцензії, виявив відсутність можливості створення динамічних моделей, що дозволяли б досліджувати поведінку систем у часі. Тому було вирішено поєднати можливості програмних продуктів GeNIe та Bayes Net Toolbox. Обидва продукти, як уже зазначалось у [5], є повністю безкоштовними. GeNIe має зручний графічний інтерфейс та дозволяє користуватись алгоритмами для автоматичної побудови байєсівських мереж, однак отримані мережі неможливо включити до складу більш складних динамічних моделей. У Bayes Net Toolbox відсутній графічний інтерфейс, але є можливість організації взаємодії з іншими модулями пакету MATLAB.

Серед таких модулів найбільший інтерес викликає надбудова MATLAB/Simulink, що використовується для моделювання найрізноманітніших систем та процесів. У ньому є набори інструментів для моделювання аерокосмічних систем, нейронних мереж, нечіткої логіки, систем керування та багатьох інших.

Створення користувацьких блоків для Simulink передбачає написання спеціальних функцій, – так званих S-Function [4]. Крім того, для перетворення моделей, отриманих у GeNIe, до вигляду, що використовується у Bayes Net Toolbox, був написаний окремий скрипт. Завдяки цьому скрипту можна створювати моделі за допомогою графічного інтерфейсу у GeNIe, а потім конвертувати у структуру Bayes Net Toolbox.

S-Function приймає на вхід значення певних вершин та видає ймовірності для інших вершин мережі. Загальний алгоритм запропонованого рішення можна представити наступним чином:

- 1) створення байєсівської мережі у середовищі GeNIe;
- 2) експорт структури та параметрів створеної мережі;
- 3) визначення вхідних та вихідних величин мережі Байєса;

4) побудова динамічної моделі у Simulink на основі S-Function.

Середовище GeNIe зберігає структури і параметри байесівської мережі у форматі XML. На початку моделювання у MATLAB/Simulink, скрипт зчитує вміст XML-файлу, вибирає необхідні параметри та виконує команди для створення мережі у форматі тулбоксу Bayes Net. Потім, у ході моделювання, дані зі входів S-Function додаються у модель у вигляді свідчень (evidence). На основі цих свідчень формується ймовірнісний висновок. Цей висновок, у вигляді числових значень імовірностей необхідних вершин, подається на вихід S-Function.

На рис. 1 наведено структуру байесівської мережі діагностування несправностей автомобіля.

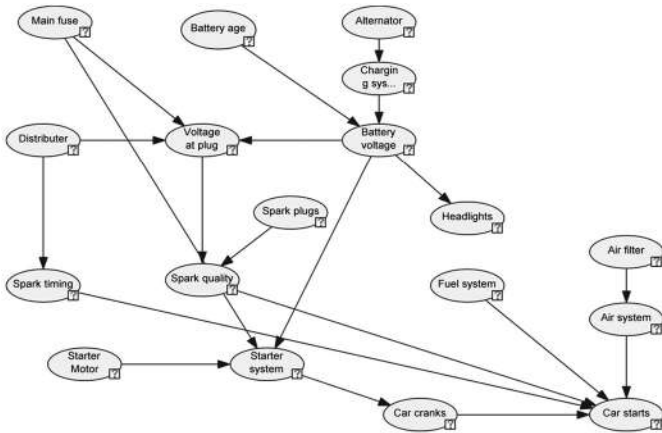


Рис. 1 – Модель Car Diagnosis

Вершини мережі представляють ключові вузли автомобіля або їх параметри. Для кожного вузла визначено перелік станів, у яких він може перебувати, а також імовірність з якою цей вузол перебуває у тому чи іншому стані. Маємо наступні вершини:

- Main fuse – головний запобіжник.
- Alternator – генератор змінного струму.
- Charging system – система зарядки акумулятора.
- Battery voltage – напруга на акумуляторі.
- Battery age – вік акумулятора.
- Headlights – фари.
- Voltage at plug – напруга на свічках запалювання.
- Distributer – розподільювач запалювання.
- Spark timing – час запалювання.
- Spark quality – якість іскри запалювання.
- Spark plugs – правильність встановлення свічки запалювання.
- Starter Motor – стартер.
- Starter system – система старту двигуна.
- Fuel system – паливна система.
- Air filter – повітряний фільтр.
- Air system – повітряна система.
- Car cranks – стартер обертає двигун, але автомобіль ще не завівся.
- Car starts – автомобіль завівся.

У якості вхідних вершин було обрано: Battery voltage, Voltage at plug, Starter Motor, Car cranks та Car starts, оскільки

ці величини підлягають безпосередньому вимірюванню під час функціонування транспортного засобу.

Battery voltage має три стани : strong, weak та dead. Стан strong відповідає значенню напруги від 10 до 12 В, weak від 9 до 10 та dead 0-9 В.

Voltage at plug – strong (20-25 кВ), weak (17-20 кВ) та none (0-17кВ).

Starter Motor має два стани, які визначаються обертами на валу: Okay (200-300 об/хв) та Faulty (0-200 об/хв).

Car cranks – True (400-700 об/хв), та False (0-400 об/хв).

Car starts – True (1500-2000 об/хв), False (0-1500 об/хв).

Модель підсистеми діагностики у MATLAB/Simulink представлена на рис. 2.

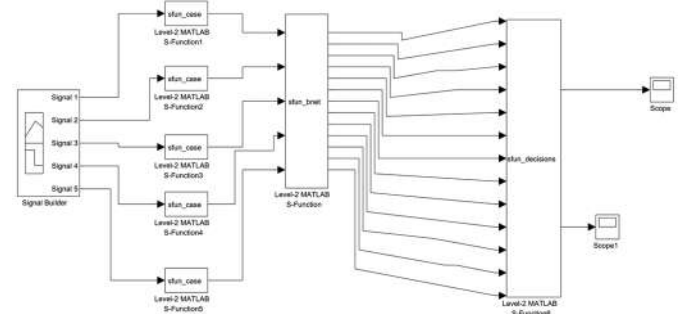


Рис. 2 – Модель підсистеми діагностики транспортного засобу у MATLAB/Simulink

Процес вимірювання значень вхідних параметрів моделюється за допомогою блоку Signal Builder (рис. 3). В подальшому замість нього можна подати сигнали з моделі транспортного засобу або дані, отримані з автомобіля.

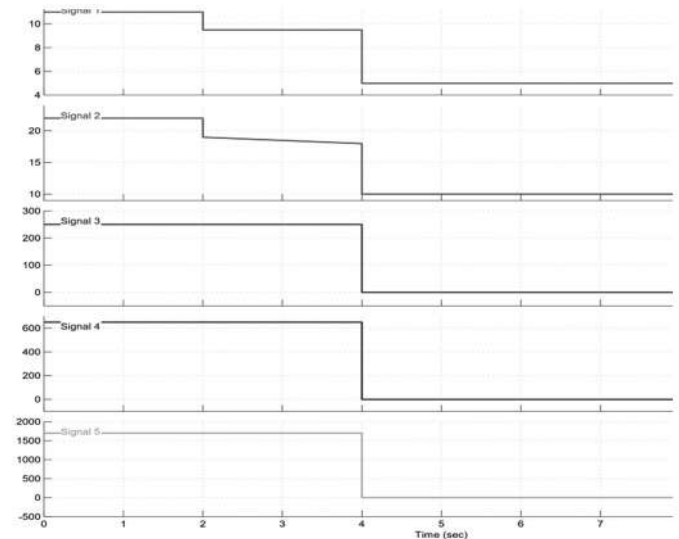
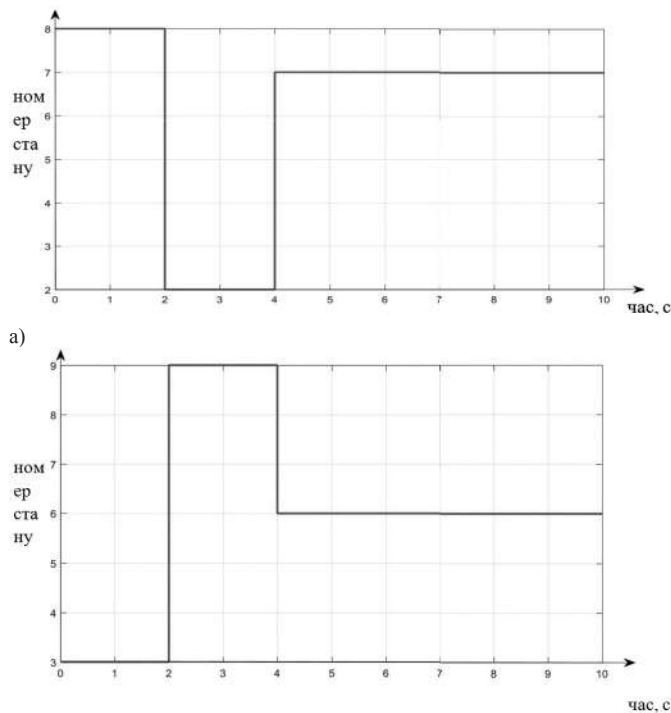


Рис. 3 – Значення вхідних величин. Signal 1 – Battery voltage, Signal 2 – Voltage at plug, Signal 3 – Starter Motor, Signal 4 – Car cranks, Signal 5 – Car starts

Для перетворення числових значень параметрів у номер стану використовуються блоки sfun_case.

Байесівська мережа міститься у блоці sfun_bnet.



б)
Рис. 4 – Результати моделювання

Остаточний висновок формується блоком `sfun_decisions`. Вершини мають два або три стани. Крім того, деякі вершини передбачають градацію станів від «найкращого» до «найгіршого», а інші мають декілька «поганих» станів та один «добрий». Тому у висновку вказуються одразу дві вершини – з максимальною вірогідністю другого (рис. 4а) та останнього стану (рис. 4б). Для деяких вершин другий стан і є останнім, тому вони враховуються у обох порівняннях.

Результатом є номер вершини, яка є найвірогіднішою причиною поломки. У разі, якщо автомобіль заводиться, то імовірнісний висновок вказує на ту вершину, яка з

найбільшою вірогідністю може стати причиною поломки. Наприклад, у період часу 2-4 с за останнім станом найвірогіднішою є вершина під номером 2 (Charging system), а за другим – під номером 9 (Spark quality).

Таким чином, представлена модель підсистеми діагностики на основі мережі Байєса, дозволяє застосовувати її для формування висновку про причину несправності транспортного засобу. Запропоноване рішення дуже гнучке, оскільки дозволяє обирати які величини є вхідними (можуть бути вимірними), а які є вихідними (висновок по причинам несправності автомобіля). Перспективними кроками є включення розробленої підсистеми у модель, що дозволить на основі отриманих результатів впливати на проблемні вузли та підтримувати або повертати (за можливості) працездатність всієї системи.

ПЕРЕЛІК ПОСИЛАНЬ

1. Diagnostic Expert Systems: From Expert's Knowledge to Real-Time Systems [Електронний ресурс] // TMRF e-Book Advanced Knowledge Based Systems: Model, Applications & Research. – 2010. – Режим доступу до ресурсу: <http://www.tmrfindia.org/eseries/ebookv1-c4.pdf>.
2. Wesley A. Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project / Addison Wesley. – MA, 1984. – 754с.
3. Байєсівські мережі в системах підтримки прийняття рішень / М. Згуровський, П. І. Бідюк, О. М. Терентьев, Т. І. Просянкін-Жарова. – Київ: ТОВ «Видавниче Підприємство «Едельвейс», 2015. – 300 с.
4. Write Level-2 MATLAB S-Functions [Електронний ресурс] – Режим доступу до ресурсу: <https://www.mathworks.com/help/simulink/sfg/writing-level-2-matlab-s-functions.html>.
5. Писаренко А. Інструменти побудови динамічних моделей інформаційних систем із застосуванням байєсівських мереж / А. Писаренко, Д. Тищенко. // Winter InfoCom 2016: Матеріали III Міжнародної науково-практичної конференції. – 2016.

Міжнародні нормативні засади людино-машинного інтерфейсу систем управління технологічними процесами

Юрчук Леонід Юрійович
доцент, к.т.н., доцент
КПІ ім. Ігоря Сікорського
м. Київ, Україна

Розглядаються питання розробки людино-машинного інтерфейсу систем управління технологічними процесами відповідно до вимог та рекомендацій міжнародного стандарту ANSI/ISA-101.01-2015 - Human Machine Interfaces for Process Automation Systems

Ключові слова: людино-машинний інтерфейс, життєвий цикл, організації стандартизації, ергономіка

У міжнародній практиці розробкою стандартів займається багато організацій. За звичай це об'єднання офіційних представників держав-учасників, наприклад Міжнародна організація стандартизації (ISO), або об'єднання різних юридичних та фізичних осіб як Міжнародна асоціація автоматизації (ISA). Розроблений стандарт публікується і зацікавлені держави на базі його створюють свою версію або прямим перекладом запроваджують його на своїй території. Можлива і розробка корпоративних стандартів на базі міжнародних. Цей процес особливо актуальний для України, яка відміняє радянські стандарти (випуску до 1992 р.) і хоче створити національну нормативну базу гармонізовану з міжнародною.

Однією з таких організацій є International Society of Automation (скор. ISA, укр. Міжнародна асоціація автоматизації) - глобальна некомерційна асоціація, основним напрямком діяльності якої є розробка стандартів у галузі промислової автоматизації. Заснована у 1947 році, об'єднує більше 30000 учасників, має відділення у 28 країнах [1].

У кінці 20 сторіччя виникли певні проблеми пов'язані з оперативним управлінням складними технологічними процесами. Аналіз причин великих технологічних аварій у США (Three Mile Island – 1979, Piper Alpha Oil Rig – 1988, Texaco Milford Haven Refinery – 1994 тощо) показав, що їх причини лежать у невідповідності людино-машинного інтерфейсу (ЛМІ) функціональним завданням технологічного процесу. Основні причини полягали в тому, що ЛМІ, який історично будувався на принципах приладового інтерфейсу та функціональних схем автоматизації (у західній інтерпретації - P&ID) у сферу комп'ютерної автоматизації не забезпечував як нормальні особисті умови роботи операторів-технологів (в першу чергу підвищене когнітивне навантаження), так і складні алгоритми керування у нестандартних умовах. Відповідно почались роботи по розробці нових підходів до проектування ЛМІ. У 2006 році у ISA був створений технічний комітет ТК 101, який очолює цю роботу. З того часу ним було видано кілька проміжних версій проекту стандарту, які пройшли експертизу та ши-

роке обговорення фахівців і нарешті у 2015 була оприлюднена остання версія вже стандарту - ANSI/ISA-101.01-2015 Human Machine Interfaces for Process Automation Systems - Людино-машинний інтерфейс для систем виробничої автоматизації [2]. При цьому розробники вважають що щоб мати реальну цінність, цей документ не повинен бути статичним, але повинні бути предметом періодичного огляду. З цією метою Товариство вітає всі коментарі і критичні зауваження.

ЛМІ є комплексом технічних засобів і програмного забезпечення, що використовується для моніторингу і взаємодії з системою управління та в кінцевому рахунку з процесом.

Стандарт визначає термінологію і моделі для розробки ЛМІ і робочі процеси, рекомендовані для ефективного підтримки ЛМІ протягом усього життєвого циклу. Має 64 сторінки, 8 рисунків (схем), 8 таблиць і включає 9 розділів:

- Галузь застосування (Scope);
- Нормативні посилання (Normative references);
- Визначення термінів і скорочень (Definition of Terms and Acronyms);
- Система управління ЛМІ (HMI System Management);
- Людський фактор - інжиніринг і ергономіка (Human Factors Engineering & Ergonomics);
- Стили відеокадрів (дисплеїв) і структура ЛМІ у цілому (Display Styles and Overall HMI Structure);
- Взаємодія з користувачем (User Interaction);
- Виконання (Performance);
- Навчання (Training).

Як видно перші три розділи є вступними за своїм характером. Розділ 4 представляє модель життєвого циклу для ЛМІ. Розділи 5-9 надають додаткові деталі для підтримки життєвого циклу у вигляді обов'язкових вимог і необов'язкових рекомендацій.

Мета стандарту полягає у вирішенні концепції (philosophy) ЛМІ, його проектування, реалізації, експлуатації та технічного обслуговування для систем автоматизації технологічних протягом усього життєвого циклу.

Розробники вважають, що якщо слідувати стандарту, рекомендованим практикам та методології, результат повинен дозволити користувачам більш ефективно працювати, забезпечуючи підвищену безпеку, якість, продуктивність і надійність.

Цільовою аудиторією є кінцеві користувачі, дизайнери, розробники і виконавці систем людино-машинного інтерфейсу. Проте коло зацікавлених осіб цим не обмежується - є ще менеджери різного рівня, технологи, фахівці з виробничої безпеки, технічного обслуговування, системи контролю якості тощо.

У першому розділі «Галузь застосування» визначається що рекомендації цього документу, можуть бути застосовані до будь-якого процесу для взаємодії з системою управління за допомогою ЛМІ, включаючи, але не обмежуючись безперервними, порційними, а також дискретними процесами.

Другий розділ «Нормативні посилання» включає посилання на 10 стандартів ANSI/ISA з 2007 року та три стандарти ISO присвячені розробці ЛМІ та пов'язаним питанням.

Можна відмітити що в Україні є діючим один рідний стандарт ДСТУ 2429 - 94 «Система «людина - машина». Ергономічні та техніко-естетичні вимоги. Терміни та визначення.» та ще 4 стандарти радянських часів 80 років які відмінюються.

Третій розділ присвячено визначенню термінів та скорочень. Розділ є дуже важливим тому що дає нормативне тлумачення 46 термінам та 16 аббревіатур. І хоча ця термінологія в основному знайома і використовується фахівцями тим не менш її нормативне закріплення має певний методичний зиск. На рис.1 представлено графічне відображення деяких термінів, пов'язаних з робочим місцем оператора.

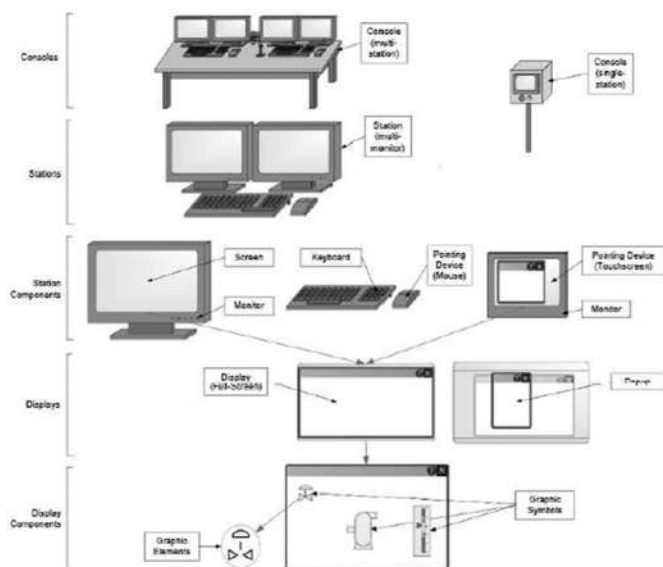


Рис. 1 – Графічне відображення деяких термінів

Четвертий розділ «Система управління ЛМІ (HMI System Management)» (16 стор.) є основним і базується

на моделі життєвого циклу системи що включає 5 стадій (рис.2):

- Стандарти системи (System Standards) (Концептуальне проектування) – розробка основних документів, які встановлюють основу для всіх проектних рішень ЛМІ.

- Проектування (The Design Process) - всі апаратні і програмні аспекти проектування ЛМІ (огляд).

- Впровадження (The Implementation Stage of the HMI Lifecycle) - створення ЛМІ в цільовій платформі і апаратному забезпеченні від розробки до впровадження в експлуатацію (через випробування, тренування, комісії і кваліфікацію).

- Експлуатація (The Operate Stage of the HMI Lifecycle) – розглядає процеси експлуатації, технічного обслуговування, зняття з експлуатації

- Стадія «Безперервна підтримки змін (Continuous Work Processes)» - управління змінами, аудит та валідація – працює практично на всіх стадіях життєвого циклу забезпечуючи можливість своєчасного внесення змін та контролю їх ефективності.

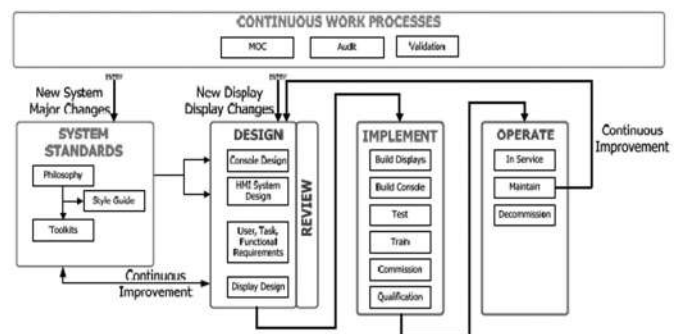


Рис. 2 - Життєвий цикл HMI

На стадії системної стандартизації розробляються ключові аспекти майбутнього інтерфейсу:

- концепція (філософія) ЛМІ;
- керівництво по стилю ЛМІ;
- набір інструментів розробки ЛМІ (документація).

Концепція є фактично тим стратегічним документом, в якому розглядаються основні принципи, що регулюють проектування структури ЛМІ для того, щоб забезпечити високу ефективність та безпечність управління технологічним процесом. Крім врахування формальних показників таких, як експлуатаційні вимоги, стандарти проектування, методи розробки тощо, велика увага приділяється врахуванню зручності роботи оператора.

Керівництво по стилю забезпечує корпоративні вимоги до розробки мнемосхем (відеокадрів, дисплеїв) автоматизованого робочого місця оператора. Воно повинно відповідати та розвивати керівні принципи концепції, базуватись на специфікаціях виробництва, включати загальні правила проектування для мнемосхем (дисплеїв) і пов'язаних з ними програм, а також забезпечити керівництво зі створення сценаріїв, вбудовано логіки, використання кольору тощо.

Інструментарій. Визначаються засоби з допомогою яких буде розроблено мнемосхеми відповідно до закладених у попередніх розділах принципів побудови інтерфейсу. Засоби повинні включати сучасні підходи до проектування, бібліотеки шаблонів, статичних і динамічних графічних символів тощо.

Аналогічно представляються наступні стадії з детальним розглядом вимог та рекомендацій для кожного етапу.

П'ятий розділ «Людський фактор - інжиніринг і ергономіка» (Human Factors Engineering & Ergonomics) (6 стор.) - присвячено розгляду принципів побудови ЛМІ з урахуванням когнітивних і сенсорних можливостей і обмежень користувачів-операторів.

При проектуванні ЛМІ повинні враховуватися загальні принципи:

- бути ефективним інструментом для безпечного і ефективного управління процесом;
- допомагати в ранньому виявленні, діагностиці та правильній реакції на нештатні ситуації;
- допомагати операторам визначати пріоритети реагування на основні або декілька одночасних системних збоїв;
- відеокадру (дисплея) або елементів на ньому відразу очевидні для оператора;
- засновані на вимогах завдань і потреб оператора і мати пріоритет над інтересами всіх інших користувачів.

