

**Міністерство освіти і науки України
Львівська філія Європейського університету**



**СУЧАСНІ ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ В
ЕКОНОМІЦІ, МЕНЕДЖМЕНТІ ТА ОСВІТІ**

Матеріали
всеукраїнської науково-практичної конференції

Львів – 2010

Організаційний комітет

Голова оргкомітету:

Гаршнев Ю.Г., к.т.н., доцент, академік ПТАНУ, директор Львівської філії Європейського університету

Заступник голови оргкомітету:

Сікорський П.І., д.п.н., професор, заступник директора з навчальної та наукової роботи Львівської філії Європейського університету

Члени оргкомітету:

Цегелик Г.Г., д.ф.-м.н., професор, завідувач кафедри математичного моделювання соціально-економічних процесів Львівського національного університету ім. Ів.Франка;

Притула М.М., д.ф.-м.н., професор, завідувач кафедри дискретного аналізу та інтелектуальних систем Львівського національного університету ім. Ів.Франка;

Пашко А.О., к.ф.-м.н., доцент, завідувач кафедри інформаційних систем і технологій Європейського університету

Вітер М.Б., к.ф.-м.н., доцент, начальник відділу організації обміну та аналізу інформації МФ України

Томашевський О.М., к.т.н., доцент, завідувач кафедри математики і комп'ютерних дисциплін Львівської філії Європейського університету

Секретар конференції:

Дубук В.І., к.т.н., доцент, доцент кафедри математики і комп'ютерних дисциплін Львівської філії ЄУ

Адреса оргкомітету

79019, Україна, Львів, вул. С.Кушевича, 5
Львівська філія Європейського Університету
тел.:+38 (0322) 97 50 73
e-mail: conf_eu_lviv@ukr.net
www: <http://lviv.e-u.in.ua/>

ЗМІСТ

Пленарне засідання

П.І. Сікорський. Методика впровадження кредитно-модульної технології навчання у вищих навчальних закладах.....	7
Г.Г. Цегелик. Використання математичних методів і моделей для дослідження економічних процесів.....	15

Секція 1. Навчально-методичні основи освіти, обмін досвідом в ІТ-навчанні

Ю.Г. Гаршнев, Н.Ф. Клочко. Міжпредметні зв'язки як складова формування компетентнісно-орієнтованої освіти у вищій школі економічного спрямування.....	23
А.М. Пелешишин, Ю.О. Сєров, О.Ю. Тимовчак-Максимець. Роль віртуальних соціальних спільнот у реалізації інформаційних потреб їх учасників.....	27
О.Є. Білас, Г.Б. Кришталь. Адаптування завдань з галузі програмної інженерії до комп'ютерного тестування.....	30
Л.І. Хмілярчук. Деякі аспекти тестування студентів з комп'ютерних дисциплін на прикладі вивчення об'єктно-орієнтованого програмування.....	34
Н.В. Дорош, О.В. Бойко, Г.Л. Кучмій, О.І. Дорош, А. Денис, М.В. Коцаренко. Розроблення та впровадження інтерактивних систем ІТ навчання.....	40

*Секція 2. Прийняття оптимальних рішень в економіці,
техніці, підприємництві та бізнесі*

О. Bihun, M. Prytula. Are Lie-Algebraic Approximations of Initial Value Problems Well-Posed?.....	44
М.Б. Вітер. Автоматизація документообігу в організації.....	52
А. Батюк, Л. Мандзюк. Доцільність застосування автоматизованого тестування з точки зору економічних затрат.....	55
А.В. Гунькало, О.В. Бойко, Н.В. Дорош. Управління якістю продукції (послуг) за допомогою експертних систем.....	62
М.О. Цибинога. Прийняття рішень щодо фінансування наукових розробок за допомогою “Технологічного аудиту”.....	64
В.І. Дубук, Т.З. Цюра. Особливості функціонального аналізу даних під управлінням операційних систем linux.....	67
О.Р. Зав’ялова. Сучасні підходи до розробки інвестиційних проектів.....	69
Л.І. Гальків. Багатовимірні середні в оцінці часових рядів втрат людського капіталу України.....	73
І.А. Маринич, Л.О. Гелей. Обґрунтування планування на підприємстві як фактор підвищення ефективності його діяльності.....	78
A. Dichko, Y. Sadova. Decision-making in wastewater treatment.....	82
С. Голубка. Функціональні аспекти національної політики щодо використання інформаційних технологій.....	84

Секція 3. Моделювання інформаційних, економічних і соціальних процесів, інформаційні технології в економіці

А.В. Мельничин, Г.Г. Цегелик. До побудови оптимальних стратегій пошуку записів у послідовних файлах баз даних за використання блочного пошуку.....	87
M. Gusti. An algorithm for simulation of forest management response to carbon price incentive.....	93
О.П. Пелешишин. Моніторинг активності конкурентів та захист від агресивних дій в онлайн-спільнотах.....	96
О.Я. Прядко, Г.Г. Цегелик, Л.М. Шевчук. До оптимізації розподілу фінансових ресурсів вузу з приватною формою власності своїм філіям.....	100
Н.І. Бойко. Методики моделювання логістичної web-системи.....	103
А. Батюк, І. Хомишин, С. Пилипчук. Система автоматизованої побудови програмних проектів Maven.....	108
Т.М. Степура. Моделювання соціального забезпечення населення України для пошуку шляхів підвищення його соціального захисту.....	115
К.Ф. Базилюк, С.Р. Ладик. М'які обчислення в економіці та фінансах.....	120
В. Голубка. Інвестування в інформаційні технології: проблеми та пріоритети розвитку.....	125
Т.З. Цюра. Інтерпретація алгоритму Лі для виконання на багатопоточній архітектурі nvidia® cuda.....	127

**Секція 4. Новітні комп'ютерні технології. Прикладні
додатки, засоби захисту та інтелектуальні методи
у телекомунікаціях**

О.В. Марковець, А.М. Пелещишин. Використання сучасних інформаційних технології для забезпечення виборчого процесу органів місцевого самоврядування.....	131
А. Батюк, С. Гринєць, С. Пилипчук. Огляд багатомодульного каркасу розробки Spring Framework.....	133
А.В. Костенко. Проблеми захисту від несанкціонованого доступу до баз даних SQL-сервера Firebird.....	139
О.М. Томашевський. Проблема збору статистичної інформації при моніторингу обчислювального кластера.....	143
М.Р. Гладун, Д.Ю. Волинюк, С. Михайлівський. Оптичні параметри світловипромінюючого елемента для створення пристроїв відображення інформації на його основі.....	147
Х.О. Засадна. Фішингатаки, їх види та характеристика.....	150
Ю.І. Макогон. Захист інтелектуальної власності (винаходів та корисних моделей) в господарських судах України.....	155
А. Батюк, В. Голощанов, С. Пилипчук. Google Web Toolkit (GWT) – платформа програмування для створення веб-ужитків на основі технології AJAX.....	158
S.S. Fedushko. Disclosure of Web-members Personal Information in Internet.....	162

ПЛЕНАРНЕ ЗАСІДАННЯ

МЕТОДИКА ВПРОВАДЖЕННЯ КРЕДИТНО-МОДУЛЬНОЇ ТЕХНОЛОГІЇ НАВЧАННЯ У ВИЩИХ НАВЧАЛЬНИХ ЗАКЛАДАХ

П.І. Сікорський

д.п.н., професор, заступник директора з ННР ЛФ СУ

Метою доповіді є визначення поняття кредитно-модульної технології навчання у вищих навчальних закладах та обґрунтування методики її реалізації.

Аналізуючи різні дефініції поняття технології навчання в дидактиці, можна виділити такі його основні ознаки:

- постановка конкретних цілей, планування та організація їх виконання;
- використання системи раціональних способів у досягненні поставлених цілей навчання;
- система науково обґрунтованих дій активних учасників процесу навчання;
- взаємодія і цілісність трьох компонентів: організаційної форми, дидактичного процесу і кваліфікації вчителя;
- проектування навчального процесу й гарантований кінцевий результат;
- педагогічна система, яка підвищує ефективність навчання через використання відповідних засобів;
- системний метод організації процесу навчання й засвоєння знань через взаємодію технічних і людських ресурсів;
- алгоритмізація спільної діяльності вчителя та учнів.

Оскільки навчання – це цілеспрямована педагогічна взаємодія, яка забезпечує виховання, розвиток і саморозвиток суб'єктів учіння, засвоєння ними необхідних знань, навичок і вмінь, методів і способів пізнання навколишнього світу, а навчальний процес –

конкретна, певною мірою організована модель навчання зі своєю внутрішньою алгоритмізованою структурою, потрібно визначити такі істотні ознаки технологічності навчання, які не дублюють його процесуальності, а посилюють її. Оскільки процесуальність у навчанні є сукупністю послідовних навчальних дій і кроків, то технологізація передбачає цілісне використання в навчальному процесі оптимальної кількості певним чином упорядкованих і адаптованих до суб'єктів учіння навчальних компонентів, які доводять навчання до відповідного рівня майстерності та досконалості, що оцінюється **гарантованим досягненням** наперед заданих результатів у вихованні, розвитку й засвоєнні знань. Причому залежно від віку дітей, їхніх індивідуальних особливостей, цільових настанов навчальних закладів у структурі кінцевих результатів можуть переважати розвивальні або освітні компоненти, проте виховання актуалізується в усіх закладах освіти.

Усе ж для розрізнення технологій навчання встановимо, які з основних навчальних компонентів (суб'єкти навчання – педагог, учні, зміст, методи, форми, засоби) є визначальними в технологізації навчального процесу. На нашу думку, до таких компонентів належать організаційні форми й домінуючі методи навчання. Важливе значення має також ефективна структурна взаємодія усіх навчальних елементів, використання раціональних і надійних засобів, рівень педагогічної майстерності та кваліфікації вчителя, викладача.

На різновид технології навчання істотно впливають **обрана навчальна форма** (індивідуальна, індивідуально-групова чи групова), домінуючі методи: лекція чи діалогічні методи (бесіда, пояснення, розповідь, вправлення тощо). Наприклад, у найпоширенішій у шкільництві класно-урочній технології навчання характерною ознакою є групова форма (клас) з дозованим часом (45 хв) та домінуючими методами (бесіда, пояснення, вправи тощо); у вищих навчальних закладах переважає лекційно-семінарська технологія навчання, в якій також використовуються групові форми, а лекція і семінар – це основні навчальні методи. Отже, прийоми і методи, форми навчання найістотніше урізноманітнюють навчальний процес, визначаючи рівень його технологічності.

Очевидно, що комбінації навчальних компонентів у їх структурній взаємодії утворюватимуть різні педагогічні технології. Поняття технології власне вбирає в себе найістотніші ознаки

(методи і форми), що й визначають назву, суть педагогічної технології, яка дає змогу досягати оптимальних кінцевих результатів.

Таким чином, **навчальна технологія** – це цілісний алгоритм організації ефективного засвоєння знань, умінь і навичок, який характеризується оптимальною комбінацією основних навчальних компонентів (зміст, прийоми і методи, форми і засоби), і з урахуванням вимог наукової організації праці, збереження і зміцнення здоров'я суб'єктів навчання забезпечує досягнення запланованих навчально-виховних результатів.

Основними ознаками, визначальними елементами кредитно – модульної технології навчання є дидактичні поняття *кредиту* і *модуля*. Кредит спонукає студента добровільно вибирати ті чи інші навчальні дисципліни в структурно – логічній послідовності, визначає оптимально повне навчальне навантаження, що виділяється на вивчення обраного предмета і зобов'язує студента продемонструвати стан засвоєння дисципліни під час контролю (екзамену). Отже, введення в нову навчальну технологію поняття дидактичного кредиту висуває насамперед певні вимоги до формування освітньо-професійних програм.

В умовах традиційних технологій навчання освітньо-професійна програма складалася з інваріантної і варіативної частин. Інваріантна складова ділилася, як правило, на три цикли дисциплін. Наприклад, для напряму підготовки «Економіка підприємництва» такі цикли: гуманітарної підготовки ($\approx 17\%$); природничо-наукової та загальноекономічної підготовки ($\approx 39\%$); професійної підготовки ($\approx 14\%$).

Варіативна складова становить 30% і ділиться на цикл дисциплін за вибором навчального закладу ($\approx 18\%$) і цикл дисциплін на вибір студента ($\approx 12\%$).

Таким чином, у традиційній навчальній системі студент обирає цикл навчальних дисциплін, які становлять $\approx 12\%$ сукупного навчального навантаження, з метою поглиблення своїх знань у вузькій сфері даної спеціальності.

Тому студентам потрібно запропонувати цикли дисциплін з різних вузьких спеціалізацій, що впливають з основної спеціальності, і дати можливість обрати той напрям (цикл дисциплін), який найбільше їх цікавить. Очевидно, що саме з таких міркувань обирає цикл вибіркових дисциплін і навчальний заклад.

Відомо, що вибір навчальним закладом дисциплін залежить від можливостей професорсько-викладацького складу, наукових шкіл, які проводять дослідження за певними напрямками.

Отже, у кредитній системі студенту потрібно навчитися обирати для послідовного вивчення усі навчальні дисципліни (інваріантну й варіативну частини), зважаючи лише на свої інтелектуально-фізичні можливості й послідовність їх вивчення (структурно-логічні схеми). Тобто кредитна система, з одного боку, дає змогу максимально демократизувати, диференціювати та індивідуалізувати навчальний процес, а з другого – зобов'язує якісно засвоювати зміст вищої освіти й тим самим забезпечити об'єктивне визнання результатів навчання в різних навчальних закладах і країнах.

Не менш важливим у новій технології навчання є дидактичне поняття модуля – логічно завершеної частини елементів знань з навчальної дисципліни. Якщо в кредитах визначають оптимальний час повного навчального навантаження студента для засвоєння змісту навчальної дисципліни чи курсу, то модуль як одиниця вимірювання змісту дає можливість моделювати іншу, за нашими дослідженнями, більш продуктивну технологію навчання, яка дає змогу засвоювати зміст навчального предмета не урочними порціями (шкільництво), не семестровим відрізком (традиційна вища школа), а логічно завершеними частинами (модулями). Кількість змістових одиниць модуля резонує з виділеним часом на їх засвоєння, протягом якого суб'єкти навчання спроможні реалізувати психолого-педагогічні й кібернетичні вимоги до цілісного процесу засвоєння модульної сукупності знань і тим самим досягти оптимального ефекту в навчанні.

Модульне навчання вимагає складати навчальні програми дисципліни на інших засадах. З одного боку, виходячи з кількості виділених кредитів на його вивчення, потрібно поділити навчальну дисципліну на скінчену кількість логічно завершених частин (модулів), а з іншого – визначити сукупність елементів знань модуля у структурних взаємозв'язках, а також визначити найголовніші, тобто ті знання, які треба засвоїти (генералізація знань).

Упровадження модульного навчання в навчальних закладах України здебільшого зводиться до модульного контролю, який є останньою ланкою в технології навчання. На нашу думку, основну

увагу в модульній технології навчання потрібно зосередити на психолого-педагогічному забезпеченні організації засвоєння знань з урахуванням кібернетичних вимог і паралельно відпрацьовувати ефективні підходи до контролю й оцінювання засвоєних знань.

До психолого-педагогічного забезпечення модульної технології відноситься: складання модульних навчальних програм і модульного планування; модульний розподіл навчальних дисциплін і визначення оптимального часу на засвоєння кожного модуля; структурування й генералізація навчального матеріалу кожного модуля, навчальної дисципліни, міждисциплінарних курсів; підготовка модульних посібників і підручників; визначення системи контрольних параметрів з кожної навчальної дисципліни; напрацювання, адекватних до генералізованої структури знань і вмінь, матеріалів для проведення контролю й оцінювання, в тому числі й за допомогою комп'ютерних технологій.

Для моделювання кредитно-модульної технології навчання використаємо своє трактування модульно-рейтингової технології та принципу модульності.

Принцип модульності передбачає організацію засвоєння навчального матеріалу в дискретно-неперервному полі за програмою, яка складається з логічно завершених частин (модулів) із структурованим змістом та цілісним навчально-управлінським циклом кожного модуля.

Модульно-рейтингова технологія навчання – це цілісний алгоритм організації повного засвоєння знань та вмінь за модульним принципом з дотриманням психолого-педагогічних і кібернетичних вимог до навчального процесу та рейтинговим підсумовуванням результатів навчання за семестр, рік і весь період навчання.

Оскільки дидактичний кредит не лише визначає повне навчальне навантаження для здобуття того чи іншого освітньо-кваліфікаційного рівня чи засвоєння навчальної дисципліни, а й спонукає студентів виявляти ініціативу, добровільно обираючи для засвоєння систему навчальних дисциплін, тому він істотно впливатиме на модульну технологію. Якщо модуль регулює в навчальній технології змістову компоненту і з урахуванням кількості елементів знань модуля, рівня їх складності визначаються прийоми й методи навчання, то кредит передусім впливає на вибір форми навчання (індивідуальна, індивідуально-групова, групова, колективна). Форма організації засвоєння знань з дисципліни

залежатиме від того, скільки студентів обере її у своїх навчальних зобов'язаннях. Крім того, кількість кредитів, що виділяється на засвоєння тієї чи тієї дисципліни, впливатиме на модульний розподіл часу. Отже, між кредитом і модулем існують складні діалектичні взаємозв'язки. Час оптимального повного навантаження студента, наприклад, для здобуття бакалаврського ступеня, є сталою величиною і становить 5760 год (240 кредитів). За цей період потрібно засвоїти визначену кількість елементів знань інваріантної і варіативної частин освітньо-професійної програми. Ефективність навчання залежатиме від адекватного (складності та важливості дисциплін у підготовці фахівця) розподілу кредитів на той чи інший предмет. Якщо навчальний заклад збільшуватиме кількість навчальних дисциплін, то слід або збільшувати час навчального навантаження студента за рахунок його вільного часу, або підвищувати ефективність навчальних технологій.

Таким чином, кредитно-модульна технологія навчання – це цілісний алгоритм повного засвоєння знань та вмінь майбутніми фахівцями за структурно-інтегрованими освітньо-професійними програмами в кредитних вимірах, свідомим самостійним вибором студентами навчальних дисциплін з метою докладання максимальних інтелектуальних зусиль для їх вчасного засвоєння та модульним принципом з дотриманням психолого-педагогічних та кібернетичних вимог до навчального процесу в межах модуля, навчальної дисципліни, міждисциплінарного курсу і ступеня підготовки загалом.

Запровадження модульно-рейтингових технологій навчання у старших класах загальноосвітньої школи (ліцеїв, гімназій), за нашими дослідженнями, спонукає учнів до систематичної навчальної праці, максимально об'єктивізує оцінювання знань, дає змогу досягати оптимальних навчальних результатів, уможливорює продовження навчання випускників у вищих навчальних закладах без вступних іспитів, а також забезпечує психолого-педагогічну підготовку першокурсників до плавного переходу на кредитно-модульну технологію навчання.

Для переходу на кредитно-модульну технологію навчання потрібно, на нашу думку, реалізувати такі першочергові заходи:

- порівняти навчальні плани вищих закладів освіти України з навчальними планами країн Болонського процесу й узгодити їх інваріантні частини (до 70 %);

- централізовано затвердити єдині терміни аудиторного навчання у вищих навчальних закладах протягом 36 тижнів щорічно: I семестр – з 1.09 по 29.12 (17 тижнів), II семестр – з 25.01 по 6.06 (19 тижнів);
- виристовуючи можливості дидактичної інтеграції, зменшити кількість різних навчальних дисциплін до п'яти для кожного курсу й розмежувати поняття навчального предмета і дисципліни: дисципліна складається з окремих автономних предметів;
- централізовано для кожного напряму підготовки затвердити інваріантну частину навчальних планів і програм, а також концепцію формування навчальними закладами варіативної частини програми;
- уніфікувати поняття дидактичного кредиту, виходячи з європейських вимог (I кредит 24 год);
- конкретизувати державний стандарт вищої освіти, визначивши структуру елементів знань для повного засвоєння даного навчального предмета, дисципліни, курсу і ступеня підготовки;
- організувати конкурси з написання інтегрованих навчальних підручників за модульним принципом та структурованими знаннями, а також матеріалів для контрольних зрізів;
- у провідних університетах організувати кафедри кредитно-модульних технологій навчання, а для координації їх наукових зусиль – Інститут кредитно-модульних навчальних технологій Міністерства освіти і науки України, відкрити відповідні лабораторії та відділи в системі наукових установ АПН України;
- зменшити річне аудиторне навчальне навантаження науково-педагогічних працівників: доцентів до 432 год (12 год на тиждень), а професорів до 288 (8 год на тиждень);
- укласти перехідні структурно-логічні схеми – викладання навчальних дисциплін у кредитних вимірах, за модульним принципом і перейти на їх реалізацію в експериментальному режимі.