Так як існують різні користувачі ЛМІ, що мають різні особисті та функціональні потреби, то кожен з їхніх потреб слід враховувати при проектуванні

Велика увага приділяється ергономічним засадам взаємодії людини з технічними засобами. Наприклад, надаються рекомендації з використання колірної гамми:

- сірі фони використовуються, щоб звести до мінімуму відблиски і забезпечують низький контрастне зображення;
- яскраві кольори повинні використовуватися тільки для виділення сигналів тривоги і позаштатних ситуацій;
- кольори, які використовуються для сигналізації не повинні використовуватися в інших місцях (для позначення технологічного обладнання, трубопроводів тощо).

Наступний розділ «Стилі відеокадрів (дисплеїв) і структура ЛМІ у цілому (Display Styles and Overall HMI Structure)» (7 стор.).

Відеокадри (дисплеї) є основними будівельними блоками, які створюють загальну структуру ЛМІ. Стиль відображення показує як буде представлена інформація на відеокадрі. При виборі стилю необхідно враховувати загальну концепцію ЛМІ, функціональні вимоги до відеокадру, технічні можливості моніторів, психофізіологічні характеристики операторів тощо.

Відповідно до загальної структури ЛМІ приділена увага ієрархії відеокадрів - рекомендується не більше чотирьох рівнів (але може бути менше або більше рівнів).

Перший рівень забезпечує огляд всієї зони відповідальності оператора.

Другий рівень як правило є головним керуючим інтерфейсом оператора забезпечуючи видимість стану процесу, сигналізацію нестандартних та аварійних ситуацій тощо, має особливо високий пріоритет.

Рівень три складають докладні відеокадри, що використовуються для виконання нестандартних операцій, забезпечують видимість всіх нестандартних ситуацій (тривог).

Рівень чотири використовується рідко – в основному для цілей діагностики і усунення неполадок.

Розділ «Взаємодія з користувачем (User Interaction)» (12 стор.) присвячений методам програмного забезпечення та апаратних засобів, що дозволяють користувачам взаємодіяти з ЛМІ.

Розглядаються методи взаємодії з користувачем - введення і переміщення даних, формування цифрових та текстових повідомлень, навігації, запобігання помилок, забезпечення безпеки доступу, використання спливаючих і інших вікон тощо.

Апаратне забезпечення – монітори, кнопки, пристрої введення інформації тощо.

У розділі «Виконання (Performance)» (3 стор.) розглядаються технічні питання реалізації ЛМІ – необхідність визначення оптимальних параметрів взаємодії людини та техніки – час реакції людини та системи, час оновлення інформації та її запису, розподіленість системи, реалізацію на базі SCADA-системи тощо.

Останній розділ «Навчання (Training)» розглядає питання навчання різних категорій користувачів – визначає матеріальну та інформаційну базу, основні завдання тощо. Питання підготовки операторів також розглядаються у інших розділах.

За загальною характеристикою фахівців цей стандарт є дорожньою картою для створення високоефективного ЛМІ.

ПЕРЕЛІК ПОСИЛАНЬ

1. Міжнародна асоціація автоматизації (ISA). [Електронний ресурс] – Режим доступу до ресурсу: <https://www.isa.org/>.
2. Стандарт ANSI/ISA-101.01-2015 Human Machine Interfaces for Process Automation Systems [Електронний ресурс]. – 2015. – Режим доступу до ресурсу: <https://www.isa.org/store/ansi/isa-10101-2015,-human-machine-interfaces-for-process-automation-systems/41921133>.

Компоненти систем управління з розширенням реального часу

Николайчук Микола Ярославич
доцент

Івано-Франківський національний технічний університет нафти і газу
Україна, Івано-Франківськ

Комп'ютеризовані системи керування процесами реального часу (Run-Time Extended) базуються на сучасних інформаційних технологіях, операційних системах, апаратно-програмних засобах, комунікаціях і забезпечують детерміновані інтервали часу на збір, обробку і передавання даних в процесі керування технологічними об'єктами.

При цьому застосовуються WEB-орієнтовані технології побудови систем управління на базі PLC та SCADA-систем з елементами віртуалізації і тунелювання каналів промислового зв'язку.

Враховуючи сучасні вимоги до систем управління технологічними процесами з високою динамікою інформаційних процесів, системи реального часу в повній мірі їх забезпечують.

Ключові слова: Програмний PLC, операційна система, розширення реального часу, управління в реальному часі, SCADA-система.

Розширення IntervalZero RTX корпорації «IntervalZero, Inc.» [1] - програмний пакет, призначений для інтеграції функціональності «жорсткого» реального часу в системи під управлінням операційних систем Microsoft Windows. Програмний продукт RTX був з успіхом випробуваний в тисячах різних автоматизованих систем управління, роботах та інш. Це дозволило добитися підвищення їх ефективності, можливостей, ступеня масштабованості і надійності функціонування при одночасному скороченні термінів і вартості розробки нової продукції.

IntervalZero RTX є справжнім розширенням OS без інкапсуляції Windows і будь-якої взаємодії з інфраструктурою для Windows або її зміни [2]. Завдяки подібному поділу, додатки підсистеми реального часу RTX (real-time subsystem, RTSS) будуть виконуватися навіть при повному збої для Windows.

Розширення RTX IntervalZero лежить в основі нового підходу до проектування систем реального часу, що отримав назву Архітектура Soft-Control (архітектура програмного управління). Його застосування здатне змінити принципи конкуренції в таких областях, як виробництво засобів промислової автоматизації, медичних систем, контрольно-виміральної техніки та інших систем завдяки підвищенню вимог до можливостей і точності управління.

Основа архітектури програмного регулювання для систем «жорсткого» реального часу - інтегрована інструментальна платформа на базі багатоядерної архітектури x86, що працює під управлінням операційної системи Microsoft Windows і розширення IntervalZero RTX. Soft-Control Architecture забезпечує OEM-виробникам значну конкурентну перевагу. Soft-Control Architecture дозволяє виробникам відмовитися від застосування спеціалізованих апаратних засобів на основі PLC або додаткових пристроїв управління на користь програмних систем. Використання сучасних платформ спільно з RTX - економічне і багатофункціональне рішення для побудови систем реального часу на базі Windows, з архітектурою x86.

Система INtime є розширенням реального часу Windows, яке було розроблено корпорацією «Radisyс

Corporation», а в даний час підтримується корпорацією «TenAsys» [3].

INtime комбінує можливості OS реального часу зі стандартними OS Windows, включаючи версії Embedded, без додаткових апаратних засобів. INtime спеціально розроблена під архітектуру процесора x86. Додатки реального часу і не реального часу виконуються на різних віртуальних машинах на єдиному комп'ютері (рис. 1).

INtime, на відміну від RTX, слабо пов'язана з NT. Архітектура INtime базується на механізмі апаратного обслуговування задач (hardware tasking), яке забезпечується процесором Intel. В результаті, два ядра виконуються на одних апаратних засобах. Враховуючи, що вони розділяють одні апаратні засоби, стали потрібними деякі модифікації NT HAL. Такий підхід дозволяє захистити і відокремити середовище виконання і область пам'яті від Windows. Всередині INtime кожний процес додатку має свій власний адресний простір. Крім того, ядро і додатки виконуються на різних пріоритетних рівнях, що дозволяє захистити їх одне від одного.

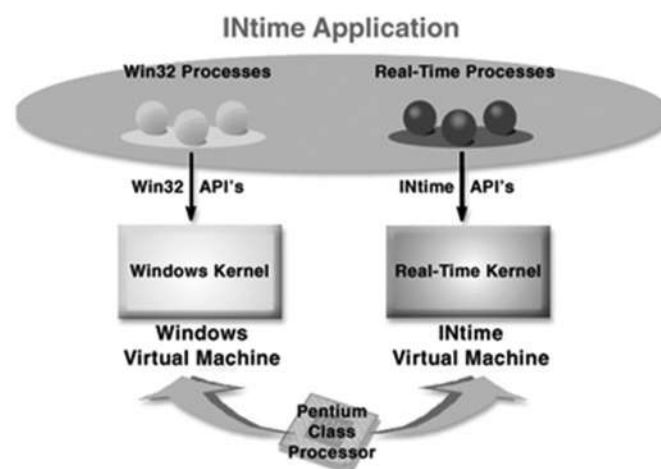


Рис. 1 – Архітектура розширення реального часу INtime

INtime показує передбачувану поведінку, але його складна архітектура не дозволяє системі досягнути значної продуктивності. Внаслідок сегментаційних обмежень INtime підходить не для всіх систем реального часу.

Системи управління реального часу на базі PC WORX RT BASIC SOFTWARE PLC «Phoenix Contact» є програмним PLC з вбудованими шинами PROFINET, Modbus і підтримкою INTERBUS, на основі промислового або персонального комп'ютера [4].

Simatic WinAC (Windows Automation Center) - це програмне забезпечення з підтримкою функцій S7-сумісних PLC в середовищі операційних систем Windows [5]. Програмне забезпечення WinAC є додатком Windows і може використовуватися на комп'ютері оператора паралельно з іншими програмами. Ця особливість дозволяє одночасне вирішення завдань автоматизованого управління, візуалізації, комп'ютерної обробки та архівування даних, організації промислового зв'язку та інш.

Після запуску програми відкривається панель параметрів CPU WinLC, як показано на рис. 2 - ця панель використовується для управління роботою WinLC.

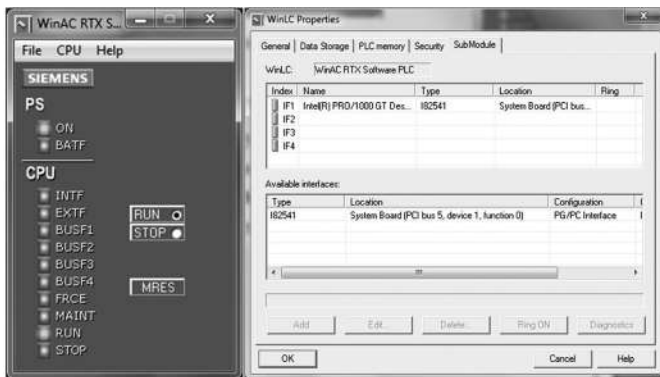


Рис. 2 - Вікно налаштувань програмного контролера WinAC RTX

Панель налаштування призначена для параметрування та перевірки роботи WinLC. Параметрування відповідних значень забезпечує продуктивність та інші параметри контролера.

Крім того, WinLC RTX має розширену панель параметрів, як показано на рис. 3.

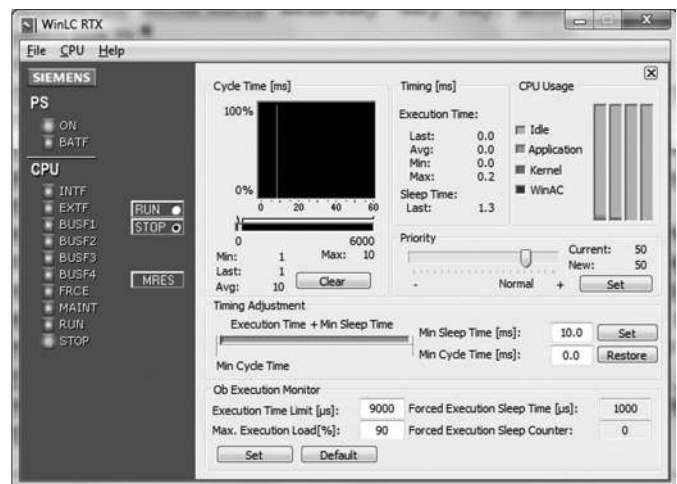


Рис. 3 – Розширена панель параметрів WinLC RTX

Таким чином, до основних компонентів систем керування реального часу відносяться:

- операційні системи (OS) з розширенням реального часу (RTX – Run Time Extension);
- PLC реального часу (RTX PLC);
- SCADA-системи;
- комунікаційне обладнання для забезпечення режимів реального часу.

Розширення реального часу модифікують ядро операційної системи і забезпечують детерміновані інтервали часу обробки запитів. В системах керування технологічними процесами реального часу найбільш поширеними є RTX для OS Windows.

ПЕРЕЛІК ПОСИЛАНЬ

1. IntervalZero RTX Quick Start Guide (Document Number DOC-RTX-912-010). IntervalZero, June 2010. - 28 p.
2. Hard Real-Time with IntervalZero RTX® on the Windows® Platform (DOC-RTX-004). IntervalZero, January 2010. - 19 p.
3. INtime® 4.0 Software. User's Manual (31001-6). - TenAsys Corporation, September 2009. – 213 p.
4. Installing and operating the PC WORX RT BASIC software PLC with Valueline industrial PCs and PROFINET. User manual (8047_en_03). - PHOENIX CONTACT, 2016-04-14. – 84 p.
5. Windows Automation Center RTX WinAC RTX (F) 2010. Operating Instructions (A5E00486536-04). – Siemens, 07/2010 - 296 p.

Система керування механізмом видалення металевих включень в умовах виробництва керамічної цегли

Заміховський Леонід Михайлович
завідувач кафедри

Івано-Франківський національний технічний університет
нафти і газу
Україна, Івано-Франківськ

Левицький Іван Теодорович
асистент

Івано-Франківський національний технічний університет
нафти і газу
Україна, Івано-Франківськ

Наведено структурну схему системи керування механізмом видалення металевих включень у умовах виробництва керамічної цегли на основі конвеєрної лінії і системи виявлення металевих включень. Розробка включає блок контролю металевих включень, апаратно-програмні засоби на базі PLC Simatic S7-1200, систему передавання даних, алгоритм видалення металовключень із вхідної сировини.

Ключові слова: Система керування, виконавчі пристрої, вимірювання, металеві включення, системи передавання даних, конвеєрна лінія.

Сучасний процес виготовлення керамічної цегли відрізняється високим рівнем складності, технологічний процес якого включає: підготовку суміші, формовку, сушку, випалювання. Найбільш відповідальним і впливовим на якість вихідної продукції є процес підготовки суміші. Саме на даному етапі і до нього є велика ймовірність потрапляння у сировину сторонніх металевих предметів, які не тільки впливають на якість вихідної продукції, але і стають причиною виходу з ладу основного технологічного обладнання [1,2].

Одним із основних методів інтенсифікації виробництва в керамічній промисловості є створення і впровадження великих технологічних агрегатів і комплексів агрегатів з форсованими режимами технологічних процесів. Важливою задачею є забезпечення високої продуктивності теплових агрегатів, зниження енергетичних затрат при високій якості готової продукції. Цього не можливо добитися за наявності значної кількості і тривалості простоїв, що виникають в результаті потрапляння по сторонніх металевих предметів у сировину. Саме тому наявність автоматизованої системи виявлення і видалення сторонніх предметів є надзвичайно актуальною задачею. Було розроблено метод скануючого сигналу, який базується на формуванні скануючого сигналу з рухомим максимумом амплітуди напруженості магнітного поля «дзвоноподібної» форми в межах ширини конвеєрної лінії з наступною обробкою прийнятих сигналів, які ідентифікують наявність, локалізацію і розміщення металевих включень [3].

З метою подальшої розробки структури і технічної реалізації системи контролю металевих включень у сировині для виготовлення керамічних виробів, беручи до уваги досвід експлуатації конвеєрних ліній на цегельному заводі ТзОВ «Керамікбудсервіс», було сформульовано наступні вимоги щодо даної системи:

- неперервний контроль на рухомій конвеєрній стрічці;

- чутливість щодо виявлення металевих включень розміром більше 5 мм;
- інтеграція в існуючу систему автоматики промислового виробництва;
- цифрова обробка сигналу з метою зменшення впливу завад і збільшення чутливості;
- відбраковка сировини (видалення металевих включень);
- підтвердження відбраковки сировини із вмістом металевих включень;
- забезпечення захисту від промислових завад;
- робота чутливих елементів поряд із металевими конструкціями;
- можливість визначення геометрії металевих включень;
- зменшення «ефекту продукту» (вплив самої сировини на чутливість чи результати контролю);
- наявність захисної сигналізації.

На рис. 1 представлено схему конвеєрної лінії. Відвантажувальний бункер подає сировину на конвеєр №1, а згодом сировина потрапляє на конвеєр №2, де відбувається контроль наявності у сировині металевих включень. У випадку відсутності у сировині металевих включень, сировина потрапляє у приймальний бункер. А у випадку наявності у сировині металевих включень, конвеєр №1 зупиняється, а конвеєр №2 вмикається системою у реверс. Тим самим сировина, що містить металеві включення потрапляє у ємність для забрудненої сировини. Після завершення процесу видалення забрудненої сировини конвеєр №2 вмикається у нормальний режим роботи, що аналогічно відбувається з конвеєром №1. Час на який конвеєр №2 вмикається у реверс повинен розраховуватись системою і залежить від величини і розмірів металевих включень що знаходяться у сировині.



Рис. 1 – Схема конвеєрної лінії і принципу роботи відбраковування

При вмиканні конвеєрів у нормальний режим, після процесу видалення металевих включень, відбувається повторний контроль наявності металевих включень. У випадку їх наявності, процес видалення повторюється. Але уже при повторній наявності металевих включень, вмикається захисна сигналізація і обидва конвеєри зупиняються. Такою причиною зупинки конвеєрів може спричинити наявність металевих включень значних габаритних розмірів (значної протяжності – металевий дріт, прут, арматура), що потребує втручання обслуговуючого персоналу.

Як допоміжний елемент можна використати в якості пристрою відбраковування потужного електромагніту. Його робота функція полягає в тому, що при виявленні у сировині металевих включень, у розрахований центральним процесором час (в момент часу коли металеві включення будуть знаходитись під магнітом) вмикається електромагніт на короткий час на максимальну потужність, а згодом потужність електромагніту знижується до мінімальної, необхідної для підтримки на осерді електромагніту металевих включень. Таке рішення має ряд позитивних і негативних сторін. До позитивних слід віднести те, що за таким методом видалення металевих включень конвеєр не вмикається у реверс і тим самим не має пере розходу сировини, адже вона залишається на конвеєрі. До негативних моментів слід віднести те, що існують випадки, коли металевими включеннями є предмети значної ваги або які знаходяться під значним шаром сировини, тоді електромагніт їх видалити не здатен. Окрім цього при такому методі не має змоги здійснити підтвердження відбраковування. Як висновок, даний метод відбраковування потребує доопрацювання і вдосконалення.

Блок підтвердження відбраковування використовується як пристрій що підтверджує видалення із сировини металевих включень блоком відбраковуванням. Для цього необхідно провести повторно контроль наявності металевих включень, а тому необхідно або монтувати ще один чутливий елемент, що збільшує вартість самої системи. Пропонується у якості блока підтвердження відбраковування повторно використати чутливий елемент, адже конвеєр під час процесу відбраковування здійснює реверс, а згодом відновлює свій напрям руху, тим самим сировина знову потрапляє під чутливий елемент (рис.1). Чутливий елемент знову має можливість перевірити наявність металевих включень у сировині, тим самим підтвердити факт видалення металевих включень. Таким чином здійснюється підтвердження відбраковування. Такий

принцип роботи дає перевагу у відсутності необхідності встановлення додаткового чутливого елемента і в результаті зменшує як габарити так і кінцеву вартість всієї системи.

Для забезпечення безупинного виробництва узгодження усіх технологічних і виробничих операцій дану систему інтегровано в діючу систему управління виробництва керамічної цегли, рис.2.

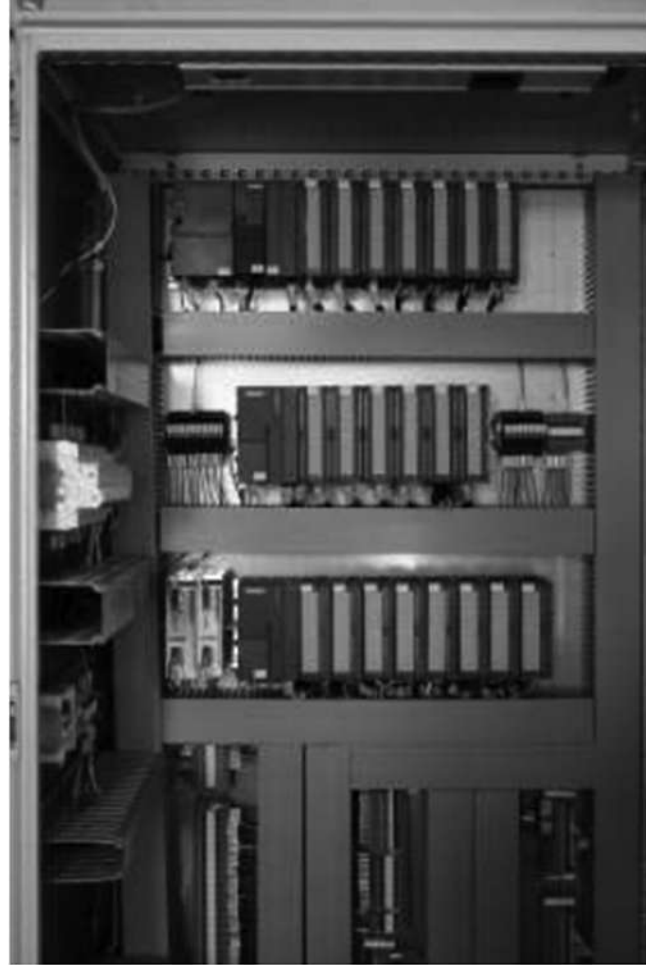


Рис. 2 – Шафа управління і автоматики

У підсумку можна зазначити, що в результаті впровадження система дозволить форсувати технологічний процес виготовлення керамічної цегли, зменшити енергозатрати на її виготовлення а в результаті і собівартість готової продукції.

ПЕРЕЛІК ПОСИЛАНЬ

1. Левицький І.Т. Сучасний стан контролю складу сировини (глини) для керамічної промисловості // «Наукові вісті» Інституту менеджменту та економіки «Галицька академія» №1(12) – 2007. – с. 101-104.
2. Вороб'єв В.А. Строительные материалы. М.: Стройиздат, 1971.
3. Патент 106959 України, МПК В07В15/00. Спосіб локалізації та ідентифікації металовключень у сировині на стрічковому конвеєрі та система для його реалізації [Текст] / Заміховський Л. М., Євчук О. В., Левицький І. Т. – заявник і патентовласник. – № а201315542 ; заявл. 30.12.13 ; опубл. 27.10.14, Бюл. № 20.

Проблеми управління групою автономних рухомих об'єктів у 3D просторі

Долина Віктор Георгійович
доцент, к.т.н.
КПІ ім. Ігоря Сікорського
Україна, Київ

Пріліпухов Євгеній Вадимович
студент
КПІ ім. Ігоря Сікорського
Україна, Київ

В роботі розглядається проблема керування групою безпілотних апаратів. Описані тенденції розвитку безпілотних систем. Розглянуті два методи вирішення проблем з керуванням об'єктів у групах, а саме мультиагентний метод і метод ройового інтелекту.

Ключові слова: *безпілотні системи, БПЛА, ройовий інтелект, мультиагентний метод, мікрокомп'ютер.*

В сучасному світі безпілотні літальні апарати (БПЛА) користуються все більшою популярністю в якості малих, легких і недорогих інструментів для доставки вантажів, наукових цілей, геологічної розвідки, повітряних зйомок, військової та правоохоронної справи. До недавнього часу БПЛА використовувались переважно у військовій справі, але з плином часу і при швидкому розвитку техніки постають виникають нові можливості та з'являються різноманітні задачі цивільного призначення. У всьому світі ведуться і впроваджуються розробки з використанням БПЛА в області комерційних вантажоперевезень, сільськогосподарській діяльності, у пошуково-рятувальних і різних моніторингових задачах[1]. Однією із головних переваг використання БПЛА є відсутність людського фактору при виконанні поставленої задачі, який може зіграти негативну роль в надзвичайних ситуаціях.

На теперішній час для виконання поставлених задач досить розповсюджене використання не одного БПЛА, а декількох, які утворюють певну групу в якій окремі БПЛА виконують власні, проте, схожі між собою, задачі. Більшість сучасних систем управління групою БПЛА характеризуються відсутністю автономного прийняття нових, складних задач, що дозволяють групі БПЛА оперативно приймати ефективні рішення по змінній сценарію виконання поставлених задач. Типовими прикладами подій, що визивають необхідність в постановці нових задач є: поява нової вигідної інформації, для більш ефективного виконання задач, вихід зі строю частини деталей або ресурсів, а також зміна критеріїв прийняття рішень. Чим більша неоднозначність, більш розподілений характер процесів прийняття рішень і чим частіше трапляються незаплановані події, тим нижча ефективність сучасних систем, які не можуть самостійно приймати рішення і в автоматичному режимі перебудовуватися під зміни навколишнього середовища. Крім того, бідь-яка модифікація схем прийняття рішень в традиційних системах представляє собою дуже складний в розумінні і виконанні процес, що потребує високої кваліфікації розробників[2], що, в свою чергу, суттєво збільшує вартість розробки і експлуатації таких систем.

На теперішній час для вирішення проблем управління групою безпілотних рухомих об'єктів (БРО) існує два гра-

ничних підходи: мультиагентний метод і метод ройового інтелекту. Між цими двома підходами існує поле можливих їхніх комбінацій, які необхідно досліджувати і визначати оптимальні з урахуванням вирішуваних задач. БПЛА є різновидом БРО і всі ці підходи також можуть бути застосовані для керування групою БПЛА.

В основі мультиагентного методу лежить поняття «агента», програмного об'єкта, розташованого на кожному окремому БПЛА, який за допомогою датчиків дозволяє сприймати ситуацію, приймати рішення і взаємодіяти з подібними системами. Характерними особливостями технології є:

- здатність до колективної, цілеспрямованої поведінки в інтересах вирішення загальної задачі;
- автономність, тобто здатність самостійно вирішувати локальні задачі;
- здатність до активних дій заради досягнення загальних або локальних цілей;
- адаптивність, тобто можливість автоматично пристосовуватися до динамічно-змінюваного середовища;
- можливість активно переміщуватися, цілеспрямовано шукати і знаходити інформацію, енергію, об'єкти і ресурси які необхідні для колективного вирішення задачі[3].

Ці можливості кардинально відрізняють мультиагентні системи від інших існуючих для управління групою незалежних, автономних об'єктів.

Таким чином, для створення мультиагентної системи для групи БПЛА кожен БПЛА має бути обладнаний наступними приладами:

- невеликий за масою, але потужний мікрокомп'ютер для роботи в реальному часі;
- датчики для визначення поточного стану БПЛА;
- датчики для визначення положення БПЛА та дослідження оточуючого простору;
- засоби зв'язку для організації впевненого обміну даними «агентів» між собою;
- виконавчі пристрої та джерело живлення.

На мікрокомп'ютер одного «агента» серед групи БПЛА покладається вирішення таких задач:

- визначення точного місцезнаходження;

- визначення ступеня виконання задач;
- обмін даними з іншими «агентами» системи;
- порівняння своїх даних з поставленою задачею;
- корегування власних дій за прийнятими даними;
- управління виконавчими пристроями.

Мікрокомп'ютер у БПЛА відповідає за функцію низького та високого рівня управління. Низькорівнева система керування відповідає за роботу давачів і виконавчих пристроїв. Система керування високого рівня відповідає за стійкий рух, вирівнювання апарату, направлення руху, корегування та підтримання потрібного курсу, а також, в цілому, за виконання своєї задачі. Керування людиною обмежується постановкою глобальних задач[4].

Іншим підходом вирішення проблем управління групою безпілотних об'єктів є метод ройового інтелекту.

Ройовий інтелект – це аналог природного поведіння тварин у великих децентралізованих групах, в яких керування відбувається на принципах самоорганізації.

В групі безпілотних апаратів, які взаємодіють на основі ройового інтелекту, кожен апарат здійснює взаємодію лише з декількома, найближчими до нього в даний момент, апаратами. При цьому дальність зв'язку та енерговитрати на інформаційну передачу відносно невеликі. Безпілотні апарати приймають рішення щодо поточної поведінки, спираючись на самостійно зібрані дані про навколишнє середовище, а також на дані сусідніх апаратів. Енерговитратний зв'язок з центральним пристроєм керування відбувається дуже рідко, і не зі всіма апаратами групи, і лише для того, щоб отримати інформацію по задачах, які стоять перед групою та для передачі інформації про стан групи, хід виконання задачі та інших даних.

Методи ройової взаємодії вже знаходять своє використання в задачах управління великими групами наземних мобільних роботів (Swarm-bots), що може слугувати підтвердженням того, що використання цього методу для груп БПЛА може виявитися дуже ефективним[5].

Використання ройових методів взаємодії в групах БПЛА допоможе вирішити ще одну проблему – збір даних про навколишнє середовище. Габарити БПЛА можуть істотно обмежувати доступний набір бортових датчиків та їхній діапазон дії. Обмежений енергоресурс також погано впливає на допустимі енерговитрати засобів збору даних про навколишнє середовище, що призводить до зменшення радіусу роботи активних засобів збору даних, таких як лазерні далекоміри, ультразвукові датчики, системи орієнтації тощо. Ці обмеження призводять до того, що БПЛА може самостійно зібрати інформацію лише з певної області навколо себе. Хоча для забезпечення стійкого руху апарат потребує даних про нерухомі і рухомі об'єкти, що розташовані в значно більшій зоні, особливо по курсу руху. При ройовій взаємодії в групі, сусідні апарати обмінюються інформацією про навколишнє середовище, розширюючи таким чином доступні всім дані про перешкоди, повітряні потоки та інші важливі параметри середовища.

Виконання групою безпілотних апаратів загальної задачі потребує дотримання певних дистанцій між апаратами. Наприклад, при виконанні задач збору інформації на певній території, БПЛА повинні літати на

таких дистанціях, щоб мінімізувати перекривання робочих зон сенсорних датчиків і, в той же час, не допускати «пробілів» на мапі[6]. При ройовій взаємодії, кожен БПЛА визначає дистанції до сусідів і сам корегує свій курс таким чином, щоб дотримуватись потрібних дистанцій між апаратами, і не зближуватись з перепонами.

Таким чином використання мультиагентного підходу дозволяє управляти групою БПЛА умовно незалежних, що виконують задачі, які істотно відрізняються одна від одної. При використанні ройових методів всі БПЛА покликані виконувати одну загальну задачу і діють як розподілений об'єкт. Проаналізувавши відмінності обох підходів основні висновки зведено в таблицю 1.

Таблиця 1 – Особливості використання мультиагентного і ройового підходів до управління групою БПЛА

<i>Критерій</i>	<i>Мультиагентний підхід</i>	<i>Ройовий підхід</i>
Автономність групи	Група БПЛА може діяти автономно	Група може діяти автономно
Автономність окремого БПЛА	Окремий БПЛА може діяти цілком автономно	Окремому БПЛА необхідна постійна комунікація з членами групи
Розміри групи	Від одного БПЛА	Група має складатись з кількох БПЛА, що дадуть змогу їй ефективно функціонувати
Кількість виконуваних групою задач	Фактично кожен БПЛА в групі може виконувати окрему задачу	Група виконує одну розподілену або невелику кількість схожих задач
Розміри окремого БПЛА	БПЛА повинен мати достатні розміри, аби нести на собі всі необхідні для автономного польоту датчики і засоби комунікації	БПЛА може мати мінімальні розміри, проте має нести на собі деякі датчики і засоби комунікації
Комунікаційні можливості	Кожен БПЛА повинен мати можливість комунікації з координаційним центром	Кожен БПЛА повинен мати можливість комунікації з членами групи
Можливість використання БПЛА різних модифікацій	Можуть бути використані БПЛА різних модифікацій для виконання різних окремих завдань	Можуть бути використані БПЛА різних модифікацій для забезпечення ефективності функціонування групи
Вартість окремого БПЛА	Вартість окремого БПЛА висока	Вартість окремого БПЛА невисока

Отримані результати ілюструють те, що мультиагентний підхід варто використовувати у разі виконання БПЛА в складі групи окремих задач і забезпечення обміну між ними даними, що не мають критичної терміновості. Проблема уникнення несподіваних зіткнень між БПЛА лежить на датчиках і знанні маршрутів руху інших БПЛА.

Ройовий підхід слід використовувати, коли для виконання окремої задачі або групи схожих задач використовується велика кількість порівняно недорогих БПЛА, які функціонують як розподілений у просторі об'єкт, частини якого постійно обмінюються між собою інформацією про свій стан.

Подальшими шляхами дослідження є створення моделей впливу тих чи інших критеріїв на ефективність автономної роботи групи БПЛА які виконують конкретні задачі як то доставка вантажів, правоохоронна діяльність, тощо.

ПЕРЕЛІК ПОСИЛАНЬ

1. Трубіков Г. В. Применение беспилотных летательных аппаратов в гражданских целях / Г. В. Трубіков. – 2008. – С. 1.
2. Baxter J. W. Fly-by-Agent: Controlling a Pool of UAVs via a Multi-Agent System / J. W. Baxter, G. S. Horn, D. P. Leivers. // QinetiQ Ltd Malvern Technology Centre St Andrews Road. – 2007.
3. Скобелев П. О. Открытые мультиагентные системы для оперативной обработки информации в процессах принятия решений / П. О. Скобелев. – Самара, 2003. – 418 с. – (Автометрия). – (6).
4. Адаптивное управление автономной группой беспилотных летательных аппаратов / К. С.Амелин, Е. И. Антал, В. И. Васильев, Н. О. Граничина. // СПГУ. – 2009. – С. 160–161.
5. Попов В. А. Развитие направления миниатюрных беспилотных летательных аппаратов за рубежом / В. А. Попов, Д. В. Федутин. // ФГУП «ГосНИИАС».
6. Иванов Д. Я. Методы роевого интеллекта для управления группами малоразмерных беспилотных летательных аппаратов / Д. Я. Иванов. // Известия ЮФУ. – С. 222–225.

Синтез нечітких регуляторів на основі асоціативної пам'яті з апаратною реалізацією на ПЛІС

В.А. Неволько
аспірант
КПІ ім. Ігоря Сікорського
Україна, м.Київ

Розглянуто методику синтезу і реалізації швидкодіючих нечітких регуляторів на основі апаратної реалізації асоціативної пам'яті, призначених для побудови систем керування швидкодіючими об'єктами і процесами. Досліджено структури регуляторів, які забезпечують знаходження керуючого рішення на протязі 0.0001...0.0003 сек.

Ключові слова: регулятори, нечіткі множини, нечітка логіка, нейронні мережі, асоціативна пам'ять, програмована логіка.

В останні десятиріччя значна увага в теорії та практиці синтезу систем керування технічними об'єктами приділяється інтелектуальним системам з регуляторами, що базуються на нечітких множинах та нечіткій логіці і реалізують експертну технологію керування. Такі системи характеризуються значною робастністю по відношенню до різного роду невизначеностей пов'язаних з нелінійністю та нестационарністю характеристик об'єкта, впливом зовнішніх та внутрішніх завад, тощо [1]. В більшості випадків такі регулятори використовуються для керування складними багатомірними інерційними об'єктами та процесами і реалізуються ЕОМ або контролерами. Для керування простими, але швидкодіючими об'єктами, де інерційність процесів знаходиться на рівні десятих, а то і сотих часток секунди, наприклад, при керуванні процесами, що мають місце в радіоелектронних пристроях, в елементах систем зв'язку, в різних системах синхронізації, регулятори на нечіткій логіці (НЧР) поки що не використовуються, в силу їх програмної складності і нестачі часу на обчислення, хоч, як показали дослідження, такі регулятори забезпечують більш високу стійкість та більш високі показники процесів керування в порівнянні з традиційним регулюванням.

Аналіз можливостей значного збільшення швидкодії НЧР вбачається шляхом відмови від процедури обчислень в реальному часі і переході до процедур пошуку керуючого рішення в середовищі синтезованої для даного об'єкта асоціативної пам'яті, що моделює роботу НЧР. Метою роботи є розробка методики синтезу та дослідження таких швидкодіючих НЧ-регуляторів.

На даний час опубліковано величезну кількість робіт присвячених НЧР та системам керування на їх основі, створені спеціалізовані програмні оболонки, які можуть бути використані для синтезу, моделювання, налаштування та оптимізації НЧР і систем керування. Прикладом є відомий пакет нечіткої логіки Fuzzy Logic Toolbox та середовище моделювання систем керування Simulink системи MATLAB [1]. В переважній більшості робіт принцип функціонування НЧР базується на ідеях Мамдані, де в якості вхідних величин на НЧ-регулятор подається похибка

та її часові похідні. Далі вектор вхідних величин шляхом фазифікації перетворюється в вектор значень нечітких змінних, які використовуються в операціях нечіткого логічного висновку з використанням бази знань для одержання лінгвістичного значення керуючого рішення, яке після виконання операції дефазифікації подається на вихід регулятора, як керуючий вплив [1].

Дослідження обчислювальних процесів функціонування НЧР показало, що найбільш затратними в обчислювальному сенсі є алгоритм формування нечіткого логічного висновку та процедури дефазифікації, виходячи з чого в ряді робіт запропоновано використати для реалізації НЧР нечіткі нейронні мережі (ННМ), в яких зберігається основна структура НЧР, але всі обчислення виконуються навченою нейронною мережею [2]. Такі НЧР дозволяють значно спростити процедуру налаштування регуляторів, але значно зменшити часові затрати на обчислення при програмній реалізації не вдається. Це можливо лише за рахунок апаратної реалізації ННМ, де розпаралелюються обчислення, однак така реалізація потребує значних апаратних затрат, оскільки фактично реалізує всю програму роботи НЧР.

Виходячи з того, що навчений НЧР фактично є автоматом, в якому кожному значенню вектора вхідних величин формується відповідний вектор вихідних, такий автомат можливо реалізувати нейромережею, яка може розглядатися як асоціативна пам'ять такого автомата. Розглянемо одну із можливих методик синтезу такого НЧР.

В роботі [1] запропонований метод проектування одного класу НЧР, заснований на отриманих аналітичних шляхом математичних залежностях, що пов'язують вектор вхідних величин з керуючими рішеннями на виході регулятора при різних формах функцій приналежності з двома і більше термами. При проектуванні і реалізації таких НЧР немає необхідності у використанні пакета нечіткої логіки і процедура їх проектування зводиться до програмування відповідних математичних залежностей. Налаштування таких регуляторів відбувається шляхом моделювання поведінки системи керування, наприклад, в середовищі

Simulink системи MATLAB, за рахунок приведення до єдиного масштабу діапазонів зміни вектора вхідних і вихідних величин.

Відповідно до методики, на вхід НЧР подаються помилка системи θ , швидкість її зміни $\dot{\theta}$ та прискорення зміни помилки $\ddot{\theta}$. Нечіткий регулятор працює як цифровий пристрій в дискретному режимі з періодом квантування h . На кожному кроці квантування k помилка в системі дорівнює $\theta(k)$, а перша і друга похідні від помилки зазвичай обчислюють як першу і другу різницю за формулами:

$$\left. \begin{aligned} \dot{\theta}(k) &= [\theta(k) - \theta(k-1)] / h; \\ \ddot{\theta}(k) &= [\dot{\theta}(k) - \dot{\theta}(k-1)] / h = [\theta(k) - 2\theta(k-1) + \theta(k-2)] / h^2 \end{aligned} \right\}$$

Вектор вхідних величин $(\theta(k) \dot{\theta}(k) \ddot{\theta}(k))$ розглядаються як лінгвістичні змінні -похибка, швидкість зміни похибки, прискорення зміни похибки, відповідно. Кожна із лінгвістичних змінних описується відповідним, але однаковим, числом лінгвістичних значень – двома (позитивна, від'ємна), трьома (позитивна, нульова, від'ємна), п'ятьма і більше. Кожне із значень лінгвістичних змінних описується нечіткою множиною на універсумі $U = [0,1]$ однією із форм функцій приналежності $\mu_i(u) = f(u) \quad u \in [0,1]$, що представляють трикутні, гаусові, експоненційні, та інші форми функцій приналежності. Нечіткі множини розподіляються на універсали $U = [0,1]$ симетрично:

$$A_m = \theta_{\max} = -\theta_{\min}, B_m = \dot{\theta}_{\max} = -\dot{\theta}_{\min}, C_m = \ddot{\theta}_{\max} = -\ddot{\theta}_{\min}.$$

База знань регулятора формується відповідно з продукційним правилом:

$$\text{Якщо } (\theta^* = a_1^j) \text{ і } (\dot{\theta}^* = a_2^j) \text{ і } (\ddot{\theta}^* = a_3^j), \text{ то } (m^* = a_n^j), j = \overline{1, l},$$

де a_1^j, a_2^j і a_3^j – лінгвістичні оцінки похибки, її першої та другої похідних, які розглядаються як терм-множини на універсальній множині $U = [0,1]$, $j = \overline{1, l}$; a_n^j – лінгвістичні оцінки керуючої дії регулятора, що вибирається із терм-множини змінної m .

При подачі на регулятор в деякий момент часу значень вектора вхідних змінних $\theta^*, \dot{\theta}^*$ і $\ddot{\theta}^*$ с шагом квантування h відбувається перерахунок (нормування) значень вхідних змінних в змінні u_1^*, u_2^*, u_3^* і розрахунок значень їх функцій приналежності $\mu_1(u_1^*) \mu_2(u_2^*) \mu_3(u_3^*)$

Подальші обчислення значень керуючої дії u_c (обчислення, що включають в себе дії над нечіткими множинами, що описують правила бази знань, та процедури дефаззифікації методами Мамдані чи Сугено) проводиться у відповідності з аналітичними виразами приведеними в роботі [1]. Наприклад, для регулятора, де лінгвістичні значення вектора вхідних величин представлені двома термами (позитивна, від'ємна) з трикутними функціями приналежності формули для обчислень керуючої дії u_c мають вигляд

$$u_c = \frac{B/2 + (A^3 - B^3)/6}{B + (A^2 - B^2)/2} \text{ при } A \leq B$$

та

$$u_c = \frac{(A - A^2 + B^2)/2 + (A^3 - B^3)/6}{A - (A^2 - B^2)/2} \text{ при } A \geq B,$$

де $A = \min[\mu_1(u_1^*) \mu_1(u_2^*) \mu_1(u_3^*)]$; $B = \min[\mu_2(u_1^*) \mu_2(u_2^*) \mu_2(u_3^*)]$.

Одержане значення u_c далі перераховується (нормується) у вихідний сигнал регулятора у відповідності з формулою

$$m^* = m_{\max}(1 - 2u_n),$$

де m_{\max} – максимальне значення вихідного сигналу.

Розгорнута структурна схема системи керування з НЧР та параметрами налаштування регулятора $F(\theta_{\max}, \theta_{\min}, \dot{\theta}_{\max}, \dot{\theta}_{\min}, \ddot{\theta}_{\max}, \ddot{\theta}_{\min}, m_{\max}, m_{\min}, h)$ показана на рис.1.

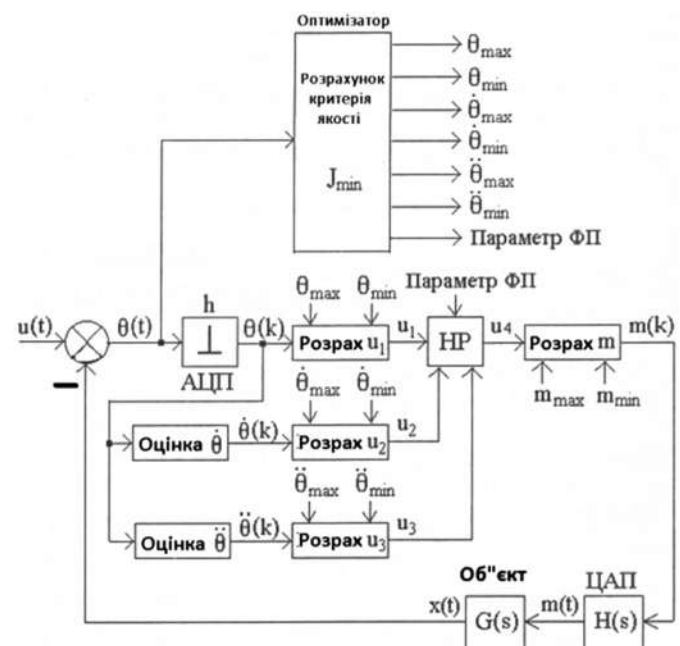


Рис. 1 Розгорнута структурна схема системи керування з НЧ-регулятором

Налагодження та дослідження системи керування з НЧР (рис.1) дозволяє отримати масив даних, що характеризує роботу регулятора і синтезувати на його основі нейромережову модель НЧР регулятора. Виходячи з того, що вектор вхідних величин НЧР має три складові $(\theta, \dot{\theta}, \ddot{\theta})$ і один вихід (m^*) , то зовнішня структура нейромережі завжди буде мати три входи і один вихід. Внутрішня структура нейромережі може бути з одним прихованим шаром, оскільки нейромережа прямого поширення з трьома шарами дозволяє змодельовати будь яку нелінійну функцію відповідно кількості нейронів в прихованому шарі [3]. Виходячи з того, що складність функції яку реалізує НЧР залежить від кількості терм-множин, що описують вектори вхідних і вихідних величин та форми нечітких множин, синтез нейромережевого регулятора зводиться до вибору із можливих варіантів структури з мінімальною кількістю нейронів в прихованому шарі і підбору функцій активації нейронів. Оскільки на даний час не існує методик або чітких рекомендацій по вибору оптимальної структури нейромережі, використовується технологія створення ряду можливих варіантів нейромереж-кандидатів і відбір найкращої з них, яка з мінімальною похибкою відтворює НЧР.