На основі наведених теоретичних викладок представимо модель дидактичного структурування навчального матеріалу:



Підсумовуючи, зазначимо, що дидактичне структурування навчального матеріалу можна реалізувати, використовуючи такий алгоритм:

- за допомогою методів логічного структурування навчального матеріалу визначаємо базові поняття фундаментальних дисциплін;
- на основі аналізу дисциплін професійного циклу виділяємо елементи знань, які складають основу фундаментальних дисциплін;
- узгоджуємо елементи знань фундаментальних дисциплін і дисциплін професійного циклу.

Отже, за допомогою дидактичного структурування, використовуючи принцип перспективних ліній, можна суттєво поліпшити підготовку висококваліфікованих фахівців.

Література

1. Сікорський П.І. Кредитно-модульна технологія навчання: Навч.посіб. – К.:Вид-во Європ. Ун-ту, 2004. -127 с.
2. Сохор А.М. Логическая структура учебного материала. Вопросы дидактического анализа. – М.: “Педагогика”, 1974.

ВИКОРИСТАННЯ МАТЕМАТИЧНИХ МЕТОДІВ І МОДЕЛЕЙ ДЛЯ ДОСЛІДЖЕННЯ ЕКОНОМІЧНИХ ПРОЦЕСІВ

Г.Г. Цегелик

д. ф.-м. н., проф., професор кафедри МКД ЛФСУ

Побудовано математичні моделі планування замовлень на виготовлення продукції з найвищою якістю та розподілу кредитів банку з мінімальною величиною ризику. Використано метод динамічного програмування для оптимального розподілу завдань серед комп'ютерів мережі та визначення найкоротших шляхів в транспортних мережах.

З розвитком ринкових відносин, удосконаленням управління в усіх сферах цілеспрямованої людської діяльності (промисловість, сільське господарство, торгівля, побутове обслуговування, транспорт, охорона здоров'я, охорона природи і т. ін.) виникають задачі, для розв'язання яких треба приймати рішення, які є досить складними і суттєво впливають на результат. Зрозуміло, що без наукового обґрунтування рішень в таких ситуаціях обійтись не можна. Чим більш складною і вартісною є організація заходу, тим менш допустимим є прийняття “вольового” рішення і тим більшого значення набувають наукові методи, які дають змогу наперед оцінити наслідки кожного рішення, відкинути недопустимі варіанти і рекомендувати оптимальні або, принаймні, найбільш раціональні.

Для наукового обґрунтування рішень в залежності від заходу може використовуватись той чи інший математичний апарат: методи лінійного і нелінійного програмування, динамічне програмування, стохастичне програмування, теорія ігор, теорія масового обслуговування, теорія графів та ін. Крім того, може використовуватись статистичне й імітаційне моделювання, регресійний аналіз тощо.

Приведемо декілька прикладів реальних задач з різних предметних областей, для розв'язування яких нами розроблені математичні моделі або використано метод динамічного програмування.

Математична модель планування замовлення на виготовлення продукції. Розглянемо задачу про планування

замовлення на виготовлення продукції. Незважаючи на наявність досить ґрунтовної і різноманітної літератури з даного питання, проблема покращення існуючих алгоритмів розв'язування такої задачі продовжує бути актуальною.

Припустимо, що торговельна фірма хоче замовити на підприємствах малого бізнесу в певному обсязі продукцію декількох видів. Нехай відомо: малі підприємства, що займаються виготовленням потрібної продукції; кількість видів продукції, потрібної замовнику; пріоритет продукції кожного виду; рейтинг продукції кожного виду кожного підприємства; максимальна потреба в замовленні продукції кожного виду; собівартість продукції кожного виду на кожному підприємстві; обсяг коштів, що може використати кожне підприємство. Задача полягає в складанні такого плану замовлення, за якого замовлена продукція була б найякіснішою.

Нехай

n – кількість видів продукції, що цікавлять замовника;

m – кількість малих підприємств, що виготовляють продукцію, потрібну замовнику;

α_i – пріоритет i -ої продукції;

r_{ij} – рейтинг i -ої продукції на j -му підприємстві;

p_i – максимальна потреба в продукції i -го виду;

c_{ij} – собівартість одиниці продукції i -го виду на j -му підприємстві;

v_j – обсяг грошових коштів, що може використати j -те підприємство;

x_{ij} – кількість одиниць продукції i -го виду, що планується замовити на j -му підприємстві (шукані величини).

Тоді математична модель задачі матиме вигляд:

$$L = \sum_{i=1}^n \sum_{j=1}^m \alpha_i r_{ij} x_{ij} \rightarrow \max$$

за умов

$$\sum_{j=1}^m x_{ij} \leq p_i, i = 1, 2, \dots, n;$$

$$\sum_{i=1}^n c_{ij} x_{ij} \leq v_j, \quad j = 1, 2, \dots, m;$$

$$x_{ij} \geq 0, \quad i = 1, 2, \dots, n; \quad j = 1, 2, \dots, m.$$

Тут L виражає сумарну якість продукції; перше обмеження передбачає, що обсяг замовленої продукції не може перевищувати потребу в цій продукції; друге обмеження означає, що кошти, затратені на виготовлення продукції, не можуть перевищувати наявних коштів підприємства.

Оптимальний розподіл завдань серед комп'ютерів мережі.

Нехай треба розподілити m завдань однакової складності серед n комп'ютерів $K_1, K_2, \dots, K_n$ мережі (комп'ютери можуть мати неоднакову потужність). Відомо час розв'язування завдань на кожному комп'ютері. Задача полягає в такому розподілі завдань серед комп'ютерів, щоб загальний час розв'язування завдань був мінімальний.

Для розв'язування поставленої задачі використаємо метод динамічного програмування. Нехай $t_i(x_j)$, $i = 1, 2, \dots, n$, – час виконання $x_j = j$, $j = 0, 1, \dots, m$, завдань на i -му комп'ютері K_i . Позначимо через $T_i(x_j)$, $i = 1, 2, \dots, n$, – загальний час у випадку розв'язування $x_j = j$, $j = 0, 1, \dots, m$, завдань на перших i комп'ютерах $K_1, K_2, \dots, K_i$, а через $T_i^*(x_j)$, $i = 1, 2, \dots, n$, – загальний час розв'язування завдань у разі оптимального розподілу $x_j = j$, $j = 0, 1, \dots, m$, завдань серед перших i комп'ютерів $K_1, K_2, \dots, K_i$. Процес розв'язування задачі розіб'ємо на n кроків.

На першому кроці визначаємо мінімальний час у разі розв'язування $x_j = j$, $j = 0, 1, \dots, m$, завдань на першому комп'ютері K_1 . На другому кроці визначаємо мінімальний час у випадку розв'язування $x_j = j$, $j = 0, 1, \dots, m$, завдань на перших двох комп'ютерах K_1, K_2 . І т. д. Нарешті, на n -му кроці визначаємо мінімальний час у разі розв'язування m завдань на n комп'ютерах $K_1, K_2, \dots, K_n$.

На першому кроці

$$T_i(x_j) = t_i(x_j), \quad T_i^*(x_j) = t_i(x_j), \quad j = 0, 1, \dots, m.$$

На другому кроці

$$T_2(x_j) = \begin{cases} t_2(0) + T_1^*(x_j - 0), \\ t_2(1) + T_1^*(x_j - 1), \\ \dots\dots\dots \\ t_2(x_j) + T_1^*(0) \end{cases}$$

і

$$T_2^*(x_j) = \min_{0 \leq k \leq j} \{t_2(k) + T_1^*(x_j - k)\}$$

для $j = 0, 1, \dots, m$.

Взагалі, на s -му кроці ($s = 3, 4, \dots, n-1$)

$$T_s(x_j) = \begin{cases} t_s(0) + T_{s-1}^*(x_j - 0), \\ t_s(1) + T_{s-1}^*(x_j - 1), \\ \dots\dots\dots \\ t_s(x_j) + T_{s-1}^*(0) \end{cases}$$

і

$$T_s^*(x_j) = \min_{0 \leq k \leq j} \{t_s(k) + T_{s-1}^*(x_j - k)\}$$

для $j = 0, 1, \dots, m$.

На останньому n -му кроці досить обчислити $T_n(m)$ і $T_n^*(m)$, де

$$T_n(m) = \begin{cases} t_n(0) + T_{n-1}^*(m), \\ t_n(1) + T_{n-1}^*(m-1), \\ \dots\dots\dots \\ t_n(m) + T_{n-1}^*(0) \end{cases}$$

і

$$T_n^*(m) = \min_{0 \leq k \leq m} \{t_n(k) + T_{n-1}^*(m - k)\}.$$

Оптимальний розподіл m задач серед n комп'ютерів мережі визначаємо так.

Нехай $T_n^*(m)$ досягає мінімуму для $k = l_1$, тоді l_1 завдань треба виконувати на останньому n -му комп'ютері K_n . Далі потрібно розподілити $m - l_1$ завдань серед перших $n-1$ комп'ютерів

$K_1, K_2, \dots, K_{n-1}$. Припустимо, що $T_{n-1}^*(m-l_1)$ досягає мінімуму для $k=l_2$. Це означає, що l_2 завдань треба виконувати на $(n-1)$ -му комп'ютері K_{n-1} . Якщо $T_{n-2}^*(m-(l_1+l_2))$ досягає мінімуму для $k=l_3$, то l_3 завдань треба виконувати на $(n-2)$ -му комп'ютері K_{n-2} . І т.д. Нехай $T_2^*(m-(l_1+l_2+\dots+l_{n-2}))$ досягає мінімуму для $k=l_{n-1}$. Тоді l_{n-1} завдань треба виконувати на другому комп'ютері K_2 . Нарешті, $l_n = m-(l_1+l_2+\dots+l_{n-1})$ завдань треба виконувати на першому комп'ютері K_1 . Мінімальний час виконання всіх завдань становитиме $T_n^*(m)$ одиниць.

Задача мінімізації ризиків в банківському кредитуванні.
Розглянемо математичну модель, реалізація якої може мінімізувати кредитний ризик.

Нехай

S – обсяг коштів банку, виділених для кредитування;

n – кількість бажаючих взяти кредит;

s_i – обсяг коштів, потрібних i -му позичальнику;

p_i – величина ризику від надання кредиту i -му позичальнику;

q_i – дохід від кредитування i -го позичальника;

Q – мінімальний загальний дохід від кредитування;

$x_i = \begin{cases} 1, & \text{якщо } i\text{-му позичальнику надається кредит,} \\ 0, & \text{в протилежному випадку.} \end{cases}$

Тоді математична модель мінімізації кредитного ризику матиме вигляд:

$$L = \sum_{i=1}^n p_i s_i x_i \rightarrow \min$$

за умов

$$\sum_{i=1}^n s_i x_i \leq S,$$

$$\sum_{i=1}^n q_i x_i \geq Q,$$

$$x_i = \{0,1\}, i = 1,2,\dots,n.$$

Тут L виражає сумарний ризик від надання кредитів; перше обмеження передбачає, що загальна сума кредитів не повинна перевищувати обсяг коштів банку, виділених для кредитування; друге обмеження означає, що загальний дохід від кредитування повинен бути не менший за певну величину.

Задача визначення найкоротших шляхів в транспортних мережах. Розглянемо задачу про визначення найкоротшої відстані від будь-якого пункту (вершини) до інших у заданій транспортній мережі.

Нехай на деякій поверхні задана скінченна кількість точок $P_1, P_2, \dots, P_n$, які з'єднані дугами (зв'язками) (P_i, P_j) , що не перетинаються. Сукупність точок і дуг, які їх з'єднують, називатимемо мережею. Мережу будемо називати достатньо зв'язною, якщо для будь-яких двох точок існує шлях (сукупність вершин і дуг, що їх з'єднують), по якому можна пройти з однієї точки в іншу. При цьому, дві точки мережі називатимемо сусідніми, якщо існує дуга, що їх з'єднує.

Постановка задачі. Нехай задана достатньо зв'язна мережа, кожній дузі якої, що виходить із точки P_i і входить у точку P_j , поставлене у відповідність деяке дійсне невід'ємне число l_{ij} – її довжину, причому $l_{ij} = l_{ji}$. Треба визначити найкоротші шляхи в мережі від довільної точки до всіх інших і вказати відповідні їм відстані.

Для розв'язування задачі використаємо метод динамічного програмування, згідно з яким будемо відшукувати найкоротші шляхи не від фіксованої точки до всіх інших, а найкоротші шляхи від усіх інших точок до фіксованої (заданої) через сусідні точки. Дугу, що міститься в найкоротшому шляху, позначатимемо стрілкою в напрямку до фіксованої точки. Вершини мережі позначатимемо кружечками (цифра всередині кружечка вказує номер точки), а в дужках біля них записуватимемо найкоротші відстані від цих точок до фіксованої точки. Відповідні найкоротші відстані називатимемо характеристиками точок.

Нехай

P_i – фіксована точка, до якої необхідно визначити найкоротші відстані та відповідні шляхи від усіх інших точок;

p_{ij} – кількість дуг, із яких складається найкоротший шлях від фіксованої точки P_i до поточної точки P_j ;

k_s ($s = 1, 2, \dots, m$) – кількість точок, найменша кількість дуг до яких від фіксованої точки P_i складає s , причому

$$m = \max_{j \neq i} p_{ij};$$

$P_1^{(s)}, P_2^{(s)}, \dots, P_{k_s}^{(s)}$ ($s = 1, 2, \dots, m$) – точки, найменша кількість дуг до яких від фіксованої точки P_i складає s .

Алгоритм розв'язування задачі. Алгоритм складається з початкового кроку та загального, що повторюється $m-1$ разів.

Початковий крок. Біля кружечка, що позначає точку P_i , записуємо нуль – характеристику цієї точки, оскільки відстань від точки P_i до неї самої дорівнює нулю. Визначаємо сусідні з P_i точки і біля кружечків, якими позначені ці точки, записуємо їх характеристики, тобто $0 + l_{ij}$, якщо P_j є сусідньою точкою, а на дугах ставимо стрілки, спрямовані в сторону точки P_i . Після цього відмічаємо точку P_i символом $+$, який означатиме, що операція над цією вершиною завершена.

Загальний крок. Припустимо, що ми вже виконали r кроків: знайшли характеристики всіх точок, найменша кількість дуг до яких від фіксованої точки P_i є не більшою за r ; точки $P_1^{(s)}, P_2^{(s)}, \dots, P_{k_s}^{(s)}$ ($s = 1, 2, \dots, r-1$), помічені символом $+$; характеристики точок $P_1^{(r)}, P_2^{(r)}, \dots, P_{k_r}^{(r)}$ відповідно дорівнюють $C_1^{(r)}, C_2^{(r)}, \dots, C_{k_r}^{(r)}$ і від цих точок проведені стрілки в напрямку до точки P_i .

Тоді на $(r+1)$ -му кроці беремо будь-яку точку $P_i^{(r)}$ і визначаємо характеристики всіх сусідніх до неї точок (крім точки, до якої спрямована стрілка від $P_i^{(r)}$) як суму $C_i^{(r)}$ і відстані від точки $P_i^{(r)}$ до кожної з них. Після цього точку $P_i^{(r)}$ відмічаємо символом $+$.

Зауважимо, що при визначенні характеристик сусідніх із $P_i^{(r)}$ точок може виявитися, що характеристика сусідньої точки вже була обчислена раніше. Нехай це буде точка P_q , стара характеристика якої C_q , а наново обчислена характеристика цієї точки C'_q . У такому випадку порівнюємо між собою величини C'_q із C_q . Якщо $C'_q \geq C_q$, то характеристику C_q залишаємо без зміни. Якщо ж $C'_q < C_q$, то старе C_q замінюємо на C'_q . Відповідно зміниться зв'язок, через який проходить найкоротший шлях до точки P_q , а стрілка на дузі, що виходить із точки P_q , заміниться на стрілку на дузі $(P_q, P_i^{(r)})$, що виходить із цієї точки.

Якщо при цьому змінилася характеристика якоїсь конкретної точки P_q , яка раніше вже була відмічена символом +, то перераховуємо характеристики сусідніх із нею точок (крім точки $P_i^{(r)}$) і змінюємо при потребі їхні характеристики і напрям стрілок.

Якщо ж при визначенні характеристик сусідніх із $P_i^{(r)}$ точок виявиться, що характеристика якоїсь сусідньої точки раніше не обчислювалася, то на відповідній дузі, що виходить із цієї точки, ставимо стрілку в напрямку до точки $P_i^{(r)}$.

Нарешті, відзначимо, що $(r+1)$ -ий крок виконуватимемо доти, доки послідовно не будуть перебрані всі вершини мережі, тобто точки $P_i^{(r)}$ ($i=1,2,\dots,k_r$).

СЕКЦІЯ 1

Навчально-методичні основи освіти, обмін досвідом в ІТ-навчанні

МІЖПРЕДМЕТНІ ЗВ'ЯЗКИ ЯК СКЛАДОВА ФОРМУВАННЯ КОМПЕТЕНТІСТНО-ОРІЄНТОВАНОЇ ОСВІТИ У ВИЩІЙ ШКОЛІ ЕКОНОМІЧНОГО СПРЯМУВАННЯ

¹Ю.Г. Гаринев, ²Н.Ф. Ключко

¹к.т.н., доцент, академік ПТАНУ, директор Львівської філії ЄУ

²ст.в. кафедри МКД ЛФ ЄУ

Розглядаються міжпредметні зв'язки (математика, інформатика, економічні дисципліни) і їх вплив на якість освіти в процесі підготовки спеціаліста економічного профілю.

В умовах ринкової економіки традиційна підготовка студентів економічних спеціальностей не задовольняє сучасні потреби ринку праці. Якщо раніше випускник вузу міг поступово адаптуватись до умов роботи, то тепер від нього вимагається повна професійна реалізація відповідного рівня компетентності.

Формування професійної компетентності здійснюється через зміст освіти. Сучасний спеціаліст для розв'язування професійних задач повинен орієнтуватись в інформаційних потоках, грамотно використовувати засоби обчислювальної техніки, тобто володіти інформаційною компетентністю.

Базові форми підготовки спеціаліста – це навчальна діяльність академічного типу і навчально-професійна діяльність, до якої відноситься науково-дослідна робота, практика, дипломне проектування. Проектування технології навчання у відповідності до сучасних вимог до спеціалістів здійснюється на основі диференційованих завдань і міжпредметних зв'язків.

Міжпредметні зв'язки – це дидактична категорія, яка

відображається у взаємопов'язаному і взаємообумовленому вивченні навчальних предметів, це інтеграція знань і вмінь з різних предметів в процесі навчання. Сучасний етап розвитку науки характеризується взаємопроникненням наук і особливо проникненням математики і інформатики в економічні галузі знання. Зв'язок між навчальними предметами є перш за все відбиттям об'єктивно існуючого зв'язку між окремими науками і зв'язку науки з практичною діяльністю людей. Міжпредметні зв'язки потрібно розглядати як відбиття в учбовому процесі міжнаукових зв'язків, що складають одну з характерних рис сучасного наукового знання.

Компетентність – це здатність спеціаліста адекватно розуміти реальність, правильно оцінювати ситуацію, у якій потрібно діяти, і правильно використовувати свої знання. Оскільки бакалавр є спеціалістом першого ступеня вищої освіти, то потреба в ньому на ринку праці головним чином визначається професійною компетентністю.

Бакалавр економічного профілю повинен вміти планувати структуру дій, вміти будувати інформаційні моделі, вміти організувати пошук інформації, володіти знаннями про структури даних, процедури, макроси, вміти аналізувати явища, що виникають у матеріальній сфері суспільного життя, вирішувати виробничі та організаційно-управлінські задачі, розуміти роль і місце математики і математичного моделювання в економіці, бути знайомим з комп'ютерним інструментарієм, який можна використовувати для розв'язування тієї чи іншої задачі, використовувати професійні пакети прикладних програм (бухгалтерські, правові, банківські, статистичні, фінансового менеджменту, тощо).

Ці вміння можуть бути сформовані за допомогою міжпредметних зв'язків математики та дисциплін економічного циклу. Один із шляхів реалізації цих міжпредметних зв'язків – це відбір матеріалу для вивчення з точки зору професійної значущості. Важливим є формування у студентів математичної інтерпретації основних закономірностей і зв'язків, що розглядаються в дисциплінах економічного циклу.