Для реалізації такого підходу в середовищі MATLAB розроблено програмний пакет «MIMO-Plant», який дозволяє спочатку створити із одержаного масиву даних еталон-

ну модель регулятора як об'єкта у просторі станів. Далі із масиву даних в пакеті «MIMO-Plant» генерується навчальна вибірка даних для створення і навчання нейромережі. Після генерації навчальної вибірки в пакеті формується ряд нейромережевих структур з трьома входами і одним виходом та різною кількістю нейронів (2-10) в прихованому шарі, вибирається алгоритм навчання та кількість навчальних ітерацій, після чого запускається процес навчання. Навчені нейромережі оцінюються на адекватність порівнянням їх реакцій з реакцією еталонної моделі регулятора на тестові сигнали і обчислення сумарної середньоквадратичної помилки:

$$\varepsilon = \sqrt{\frac{\sum_{i=1}^n (\hat{y} - y)^2}{n}}$$

де y – вихід еталонної моделі регулятора; \hat{y} – вихід нейромережевої моделі регулятора.

По результатах оцінки вибирається структура, в якій середньоквадратична помилка найменша. Вибрана нейромережа вважається адекватною НЧР, на її основі створюється модель системи керування аналогічна системі з НЧР і порівнюються показники їх роботи. При підтвердженні адекватності нейромережевого НЧР в складі моделі системи керування вирішується завдання реалізації його на елементах програмованої логіки (ПЛІС).

Для цього можна використати пакет ISE Design Suite 13.2, який забезпечує синтез моделі нейромережі у вигляді програмного коду, і середовище ISim - для моделювання її роботи. Всі апаратні елементи нейромережі синтезуються на мові VHDL. Кожний нейрон представляється окремим процесором, має свою матрицю констант, яка формується при налагодженні нейромережі, і представлену таблицею функцію активації. Всі нейрони одного шару запускають-

ся на обчислення одночасно. Розрядність представлення вхідних і вихідних величин визначається необхідною точністю представлення даних і може бути, наприклад, 8-розрядним чи 16-розрядним двійковим числом.

Моделювання нейромереж і оцінка відповідно займаного ними ресурсу ПЛІС в кількості вентилів логічної матриці LUTs показав, що нейромережі з трьома входами і одним виходом та сигмоїдними функціями активації в прихованому шарі і двома нейронами займає ресурс в 2544 вентилів, з трьома – 3497, з чотирма – 4438. Отримані данні в LUTs можуть дещо змінюватись при зміні констант, що задають синаптичні ваги нейронів. Часові затрати на обчислення знаходяться в межах 150 – 200 наносекунд.

ВИСНОВОК

Дослідження затрат часу на обчислення керуючого впливу різними реалізаціями НЧ-регулятора показало, що основна структура витрачає (0.1...0.25) сек, нейромережева – (0.01...0.03) сек, апаратна – (0.0001...0.0003) сек, що дозволяє їй керувати процесами зі швидкодією (0.5...1.0) 10^{-3} сек.

ПЕРЕЛІК ПОСИЛАНЬ

1. Гостев В.И. Нечеткие регуляторы в системах автоматического управления. К.:Издательство "Радиоаматор", 2008. - 972 с
2. Рутковская Д. и др. Нейронные сети, генетические алгоритмы и нечеткие системы: Пер. с польск. И.Д. Рудинского. – М.: Горячая линия – Телеком, 2007. – 452 с.
3. Саймон Хайкин. Нейронные сети. Полный курс. – М.: Вильямс, 2006. – 1104с.

Рецензент: к.т.н., доц. каф. АУТС, КПІ ім. Ігоря Сікорського,
П.І. Кравець

ТЕХНОЛОГІЇ ПРОГРАМУВАННЯ
PROGRAMMING TECHNOLOGIES

Реалізація віртуальних змінних

Крак Ю.В.
професор

Київський національний університет ім. Тараса Шевченка
Україна, Київ

Коваль Ю.В.
асистент

Київський національний університет ім. Тараса Шевченка
Україна, Київ

В цій статті розглянуто поняття змінної, запропоновано його формалізацію. Розширено поняття змінної до поняття віртуальної змінної та наведено перелік операцій для віртуальної змінної.

Ключові слова: віртуальний процес, віртуальна змінна.

1. ВСТУП ТА ПОСТАНОВКА ЗАДАЧІ

Віртуальний процес складається з атомарних процесів [1]. Термін атомарний вживається для процесів виконання програми локалізованих в межах однієї операційної системи та ідентифікованих ідентифікатором процесу в цій системі. Кожен атомарний процес має свої локальні змінні, що визначені та застосовуються відповідно до правил тієї мови програмування, якою написаний код атомарного процесу. Віртуалізація процесу дозволяє атомарному процесу окрім власних змінних використовувати змінні всього віртуального процесу. Зрозуміло, що з точки зору конкретної мови програмування ці віртуальні змінні мають специфічне синтаксичне оформлення, яке може суттєво відрізнятися від відповідного оформлення в самій мові програмування. Тож необхідно формалізувати загальне поняття змінної та запропонувати його конкретизацію для віртуальних змінних, що і буде задачею цієї публікації.

2. ФОРМАЛІЗАЦІЯ ПОНЯТТЯ ЗМІННОЇ

2.1 Змінна оперативної пам'яті

З точки зору атомарного процесу змінна це частина адресного простору процесу наперед визначеного розміру. Структури даних такі, як масив, можуть міняти свій розмір, проте в кожен момент часу їх розмір чітко визначений. Фіксований розмір пам'яті, виділений для змінної, визначає та обмежує множину значень, що може бути застосована, та традиційно називається типом. Ідентифікується така змінна адресою, починаючи з якої віділяється пам'ять для неї.

2.2 Змінна мови програмування

З точки зору мови програмування, змінна це іменоване значення іноді певного типу. Де і як зберігається це значення переважно не відомо. Втім, мова програмування C [2], та всі інші мови, які дотримуються визначеного нею підходу, визначає поняття поінтера (вказівника), що, власне, і є адреса. Вцілому ж, операція встановлення значення визначається як операція іменування значення. Це принципово відрізняється від того, що відбувається в пам'яті процесу. В такому випадку значення існує завжди, навіть коли не використовується. До нього лише приписують ім'я. Зміна стану змінної полягає не в тому, що змінилося наповнення пам'яті, а в тому, що змінилося значення, до якого припи-

сано ім'я. Тобто змінна не має ні адреси, ні, як такого, розміру. Розмір має значення, но не змінна.

Операція розіменування не знищує зв'язок між ім'ям та значенням (як це звучить в самому слові), а надає доступ до значення, з яким встановлено зв'язок у імені.

Спираючись на такі означення отримаємо формалізацію поняття імені а не змінної. Тож для формалізації поняття змінної оберемо підхід, визначений в мові програмування C, та такий, що узгоджений з поняттям пам'яті процесу та адресного простору.

2.3 C-змінна

Як відомо [2], в мові програмування C розрізняють глобальні, статичні, динамічні, локальні автоматичні та локальні статичні змінні. В C++ [3] до цього додаються статичні поля класу. (По суті вони подібні статичним змінним.) Крім того є параметри значення (в C) та параметри посилання (референс) (в C++). (Поінтер як параметр принципово не відрізняється від параметру значення). Глобальні та статичні змінні створюються під час запуску процесу. В кодї навіть можуть бути відсутні команди виділення пам'яті для них. Автоматичні змінні та параметри створюються автоматично (що, власне, і дало їм назву). Те саме стосується знищення змінних. Для динамічних змінних операція створення завжди доступна програмісту. Операція знищення іноді закривається процедурою збору сміття. Єдина операція яка не передбачена для цих змінних – це витирання (втрата значення). Сама змінна при цьому залишається існувати. Проблема з реалізацією цієї операції полягає в тому, що в пам'яті не існує неіснуючого значення. Як деяке виключення існує значення NULL або nullptr що є еквівалентом неіснуючого значення. Проте це лише домовленість. Насправді, неіснуюче значення реалізовувалося в ЕОМ «Ельбрус» [4].

2.4 Формалізація загального поняття змінної

Таким чином в загальному розумінні змінна це сховище значень, для якого можуть бути реалізовані та застосовуватись наступні п'ять операцій:

- Створення,
- Знищення,
- Встановлення значення,
- Зчитування значення,
- Витирання значення.

Слід зауважити, що не всі операції можуть бути реалізованими, проте операція зчитування значення має бути реалізована завжди. Ім'я змінної є необов'язковим параметром який є еквівалентом адреси в адресному просторі процесу.

3. КОНКРЕТИЗАЦІЯ ВІРТУАЛЬНОЇ ЗМІННОЇ

Віртуальна змінна реалізує всі наведені базові операції для змінної а також наступні додаткові операції, дуже корисні та притаманні інтерпретованим мовам програмування, в тому числі препроцесором мови C,:

- Перевірка існування змінної,
- Перевірка наявності у змінної значення,
- Обчислення розміру змінної,
- Обчислення розміру значення,
- Перевірка, чи є у змінної ім'я,
- Встановлення імені змінної,
- Отримання імені змінної.

Наступні операції не типові для змінних мов програмування, проте є типовими для баз даних [5] та файлових систем [6]:

- Встановити видимість змінної,
- Отримати видимість змінної,
- Встановити власника змінної,
- Отримати власника змінної,
- Встановити права доступу змінної,
- Отримати права доступу змінної,
- Встановити захист для змінної.

4. ВИСНОВКИ

В цій статті наведено формалізацію поняття «змінна» та розширено його для створення віртуальних змінних для

віртуального процесу. Наведено перелік операцій для віртуальних змінних.

ПЕРЕЛІК ПОСИЛАНЬ

1. Крак Ю. В. Віртуальний процес: означення та застосування в створенні системи жестового інтерфейсу / Ю. В. Крак, Ю. В. Коваль, А. Б. Ставровський. // Вісник Київського національного університету імені Тараса Шевченка Серія фізико-математичні науки. – 2015. – №1. – С. 141–144.
2. Ritchie D. M. C Reference Manual [Електронний ресурс] / Dennis MacAlistair Ritchie // Bell Telephone Laboratories, Murray Hill, New Jersey, USA, p. 31. – 1974. – Режим доступу до ресурсу: <https://www.bell-labs.com/usr/dmr/www/cman.pdf>.
3. Stroustrup B. The C++ Programming Language [Електронний ресурс] / Bjarne Stroustrup. – 2016. – Режим доступу до ресурсу: <http://www.stroustrup.com/C++.html>.
4. Пентковский В. М. Автокод Эльбрус. Эль-76. Принципы построения языка и руководство к использованию / В. М. Пентковский. – Москва: Наука, 1982. – 352 с.
5. PostgreSQL 9.6.3 Documentation [Електронний ресурс]. – 2017. – Режим доступу до ресурсу: <https://www.postgresql.org/docs/9.6/static/index.html>.
6. Thompson K. UNIX PROGRAMMER'S MANUAL [Електронний ресурс] / K. Thompson, D. M. Ritchie // Bell Labs, USA, p. 194,. – 1971. – Режим доступу до ресурсу: <https://www.bell-labs.com/usr/dmr/www/1stEdman.html>.

Причины отказа от Hibernate и как правильно использовать JDBC

Шишкин Владислав Игоревич
студент
КПІ ім. Ігоря Сікорського
Україна, м.Київ

Суханюк Марина Валентиновна
студентка
КПІ ім. Ігоря Сікорського
Україна, м.Київ

Доклад состоит из одного раздела, содержит 6 страниц, 3 таблицы, 1 рисунок, 7 источников.

В данном докладе ведется короткий экскурс по технологиям JDBC и Hibernate, где приводятся сравнительные характеристики, а также перечисляются их достоинства и недостатки. Особое внимание уделяется разработанной технологии MySQL-wrapper: описание и обоснование ее необходимости. Также прилагаются материалы тестирования трех технологий при разных нагрузках, после которых сделаны выводы.

Ключевые слова: соединение с базами данных на java, Hibernate, система управления базами данных, объектно-реляционное отображение, MySQL, java.

На сегодняшний день, очень сложно найти разработчика, который бы при программировании приложений, не работал с базами данных. Потому, для упрощения взаимодействия с СУБД начали писаться различные фреймворки [3]. JDBC (Java DataBase Connectivity) является платформенно-независимым промышленным стандартом взаимодействия Java-приложений с различными СУБД, реализованным в виде пакета java.sql, входящего в состав Java SE [1]. После этого стали появляться ORM-решения, которые упрощали взаимодействия с базой данных и объектами Java. ORM (Object-relational mapping) позволяет заносить, получать, изымать данные из базы данных. ORM-решением для языка Java, является технология Hibernate, которая помогает сэкономить значительную часть времени при построении запросов и извлечении данных.

Однако, существует множество споров на тему, что лучше использовать при разработке проектов: JDBC или Hibernate. Данное сравнение нельзя назвать верным, поскольку Hibernate - это оболочка для JDBC, которая обладает огромным набором функций и возможностей.

Hibernate имеет целый ряд преимуществ, которые являются решающими при выборе данной технологии.

- Hibernate устраняет множество спагетти кода (повторяющегося), который постоянно преследует разработчика при работе с JDBC. Скрывает от разработчика множество кода, необходимого для управления ресурсами и позволяет сосредоточиться на бизнес логике.
- Hibernate поддерживает XML так же как и JPA аннотации, что позволяет сделать реализацию кода независимой.
- Hibernate предоставляет собственный мощный язык запросов (HQL), который похож на SQL. Стоит отметить, что HQL полностью объектно-ориентирован и понимает такие принципы, как наследование, полиморфизм и ассоциации (связи).
- Hibernate — широко распространенный open source проект. Благодаря этому доступны тысячи открытых статей, примеров, а так же документации по использо-

ванию фреймворка.

- Hibernate поддерживает ленивую инициализацию используя проху объекты и выполняет запросы к базе данных только по необходимости.
- Hibernate поддерживает разные уровни cache, а, следовательно, может повысить производительность.
- Важно, что Hibernate может использовать чистый SQL, а, значит, поддерживает возможность оптимизации запросов и работы с любым сторонним вендором БД и его возможностями [6].
- Лёгкость разработки: разработчик может не знать специфики базы данных, с которой работает.

Однако, далеко не все разработчики являются поклонниками Hibernate или JDBC. Большинство компаний склоняются к использованию различных ORM для ускорения работы. Важным недостатком является то, что использование Hibernate ускорит роботу, но при этом замедлит приложение.

Каковы же основные причины отказа от Hibernate и JDBC без оболочки? В таблице 1 наведены основные данные по этому поводу.

Таблица 1 – Причины отказа от Hibernate и JDBC

Почему не чистый JDBC?	Почему не Hibernate?
При использовании обычного JDBC, мы получаем огромное количество запросов, каждый из которых необходимо покрыть тестами, чтобы быть уверенными в правильности выполнения программного продукта. В случае использования своей ORM, необходимо проверить базовые функции, так как, человеческая ошибка, пропуск символа в запросе, исключаются.	Практически сразу, не тратя время и не зная SQL, есть возможность начать писать читабельный код для работы с базой данных. Hibernate позволяет перейти на другую СУБД, изменив только файл настройки. Однако, Hibernate - это огромный поезд с достаточно большим количеством вагонов, которые в разрабатываемом проекте могут быть совершенно не нужны. Очень часто бывает, что главным приоритетом является наличие быстрого фреймворка для работы конкретно с MySQL.

Таким образом, для разработки нашего проекта, было решено использовать свой собственный, быстрый и небольшой ORM для работы с базой данной MySQL.

Во время разработки, первым, с чем мы столкнулись, была проблема, связанная с большими затратами при процессе открытия подключения к базе данных. В Hibernate используется технология Connection Pool, которая при запуске открывает заданное количество подключений и, когда достаточно большая нагрузка, открываются дополнительные подключения. Взяв за основу вышеописанную технологию, мы решили написать собственный Connection Pool, схема работы которого описана на рисунке 1.

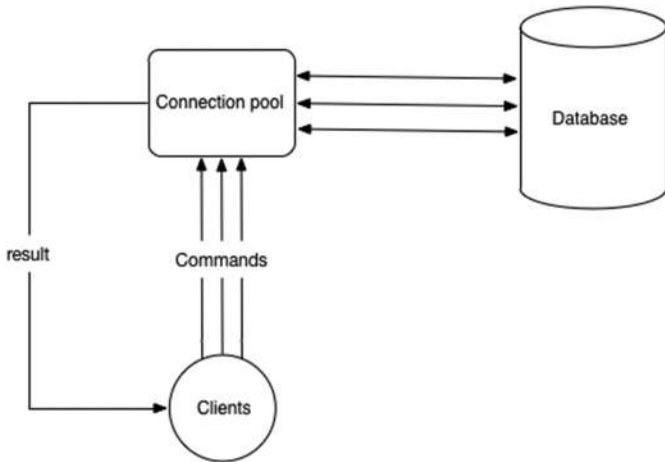


Рис. 1 - Схема работы Connection Pool

Как видим с рисунка 1, Connection Pool играет роль связывающего компонента между базой данных и клиентом. Что является логичным, поскольку благодаря этому компоненту есть возможность не тратить лишнее время на открытие нового подключения к базе данных, потому что несколько свободных подключений всегда будут наготове для использования.

Что же происходит, если Connection Pool занят?

Если есть возможность открыть еще подключения, Connection Pool автоматически создаст несколько подключений к базе данных и выполнит пользовательский запрос в новом подключении. Иначе, клиент будет ожидать, когда соединение освободится по принципу FIFO (first in first out).

Также было важно написать оболочку под JDBC. Java-класс должен иметь пустой конструктор и быть описан с помощью аннотаций, благодаря которым есть возможность построить такие запросы, как insert, delete, update, select.

Далее приведен пример модели, которая была использована как сущность для подключения к базе данных.

```

@DBModel(tableName = "users")
public class User implements DBEntity {
    @DBField(fieldName = "id", isAutoIncrement = true)
    private int id;
    @DBField(fieldName = "login")
    private String login;
    @DBField(fieldName = "password")
    private String password;
}
    
```

```

public User() {
}
}
    
```

С помощью аннотаций мы указываем поля и имя таблицы соответственно в базе данных. Благодаря такому подходу мы имеем возможность делать любые запросы в базу данных. Пример написания аннотаций наведен ниже.

```

@Target(ElementType.FIELD)
@Retention(RetentionPolicy.RUNTIME)
public @interface DBField {
    @NotNull
    String fieldName();
    boolean isAutoIncrement() default false;
    int maxLength() default 3;
    int minLength() default Integer.MAX_VALUE;
}

@Target(ElementType.TYPE)
@Retention(RetentionPolicy.RUNTIME)
public @interface DBModel {
    @NotNull
    String tableName();
}
    
```

Если говорить о времени обработки, то возможно проинформировать следующую проверку.

Пусть имеем выборку в базе данных с заданным количеством пользователей в таблице равным 1000. Выполняемые запросы: select, insert, delete, update. Количество клиентов (потоков), которые выполняли запросы – 5. Будем считать, что клиент уже открыл подключение к базе данных.

В таблице 2 наведено сравнение скоростей обработки запросов с помощью технологий JDBC, Hibernate, разработанного MySQL wrapper в зависимости от количества повторений запроса.

ТАБЛИЦА 2 – СРАВНЕНИЕ СКОРОСТЕЙ

Количество повторений запроса	JDBC	Hibernate	Разработанный MySQL wrapper
100	AVG time : 37 millisecond	AVG time : 787 millisecond	AVG time : 88 millisecond
1000	AVG time : 146 millisecond	AVG time : 868 millisecond	AVG time : 279 millisecond
100000	AVG time : 7699 millisecond	AVG time : 11073 millisecond	AVG time: 9588 millisecond
1000000	AVG time : 75515 millisecond	AVG time : 93989 millisecond	AVG time: 91832 millisecond

Как видно из таблицы 2, JDBC-технология работает быстрее двух остальных, но это происходит в том случае, когда подключение открывается один раз. То есть, клиент, открыв подключение, использует его постоянно - такой подход нельзя звать правильным. Однако, поскольку и Hibernate, и разработанный MySQL wrapper используют Connection Pool, это было бы неверным - учитывать время закрытия и открытия подключения каждый раз. Если

же их сравнивать в одинаковых условиях, то есть, учитывая время подключения у всех трех технологий, или же, не учитывая ни у кого, то скорость у JDBC будет намного хуже, чем у Hibernate и разработанного фреймворка.

Как видим, на большом количестве запросов, Hibernate начинает работать в разы быстрее из-за системы кэширования. Однако, на последнем приведенном количестве повторений MySQL wrapper работает несколько быстрее, и с увеличением повторений, разница будет увеличиваться.

Ниже представлены параметры компьютера, на котором производилось тестирование.

ТАБЛИЦА 3 – ПАРАМЕТРЫ КОМПЬЮТЕРА ПРИ ТЕСТИРОВАНИИ

Тип процессора	Intel Core i7-4870HQ
Частота, ГГц	2,5
Количество ядер процессора	4
Предустановленная ОС	OS X
Объем оперативной памяти, ГБ	16
SSD, ГБ	512

Благодаря тестированию скорости работы технологии, мы можем видеть необходимость в использовании своего ORM-фреймворка, который справляется с задачей быстрее, чем Hibernate, и если учитывать время подключения для JDBC, то и быстрее последнего. Свой ORM-фреймворк является необходимым в том случае, когда приоритетом является скорость и легкость приложения. В случае же с большим проектом, где есть вероятность перехода с одной СУБД на другую, больше подойдет использование Hibernate.

ПЕРЕЧЕНЬ ССЫЛОК

1. Java Database Connectivity [Электронный ресурс] - Режим доступа к ресурсу: https://ru.wikipedia.org/wiki/Java_Database_Connectivity
2. Hibernate [Электронный ресурс] - Режим доступа к ресурсу: [https://ru.wikipedia.org/wiki/Hibernate_\(%D0%B1%D0%B8%D0%B1%D0%BB%D0%B8%D0%BE%D1%82%D0%B5%D0%BA%D0%B0\)](https://ru.wikipedia.org/wiki/Hibernate_(%D0%B1%D0%B8%D0%B1%D0%BB%D0%B8%D0%BE%D1%82%D0%B5%D0%BA%D0%B0))
3. JDBC - средство общения между Java и базами данных [Электронный ресурс] - Режим доступа к ресурсу: http://www.javaportal.ru/java/articles/JDBC_java_BD.html
4. Dipti Phutela. Hibernate Vs JDBC [Электронный ресурс] / Dipti Phutela, Mindfire Solutions - Режим доступа к ресурсу: http://www.mindfiresolutions.com/mindfire/Java_Hibernate_JDBC.pdf.
5. Hibernate vs JDBC performance [Электронный ресурс] - Режим доступа к ресурсу: http://phpdao.com/hibernate_vs_jdbc/
6. Вопросы и ответы на собеседование по теме Hibernate Framework. Часть 1. [Электронный ресурс] – Режим доступа к ресурсу: <http://javastudy.ru/interview/jee-hibernate-questions-answers/>
7. Java Database Connectivity - Преимущества [Электронный ресурс] – Режим доступа к ресурсу: <http://bourabai.kz/dbt/JDBC02.htm>Бл

Рецензент: к.т.н., ст. викл. каф. АСОІУ, КПІ ім. Ігоря Сікорського,
М.О. Сперкач

Конструювання програмного забезпечення для вбудованих систем з використанням типових шаблонів

Катин Павло Юрійович
доцент
КПІ ім. Ігоря Сікорського
Україна, Київ

Розглянуті особливості побудови архітектури і конструювання програмного забезпечення, призначеного для вбудованих систем на основі малопотужних мікроконтролерів. Приведені результати, що відображають вплив на потрібну оперативну пам'ять і пам'ять програм архітектури програмного забезпечення. Розглянуті архітектури програми у вигляді нескінченного циклу, модифікованого кінцевого автомату і класичного кінцевого автомату. Надані рекомендації конструювання програмного забезпечення для малопотужних мікроконтролерів і напрямки подальших досліджень.