Нажаль, учбово-методична література з вищої математики для студентів економічних спеціальностей не відповідає сучасним вимогам до їх математичної підготовки. Нові західні підручники

спираються на більш широку математичну базу.

Використання міжпредметних зв'язків математики і економічних дисциплін в процесі фундаментальної підготовки студентів Львівської філії Європейського університету в курсі вищої математики відображене в учбово-методичному комплексі дисципліни і проявляється в розв'язуванні задач прикладного характеру з кожної теми. Головна мета прикладної спрямованості викладання математики в економічному вузі – формування математичного аспекту готовності випускників вузу до професійної діяльності у сфері ринкової економіки. З цієї точки зору найважливішим методом навчання є метод математичного моделювання.

Математичні моделі практичних задач економіки часто мають складну структуру і вимагають значних витрат часу на своє розв'язання, тому доцільно будувати і аналізувати такі моделі з використанням засобів обчислювальної техніки.

Міжпредметні зв'язки математики та інформатики починаються з використання офісних програм (Word, Excel, Access) і реалізуються в процесі практичної роботи студентів у комп'ютерних аудиторіях не тільки під час аудиторних занять, де викладач інформатики демонструє принципи розв'язування комп'ютерно-орієнтованих математичних задач, зміст яких відповідає навчальній програмі з вищої математики, але й у процесі самостійної роботи над індивідуальними завданнями під час комп'ютерної практики, а також при вивченні курсу обчислювальної математики.

Математика та інформатика мають *глибокі внутрішні взаємозв'язки*: інформатика виникла як розділ прикладної математики і лише поступово сформувалася як окрема наука, але двосторонні зв'язки між цими предметами актуальні. Інформатика має справу з інформаційно-обчислювальними моделями, методами їх побудови і аналізу, алгоритмами, які розроблені на базі математичних методів. Традиційна система математичної освіти потребує змін, оскільки виникло протиріччя між обсягом професійної інформації, потрібної для майбутнього спеціаліста і часом, що відводиться на освіту студента.

Однією з головних задач, що стоять перед системою підготовки майбутніх спеціалістів, є підвищення якості

математичної підготовки з використанням інформаційних технологій у професійній діяльності. Ця задача може бути вирішена за допомогою сучасних інтегрованих математичних систем, які використовуються з метою максимального спрощення для користувачів комп'ютерної реалізації математичних алгоритмів і методів, що використовуються в економічному аналізі.

У Львівській філії Європейського університету використовується спеціалізований пакет математичних програм MathCAD. Професійний математичний пакет з точки зору педагогіки є сучасним дидактичним засобом навчання, який дозволяє оптимізувати учбовий процес, надати йому якісно новий рівень. Професійний математичний пакет з точки зору інформатики – це інформаційна технологія, призначена для автоматизації розв'язування математичних задач із сучасним інтерфейсом для користувача, із засобами візуалізації результатів обчислень, що дозволяє аналізувати одержані результати, надавати їм економічну інтерпретацію, приймати управлінські рішення.

Таким чином, відбувається перехід на якісно новий рівень оволодіння методологічними поняттями теорії математики із застосуванням ІТ. Починаючи з виконання перших лабораторних робіт із застосуванням пакету MathCAD, у студентів виробляються навички самостійного дослідження, редагування процесу розв'язування і грамотного оформлення одержаних результатів.

У результаті використання в навчальному процесі інформаційних технологій суттєво підвищується зацікавленість студентів у глибокому вивченні математики, полегшується засвоєння структурних зв'язків між різними розділами курсу математики. Можливість не використовувати великих за обсягом перетворень і обчислень дозволяє їм мислити крупними блоками, не гублячи головної лінії міркувань. У процесі використання інформаційних технологій у навчанні математики встановлюються і посилюються між предметні зв'язки математики та інформатики, формуються основи математичної та стилістичної культури.

Реалізація міжпредметних зв'язків математики, дисциплін економічного циклу та інформатики в процесі навчання в економічному вузі є важливим фактором підвищення ефективності учбового процесу.

РОЛЬ ВІРТУАЛЬНИХ СОЦІАЛЬНИХ СПІЛЬНОТ У РЕАЛІЗАЦІЇ ІНФОРМАЦІЙНИХ ПОТРЕБ ЇХ УЧАСНИКІВ

¹А.М. Пелецишин, ²Ю.О. Серов, ³О.Ю. Тимовчак-Максимець

¹д.т.н., доцент, професор кафедри ІСМ, НУ “Львівська політехніка”

²к.т.н., ст. викладач кафедри ІСМ, НУ “Львівська політехніка”

³аспірант кафедри ІСМ, НУ “Львівська політехніка”

У статті розглядаються віртуальні соціальні спільноти як середовище комунікації та забезпечення інформаційних потреб учасників цих спільнот. Вводяться поняття інформаційного донора та інформаційного реципієнта, які характеризують інформаційні ролі учасників дискусії. Розглядається вплив учасників на тематику та розгортання дискусій на Веб-форумах.

Віртуальні соціальні мережі останнім часом набули поширення як неформальні спільноти – середовище спілкування, обміну думками, а також отримання інформації. Вони сприяють, по-перше, організації соціальних комунікацій між людьми, а, по-друге, реалізації їх базових соціальних потреб.

Віртуальні соціальні спільноти формуються довкола певного Інтернет-середовища комунікації, наприклад, Веб-форуму, і забезпечують задоволення комунікаційних та інформаційних потреб їх учасників. Спільнота (англ. *community*), із точки зору соціології, це група людей, які вступають у соціальну взаємодію і мають певні зв'язки між собою та іншими членами групи і принаймні певний час володіють спільною територією [1]. У віртуальній спільноті зберігаються ознаки соціальної взаємодії та зв'язків між членами групи. Однак, віртуальна спільнота принципово відрізняється тим, що спільною територією для групи людей є віртуальний простір, а точніше певний Інтернет-ресурс або середовище спілкування.

Участь у віртуальній спільноті дає такі можливості учасникам:

1. отримання інформації від інших учасників спільноти;
2. верифікації ідей шляхом соціальної взаємодії у спільноті;
3. отримання соціальної користі від контактів (визнання, самоідентифікація, підвищення самооцінки тощо);
4. рекреації (проведення вільного часу).

Кожен із учасників реалізує власну потребу шляхом взаємодії з іншими учасниками спільноти. Взаємодія учасників реалізується у тривалій в часі комунікації, тобто у формі полілогу учасників на певну задану тематику. Життєдіяльність віртуальної спільноти визначається бажанням учасників долучитися до обговорення. Таким чином, обов'язковою умовою існування віртуальної спільноти є наявність у достатньої кількості її учасників бажання вступати в комунікативну взаємодію, для якої ключовими словами є “кількість” і “бажання” [1].

З точки зору обміну досвідом, будемо виділяти такі дві ролі учасників спільноти: інформаційний донор та інформаційний реципієнт. **Інформаційний донор** це учасник спільноти, який має на меті поділитися (в англ. мові – *to share*) власним досвідом або точкою зору стосовно певного об'єкту чи ситуації з іншими учасниками спільноти. **Інформаційні реципієнти** це особи, які зацікавлені в отриманні інформації про досвід учасників спільноти стосовно певного об'єкту чи ситуації. Інформаційними реципієнтами можуть виступати як учасники спільноти, так і сторонні особи, які потрапили на форум, наприклад, за запитом у пошуковій системі.

Учасники спільноти можуть мати або строго визначені ролі, або приймати іншу роль, залежно від інформаційних потреб. Так, в одній дискусії учасник може бути реципієнтом, а в іншій – донором інформації. На форумах технічної підтримки, наприклад, донор (фахівець, який надає консультацію) та реципієнт (користувач, який потребує фахової консультації) не можуть помінятися ролями.

Обмін інформацією, зокрема особистим досвідом учасників, реалізується у формі дискусій. Первинна тематика задається ініціатором дискусії, первинним реципієнтом, і формулюється таким чином, щоб задовольнити інформаційні потреби ініціатора. У дискусію вступають учасники, які мають намір надати певну інформацію. Мінімальна кількість учасників дискусії це та кількість учасників дискусії, при якій у самих її учасників не втрачається інтерес до дискусії. З точки зору обміну досвідом, для задоволення інформаційних потреб ініціатора дискусії, часто достатньо одного учасника, який може задовольнити інформаційні потреби ініціатора. Результуючий рівень задоволеності інформаційних потреб ініціатора залежить від фаховості і/або досвідченості донора.

До процесу розгортання дискусії можуть долучатися не лише інформаційні донори, але й інші інформаційні реципієнти, окрім первинного. Вторинні інформаційні реципієнти зацікавлені в первинній тематиці дискусії, але крім цього переслідують власну мету – отримання додаткової або суміжної інформації. Таким чином, первинна тематика дискусії може переходити у похідну.

Кількість похідних тем дискусії не є обмеженою і залежить від актуальності обговорюваного питання та активності учасників у дискусії. Актуальна тематика породжує широке зацікавлення, велику кількість суперечливої інформації, антагоністичні настрої тощо, що привертає широке коло зацікавлених осіб, тому насиченість та динаміка дискусії залишається високою, поки ажіотаж і поживлене обговорення не підуть на спад.

Відвідуваність певної віртуальної спільноти опосередковано впливає на ранг сторінки форуму у пошукових системах. Висока відвідуваність сторінки віртуальної спільноти підвищує ймовірність того, що за релевантним запитом у пошуковій системі, у спільноту прийде новий учасник, який надасть дискусії похідну тему і стане вторинним інформаційним реципієнтом. Вторинними реципієнтами можуть бути також дійсні члени спільноти, інформаційні потреби яких співпадають з інформаційними потребами первинного реципієнта. Таких членів спільноти будемо назвати явними вторинними реципієнтами, якщо вони задекларували у дискусії свою зацікавленість первинною тематикою, і неявними вторинними реципієнтами, якщо реципієнт отримав інформаційну вигоду, але спільнота не отримала зворотного зв'язку.

Отже, дискусія служить інструментом задоволення не лише комунікаційних, але й інформаційних потреб. Участь у віртуальних соціальних спільнотах дозволяє учасникам реалізувати власну мету участі у спільноті шляхом безпосереднього впливу на розгортання та тематику дискусій.

Література

1. George H. A., Jr. Definitions of Community: Areas of Agreement. *Rural Sociology* 20, 1955. - p.111 – 123
2. Щербина В. Соціальні риси мережних спільнот // *Соціальна психологія*. - 2005. - № 2 (10). - с.139-149

АДАПТУВАННЯ ЗАВДАНЬ З ГАЛУЗІ ПРОГРАМНОЇ ІНЖЕНЕРІЇ ДО КОМП'ЮТЕРНОГО ТЕСТУВАННЯ

¹О.Є. Білас, ²Г.Б. Кришталъ

¹к.т.н., доцент НУ “Львівська політехніка”

²студент НУ “Львівська політехніка”

Необхідність швидкого та якісного оцінювання знань студентів вимагає застосування комп'ютерних технологій тестування. Певні галузі знань чи навчальні дисципліни мають свою специфіку підготовки та представлення завдань. Розглядаються особливості тестових завдань із дисциплін програмної інженерії.

На сучасному етапі розвитку інформаційних технологій з'явилися нові методики вивчення та представлення матеріалів. Також виникає необхідність перевірки отриманих знань та показника засвоєної інформації, саме для цього існують різноманітні тестові завдання, які дають змогу перевірки вивченого матеріалу, а також прослідкувати і визначити рівень наявних знань.

Все ширшого поширення набувають різноманітні засоби перевірки знань, серед яких значне місце займають електронні комп'ютерні тести. У порівнянні із уже існуючими інструментами перевірки здобутих знань електронні тести володіють рядом переваг, які виділяють їх поміж інших. Наприклад, тестовий модуль володіє переліком питань, які пропонується пройти в межах вказаного часу, результати тестування зберігаються за конкретним студентом. Ряд систем пропонує перевірити рівень засвоєних знань та якісно їх оцінити, інші ж пропонують здійснити перевірку знань, а також при необхідності вивести підказку користувачу для повторення або інформування коректної відповіді. Відсутня необхідність перевірки відповідей викладачем оскільки це здійснює система та виводить інформацію із повідомленням результатів. Дані із інформацією про тестування зберігаються за конкретною особою та займають незначний об'єм пам'яті на дисковому просторі, а отже відсутня необхідність зберігання і ручної обробки паперової документації. Завдяки застосуванню автоматизованого підходу для обробки інформації та її збереження є значна економія часу на оцінювання. Викладач може переглянути та проаналізувати інформацію на основі отриманих відповідей та презентувати

студенту результати засвоєного матеріалу, а також визначити рівень підготовки матеріалу як за певним студентом так і за групою студентів, які вивчали даний розділ чи окрему дисципліну. Автоматизований контроль знань, який має забезпечити ефективність організації тестового контролю через автоматизацію процесу проведення контролю та обробки результатів тестування здійснюється використовуючи розроблені та досліджені підходи організації тестів.

В моделях програмованого навчання: **лінійної** (Б.Скіннера) і **розгалуженої** (Н.Краудера) – для оцінки якості засвоєння учбового матеріалу рекомендувалися різні форми відповідей. Скіннер користувався відповіддю, що конструюється, а Краудер – вибірковою. В першому випадку учень повинен доповнити текст завдання пропущеним словом або словами, в другому – вибрати правильну відповідь з ряду запропонованих. При другому підході, в пропонованому наборі відповідей одна – вірна, а інші – помилкові. По-можливості, в число невірних відповідей включаються типові помилки тих, хто навчається, відомі з практики навчання [1]. Саме вибіркового типу відповіді на сьогоднішній день найбільш часто використовують при автоматизованому контролі, оскільки його легше реалізувати в порівнянні з іншими типами. На користь вибірових відповідей зазвичай наводяться *такі аргументи*: а) в практичному житті найчастіше зустрічаються ситуації, де вирішальне значення має вибір відповідної альтернативи; б) оскільки студентові потрібно навести явну відповідь в звичній для нього формі, не має суттєвого значення, сформулює він її сам чи вибере готову з наведеного списку; в) необхідність вибору відповіді з декількох можливих варіантів допомагає студенту краще розібратися в матеріалі, що вивчається; г) програму з вибіровими відповідями легше реалізувати; д) вибірові відповіді дозволяють набагато простіше побудувати адаптивне навчання; та ін.

Реалізацію даних підходів дає можливість втілити OpenTest 2.0 – комп'ютерна система тестування знань створена для очного підсумкового контролю якості засвоєного теоретичного матеріалу, здобутих знань і практичних навиків у великих організаціях масштабу підприємства із складною розподіленою структурою [2]. Основною особливістю системи OpenTest 2.0 є її спрямованість на забезпечення тестувань з максимально строгою звітністю. Сферою застосування можуть бути різноманітні підсумкові тестування,

заліки, іспити, кваліфікаційні тести і будь-які інші види контролю знань учнів, в яких головну роль відіграє максимально об'єктивна оцінка знань.

Особливістю комп'ютерної системи тестування OpenTest 2.0 є низькі вимоги до апаратних і програмних ресурсів, що дозволяє проводити тестування навіть в погано устаткованих комп'ютерних класах. OpenTest 2.0 створена для повноцінної роботи під управлінням різних операційних систем на клієнтських комп'ютерах можуть бути встановлені різні інтернет-оглядачі, допускається низька пропускна здатність мережі між сервером і клієнтськими комп'ютерами.

В системі використовуються такі види тестових завдань:

- **Завдання закритої форми** – завдання тесту з вибором відповіді з декількох запропонованих варіантів.
- **Завдання відкритої форми** – завдання тесту, в якому учень повинен самостійно формулювати свою відповідь.
- **Завдання з короткою вільною відповіддю** – це тестові завдання, на які учень повинен написати відповідь словом, словосполученням або числом. Є окремим випадком відкритих завдань.
- **Завдання з розгорнутою відповіддю** – тестові завдання, на які учень повинен записати відповідь у вигляді одного або декількох речень чи формул. Є окремим випадком відкритих завдань.
- **Завдання перестановки** – тестові завдання, в яких потрібно переставити елементи списку в потрібній послідовності. Наприклад, розставити події у фонологічному порядку або співставити відповідності із запропонованого ряду у відповідності до завдання.
- **Завдання на відповідність**. У даному вигляді завдань необхідно встановити відповідність між лівими і правими елементами завдання і відповідь записати у вигляді правильної комбінації цифр і букв.

У системі OpenTest 2.0 приділено багато уваги проблемам безпеки при проведенні тестувань. Оскільки при контрольних тестуваннях кінцева оцінка грає велику роль, об'єктивність її виставляння має бути максимальна, а також всі можливі варіанти фальсифікації результатів мають бути виключені.

Система дає можливість здійснювати тематичний аналіз тесту використовуючи засоби математичної статистики та враховувати відповідність результатів нормальному закону розподілу із відповідними зміщеннями та коефіцієнтами.

Таблиця 1

Тематичний аналіз тесту

Тема	K	N	T	A	R
Визначення, поняття	15	40	0.7	4.17	$R_{xy} = 0.849$ $SS_x = 12.8$ $SP_{xy} = 378.8$
Якість ПЗ	10	40	0.53	8	$R_{xy} = 0.899$ $SS_x = 8.7277778509593$ $SP_{xy} = 331.35000202211$
Статичне тестування	10	40	0.51	8	$R_{xy} = 0.868$ $SS_x = 6.5500000427169$ $SP_{xy} = 277.11666954607$
Рівні тестування	10	40	0.58	6	$R_{xy} = 0.807$ $SS_x = 9.2222223050062$ $SP_{xy} = 305.50000328567$

У таблиці відображено загальну кількість запитань з кожної теми, кількість запитань, що включались в тест, відносна складність запитань, відхилення від середнього арифметичного бала та коефіцієнти кореляції із результатами відповідей інших тем. Можливості аналізу тесту дають змогу оцінити якість тестових запитань та відповідно коригувати їх. Передбачається використання у завданнях графічних елементів діаграм, стандартних нотацій чи дескрипторів, що дасть можливість краще адаптувати тести до дисциплін з програмної інженерії. Такий підхід дасть змогу розробити тести різних рівнів складності та надати можливість більш творчих відповідей на запитання.

Література

1. Аванесов В.С. Формы тестовых заданий : Учебное пособие для учителей школ, лицеев, преподавателей вузов и колледжей. – М.: Центр тестирования, 2005. – 156 с.

2. Напрасник С.В., Цимбалюк Е.С., Шкиль А.С. Компьютерная система тестирования знаний OpenTEST 2.0 // Сборник научных трудов 10-й международной конференции УАДО Образование и виртуальность–2006. Харьков: ХНУРЭ.– 2006.– С. 454–461.

ДЕЯКІ АСПЕКТИ ТЕСТУВАННЯ СТУДЕНТІВ З КОМП'ЮТЕРНИХ ДИСЦИПЛІН НА ПРИКЛАДІ ВИВЧЕННЯ ОБ'ЄКТНО-ОРІЄНТОВАНОГО ПРОГРАМУВАННЯ

Л.І. Хмілярчук

асистент кафедри ПЗІ, Львівська комерційна академія

Сьогодні в Україні питанням комп'ютерних технологій навчання, створенню і впровадженню в навчальний процес тестових систем навчання і контролю знань студентів приділяється багато уваги. Тестування – достатньо об'єктивний інструмент, використання якого дозволяє цілком об'єктивно оцінити якість освіти. У роботі розглядаються деякі особливості тестування студентів з комп'ютерних спеціальностей на прикладі об'єктно-орієнтованого програмування, подаються приклади тестових завдань при вивченні таких важливих тем ООП, як “Визначення класів” та “Перевантаження операцій”.

У сучасних соціально-економічних умовах вищі навчальні заклади повинні випускати не просто висококваліфікованих фахівців, але осіб з високим розвитком інтелекту.

У зв'язку з цим традиційні класичні методи навчання у вузі виявляються недостатніми. Потрібні нові підходи до процесу навчання, що сприяють розвитку творчої ініціативи та пізнавальної діяльності. Одним з таких підходів є технологія тестування, використовуючи яку можна постійно відслідковувати характер досягнень студентів з усіх навчальних дисциплін. Для цього кожна кафедра повинна розробити банк тестових завдань, зміст яких повністю б охоплював навчальну програму з конкретної дисципліни.

Різні аспекти тестування, як проблеми, історія, світовий досвід [3], труднощі, зумовлені специфікою використання у

підготовці спеціалістів різного фаху [4, 6], мотивація студентів [1] тощо, висвітлюються у низці публікацій.