Ключові слова: малопотужні мікроконтролери, програмне забезпечення, кінцевий автомат.

У статті будуть розглянуті особливості побудови архітектури програм для 8, 16 розрядних мікроконтролерів, призначених для вбудованих систем. Далі будемо називати ці мікроконтролери малопотужними. На сьогоднішній день малопотужні мікроконтролери (ММ) є актуальними для простих і надійних пристроїв. Особливості побудови архітектури, розробка та налагодження програмного забезпечення для вбудованих систем на основі ММ передбачають необхідність враховувати ряд факторів. У деяких типах мікропроцесорних систем (МПС) дані фактори не мають особливого значення. Прикладом таких МПС є відкриті системи на базі потужних мікропроцесорів. Прикладом архітектури таких мікропроцесорів являється архітектура IA-32, AMD 64 або інша, аналогічна за потужністю і можливістю адресувати пам'ять [1]. В цьому випадку важливим фактором є читабельність програм, можливість їх легкого супроводження, тестування і налагодження та інші питання вдосконалої розробки програмного забезпечення (ПЗ). Для вирішення вищезазначених питань використовується сучасні IDE. Програма будується з використанням типових шаблонів (патернів) [2-5], робиться формалізація ПЗ у вигляді діаграми класів та інших типів діаграм. При цьому збільшується обсяг потрібної оперативної пам'яті і вихідного коду.

На відміну від потужних МПС, в системах на базі ММ, важливим фактором є збереження оперативної пам'яті мікроконтролера (Random Access Memory (RAM)). Другим важливим фактором є обмеженість пам'яті програм (Read Only Memory (ROM)). Крім того, для ММ існує багато інших особливостей, які є суттєвими при конструюванні програм. При цьому, з огляду на значні відмінності в широкій номенклатурі сучасних ММ, дані особливості враховуватися не будуть. З урахуванням цього, введемо обмеження на проведення досліджень з питань побудови архітектури програмного забезпечення для МПС на базі ММ:

- в дослідженні розглядаються питання побудови

архітектури програмного забезпечення на мові програмування С, С ++;

- архітектурні особливості побудови ПЗ з використанням Ассемблера або машинного коду необхідно виносити в окреме дослідження, у данному розглядатися не будуть;

- до факторів, які ураховуються в дослідженні, відноситься архітектура ПО і її вплив на потрібну ROM та обмеження пам'яті типу RAM;

- для дослідження були обрані приклади програми, що моделюють 3 стана ММ мікроконтролеру, це не обмежує ефективність результатів, проте потребує подальших досліджень.

Метою дослідження є порівняння простої архітектури ПЗ у вигляді нескінченного циклу [3], що звичайно підлягає критиці з боку прихильників класичної архітектури типових паттернів (шаблонів), і архітектури у вигляді кінцевого автомату (КА).

Розглянемо перший варіант реалізації архітектури ПЗ для вбудованих систем на базі ММ. Першим простим варіантом є реалізація програми з використанням нескінченного циклу і спрощеного варіанту КА [3]. У такої архітектури є свої плюси і мінуси. Плюси полягають у простоті архітектурної реалізації. Такий варіант архітектури ПЗ часто зустрічається у академічних (навчальних) рішеннях і простих типах МПС, описаний у [3] і особливих пояснень не вимагає. Приведемо демонстраційних приклад коду, що реалізує дану архітектуру. У данному коді реалізована спрощена модель роботи світлофора, для демонстрації. У коді реалізовано два вкладених кінцевих автомата. Один автомат керує роботою світлофору, інший автомат моделює роботу денного і нічного режиму. Для спрощення коду і зменшення обсягу файлу, що завантажується у мікроконтролер, стани програмно реалізовані у вигляді констант. Програма реалізована у процедурному стилі. Пояснення надані у коментарях.

```

Приклад вихідного коду 1
#include "io430.h"
#include <stdint.h>
#include <stdlib.h>
#include <intrinsics.h>
#define IN_ 0
#define OUT_ 1
#define ON_ 1
#define OFF_ 0
// Моделювання станів світлофору
#define RED 3
#define GREEN 5
#define YELLO_DOWN 7
#define YELLO_UP 9
// Моделювання денного і нічного режимів
#define DAY 23
#define NIGHT 25
// Глобальні змінні для зберігання поточного стану світло-
фору, денного і нічного режимів
volatile unsigned int Blink_State;
volatile unsigned int Semo_State;

void Init(void) //Функція ініціалізації
{
    WDTCTL = WDTPW + WDTHOLD;
    Blink_State = RED;
    Semo_State = DAY;

    P2OUT_bit.P0 = OFF_;
    P2OUT_bit.P1 = OFF_;
    P2OUT_bit.P2 = OFF_;

    P2DIR_bit.P0 = OUT_;
    P2DIR_bit.P1 = OUT_;
    P2DIR_bit.P2 = OUT_;

    P1IE |= BIT1 + BIT0;// Interrupt inable from P1.3
    P1IES |= BIT1 + BIT0;// Pulse front. From h to 0
    P1IFG &= ~(BIT1 + BIT0);// Clear interrupt flag. If
not may be go to isr

    __enable_interrupt();
}

void BlinkDAY(void) // Функція - автомат роботи світлофо-
ру для денного режиму
{
    if (Blink_State == RED){

        P2OUT_bit.P0 = ON_;
        P2OUT_bit.P1 = OFF_;
        P2OUT_bit.P2 = OFF_;
        Blink_State = YELLO_DOWN;
        __delay_cycles(1000000);

    }    if ((Blink_State == YELLO_UP)||(Blink_
State == YELLO_DOWN)){
        P2OUT_bit.P0 = OFF_;
        P2OUT_bit.P1 = ON_;
        P2OUT_bit.P2 = OFF_;

        if(Blink_State == YELLO_DOWN) {
            Blink_State = GREEN;
        } else if (Blink_State == YELLO_UP){
            Blink_State = RED;
        }
        __delay_cycles(1000000);
    }

    if (Blink_State == GREEN) {
        P2OUT_bit.P0 = OFF_;
        P2OUT_bit.P1 = OFF_;
        P2OUT_bit.P2 = ON_;
        Blink_State = YELLO_UP;
        __delay_cycles(1000000);
    }
}

void BlinkNIGHT(void) // Функція для моделювання ніч-
ного режиму
{
    P2OUT_bit.P0 = OFF_;
    P2OUT_bit.P1 = ON_;
    P2OUT_bit.P2 = OFF_;
    __delay_cycles(1000000);
    P2OUT_bit.P0 = OFF_;
    P2OUT_bit.P1 = OFF_;
    P2OUT_bit.P2 = OFF_;
    __delay_cycles(1000000);
}

int main( void )
{
    Init();

    for( ; ; ) { // Нескінченний цикл для визначення
денного або нічного режиму
        if (Semo_State == DAY)
            BlinkDAY(); // Функція- автомат денного
режиму
        else
            if (Semo_State == NIGHT)
                BlinkNIGHT();// Функція нічного режиму
            }
    return 0;
}

#pragma vector=PORT1_VECTOR
__interrupt void isr_port_1(void)
{
    __delay_cycles(10);
    if (Semo_State == DAY)
        Semo_State = NIGHT;
    else
        Semo_State = DAY;
}

```

```
P1IFG &= ~(BIT1 + BIT0);
    __delay_cycles(10);
}
```

Програма побудована для мікроконтролерів родини MSP430 і є робочим прикладом. Під час нескінченного циклу здійснюється перевірка глобальної змінної Semo_State, значення якої змінюються через апаратне переривання. Апаратне переривання зв'язано зі станом двох кнопок, див. функцію void isr_port_1(void).

В одному з варіантів відпрацьовує функція BlinkDAY(), що є функцією-автоматом денного режиму роботи. У іншому відпрацьовує функція BlinkNIGHT(), нічного режиму. Можливий інший варіант реалізації данної архітектури, проте особливого впливу на обсяг ROM і потрібну RAM не відбувається, що не впливає на якість досліджень. Компіляція здійснювалася з використанням середовища IAR system IAR Embedded Workbench for MSP430 Version 6.50.4. В результаті компіляції розмір файлу коду для завантаження, потрібної ROM складав 1,214 Кбайт. Тип файлу *.hex. Під час запуску програми ознак переповнення RAM не виявлено.

Аналогічні результати були отримані для більш простого варіанту програми на основі архітектури нескінченного циклу, приклад якої внаслідок її простоти не наводиться. При аналогічних умовах дослідження розмір файлу для завантаження дорівнював 755 байт.

Розглянемо інший варіант реалізації архітектури програми з використанням типового паттерну кінцевого автомату [2] з використанням ООП. У вихідному кодї, що представлено далі, реалізований модифікований варіант паттерну кінцевого автомату. Це експериментальний варіант, що поєднує у собі варіант нескінченного циклу і інкапсуляцію функціональності у окремому класі. У наслідок збільшення обсягу вихідного коду, у даній роботі приклад не наводиться.

Приклад вихідного коду 2

```
#include "io430.h"
#include "Context.h"
#include "Red.h"
#include "Green.h"
#include "YellowUp.h"
#include "YellowDown.h"
int main( void )
{
    // Stop watchdog timer to prevent time out reset
    WDTCTL = WDTPW + WDTHOLD;
    P1DIR = BIT0|BIT1|BIT2|BIT3|BIT4|BIT5|BIT6|BIT7;
    P1OUT = 0;
    Context *_context = new Context(new Red());
    while(1)
    {
        _context->NextState();
    }
    return 0;
}
```

Під час запуску програми ознак переповнення RAM не виявлено. Перевагою даної реалізації є можливість збіль-

шувати кількість станів автомату без впливу на інші елементи вихідного коду. Для даного прикладу функціональність КА інкапсульована у класи станів.

Недоліком є збільшення обсягу вихідного коду і певна відмінність від типового шаблону КА. Умови компіляції такі самі як і у прикладі вихідного коду 1. Розмір файлу коду для завантаження, тобто обсяг потрібної ROM складав 2,334 Кбайт. Тип файлу *.hex. Порівняння першого і другого варіантів програми дає можливість зробити висновок про незначне збільшення обсягів файлу для завантаження. Експертна оцінка даної реалізації викликала сумнівність у доцільності її використання у практичній роботі.

Розглянемо 3 варіант вихідного коду, у якому реалізований типовий варіант паттерну кінцевого автомату [2-5]. У вихідному кодї і заголовочних, що представлені далі, реалізований типовий варіант паттерну кінцевого автомату. Це класичний варіант реалізації КА з використанням ООП.

Приклад вихідного коду 3

```
#include "io430.h"
#include <stdint.h> // Integers of defined sizes
#include <stdlib.h>
#include <intrinsics.h>

#include "ConcreteState.h"
#include "TrafficLight.h"
#define IN_ 0
#define OUT_ 1
#define ON_ 1
#define OFF_ 0

int main( void )
{
    TrafficLight* trl = new TrafficLight();
    trl->on();
    return 0;
}

    Заголовочний файл ConcreteState.h
#pragma once
#ifndef CONCRETE_STATE
#define CONCRETE_STATE
#include "io430.h"
#include <stdint.h> // Integers of defined sizes
#include <stdlib.h>
#include <intrinsics.h>

#include "IState.h"
#include "TrafficLight.h"

#define IN_ 0
#define OUT_ 1

#define ON_ 1
#define OFF_ 0
```

```
class OffState : public IState
```

```
{
    virtual void changeState(TrafficLight* context);
```

```

};

class InitState : public IState
{
    virtual void changeState(TrafficLight* context);
};
//-----
class RedState : public IState
{
    virtual void changeState(TrafficLight* context);
};
class RedBlinkingState : public IState
{
    virtual void changeState(TrafficLight* context);
};
.....

#endif // !CONCRETE_STATE
        Заголовочний файл TrafficLight.h
#pragma once
#ifndef TRAFIC_LIGHT
#define TRAFIC_LIGHT
class IState;

class TrafficLight
{
public:
    TrafficLight();
    ~TrafficLight();
    void nextState();
    void on();
    void off();
    void setState(IState* state, bool withStart);
    IState* getState();
    IState* getPrevState();

private:
    IState* state;
    IState* prevState;
    bool isOff;
};
#endif // !TRAFIC_LIGHT

```

При налагодженні програми на першому етапі було виявлено утікання пам'яті. Після виявлення помилки ознак переповнення RAM не виявлено. Переваги даної реалізації показані у [2] і особливих пояснень не вимагають. Особливо уважно необхідно підходити до помилок, що

пов'язані з звільненням ресурсів. Розмір файлу коду для завантаження, тобто обсяг потрібної ROM склав 3,239 байт. Тип файлу *.hex.

Порівняння першого, другого і третього варіантів програми дає можливість зробити висновок про збільшення майже у 2.5 рази обсягу потрібної ROM при використанні класичного варіанту КА для пристроїв на ММ. Таким чином можна зробити наступні висновки:

- використання архітектури ПЗ у вигляді КА із застосуванням ООП дає перевагу у якості налагодження і супроводження ПЗ, що не суперечить теоретичним положенням [2-5], отже архітектура КА може використовуватися для розробки вбудованих систем на основі ММ;
- збільшення обсягу вихідного коду при застосуванні класичної архітектури КА із застосуванням ООП призводить до подвійного збільшення потрібної пам'яті програм ROM, це свідчить про доцільність використання архітектури нескінченного циклу і процедурної моделі КА у деяких випадках реалізації ПО для вбудованих систем на основі ММ;
- під час реалізації класичної архітектури КА із застосуванням ООП можливі небезпечні помилки - "утікання пам'яті";
- потрібно додатково дослідити доцільність комбінованого варіанту реалізації КА (приклад вихідного коду 3);
- доцільно провести дослідження більш складних програм і їх порівняльну реалізацію у різних варіантах КА;
- необхідно провести аналогічне дослідження для операційних систем реального часу для розробки ПЗ вбудованих систем на базі ММ.

ПЕРЕЛІК ПОСИЛАНЬ

1. Цилькер Б. Я. Организация ЭВМ и систем / Цилькер Б. Я. – СПб.: Питер, 2006.
2. Приемы объектно-ориентированного проектирования. Паттерны проектирования / Э.Гамма, Р. Хелм, Р. Джонсон, Д. Влссидес. – СПб.: Питер, 2016. – 366 с.
3. Фримен Э. Паттерны проектирования / Э. Фримен, Э. Фримен. – СПб: Питер, 2016. – 656 с.
4. Design Patterns – Elements of Reusable Object-Oriented Software <http://www.awprofessional.com/bookstore/product.asp?isbn=0201633612&rl=1> Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides ISBN 0-201-63361-
5. UML Tutorial: Finite State Machines <http://rd13doc.cern.ch/Atlas/DaqSoft/sde/uml/UMLFSM.pdf> Robert Martin C++ Report, June 1998

**ОБРОБЛЕННЯ ІНФОРМАЦІЇ В
СКЛАДНИХ СИСТЕМАХ**

**INFORMATION PROCESSING IN
COMPLEX SYSTEMS**

Передача даних в автомобільній LIN-мережі на базі мікроконтролерів Cypress

Самотий В.В.
професор

Краківська політехніка ім. Тадеуша
Костюшки
Львівський державний університет
безпеки життєдіяльності
Польща, Краків; Україна, Львів

Павельчак А.Г.
доцент

Національний університет “Львівська
політехніка”
Україна, Львів

Дзелендзяк У.Ю.
доцент

Національний університет “Львівська
політехніка”
Україна, Львів

Запропоновано модель даних та механізми їхнього оброблення для забезпечення роботи LIN-протоколу на базі мікроконтролерів Cypress.

Ключові слова: LIN, Cypress, мікроконтролер.

ВСТУП

Частка електронних компонентів в автомобілі зростає з року в рік. Електроніка грає вирішальну роль не тільки в задоволенні побажань клієнтів, для кращої безпеки і комфорту водіння, але і для досягнення кращої економії палива та зменшення викидів вихлопних газів. Реалізація всіх цих функцій забезпечується завдяки обміну даними між електронними компонентами за допомогою систем послідовних шин автомобіля. Завдяки конкуренції забезпечується все більше функцій безпеки і зручності в автомобілі. Це гарантується не лише постійним збільшенням електронних компонентів в транспортних засобах, але також зростанням обсягу передачі даних, без яких більшість нових функцій автомобіля не може обійтися. Для підтримки зростаючої складності автомобільної електроніки, виробники створюють різні стандарти для функціональних і комунікаційних рівнів автомобільних систем. Популярними є такі непатентовані стандарти зв'язку як CAN, LIN, MOST і FlexRay, які забезпечують чітку прозорість на рівні комунікацій [1-2].

CAN (Controller Area Network) використовується в основному в трансмісії, шасі та відповідних частинах автомобіля. LIN (Local Interconnected Network) застосовується для простої і економічно ефективної передачі даних в зоні датчика/виконавчого механізму. MOST (Media Oriented System Transport) реалізується в інформаційно-розважальних системах для передачі відео і аудіо сигналів. І, нарешті, FlexRay забезпечує найбільш складний зв'язок з критичними до безпеки розподіленими програмами. CAN був розроблений на початку 1980-х років Robert Bosch GmbH, а в 1994 році він став міжнародним стандартом (ISO 11898). LIN, MOST і FlexRay вийшли від таких організацій: The LIN Consortium (www.lin-subbus.org), MOST Cooperation (www.mostcooperation.com) і FlexRay Group (www.flexray.com). Незважаючи на те, що вони не були офіційно стандартизовані, їх можна вважати де-факто стандартами.

РЕАЛІЗАЦІЯ СИСТЕМИ ПЕРЕДАЧІ ДАНИХ ЗА LIN ПРОТОКОЛОМ

Для дослідження можливостей щодо передачі даних за LIN-протоколом [3] була розроблена система передачі даних з використанням мікроконтролерів фірми Cypress (рис. 1).

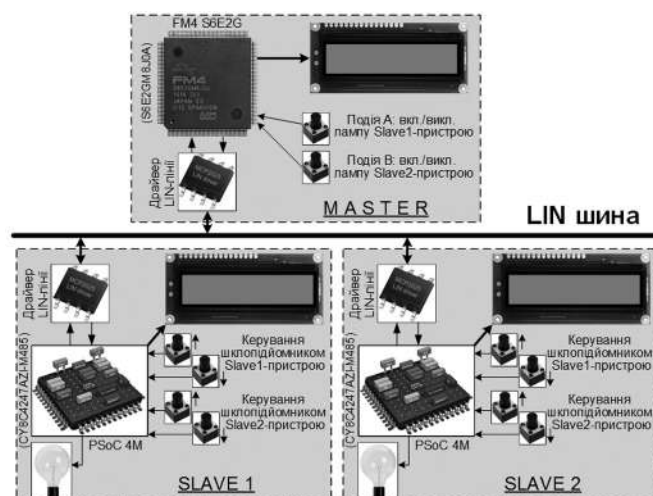


Рис. 1. Структурна схема системи передачі даних

Розроблена система передачі даних складається з трьох модулів:

- Головного master-пристрою, що відповідає за керування передачею даних на LIN шині. Крім того, він управляє включеннями/виключеннями ламп на підлеглих slave-пристроях.
- Двох підлеглих slave-пристроїв, що виконують прийом-передачу даних та управління виконавчими елементами, згідно повідомлень головного master-пристрою. Окрім цього, ці модулі можуть здійснювати управління виконавчими елементами один одного.

Головний модуль складається з ARM мікроконтролера (Cortex-M4) зі FM4 S6E2G серії [6], підлеглі модулі складаються з чіпів PSoc 4M [5], а також ці всі модулі містять драйвери LIN-лінії MCP2025 (Microchip) [4] та LCD-дис-

плеї на базі контролера HD44780 для виводу інформації щодо обміну даними на LIN шині.

PSoC 4M реалізують LIN протокол лише для slave-вузлів. Швидке налаштування передачі даних забезпечується в IDE PSoC Creator за допомогою компоненти LIN Slave. Компонент LIN Slave реалізує для підлеглого LIN пристрою протокол LIN 2.2 на PSoC 3, PSoC 4 і PSoC 5LP кристалах. Також доступні варіанти LIN 2.0, LIN 1.3 або SAE J2602-1. Цей компонент складається з апаратних блоків, необхідних для зв'язку на LIN шині, а також API-функцій, щоб код додатка міг легко взаємодіяти з LIN шиною зв'язку. Компонент надає необхідний API, що відповідає специфікації LIN 2.2. Цей компонент забезпечує хороше поєднання гнучкості і простоти використання. Наданий налаштувач для компонента дозволяє легко налаштувати всі параметри LIN Slave. Для PSoC 4 пристроїв компонент LIN Slave сертифікований C&S group GmbH на основі стандартних протоколів і випробувань на відповідність канальному рівню.

Периферія Multi-Functional Serial interface модуля мікроконтролера серії FM4 дозволяє реалізувати кілька комунікаційних послідовних протоколів: UART, CSIO (підтримує SPI та I2S), I2C, LIN. Режим інтерфейсу Local Interconnect Network (LIN) забезпечує обмін даними по протоколу LIN. Режим LIN в мікроконтролері FM4 має такі особливості: підтримка протоколу LIN версії 2.1; повнодуплексний подвійний буфер; буфер передачі / прийому FIFO (до 128 байт кожен); підтримка master/slave режимів; функція генерації поля LIN break (довжина варіюється між 13 і 16 бітів); функції виявлення різних помилок (помилки парності, кадрування і помилки переповнення буфера).

MCP2025 надає фізичний інтерфейс для напівдуплексного зв'язку між мікроконтролером та LIN шиною. Він призначений для передачі даних зі швидкістю до 20 кбод на послідовній шині. Ця мікросхема виконує перетворення CMOS/TTL логічних рівнів до LIN логічних рівнів, а також у зворотному порядку. Має два режими пониженого енергоспоживання. На рис. 2 зображено принципову схему системи передачі даних, зібрану на основі KITів фірми Supress.

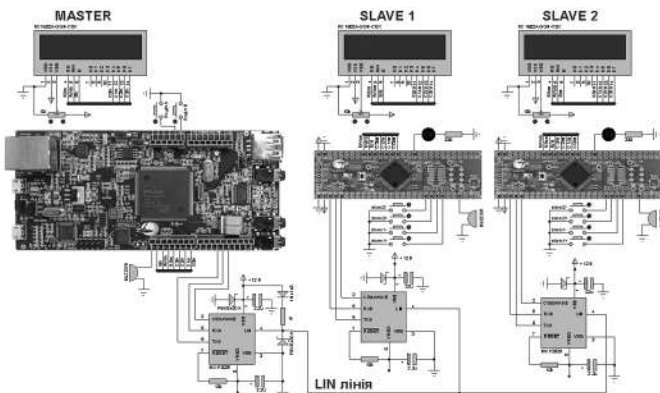


Рис. 2. Принципова схема системи передачі даних

Особливість використання LIN протоколу для передачі даних в автомобільних шинах полягає в його простоті і

дешевизні. Це забезпечується тим, що якщо простіші засоби для передачі даних і реалізовані апаратно, то увесь основний функціонал має бути виконаний програмно. Тому нами була розроблена детальна модель для передачі даних за допомогою LIN інтерфейсу.

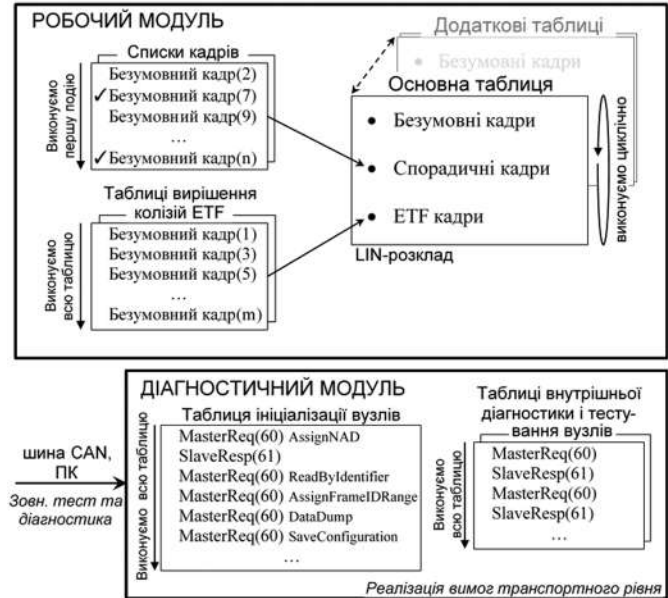


Рис. 3. LIN модель master-вузла для передачі даних

Така LIN модель для передачі даних має забезпечувати: обмін даними на LIN шині; початкову ініціалізацію slave-вузлів; діагностику та тестування вузлів. На рис. 3 зображено структуру даних master-вузла для цієї LIN моделі. Основна робота по організації передачі даних покладається на Master-вузол. Він виконує ініціалізацію усіх вузлів, присутніх на LIN шині, тобто здійснює діагностику наявних Slave-вузлів. Далі він забезпечує диспетчеризацію робочих повідомлень як між ним і робочими вузлами, так і обмін даними між самими вузлами.

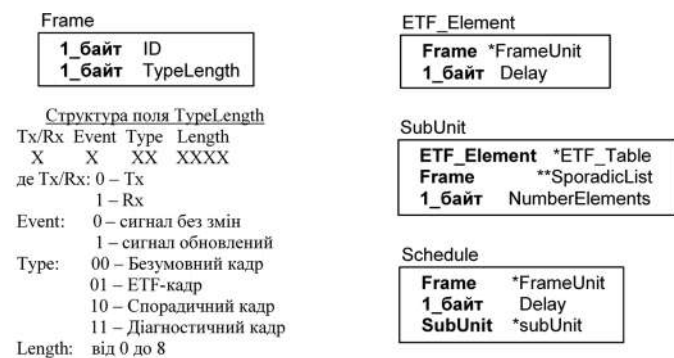


Рис. 4. Структура даних робочого модуля LIN-моделі

На рис. 4 зображено структуру даних для робочого модуля, що забезпечує передачу даних згідно диспетчера завдань (LIN-розкладу). Структура даних Frame відображає окремо взятий кадр. Масив таких елементів дає можливість створити перелік необхідних кадрів. Для спорадичних кадрів значення ідентифікатора має бути рівним 0xFF.

Для розробленої нами системи передачі даних був складений LIN розклад. Він передбачає різні варіанти передачі даних.

T	Slave-Задача Кадр	Кількість байтів, зміст	Slave-задача Master	Slave-задача 1	Slave-задача 2
3	Безумовний кадр ID=1	4 поточні налаштув.	Відправник	Приймач	
3	Безумовний кадр ID=2	4 поточні налаштув.	Відправник		Приймач
5	Спорадичний кадр ID=5	1 див. список кадрів	Відправник	?	?
3	Безумовний кадр ID=3	4 зовн+вн давач темп	Приймач	Відправник	
3	Безумовний кадр ID=4	4 зовн+вн давач темп	Приймач		Відправник
5	ETF-кадр ID=8	3 NAD+вікно1+вікно2	Приймач	?	?
4	Безумовний кадр ID=11	2 вікно1+вікно2	Відправник	Приймач (вікно1)	Приймач (вікно2)
2	Безумовний кадр ID=12	0 вказівка	Відправник	Приймач	
2	Безумовний кадр ID=13	0 вказівка	Відправник		Приймач
2	Безумовний кадр ID=14	0 вказівка	Відправник	Приймач	Приймач
3	Безумовний кадр ID=15	1 внутрішня інф орм.		Відправник	Приймач
3	Безумовний кадр ID=16	1 внутрішня інф орм.		Приймач	Відправник

Таблиця вирішення ETF-колізії для ETF-кадру з ID=8

T	Slave-Задача Кадр	Кількість байтів, зміст	Slave-задача Master	Slave-задача 1	Slave-задача 2
2	Безумовний кадр ID=9	2 вікно1+вікно2	Приймач	Відправник	
2	Безумовний кадр ID=10	2 вікно1+вікно2	Приймач		Відправник

Список приєднаних безумовних кадрів для спорадичного кадру з ID=5

Slave-Задача Кадр	Кількість байтів, зміст	Slave-задача Master	Slave-задача 1	Slave-задача 2
Безумовний кадр ID=9	2 подія A	Відправник	Приймач	
Безумовний кадр ID=10	2 подія B	Відправник		Приймач

Рис. 5. LIN-диспетчеризація

Для апробації розробленої системи передачі даних був складений макет на основі готових КІТів фірми Cypress. Для спрощення візуалізації ми вибрали завищений часовий інтервал міні-слоту в 1 секунду. Складений LIN розклад був відпрацьований на макеті коректно та в повному обсязі.

ПЕРЕЛІК ПОСИЛАНЬ

1. Eugen Mayer. Status quo and future of ECU communication in automobiles. – AUTOMOTIVE, Vector Informatik. – October 2006. – pp. 21-23.
2. Technical Papers on Embedded Network Solutions. 5th Edition. – Vector Informatik GmbH, Stuttgart. – February 2014. – 419 pages.
3. LIN Specification Package. Revision 2.2A. – LIN Consortium. – 2010. – 194 pages.
4. MCP2025. LIN Transceiver with Voltage Regulator. Datasheets – Microchip Technology Inc. – 2012. – 36 pages.
5. 10. PSoC 4100M/4200M Family. PSoC® 4 Architecture Technical Reference Manual. Application note. – Cypress Semiconductor Corporation. – 2015. – 334 pages.
6. S6E2G Series 32-bit ARM® Cortex®-M4F, FM4 Microcontroller. Datasheet – Cypress Semiconductor Corporation. – 2016. – 190 pages.
7. Павелчак А. Г., Яцук Ю. В. Програмна модель даних для забезпечення роботи автомо-більного LIN-протоколу // Автоматика, вимірювання та керування: Вісник НУ «Львівська політехніка». – 2014. – №802. – С. 71-81.

Високопродуктивний алгоритм зіставлення патернів у часових рядах для задач аналізу якості поточкових сервісів

Дорошенко Анатолій Юхимович
доктор фізико-математичних наук, професор
КПІ ім. Ігоря Сікорського
Україна, Київ

Тітов Дмитро Сергійович
аспірант
КПІ ім. Ігоря Сікорського
Україна, Київ

Розглянуте застосування алгоритмів та технік аналізу часових рядів в задачах моніторингу та прогнозування якості передачі даних в поточкових сервісах, таких як VoIP. Запропоноване покращення алгоритму знаходження подібних патернів часових послідовностей з точки зору продуктивності обчислень.

Ключові слова: VoIP; QoS; MOS; джитер; патерн; часовий ряд; PAA; SAX.

1. ВСТУП

Практична проблема забезпечення якості обслуговування (Quality of Service, QoS) істотних обсягів різноманітного мережевого трафіку в умовах невизначеності динаміки мережевих потоків та мережевого середовища набуває важливого значення у сучасних телекомунікаційних мережах [1]. QoS є необхідним для пакетних мереж, які використовуються для сервісів працюючих у режимі реального часу, насамперед VoIP. Мережеві протоколи, які обслуговують сервіси реального часу є чутливими до якості обслуговування, а саме до втрати пакетів даних, затримок у передачі пакетів та нерівномірності цих затримок [2].

Для оцінювання якості надання послуг використовують показник оцінювання якості сприйняття послуг QoE (Quality of Experience), який прямопропорційно залежить від показника якості надання сервісу QoS. В свою чергу, для оцінки QoE найчастіше використовують величину MOS (Mean opinion score). MOS – це середнє арифметичне по всіх окремих «значеннях за задалегідь визначеною шкалою, що суб'єкт приписує його враженню про продуктивність системи в цілому». Такі рейтинги, як правило, зібрані в тесті суб'єктивної оцінки якості, але вони також можуть бути алгоритмічно оцінені [3].

2. ОГЛЯД РОБІТ

Існує дуже багато праць, присвячених різноманітним підходам до аналізу трафіку, алгоритмам їх моніторингу та навіть передбачення показників якості. Наприклад, роботи [4-7] демонструють, що VoIP-трафік має властивості самоподібності та може містити у собі довготривалі залежності, а у роботі [8] розглядається побудова QoS моделей VoIP-трафіку, заснованих на мультифрактальності та Марковських моделях.

У роботі [9] розглянута можливість прогнозування продуктивності мережевих інфраструктур за допомогою Марковських процесів, а в [10] пропонується використання нейромережевих підходів та лінійної регресії, як технік прогнозування показників навантаження в мультисервісних мережах.

Низка робіт присвячена можливостям представлення VoIP-показників за допомогою нечіткої логіки [11] або передбаченню рівня QoS з використанням штучних нейронних мереж [12]. Також існують праці, що описують підходи до моніторингу QoE високорівневими засобами, наприклад, аналіз соціальних мереж у реальному часі [13].

3. ПАРАМЕТРИ VOIP-ТРАФІКУ, ЩО ВПЛИВАЮТЬ НА ЯКІСТЬ ПОСЛУГ

Голосові кодеки вимагають постійного, надійного потоку пакетів, щоб забезпечити прийнятну якість відтворення. Пакети, які прибувають занадто рано, занадто пізно, або з ривками, спотворюють відтворення аудіосигналу. Це явище називається джитером.

Джитер – це небажані фазові та/або частотні випадкові спотворення при передачі сигналу. У той час як джитер-буфер може успішно замаскувати невеликі проблеми затримки, суттєвий джитер може призвести до переповнення джитер-буферу, що в свою чергу призводить до втрати пакетів.

Найбільш важливим фактором для якості зв'язку є “burstiness” – так звані сплески. Сплески є періодами, що характеризуються високим рівнем втрати пакетів [14] (рис. 1).

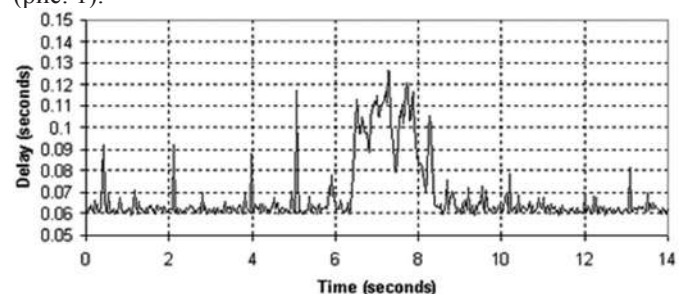


Рис. 1. Приклад спалаху на графіку затримки пакетів

4. ІДЕНТИФІКАЦІЯ ПАТЕРНІВ ПАРАМЕТРІВ ЯКОСТІ МЕРЕЖЕВОГО ТРАФІКУ

Дуже важливу роль в аналізі часових рядів, в тому числі в задачах моніторингу якості поточкових сервісів, віді-

грає pattern matching – знаходження “подібних” фрагментів у послідовності даних (рис. 2).

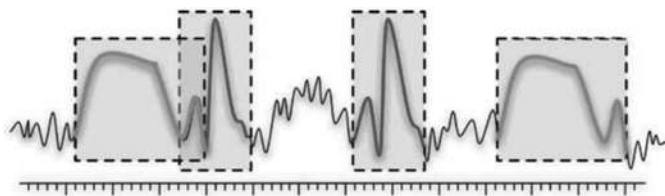


Рис. 2. Знаходження подібних фрагментів у часовому ряді

В цій роботі розглядається символічний алгоритм кодування часових послідовностей – SAX (Symbolic Aggregate approXimation). Він має низку переваг, таких як можливість хешування даних, застосування Марковських моделей, суфіксних дерев, дерев прийняття рішень та ін. [15]

Алгоритм SAX дозволяє описати часовий ряд довільної довжини n рядком довільної довжини w ($w < n$, як правило $w \ll n$). Рядок складається з символів певного алфавіту, розмір якого – довільне ціле число α , де $\alpha > 2$ [16].

Алгоритм відрізняється від інших [17] процедурою дискретизації, в якій використовується проміжне представлення часового ряду (РАА, Piecewise Aggregate Approximation).

Часовий ряд C довжини n може бути представлений у просторі розмірністю w вектором $\bar{C} = \bar{c}_1, \dots, \bar{c}_w$. i -й елемент \bar{C} обчислюється за наступною формулою:

$$\bar{c}_i = \frac{w}{n} \sum_{j=\frac{n}{w}(i-1)+1}^{n_i} c_j$$

Далі РАА-перетворенню ставиться у відповідність символічна послідовність з використанням алгоритму SAX, який на основі квантилів наявного розподілу ставить у відповідність до кожного амплітудного рівня символ із попередньо визначеного алфавіту A . Квантили відсікають в межах ряду певну частину його членів. Тобто, квантиль розподілення значень — це таке число x_p , що значення p -ї частини сукупності менше або рівне x_p (рис. 3).

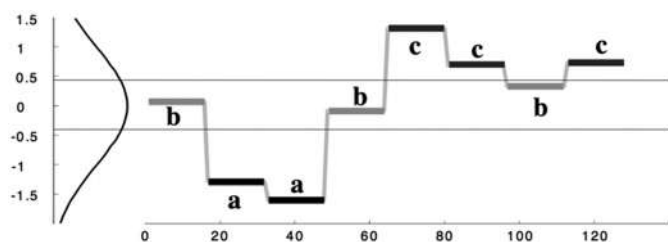


Рис. 3. Часовий ряд перетворюється у символічну послідовність “baabccbc” з параметрами $n = 128$, $w = 8$, $\alpha = 3$

Далі для знаходження певних сталих патернів часовий ряд сканується відрізком певної довжини з заданим зсувом та кожний відрізок порівнюється з оригіналом (sliding window).

5. ПОКРАЩЕННЯ ПРОДУКТИВНОСТІ ЗНАХОДЖЕННЯ ПОДІБНИХ ПАТЕРНІВ У ЧАСОВИХ РЯДАХ

Підхід “sliding window” має суттєвий недолік з огляду на продуктивність обчислень. Якщо стоїть задача для двох закодованих часових послідовностей знайти всі співпадаючі підпослідовності, доводиться виконувати повне сканування вихідних послідовностей декілька разів, оскільки попередньо ми не знаємо якої довжини подібні патерни можемо зустріти.

Очевидно, що в задачах реального часу, таких як аналіз VoIP-трафіку, така ітеративність алгоритму є неприйнятною, адже повний перебір усіх можливих вікон для співставлення даних унеможливить своєчасну обробку інформації.

Протягом досліджень нами було помічено, що якщо відобразити значення відстаней між усіма вікнами одного розміру у вигляді двовірної карти відстаней, у такому відображенні можна виділити геометричні примітиви, які вказують на подібність підпослідовностей значно більшої довжини, ніж довжина вікна, яка використовувалась для складання візуальної карти даної ітерації (рис. 4).

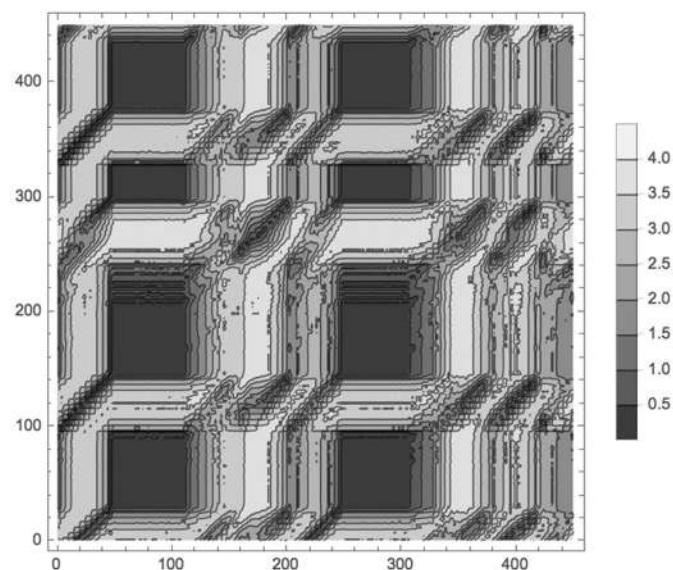


Рис. 4. Двовірні карта відстаней підпослідовностей з розміром вікна 50

Якщо зіставити, використовуючи осі абсцис та ординат, відрізки вихідних послідовностей, які відповідають діагональним відріzkам малих відстаней, то можна побачити, що ці фрагменти послідовностей є подібними. Наприклад, на рисунку 4 відрізок між точками (20; 0) та (120; 100) відповідає наведеним нижче інтервалам аналізованих часових рядів (рис. 5).

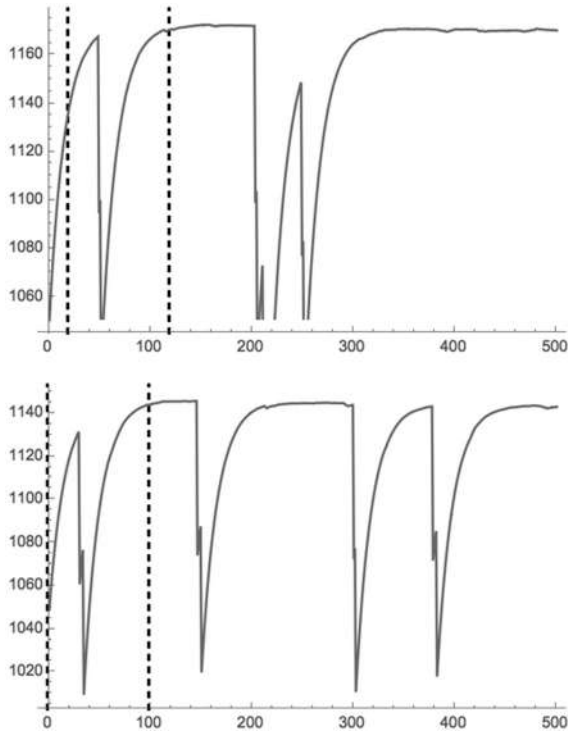


Рис. 5. Знайдені подібні підпоследності у двох часових последностях

Як бачимо, з використанням вікна однієї довжини (50) вдалося знайти подібні патерни довжиною 100 елементів (без необхідності у додаткових 50 ітераціях, щоб збільшити розмір вікна до 100).

6. ЕКСПЕРИМЕНТАЛЬНЕ ПОРІВНЯННЯ ПРОДУКТИВНОСТІ КЛАСИЧНОГО ТА УДОСКОНАЛЕНОГО ПІДХОДІВ

Розроблені в процесі дослідження Java-додатки використовують спільні алгоритми для проведення РАА-та SAX-перетворень, проте різні процедури порівняння отриманих даних. Перший додаток для знаходження схожих часових патернів використовує почергове сканування даних вікнами змінної довжини (з 50 до 140), а другий базується на підході, запропонованому у п'ятому розділі. Тестові дані для аналізу взяті з реальної VoIP-сесії та екстраговані за допомогою програми Wireshark (рис. 6).

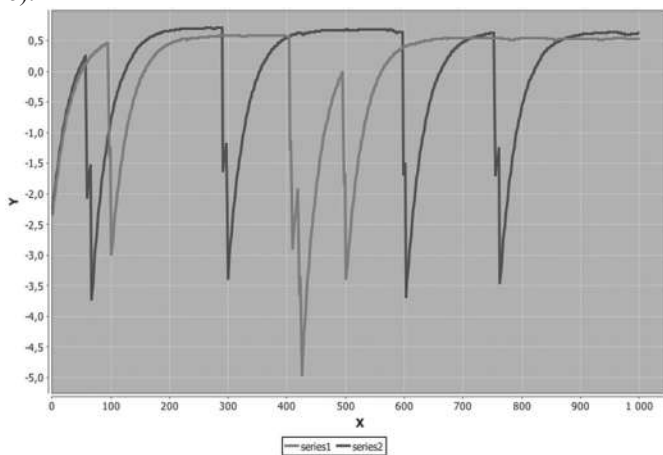


Рис. 6. Тестові дані – джитер

Обидві програми знайшли кілька співпадаючих патернів, найдовшим з яких є наступний (рис. 7).

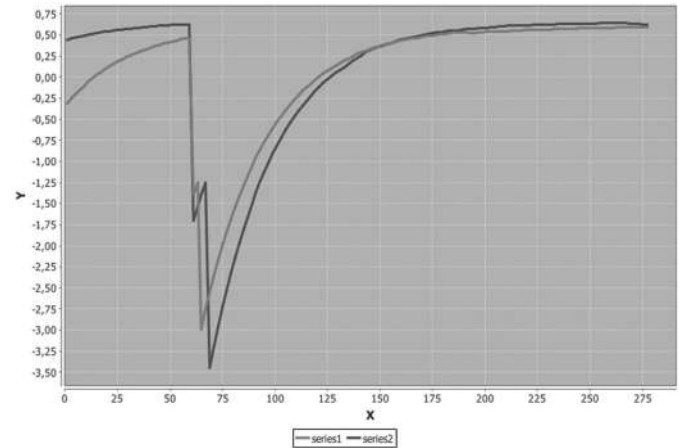


Рис. 7. Приклад знайденого патерну

Також ми провели тест існуючої реалізації SAX, що називається jMotif. Нижче наведений графік, який демонструє, скільки часу в мілісекундах пішло на сканування кожним новим вікном у класичному підході (власна імплементація та jMotif, 90 ітерацій), та скільки часу зайняв модернізований підхід (одна ітерація) (рис. 8).