Менше уваги приділялось досі питанню об'єктивності тестування як контролю знань. Тестування вважають найбільш об'єктивним методом контролю знань, оскільки воно виключає суб'єктивність перевіряючого, хоча це не зовсім так. Деякі автори зауважують, що зазвичай “слабкі” студенти “пасують” перед великим вибором наукоподібних неправильних відповідей, не можуть згадати правильну. Вони обирають випадкові відповіді і нерідко їх кількість є достатньою для позитивної оцінки. Натомість “сильні” студенти надто відповідально аналізують кожне слово завдання, довго роздумують над відповіддю, хвилюються, а в підсумку – низький результат [2].

У зв'язку з переходом освітніх програм на кредитно-модульну систему зростає обсяг самостійної роботи студентів. Тому викладач повинен ретельніше ставитися до проблеми контролю знань студентів.

Крім того, специфіка предметів, пов'язаних з інформаційними технологіями, вимагає набуття як практичних, так і теоретичних навичок, а практичні навіть більш бажані. Однак, провести їх оцінку в усіх студентів за короткий термін дуже складно. Тому все більше викладачів орієнтується на тестування, як на простий та швидкий метод опитування. Тестові завдання стають частинами поточного та підсумкового контролю і повинні відповідати певним вимогам. Завдання має бути однозначним, тобто студент не повинен задумуватися, в якому сенсі трактується те чи інше судження, зрозумілим та стислим, у ньому повинні використовуватись відомі для студентів терміни.

Об'єктно-орієнтоване програмування (ООП) займає провідне місце при розробці прикладного програмного забезпечення у будь-якій сфері діяльності, і випускник вищого навчального закладу повинен мати уявлення про сучасні технології створення програмного забезпечення своєї майбутньої професії. Крім того, особа, яка формулює завдання, повинна володіти основними знаннями і термінологією теорії комп'ютерних інформаційних систем.

Стратегія ООП – це зміщення пріоритетів у процесі програмування від функціональності програми до структур даних. Це дозволяє програмісту моделювати реальні об'єкти і ситуації.

Технологія ООП має три головні переваги: вона проста для розуміння (ООП дозволяє мислити категоріями повсякденних об'єктів); надійна і проста для супроводу – правильне проектування забезпечує простоту розширення і модифікації об'єктно-орієнтованих програм; прискорює цикл розробки – модульність тут грає важливу роль, оскільки різні компоненти об'єктно-орієнтованих програм можна легко використовувати в інших програмах, що зменшує надмірність коду і знижує ризик помилок при копіюванні.

Специфіка ООП помітно підвищує ефективність праці програмістів і дозволяє їм створювати більш потужні, масштабовані та ефективні програми.

Клас – це базове поняття в ООП. Простіше – клас – це визначений програмістом тип даних. Екземпляр класу – об'єкт – це сукупність даних (полів) і функцій (методів) для їх обробки.

Контрольні тести для студентів, які вивчають ООП, мають на меті допомогти останнім чітко розрізняти ці поняття і добре оволодіти методикою їх організації.

Розглянемо, як приклад, наступні тестові завдання:

Чому відповідає поняття “Клас” в об'єктно-орієнтованому програмуванні?

- 1) Тип даних, який визначає користувач; 2) Змінна; 3) Поле;
- 4) Метод; 5) Масив.

Поля та методи класу, які описані в розділі доступу `protected`: доступні в

- 1) методах даного класу; 2) методах класів-нащадків; 3) довільному блоці, що містить екземпляр даного класу; 4) лише у функції `main()`.

Який розділ доступу утворює за замовчуванням клас, що описаний ключовим словом `class`?

- 1) `public`:: 2) `private`:: 3) `protected`:: 4) `using`:: 5) `extern`:: 6) `throw`::.

Вищенаведені тестові запитання дають можливість оцінити рівень засвоєння студентами теоретичних знань з теми курсу “Визначення класів”. Для оцінки практичних навиків використовуються тестові завдання, які мають іншу спрямованість. Студенти повинні вміти вірно описувати клас, створювати об'єкти класів, використовуючи конструктори і без них, визначати методи поза класом, вміти викликати методи класів і т.ін.

Наприклад:

Клас Ch не містить конструкторів. Вкажіть оператори опису змінних, які створюють об'єкт A класу Ch.

1) Ch A; 2) Ch: A; 3) Ch A(5); 4) Ch A(); 5) A.Ch.

Який результат компіляції та виконання програми:

```
#include <iostream>
using namespace std;
class Ch{
public:
Ch(int i) {cout<<"i";}
Ch() {cout<<"Tak";};
void main(){
Ch A(20);
cin.get();}
```

1)вивід на консоль **Tak**; 2) вивід на консоль **i**; 3) вивід на консоль **20**;
4) вивід на консоль **20 Tak**; 5) вивід на консоль **Tak 20**;
6) помилка компіляції.

Вкажіть, які з наведених фрагментів програми для екземпляру A класу Ch здійснюють виклик методу setCh та надають полю C значення 5?

```
class Ch{
int C;
public:
void setCh(int i){c=i;}
void showCh(){cout<<c;};}
```

1)A.setCh(); 2) setCh(5); 3) A.setCh(5); 4) int k=2; A.setCh(k+3);
5) A.setCh(int 5); 6) setCh=5.

В класі Ch оголошений метод showCh(). Вкажіть синтаксично вірні визначення методу showCh() поза класом:

- 1) void Ch::showCh(){cout<<c<<"\n";}
- 2) void Ch::showCh(){cout<<c;}
- 3) Ch:: void showCh(){cout<<c<<"\n";}
- 4) void showCh(){cout<<c<<"\n";}
- 5) void Ch:: showCh():{cout<<c<<"\n";}

Вкажіть які з операторів описів змінних створюють об'єкту OS класу St?

```
class St{
```

```

char name[15];
int nogr;
int oc;
public:
void setSt(char *ns, int no, int bal);
void showSt();}

```

1) St OS; 2) St OS("Mary",121,4); 3) OS.setSt("Mary",121,4);
 4) setSt("Mary",121,4); 5) setSt()=("Mary",121,4); 6) St OS().

Коли існуючі операції, наприклад + чи =, наділяються можливістю здійснювати дії над операндами нового типу, кажуть, що вони перевантажені. Перевантаження операцій є важливим інструментом ООП. Тому тема “Перевантаження операцій” є важливою у тестуванні студентів з комп’ютерних дисциплін.

Ось деякі з тестових завдань даної тематики:

За допомогою яких функцій можливо здійснити перевантаження бінарних операцій у класі?

1) методів класу з одним формальним параметром; 2) методів класу без формальним параметрів; 3) методів класу з двома формальними параметрами; 4) дружніх до класу функцій без формальним параметрів; 5) дружніх до класу функцій з двома формальними параметрами; 6) дружніх до класу функцій з одним формальним параметром.

Який результат компіляції та виконання програми

```

#include <iostream>
using namespace std;
class COper{
public:
COper& operator+(int i){cout<<i;return *this;}
friend COper& operator+(int i,COper &x){cout<<i+3;return x;}
COper& operator+(COper &x){cout<<3;return *this;};
void main(){
COper A,B;
3+A+B+2;
cin.get();}

```

1) 632; 2) 623; 3) 236; 4) 263; 5) 362; 6) 326; 7) помилка компіляції.

Зауважимо, що в останньому з наведених завдань лише один з варіантів відповіді є правильним.

Як бачимо, тестові завдання практичного характеру об’ємні і

потребують більше часу на осмислення студентами змісту завдання. Тому, плануючи тест для контролю знань студентів, слід врахувати цю особливість, і дати студенту достатньо часу, щоб проаналізувати усі варіанти та обрати правильні відповіді.

Отже, тестова перевірка знань не завжди об'єктивна. Тому використання тестів у всіх видах контролю, а особливо в підсумковому, обов'язково повинно доповнюватися іншими видами контролю (бесідою, контрольною роботою, тощо).

Тестування стає тільки контролюючим засобом, а діяльність викладача має насамперед навчальну мету. Майбутні фахівці на робочому місці будуть виконувати реальні завдання, і життя не пропонуватиме альтернативних відповідей. У процесі спілкування з викладачем студент навчається. В цьому діалозі постають нові запитання. При цьому з боку обох учасників у великій мірі виявляються не тільки формалізовані підходи, а й творчий пошук, інтуїція і ерудиція, розвиток яких є визначальним завданням вищої школи.

Література

1. Булах І.Є. Мотивація навчання і валідизація оцінювання рівня знань / Булах І.Є., Шило І.М. // Педагогіка і психологія. Вісник АПН. – 1996. – №3. – С. 125-129.
2. Гаджиева И.Х. К вопросу о контроле знаний студентов по физиологии растений / Гаджиева И.Х., Абдурахманов А.А. // Вестник Башкирского университета. – 2001. – № 2 (II). – С. 145-146.
3. Кебуладзе В. Вивчаючи європейський досвід: освітня система Німеччини та її значення для України / В. Кебуладзе // Європейський освітній портал. – Режим доступу: <http://www.eu-edu.org/ua/news/info/163>.
4. Киринюк А. А. Тестирование студентов как метод оценки качества образования / А. А. Киринюк // Высшее юридическое образование. – Омск, 2005. – С. 7-11.
5. Лафоре Р. Объектно-ориентированное программирование в С++ / Р. Лафоре. – Классика Computer Science. – изд 4. – СПб: Питер, 2005. – 924 с.
6. Непомнящая С. Компьютерное тестирование: плюсы и минусы (к вопросу о методике использования в учебном процессе) / С.Непомнящая, Т.Хохлова, Е.Абалихина // Alma mater.– 2006.– №6.– С. 14-18.

РОЗРОБЛЕННЯ ТА ВПРОВАДЖЕННЯ ІНТЕРАКТИВНИХ СИСТЕМ ІТ НАВЧАННЯ

¹*Н.В. Дорош*, ²*О.В. Бойко*, ³*Г.Л. Кучмій*, ⁴*О.І. Дорош*,
⁵*А. Денис*, ⁶*М.В. Коцаренко*

¹*к.т.н., доцент кафедри “Електронні прилади” НУ ЛП*

²*к.т.н., ст.в. кафедри медичної інформатики ЛНМУ ім. Д.Галицького*

³*к.т.н., доцент кафедри “Електронні прилади” НУ ЛП*

⁴*магістр, Києво-Могилянська академія*

⁵*інженер, НУ ЛП*

⁶*асист. кафедри медичної інформатики ЛНМУ ім. Д.Галицького*

Проведено огляд сучасних систем керування навчальним процесом. Описано основні можливості системи Moodle та розробленого на її базі курсу дистанційного навчання з електроніки.