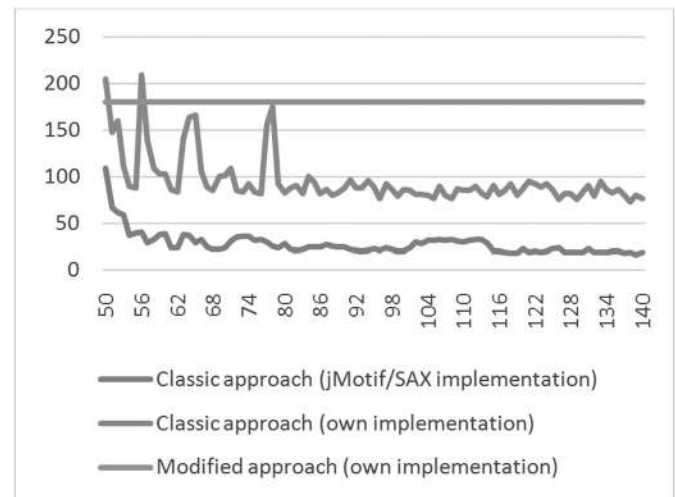


Рис. 8. Порівняльна діаграма швидкодії

Загалом повний цикл роботи програми, заснованої на класичному підході, зайняв 8740 мс та 2543 мс (власна та jMotif імплементація відповідно), у той час як виконання модернізованого алгоритму зайняло лише 180 мс, що вказує на приріст швидкодії більш ніж у 14 разів порівняно з jMotif та 48 разів порівняно з власною імплементацією.

ВИСНОВКИ

У роботі проаналізоване застосування алгоритмів та технік обробки часових рядів в задачах моніторингу та прогнозування якості передачі даних в потокових сервісах, таких як VoIP. Показано, які саме параметри впливають на якість аудіосигналу в голосових сесіях, та як можна їх представити у вигляді часового ряду.

Розглянуто символічний алгоритм кодування часових послідовностей SAX, який має низку переваг перед аналогами та оперує власною мірою відстаней між послідовностями. Запропоноване покращення алгоритму знаходження подібних патернів у часових послідовностях з точки зору продуктивності обчислень.

Створено два Java-додатки для знаходження всіх схожих підпослідовностей у часових рядах, які експериментально підтверджують перевагу у швидкодії запропонованого алгоритму порівняно з класичним та перевагу у швидкодії з існуючою імплементацією SAX – jMotif.

ПЕРЕЛІК ПОСИЛАНЬ

1. Славко О.Г. Інформаційна технологія керування перевантаженнями в мультисервісних телекомунікаційних мережах / Славко О.Г. – 2011 – Вісник Кременчуцького національного університету імені Михайла Остроградського – Вип. 2 (67), част. 1. – С. 29-34.
2. Лаврів О.А. Модель системи управління ресурсами мультисервісних мереж в умовах самоподібності трафіку / Лаврів О. А. – 2012 – Вісник Національного Університету «Львівська політехніка». Радіоелектроніка та телекомунікації. – С. 165-172
3. Чернихівський Є.М., Червенець В. В., Білик О. Б. Визначення часових параметрів обслуговування потокового трафіку з пріоритетними класами / Чернихівський Є.М. – Вісник Національного університету «Львівська політехніка». – 2011. – № 705 : Радіоелектроніка та телекомунікації. – С. 167-170.
4. Rezaul K., Grout V., A Survey of Performance Evaluation and Control for Self-Similar Network Traffic, Proceedings of the Second International Conference on Internet Technologies and Applications (ITA), 4-7 September 2007, pp. 514-524.
5. Samorodnitsky G., Long Range Dependence, Foundations and Trends in Stochastic Systems, 1 (3) (2007) 163-257.
6. Zhang G., Xie G., Yang J., Zhang D., Self-similar characteristic of traffic in current metro area network, Proceedings of the 15th IEEE Workshop on Local and Metropolitan Area Networks, 10-13 June 2007, pp. 176-181.
7. Veitch D., Hohn N., Abry P., Multifractality in TCP/IP traffic: the case against, Computer Networks, 48 (3) (2005) 293-313.
8. Toral-Cruz, Homero, Pathan, Al-Sakib Khan, Pacheco, Julio Cesar Ramirez Accurate modeling of VoIP traffic QoS parameters in current and future networks with multifractal and Markov models – Mathematical and Computer Modelling 57 , no. 11-12 (2013): 2832-2845.
9. Pacheco-Sanchez S., Casale G., Scotney B., McClean S., Parr G., Dawson S. Markovian Workload Characterization for QoS Prediction in the Cloud – 2011 IEEE 4th International Conference on Cloud Computing. 147154 (2011).
10. Islam S., Keung J., Lee K., Liu A.: Empirical prediction models for adaptive resource provisioning in the cloud. Future Generation Computer Systems. 28, 155162 (2012).
11. Mansouri, Taha, Nabavi, Ali, Ravasan, Ahad Zare and Ahangarbahan, Hamid. «A practical model for ensemble estimation of QoS and QoE in VoIP services via fuzzy inference systems and fuzzy evidence theory.» Telecommunication Systems 61 , no. 4 (2016): 861-873.
12. Ролик А.И. Оценка качества предоставления мультимедийных сервисов с использованием нейросетевого классификатора / Ролик А.И., Галушко Д.А., Барна В.В., Томашук А.В., Ясочка М.В. – Вісник НТУУ «КПІ». Інформатика, управління та обчислювальна техніка: збірник наукових праць. – К.: Століття+, 2015. – №63. – С. 25–30.
13. Тітов Д.С. Моніторинг соціальних мереж в системі реального часу / Тітов Д.С., Дорошенко А.Ю. – К.: Наукова дискусія: теорія, практика, інновації, 2015. – С. 93-96.
14. Lingfen S. Speech quality prediction for voice over Internet protocol networks: PhD diss.: 2004 / Lingfen Sun. – University of Plymouth, UK, 2004. – 218pp.
15. Lin, J., Keogh, E., Lonardi, S. & Chiu, B. (2003) A Symbolic Representation of Time Series, with Implications for Streaming Algorithms. In proceedings of the 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery. San Diego, CA. June 13.
16. Lin, J., Keogh, E., Lonardi, S. and Patel P. Finding motifs in time series. In Proc. 2nd Workshop on Temporal Data Mining, 2002. pp. 214-221.
17. Keogh, E., Chakrabarti, K., Pazzani, M., Mehrotra, S. Locally Adaptive Dimensionality Reduction for Indexing Large Time Series Databases. In proceedings of ACM SIGMOD Conference on Management of Data. Santa Barbara, CA, 2001. pp. 151-162.
18. GitHub (2017) Time series symbolic discretization with SAX [Online] Available from <https://github.com/jMotif/SAX> [Accessed: 12 May 2017]

Выбор меры различия образцового и отфильтрованного периодических сигналов при высоких уровнях флуктуационного шума

А.А. Сергеев-Горчинский
к.т.н., ассистент
КПИ им. Игоря Сикорского
Украина, Киев

Рассмотрен выбор меры различия результатов фильтрации искажённых периодических сигналов в задаче поиска оптимальных значений параметра цифрового фильтра нижних частот «простое скользящее среднее». Для сравнения были выбраны меры различия – среднеквадратичная ошибка, корень среднеквадратичной ошибки, средняя абсолютная ошибка. Выполнена серия экспериментов по генерированию и фильтрации искажённых сигналов. В результате сравнения выбрана мера, для которой характерна высокая различимость оптимальных значений параметра цифрового фильтра при высоких уровнях шума.

Ключевые слова: дискретный сигнал, цифровой фильтр, скользящее среднее, оптимальный параметр, мера различия

1. ВВЕДЕНИЕ

При решении технических задач часто возникает необходимость сравнения временных рядов, составленных из значений регистрируемых сигналов, с целью определения степени их подобия. Подобие временных рядов можно оценить при помощи мер различия значений анализируемых сигналов. К задачам, в которых применяют меры различия относятся задачи адаптивной фильтрации, классификации по образцу, обучения по образцу и некоторые другие. В цифровой обработке сигналов при настройке параметров цифровых фильтров по образцовым сигналам, требуется расчет меры различия значений образцового и отфильтрованного сигналов [1].

В [2] поиск оптимального значения параметра цифрового фильтра нижних частот (ФНЧ) «простое скользящее среднее» (ПСС) осуществлялся при сравнении отфильтрованного и аппроксимированного временных рядов. Для оценки различия временных рядов была применена мера «среднеквадратичная ошибка» (СКО). В прикладных методах анализа данных применяются и другие численные показатели различия временных рядов, которые называются «мерами ошибки» (error measures) [3] либо «метриками расстояния» (distance metrics) [4].

Чтобы проверить целесообразность применения меры СКО для оценки оптимальности фильтрации выполнено исследование характеристик известных мер различия значений дискретных квантованных сигналов.

2. ВЫБОР МЕР РАЗЛИЧИЯ

В процессе исследования была создана библиотека программных модулей для генерирования исходных неискаженных дискретных квантованных сигналов различной формы и их модифицированных копий, полученных путём добавления к исходным сигналам случайной составляющей (помехи) и последующей фильтрации искажённых сигналов. Для минимизации уровня шума был создан

«Модуль нерекурсивной фильтрации сигнала». В модуле реализован программный метод фильтрации простым скользящим средним (ПСС). В фильтре ПСС обработанное значение $y[k]$ в момент времени k определяется выражением [2]:

$$y_{\text{отфильтр}}[k] = \frac{1}{m_1 + m_2 + 1} \times \sum_{i=-m_1}^{m_2} x_{\text{искаж}}[k-i], \quad (1)$$

где $m_1 + m_2 + 1$ – параметр цифрового фильтра m (количество искажённых значений сигнала, участвующих в расчете отфильтрованного значения для заданного момента времени k), $x_{\text{искаж}}[k-i]$ – значение искажённого сигнала в момент времени дискретизации $k-i$, $y_{\text{отфильтр}}[k]$ – сигнал на выходе фильтра в текущий момент времени k .

В [3, 4, 5, 6] рассмотрены различные математические выражения для расчета меры различия числовых последовательностей, которые можно разделить на следующие группы: меры расстояния, угловые меры, корреляционные меры. Учитывая результаты экспериментов, приведенные в [6], для анализа были выбраны меры расстояния: среднеквадратичная ошибка, корень среднеквадратичной ошибки, средняя абсолютная ошибка.

Мера «среднеквадратичная ошибка» (СКО, Mean Square Error, MSE) определяется выражением [2]:

$$M_{\text{СКО}}(\{x_{\text{неискаж}}[k]\}, \{y_{\text{отфильтр}}[k]\}) = \frac{1}{N} \cdot \sum_{k=0}^{N-1} (x_{\text{неискаж}}[k] - y_{\text{отфильтр}}[k])^2, \quad (2)$$

где k – номер отсчёта, $x_{\text{неискаж}}[k]$ – значение неискажённого сигнала в момент времени дискретизации k , $y_{\text{отфильтр}}[k]$ – значение отфильтрованного сигнала в момент времени дискретизации k , N – общее количество отсчётов искажённого сигнала.

Мера «корень среднеквадратичной ошибки» (КСКО, Root Mean Square Error, RMSE) определяется выражением [3]:

$$M_{КСКО}(\{x_{неискаж}[k]\}, \{y_{отфильтр}[k]\}) = \sqrt{\frac{1}{N} \cdot \sum_{k=0}^{N-1} (x_{неискаж}[k] - y_{отфильтр}[k])^2}, \quad (3)$$

где k – номер отсчёта, $x_{неискаж}[k]$ – значение неискажённого сигнала в момент времени дискретизации k , $y_{отфильтр}[k]$ – значение отфильтрованного сигнала в момент времени дискретизации k , N – общее количество отсчётов искажённого сигнала.

Мера «средняя абсолютная ошибка» (CAO, Mean Absolute Error, MAE) определяется выражением [4]:

$$M_{CAO}(\{x_{неискаж}[k]\}, \{y_{отфильтр}[k]\}) = \frac{1}{N} \cdot \sum_{k=0}^{N-1} |x_{неискаж}[k] - y_{отфильтр}[k]|, \quad (4)$$

где k – номер отсчёта, $x_{неискаж}[k]$ – значение неискажённого сигнала в момент времени дискретизации k , $y_{отфильтр}[k]$ – значение отфильтрованного сигнала в момент времени дискретизации k , N – общее количество отсчётов искажённого сигнала.

3. ПОСТАНОВКА ЭКСПЕРИМЕНТА

Для оценки характеристик целевых функций сформированных из значений выбранных мер различия при различных значениях параметра цифрового фильтра, был сгенерирован тестовый искажённый синусоидальный сигнал со следующими характеристиками: амплитуда сигнала – 1 В; частота сигнала – 1 Гц; распределение шумовой составляющей – нормальное; среднеквадратичное отклонение – 0.7 В; частота дискретизации – 100 Гц, продолжительность регистрации – 10 с.

Чтобы оценить уровень шума было рассчитано значение отношения сигнал/шум (ОСШ), которое определяется выражением [7]:

$$ОСШ = 10 \cdot \log_{10} \left(\frac{\sum_{k=0}^{N-1} (x_{неискаж}[k])^2}{\sum_{k=0}^{N-1} (x_{искаж}[k] - x_{неискаж}[k])^2} \right), \quad (5)$$

где k – номер отсчёта, $x_{неискаж}[k]$ – значение неискажённого сигнала в момент времени дискретизации k , $x_{искаж}[k]$ – значение искажённого сигнала в момент времени дискретизации k , N – общее количество отсчётов искажённого сигнала.

Значение ОСШ для сгенерированного сигнала равно 0.14 дБ. Сгенерированный сигнал был обработан при помощи фильтра ПСС со значениями параметра m в диапазоне от 1 до 250, где m – количество искажённых значений сигнала, участвующих в расчете отфильтрованного значения для заданного момента времени k [см. выражение (1)]. Были проведены 250 сравнений и рассчитаны значения мер различия для двух временных рядов: временных рядов исходного неискажённого сигнала и 250 временных рядов отфильтрованных сигналов.

Поскольку значения мер различия могут находиться вне интервала $[0, 1]$, для приведения мер к одному диапазону значений, в их выражения было включено деление на максимальное значение (среди всех значений для каждой меры различия) и округление до сотых долей (см. рисунок 1).

Из рисунка 1 видно, что нормированные значения мер различия имеют локальные минимальные значения (экстремумы), т. е. могут быть рассмотрены в качестве целевых функций в задаче поиска оптимальных значений параметра цифрового фильтра. Из рисунка 1 следует, что для значений меры СКО визуально можно выделить два локальных оптимальных значения параметра m (см. рисунок 2): $m_1 = 28$; $m_2 = 249$.

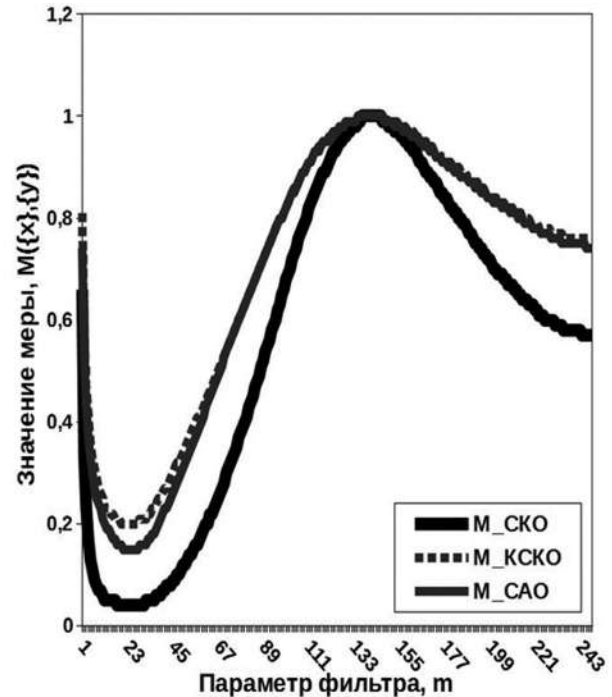


Рис. 1 – Нормированные округленные значения мер различия для значений параметра m от 1 до 250 при ОСШ = 0.14 дБ

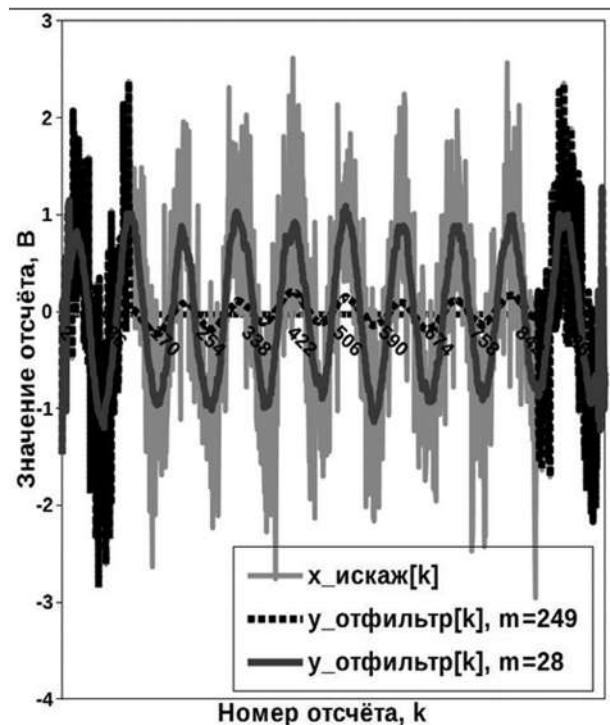


Рис. 2 – Значения отфильтрованных сигналов для заданных значений параметра m при ОСШ = 0.14 дБ

Одинаковым значениям мер различия соответствуют значения расположенные на прямолинейных участках экстремумов целевых функций (см. рисунок 1). Поскольку целью экспериментов было определение меры различия, для которой характерна высокая различимость значений экстремумов целевых функций при высоких уровнях шума, было рассчитано количество значений m с равным значением целевой функции. Количество повторений значений целевых функций изображено в виде гистограммы на рисунке 3.

4. СРАВНЕНИЕ МЕР РАЗЛИЧИЯ

Для оценки распределений количеств значений m при ОСШ от 37.04 дБ до -16.94 дБ была проведена серия экспериментов по расчету значений меры эксцесса (см. рисунок 4), которая определяется выражением (7) [8]:

$$\bar{M} = \frac{1}{N} \cdot \sum_{m=1}^N M(m), \tag{6}$$

$$\varepsilon = \frac{N \cdot \sum_{m=1}^N (M(m) - \bar{M})^4}{\left(\sum_{m=1}^N (M(m) - \bar{M})^2 \right)^2}, \tag{7}$$

где m – параметр цифрового фильтра, $M(m)$ – нормированное округленное значение меры различия для исходного и отфильтрованного (с параметром m) временных рядов, N – количество значений параметра m .

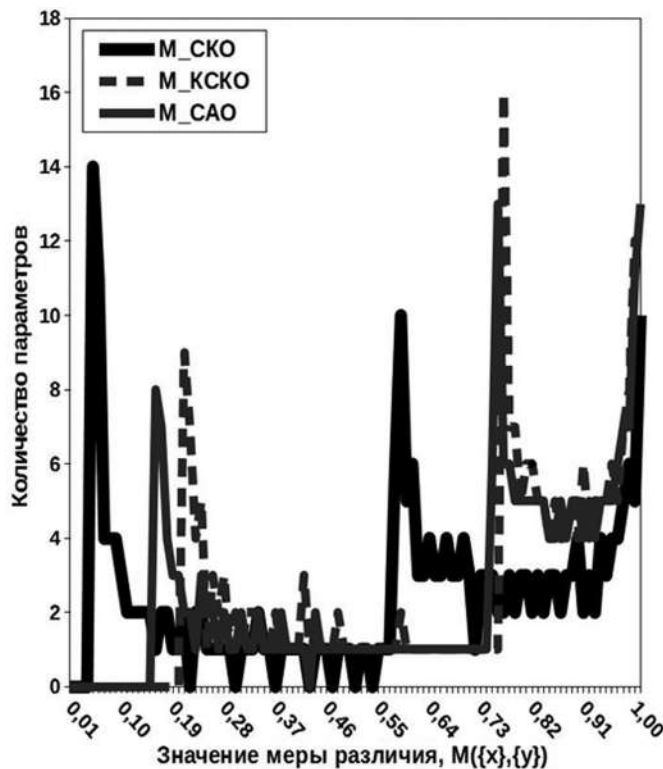


Рис. 3 – Количество значений параметра m от 1 до 250 с равными нормированными округленными значениями мер различия

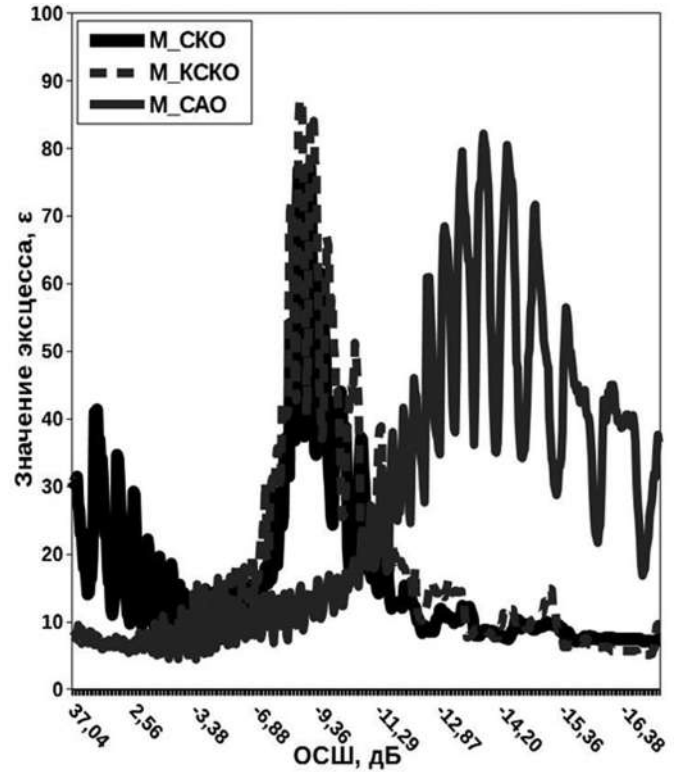


Рис. 4 – Значения эксцесса для количества повторений значений мер различия при заданных ОСШ

Из рисунка 4 следует, что в отличие от мер различия СКО и КСКО, для которых различимость локальных минимумов целевой функции уменьшается при ОСШ ≥ -8.80 дБ, для меры CAO различимость уменьшается при ОСШ ≥ -13.84 дБ, следовательно мера CAO более устойчива к высоким уровням флуктуационного шума.

В целях полноты экспериментов количество форм тестовых сигналов было расширено до четырех: синусоидального, прямоугольного, треугольного и пилообразного. В таблице 1 для всех форм сигналов приведены значения ОСШ соответствующие максимальным значениям эксцесса.

Таблица 1. Значения ОСШ, дБ, соответствующие максимальным значениям эксцесса

Мера различия	Сигнал			
	Синусоидальный	Прямоугольный	Треугольный	Пилообразный
Среднеквадратичная ошибка	-9.11	-8.61	-8.69	-7.90
Корень среднеквадратичной ошибки	-8.80	-8.28	-8.80	-7.83
Средняя абсолютная ошибка	-13.84	-13.88	-13.60	-11.63

Проанализировав значения ОСШ, приведённые в таблице 1, можно сделать вывод, что мера «средняя абсолютная ошибка» более устойчива к высоким уровням шума, поскольку максимумам значений эксцесса соответствуют наименьшие значения ОСШ среди значений, рассчитанных для трёх мер различия, следовательно применение

меры САО позволяет увеличить различимость локальных экстремумов целевых функций в задаче поиска оптимальных значений параметра цифрового фильтра «простое скользящее среднее» при высоких уровнях флуктуационного шума.

ПЕРЕЧЕНЬ ССЫЛОК

1. Сергиенко А.Б. Цифровая обработка сигналов, 3-е изд. / А.Б. Сергиенко // Санкт-Петербург, 2011. – БХВ-Петербург, 2011. – С. 593-595.
2. Sergeev-Horchynskiy A.A. Periodic signal filtration using digital filtering system calculation optimized by approximation / A.A. Sergeev-Horchynskiy, V.S. Rogoza // Science-Based Technologies. – 2014. – Т. 23, № 3. – P. 284-288.
3. Willmott C.J. Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance / C.J. Willmott, K. Matsuura // Climate Research. – Inter-Research, 2005. – Vol. 30. – P. 79-82.
4. Chai T. Root mean square error (RMSE) or mean absolute error (MAE)? – Arguments against avoiding RMSE in the literature / T. Chai, R.R. Draxler // Geoscientific Model Development. – Copernicus Publications, 2014. – Vol. 7. – P. 1247-1250.
5. Cha S.-H. Comprehensive Survey on Distance/Similarity Measures between Probability Density Functions / S.-H. Cha // International journal of mathematical models and methods in applied sciences. – NAUN, 2007. – Issue 4, Vol. 1. – P. 300-307.
6. Рогоза В.С. Выбор меры подобия цифровых сигналов для расчета оптимальных параметров цифрового фильтра / В.С. Рогоза, А.А. Сергеев-Горчинский // Автоматизированные системы управления и приборы автоматики. – 2013. – Т. 164. – С. 50-57.
7. Oppenheim A.V. Discrete-Time Signal Processing, 3rd ed. / A.V. Oppenheim, R.W. Schaffer // London, 2010. – Pearson, 2010. – P. 223-227.
8. Смирнов Н.В. Курс теории вероятностей и математической статистики / Н.В. Смирнов, И.В. Дунин-Барковский // Москва, 1969. – Наука, 1969. – С. 85-94.

ABSTRACTS

INFORMATION SYSTEMS AND TECHNOLOGIES

pp: 13-15

Synchronization in distributed information systems

Yurchyna Oleksii
student
Igor Sikorsky Kyiv Polytechnic Institute
Ukraine, Kyiv

Bugai Alexander
student
Igor Sikorsky Kyiv Polytechnic Institute
Ukraine, Kyiv

The article presents the analysis of features of data synchronization in distributed information systems. Considered distributed information systems and determines their essential characteristics. The author conducts a study of examples of synchronization.

Key words: synchronization, data synchronization, distributed information systems

pp: 16-18

Development of search robot with flexible configuration

Glybovets A.M.
assistant professor, Faculty of Informatics
National University of "Kyiv-Mohyla Academy"
Ukraine, Kyiv

Design and development of search robots - rather not trivial task. The relevance of this theme is that despite the importance of search engines for effective development of search and information systems in general, today still the problem is not completely solved, there is quite a bit of open implementations of robots that we could simply, quickly and flexibly adjust their goals and use in projects. The work proposed one approach to development of search robot with the flexible configuration.

Keywords: information retrieval systems, search robot

pp: 19-22

Approach to the problem of increasing information security companies

Igor Yu. Grishin
professor
Kuban State Technological University
Russia, Krasnodar

Rena R. Timirgaleeva
professor
Kuban State Technological University
Russia, Krasnodar

Maxim V. Mironov
postgraduate
Kuban State Technological University
Russia, Krasnodar

The analysis of the main threats to information security of the enterprise, typical for the current time, was conducted in the work. It is shown that the most important factor influencing this indicator is the choice of the method of user authentication of the information system. It is proposed to use the authentication of the user's keyboard handwriting as an effective method of biometric authentication.

Keywords: information security, threat, authentication, dynamic sign, user, keyboard handwriting.

pp: 23-24

Processor functional control in systems on chip

Zhabin V.I.
Doc. of Sc., Prof.
Dept. of Computer Engineering
Igor Sikorsky Kyiv Polytechnic Institute
Ukraine, Kyiv

Method of processor functional control maintenance on software and hardware level by means of program region flow correctness check is proposed. The method permits small delays during computing process realization and can be used for control of any program flow.

Key words: system on chip, program flow control, processor control.

pp: 25-27

Software services concept

Volodymyr Khmeliuk
Senior Lecturer
Igor Sikorsky Kyiv
Polytechnic Institute
Ukraine, Kyiv

Illia Maier
student
Igor Sikorsky Kyiv
Polytechnic Institute
Ukraine, Kyiv

Mykyta Ozerakin
student
Igor Sikorsky Kyiv
Polytechnic Institute
Ukraine, Kyiv

Oleksandr Tymoshenko
student
Igor Sikorsky Kyiv
Polytechnic Institute
Ukraine, Kyiv

Article contains description of software services concept as continuous and phaseless evolutional process of providing different software services which replace standard software life cycle (planning, coding, testing, evolution, maintenance) and support processes (payment, integration, rent, client-provider interaction)

Keys: software, development, service, provider, client

pp: 28-29

Analysis of join methods in in-memory data grids

Oleksandr Podrubailo
assistant
Igor Sikorsky Kyiv Polytechnic Institute
Ukraine, Kyiv

Yaroslav Lukyanenko
student
Igor Sikorsky Kyiv Polytechnic Institute
Ukraine, Kyiv

This report is devoted to the analysis of distributed join methods for their use in in-memory data grids. Identified factors that distinguish the join in IMDG comparatively to similar operation in the storage systems of another class. The algorithms were compared by transmission volume and computation time. Developed software environment for modelling of join operations in such systems.

Keywords: in-memory data grid, distributed join, hash join, track join, bloom join, modified track join

pp: 30-31

Program blocks parameterization of module sm1281 for constructing the GCU diagnosing system

Zamikhovsky Leonid Mykhailovych
Head of Department of “Information and communication technologies & systems”
Ivano-Frankivsk National Technical University of Oil and Gas
Ukraine, Ivano-Frankivsk

Ivanyuk Nataliya Ivanivna
Assistant of Department of “Information and communication technologies & systems”
Ivano-Frankivsk National Technical University of Oil and Gas
Ukraine, Ivano-Frankivsk

Parameterization procedure of program blocks of the technological module for measuring the vibration parameters SM1281 was examined in order to build the system diagnostics of the gas compressor units based on programmable logic controllers Simatic S7-1200 and design environment TIA Portal V13SP1 (Totally Integration Automation) «Siemens».

Keywords: parameterization, programming block, diagnostics, GCU, vibration rate, settings, compilation.

pp: 32-33

Using of discriminant analysis for selection of diagnostic characteristic of technical state of gas pumping units

Zamikhovsky Leonid Mykhailovych
Head of Department of “Information and communication technologies & systems”
Ivano-Frankivsk National Technical University of Oil and Gas
Ukraine, Ivano-Frankivsk

Pavlyk Volodymyr Vasylyovych
Chief of Service of CMD and A of Bohorodchany LVUMG UMG “Prykarpatttransgas”
Ukraine, Bohorodchany

Procedure of applying the discriminant analysis for determining the technical condition of gas compressor units (GCU) by using the observation results on change its technical parameters in service for different technical conditions of GPA was proposed.

It is shown that for chosen of diagnostic sign it is possible not merely to assess the current technical condition of the GPA, but also to predict its change in pregiven period of service.

Keywords: discriminant analysis, identifier, correlation, diagnostic characteristic, technological parameters, technical condition.

pp: 34-36

Authentication RFID tag based on the one way conversion

Valerii Pohorzhelskyi
“Igor Sikorsky Kyiv Polytechnic Institute”
Ukraine, Kyiv

Vadim Poltorak
“Igor Sikorsky Kyiv Polytechnic Institute”
Ukraine, Kyiv

There is an approach of building a system of authentication for devices with limited resources and memory, such as RFID tags, key chains, smart cards and so on. The stability of a protocol is based on the stability of used crypto primitives and also on the theoretical possibility of stable combining with logic in the process of designing schemes reader device.

Keywords - tag, RFID, symmetric algorithms, authentication systems.

pp: 37-40

Identification of the issues in the E-governance in Jordan and USA

Mohammad Alhawawsha
Taras Shevchnko National University of Kyiv
Ukraine, Kyiv

The application of the “electronic government” is one of the most trending and simplest forms of ICT (“Information and Communication Technology”) application used for governmental services. In spite of the significant advantage and benefits obtained from the e-government, various nations have failed to fully utilize the advantages and successfully implement the system. This paper evaluated the critical factors that impact and determines the failure and success of the e-government implementation project. The article focuses on the investigation of the e-government implementation case in Jordan. The various technical challenges and issues have been identified that has finally lead to the inefficient development and application of the e-government system. The evaluation of the strategies used by the Jordan e-government project team has helped in identification of the technical issues like privacy, security, integrity, reliability, interoperability, cloud management, and content management system. Based on the identified issues, an enhanced model for e-government has been proposed with data authentication, encryption and many other techniques for overcoming the technical issues.

pp: 41-42

Information modeling of transfer processes in heterogeneous environments

Prokopovich I.,

Dukhanina M.

Dobrovolska V.

Saukh I.

Daderko O.

Koshulyan S.

Conductivity in a granular medium for the heat transfer process is studied. Information transport models in heterogeneous media in the form of two interpenetrating polymer networks are proposed. The thermal conductivity of the medium was determined from the integral characteristic of the structures of the first and second order of the model.

Keywords: heterogeneous materials, conductivity, an integral characteristic.

pp: 43-44

Information Technology Management Project Logistics by virtual model

Saveleva, O.

Stanovska, I.

Malahova, D.

Sauh, I.

Gurev, I.

A discrete spatial model adaptive virtual emergency logistics. The model will continuously play the current status of available items in warehouses and resources for the project. Regardless of whether there are some risk event or not. This will provide an opportunity to carry out effective project management.

Keywords: project risks, spatial model, logistics.

pp: 45-47

On the question of assessing the quality of web-site of the company as a model of queuing

Oleg Morgal
Senior Lecturer
Igor Sikorsky Kyiv
Polytechnic Institute
Kyiv, Ukraine

Olena Savchuk
Associate Professor
Igor Sikorsky Kyiv
Polytechnic Institute
Kyiv, Ukraine

Eugene Pokrovskiy
Associate Professor
Igor Sikorsky Kyiv
Polytechnic Institute
Kyiv, Ukraine

Iryna Latash
Student
Igor Sikorsky Kyiv
Polytechnic Institute
Kyiv, Ukraine

The article offers a method of approach to the design of the site through analysis of its structure in the MATLAB environment, and given the results of its application in the design of Web-site automation of the company's automation

Keywords: website, web-service, automation of the company.

pp: 48-50

Using a component approach in the design of the services for business processes management systems

Sergii Trotskyi
Igor Sikorsky Kyiv
Polytechnic Institute
Ukraine, Kyiv

Yurii Pohorilyi
Igor Sikorsky Kyiv
Polytechnic Institute
Ukraine, Kyiv

Mykyta Kolomiichuk
Igor Sikorsky Kyiv
Polytechnic Institute
Ukraine, Kyiv

Yevgeny Vovk
Igor Sikorsky Kyiv
Polytechnic Institute
Ukraine, Kyiv

Describes usage of service-oriented architecture for the construction of the business processes execution systems. Also presents general mathematical model for such systems and the advantages of service-oriented architecture on the example of the business process which described in BPEL language.

Key words: SOA - Service-oriented architecture, SOAP - Simple Object Access Protocol, XML- Extensible Markup Language, Web-service.

pp: 51-52

Optimization of CAD equal-tensied machine parts

Stanovskiy, O.
Ukraine, Odessa

Shvets, P.
Ukraine, Odessa

Toropenko, A.
Ukraine, Odessa

Naumenko, E.
Ukraine, Odessa

Abu Shena Osama
Ukraine, Odessa

A method is proposed for the design of welded joints of cylindrical tanks containing planar circular plates of variable thickness, the shape of the diametrical cross-sections of which is modeled by the Gauss equation. To solve the bending equation for these plates, we use the degenerate hypergeometric functions of Kummer and Whittaker.

Keywords: equal-tensied machine parts, plate of variable thickness, CAD

pp: 53-55

Method to reservation and restore data in distributed data storage systems

Vinogradov Y. N.
sr. lect.

Igor Sikorsky Kyiv Polytechnic Institute
Ukraine, Kyiv

Ivanov V. G.
stud.

Igor Sikorsky Kyiv Polytechnic Institute
Ukraine, Kyiv

In this work, a method for backup and restoring data is proposed. The data is stored on different remote disks and temporary or permanent loss of access to any n of them appears. The high efficiency of the proposed method is achieved by using a theoretically proved minimum number of backup copies of the data, which is equal to the number of the disks to which access has been lost. Additionally, an extremely simple computational procedure to ensure high-speed reconstruction of lost data is employed. The theoretical justification with a methodology of backup codes and their use for data recovery.

Keywords: remote distributed information storage systems, data reconstruction, erasures codes, remove data access, reliability of remote data storage.

pp: 56-57

Method for strong abonents identification in telecommunication systems by using transformations on Galoise fields

Zacharioudakis Eleftherios
senior lecture
Macarios educational centre
Cyprus

Oliievskiyi Andrii
student
Igor Sikorsky Kyiv Polytechnic Institute
Ukraine, Kyiv

The method for high speed remote users authentication has been proposed. Proposed method realized the zero-knowledge conception of cryptographically strong authentication. Developed method is based on using irreversible transformation on Galoise fields. The procedures for remote users registration and authentication cycle were developed in detail. It has been shown that proposed identification technology allows to speed up of identification process for software and hardware implementation.

Key words: identifications schemes, zero-knowledge identification, irreversible transformation on Galoise fields.

pp: 58-60

Microcontroller with the expandable instruction set

Anatoly Sergienko
chief engineer

Igor Sikorsky Kyiv Polytechnic Institute, Computer
Engineering Department
Ukraine

Oleksii Molchanov
graduate student

Igor Sikorsky Kyiv Polytechnic Institute, Department of
System Programming and Application Specific Systems
Ukraine

The eight-bit stack processor architecture SM8 is proposed, which is designed for implementation in FPGAs. The microcontroller with this architecture has small hardware costs, reduced software amount, and ability to add up to hundred new user instructions to its instruction set.

Key words: stack processor, Forth, FPGA, VHDL.

CONTROL SYSTEMS

pp: 63-65

The model of vehicle diagnostic subsystem based on bayesian network

Pysarenko Andrii
PhD, associate professor
Igor Sikorsky Kyiv Polytechnic Institute
Ukraine, Kyiv

Tischenko Dmytro
undergraduate
Igor Sikorsky Kyiv Polytechnic Institute
Ukraine, Kyiv

The dynamic model of vehicle diagnosis based on developed tools for MATLAB/Simulink was proposed. The basis of diagnostic subsystem is a Bayesian network. The model allows to identify the causes of vehicle failure based on sensors placed on the key components of the vehicle.

Keywords: Bayesian network, MATLAB/Simulink, simulation, S-function, diagnostics

pp: 66-68

International Regulatory Principles of Man-Machine Interface for Process Control Systems

Leonid Yurchuk
Candidate of technical science, docent
Igor Sikorsky Kyiv Polytechnic Institute
Ukraine, Kyiv

The development problems of man-machine interface for process control systems according to the requirements and recommendations of the International Standard ANSI / ISA-101.01-2015–“Human Machine Interfaces for Process Automation Systems” are considered.

Keywords: man-machine interface, life cycle, standardizing organization, ergonomics.

pp: 69-70

Components of control systems with real-time extension

Nikolajchuk Mykola Jaroslavovich
docent
Ivano-Frankivsk National Technical University of Oil and Gas
Ukraine, Ivano-Frankivsk

Computerized control systems with the real-time processes are based on the up-to-date information technologies, operating systems, firmware, and communication and support the allotted time spans for collecting, processing and transmitting data under technological objects control.

Hereat the WEB-based technologies of control systems on the ground of PLC and SCADA-systems with the elements of virtualization and tunnelling of industrial communication links are applied.

Considering the present requirements for the systems of technological processes control with high dynamics of information processes, the real-time systems support them completely.

Keywords: Soft PLC, operation system, real time expansion, control of real time, SCADA-system.

pp: 71-72

Control system for removing metallic inclusions mechanism in the conditions of ceramic bricks production

Zamikhovsky Leonid Mikhailovich
Head of Department
Ivano-Frankivsk National Technical University of Oil and
Gas
Ukraine, Ivano-Frankivsk

Levitsky Ivan Teodorovich
assistant
Ivano-Frankivsk National Technical University of Oil and
Gas
Ukraine, Ivano-Frankivsk

Structural scheme of the control system by mechanism of removing the metallic inclusions in a production environment of the ceramic bricks was presented based on the conveyor line and detection system of metallic inclusions. The development includes the control block of the metallic inclusions, hardware and software means based on PLC Simatic S7-1200, data transmission system, removing algorithm of metallic inclusions from the incoming raw materials.

Keywords: control system, executive devices, measurement, metallic inclusions, data transmission system, conveyor line.

pp: 73-75

Problems of a group of autonomous moving objects in 3D space

Dolyna Viktor Georgyovych
docent, PhD
Igor Sikorsky Kyiv Polytechnic Institute
Ukraine, Kyiv

Prilipukhov Yevhenii Vadymovych
student
Igor Sikorsky Kyiv Polytechnic Institute
Ukraine, Kyiv

This paper addresses the problem of managing a group of unmanned vehicles. Described trends unmanned systems. Two methods of solving problems with the management of objects in groups, such as multi-agent method and the method of swarm intelligence.

Keywords: unmanned systems, UAV, swarm intelligence, multi-agent method, microcomputer.

pp: 76-78

Synthesis of fuzzy controllers based on associative memory with hardware implementation on FPGA

Viatcheslav Nevolko
graduate student
Igor Sikorsky Kyiv Polytechnic Institute
Ukraine, Kyiv

In the report the method of synthesis and implementation of fast fuzzy controllers based hardware implementation of associative memory designed to implement high-speed control system objects and processes. Researched structures of regulators that provide of the solution control during 0.0001 ... 0.0003 seconds.

Keywords: controllers, fuzzy sets, fuzzy logic, neural networks, associative memory, programmable logic.

PROGRAMMING TECHNOLOGIES

pp: 81-82

Virtual variable realization

Iu. V. Krak
Professor

Taras Shevchenko National University of Kyiv
Ukraine, Kyiv

Iu. V. Koval
Assistant

Taras Shevchenko National University of Kyiv
Ukraine, Kyiv

In this paper variable notion investigated, formalization for this notion is proposed. Variable notion extended to virtual variable notion and operation set for virtual variable is present.

Keywords: virtual process, virtual variable.

pp: 83-85

Reasons for not using Hibernate and how to use JDBC correctly

Shyshkin Vladysav Igorovych
student

Igor Sikorsky Kyiv Polytechnic Institute
Kyiv, Ukraine

Sukhaniuk Maryna Valentynivna
student

Igor Sikorsky Kyiv Polytechnic Institute
Kyiv, Ukraine

The report consists of one section, contains 6 pages, 3 tables, 1 picture, 7 sources.

This report includes short excursion on JDBC and Hibernate technologies, where comparative characteristics are given and their advantages and disadvantages are listed. Special attention is paid to the developed MySQL-wrapper: description and justification of its necessity are given below. Also testing materials of three technologies at different loads were included, after which conclusions are made.

Keywords: java database connectivity, Hibernate, database management system, object-relational mapping, MySQL, java.

pp: 86-89

Design software for embedded systems using standard templates

Pavlo Katin
lecturer

Igor Sikorsky Kyiv Polytechnic Institute
Kyiv, Ukraine

The article describes the features of the architecture and software designed for embedded systems based on low-power microcontrollers. The result is reflecting of the depending RAM, program memory and software architecture. The architecture of infinite loop and finite-state machine are described and studied in Article. Recommendations design software for low-power microcontrollers and directions for further research.

Keywords: low-power mikrkontrolery, software, finite-state machine.

INFORMATION PROCESSING IN COMPLEX SYSTEMS

pp: 93-95

The Data Transfer in the Automotive LIN-Network based on Microcontrollers Cypress

Samotyy V.V.
Professor
Cracow University of Technology
(Tadeusz Kościuszko)
Lviv State University of Life Safety
Poland, Cracow; Ukraine, Lviv

Pavelchak A.G.
Docent
Lviv Polytechnic National University
Ukraine, Lviv

Dzelendzyak U.Yu.
Docent
Lviv Polytechnic National University
Ukraine, Lviv

The data model and mechanisms of its processing for LIN protocol based on Cypress microcontrollers has been developed.

Key words: LIN, Cypress, microcontroller.

pp: 96-99

High-performance time-series pattern-matching algorithm for analysis of streaming services quality

Anatoliy Doroshenko
Doctor of Physics and Mathematics, Prof.
Igor Sikorsky Kyiv Polytechnic Institute
Kyiv, Ukraine

Dmytro Titov
postgraduate
Igor Sikorsky Kyiv Polytechnic Institute
Kyiv, Ukraine

The usage of algorithms and techniques of time-series analysis for the tasks of monitoring and prediction of quality of data transmitting in streaming services, such as VoIP, is investigated. The performance improvement of algorithm of similar pattern detection in time-series is proposed.

Keywords: VoIP; QoS; MOS; jitter; pattern; time series; PAA; SAX.

pp: 100-103

Selection of error measure for reference and filtered periodic signals at high levels of fluctuating noise

O.O. Serheiev-Horchynskyi
PhD, Assistant
Igor Sikorsky Kyiv Polytechnic Institute
Ukraine, Kyiv

The article covers selection of a error measure for the results of filtering distorted periodic signals when searching for optimum values of low band digital filter parameters represented by a “simple moving average”. The following error measures were selected for comparison – Mean Squared Error, Root Mean Squared Error, Mean Absolute Error. A series of experiments was carried out for the generation and filtering of distorted periodic signals with different noise levels. Ultimately, a measure was selected based on comparison results, representing high sensitivity to the optimum values of the digital filter parameter at high noise levels.

Keywords: discrete signal, digital filter, moving average, optimal parameter, error measure



Портал науково-практичних заходів

<http://www.iconfs.net>

Сучасний майданчик для організації та проведення конференцій, конгресів, круглих столів та інших заходів для обміну думками, ідеями, результатами досліджень

Можливість брати участь у онлайн обговоренні наукових та практичних публікацій

Календар майбутніх заходів дозволяє спланувати участь та підготувати публікацію вчасно

Видання збірника та одночасна публікація на сайті дозволяє розширити аудиторію

Організаторам конференцій пропонуємо співпрацю з розміщення на порталі

<http://www.iconfs.net>