Актуальною задачею сучасної освіти є вдосконалення професійного рівня спеціалістів, допомога в освоєнні нових професій і здобутті нових знань. Системи ІТ-навчання дають можливість віддаленого ознайомлення з навчальними, методичними матеріалами, виконання практичних завдань, контролю за процесом навчання та оцінювання знань за допомогою тестування. Крім того у таких системах передбачена можливість обговорення питань під час on-line конференцій, форумів у режимі реального часу.

Прикладами програмних середовищ для створення інтерактивних систем ІТ-навчання є ATutor (www.atutor.bdpu.org), Ilias (www.ilias3.bdpu.org), Interact (www.interact.bdpu.org), Ganesha (www.anemalab.org), Claroline (Classroom Online (www.claroline.net)), MOODLE (Modular Object Oriented Distance Learning Environment) (www.moodle.org).

Порівняння функціональних можливостей таких навчальних систем показує, що всі вони забезпечують добру структурованість подачі навчальних матеріалів, підтримують розмежування прав доступу до ресурсів для користувачів різного рівня, але відрізняються можливостями щодо проведення опитування та тестування, а також аналізу отриманих результатів.

ATutor – система керування навчальним матеріалом (Learning Content Management System, LCMS), перевагою якої є можливість підключення іншої не освітньої CMS. Дана система підтримує

структурованість матеріалів, проте володіє слабким тестовим модулем.

Ilias – LCMS, яка включає потужний тестовий модуль з можливістю аналізу результатів тестування, проте побудова редактора для створення сторінок є незручною для користування. Система також підтримує структурованість матеріалів та розмежування прав доступу до ресурсів.

Interact – система поєднує можливості різних CMS, натомість можливості керування навчанням є неповними. Схожою за функціональністю є система Docebo. Перевагою Docebo є добре організована політика прав доступу, реалізована можливість розподілу та обмеження прав адміністрування ресурсом, створені кращі можливості тестування та опитування.

Ganesha пропонує значні можливості для забезпечення навчального процесу як зі сторони студента, так і викладача. Система має можливість перегляду документів MS Word без необхідності встановлення додаткового програмного забезпечення. Система вирізняється з поміж інших відносною простотою інсталяції та налаштування.

Система **OpenCartable** відноситься до ENT (цифрове середовище розробки). Її основу складає набір додатків педагогічного спрямування. Додаток може використовуватися викладачами та студентами в будь-якому місці з доступом до мережі Інтернет. OpenCartable підтримує освітній мультимедійний контент, який може бути сформований викладачами, редакторами навчальних посібників, установами освітнього сектору. Після аутентифікації користувачі отримують доступ до свого робочого столу, на якому розміщується найважливіша особиста інформація: підручники, список груп учнів, нові домашні завдання, домашні завдання, завдання здані на перевірку тощо. За допомогою робочого столу викладачі і студенти можуть отримати доступ до усіх функцій ENT: системи електронних повідомлень, файлового архіву з розмежуванням прав доступу, сайту навчального закладу, форуму.

Claroline – платформа побудови сайтів дистанційного навчання, яка дозволяє створювати та редагувати лекції. Програма включає генератор вікторин, форуми, календар, функцію розмежування доступу до документів, каталог гіперпосилань, систему контролю за успіхами студента, модуль авторизації тощо. Суттєвою перевагою системи є ергономічність та зручність у

формуванні та керуванні програмою навчання. Система Claroline використовується у Національному Медичному Університеті для проведення дистанційного навчання та організації тестування для студентів денної форми навчання.

Moodle – LMS, призначена для організації online-лекцій і освітніх web-сайтів. Проект розроблений для поширення соціо-конструктивного підходу у навчанні. Цей підхід передбачає, що нові знання можуть отримуватися тільки на основі раніше придбаних знань і вже наявного індивідуального досвіду, процес навчання буде набагато ефективніший, коли студент передає або пояснює іншим отримані знання. При використанні цього підходу викладач спирається на той досвід студента, який найкраще підходить для засвоєння потрібного матеріалу, а не просто публікує і модифікує інформацію, яку учень повинен засвоїти. Такий підхід реалізує можливість надання прав кожному учаснику навчального процесу бути і вчителем, і студентом. LMS Moodle використовується для організації дистанційного навчання в Національному університеті “Львівська політехніка” та у Києво-Могилянській академії.

У системі передбачена можливість специфікації категорій (Category) навчальних курсів, таких, як Office work, Teaching and Learning (Офісна робота, Навчальний процес). Групування курсів по категоріях полегшує перегляд і пошук курсів. Навчальні курси можуть включати такі матеріали, як анотації курсів, ресурси, завдання, теми обговорень.

У даній системі реалізований розподіл прав та доступів для редагування, створення, перегляду, видалення та налаштування сторінок та курсів. При цьому, передбачена багатофункціональність користувачів – один і той же користувач може бути адміністратором системи, інструктором одного або декількох курсів і студентом одного або декількох курсів. У системі передбачена спеціальна категорія Гостей (guests), які можуть ознайомлюватись з курсами.

Зареєструвавшись на сайті користувач має право записатись на будь-яку доступну дисципліну. Для запису на курс потрібно перейти на головну сторінку сайту та в блоці “Розділи сайту” вибрати потрібний розділ, в якому вибирається або підрозділ, або бажаний курсу в залежності від структури сайту (рис.1).

Для кожного зареєстрованого користувача доступна своя сторінка профілю, на якій доступні опції перегляду даних користувача, зміни пароля та статистика відвідування сайту.

Розділи сайту: ПРЕ / Електронні прилади

Дисципліни кафедри Електронні прилади

Комерційна електроніка

Даний курс передбачає засвоєння засобів та принципів функціонування електронної комерції

Пошук дисципліни: Виконати

Рис.1. Сторінка розділу “Електронні прилади”

Для вивчення дисципліни студентів пропонується низка ресурсів та завдань. Ресурси – це статичні матеріали курсу: файли з текстами лекцій, різного роду зображення (карти, ілюстрації, схеми, діаграми), веб-сторінки, аудіо та відео-файли, анімаційні ролики, гіперпосилання на Інтернет-ресурси і т.п.

Робота з ресурсами зводиться до їх вивчення у вказані викладачем терміни. Матеріали курсу можна прочитати з екрану, зберегти на свій комп’ютер роздрукувати. Матеріали курсу можуть бути вивчені у будь-якій послідовності, проте бажано дотримуватись заданої викладачем, оскільки вивчення деяких матеріалів передбачає знання попередніх.

Для вибору ресурсу потрібно на головній сторінці курсу у блоці поточної теми вибрати необхідний матеріал. Якщо вибраний ресурс не можливо відобразити у вікні браузера, то з’явиться повідомлення про можливість завантаження даного файлу на локальний комп’ютер. Деякі формати документів можуть бути відкриті браузером за допомогою встановлених додатків.

Вивчення дисципліни передбачає також виконання різного роду завдань розміщених викладачем у темі курсу. Такими завданнями можуть бути: вправи, глосарій, форум, чат, анкета, контрольні завдання, семінар, тестування та інше.

Система електронного навчання насамперед спрямована на користувача (студента). Різноманітні доступні сервіси надають можливість отримання якісної освіти та забезпечують зручність роботи з системою.

СЕКЦІЯ 2

Прийняття оптимальних рішень в економіці, техніці, підприємстві та бізнесі

ARE LIE-ALGEBRAIC APPROXIMATIONS OF INITIAL VALUE PROBLEMS WELL-POSED?

¹O. Bihun, ²M. Prytula

¹Sc.D., Prof., Head of DAIS Department, I.Franko National University, Lviv

²Assistant Profesor, Department of Mathematics and Computer Science,
Concordia College

The Heisenberg-Weil algebra is the Lie algebra of operators $\{1, x, d/dx\}$ acting on all differentiable functions. In this paper we prove some useful properties of the representations of these operators on finite dimensional spaces of polynomials. The Lie algebraic method of discrete approximations allows to construct finite dimensional approximations of initial value problems for ordinary and partial differential equations by projecting them onto the space of polynomials of fixed degree. We prove that the Lie algebraic approximations of a class of Cauchy problems are well posed.

1. The Lie Algebraic Method of Discrete Approximations.

Every differentiable function $u : [a, b] \rightarrow \mathbb{R}$ can be approximated by its Lagrange polynomial of degree n . To construct the Lagrange polynomial $L[u]$ of a differentiable function $u : [a, b] \rightarrow \mathbb{R}$, the interval $[a, b]$ is partitioned by points $a = x_0 < x_1 < \dots < x_n = b$ and the values $L[u](x_i) = u(x_i)$ are prescribed, where $i = 0, 1, 2, \dots, n$.

The Lagrange polynomial of u is uniquely defined by its $n+1$ values at the points of the partition and can be written in the form:

$$L[u](x) = \sum_{i=0}^n u(x_i) l_i(x),$$

where

$$l_i(x) = \frac{\prod_{k=0, k \neq i}^n (x - x_k)}{\prod_{k=0, k \neq i}^n (x_i - x_k)}$$

is the i -th standard Lagrange polynomial, $0 \leq i \leq n$.

Let A be an operator acting on differentiable functions defined on $[a, b]$. We assume that A belongs to the enveloping algebra of the Heisenberg-Weyl algebra of differential operators $\left\{1, x, \frac{d}{dx}\right\}$. The value of A at a function f is denoted by $A[f]$. For a differentiable function $u : [a, b] \rightarrow \mathbb{R}$ and a partition $\sigma = \{x_0, x_1, \dots, x_n\}$ of $[a, b]$, let $u_\sigma = (u(x_0), u(x_1), \dots, u(x_n))^T$ be the vector of the values of u at the nodes of the partition. The finite dimensional representation $A_\sigma : \mathbb{R}^{n+1} \rightarrow \mathbb{R}^{n+1}$ of A can be constructed from the condition:

$$A_\sigma u_\sigma = A[L[u]]_\sigma.$$

It is easy to see that the (k, j) -th entry of the matrix A_σ is $[A_\sigma]_{kj} = A[l_j](x_k)$. The finite dimensional representation $Z = \{Z_{kj}\}_{k, j=0}^n$ of the operator $\frac{d}{dx}$ is then given by

$$Z_{jk} = \frac{dl}{dx}(x) = \begin{cases} \sum_{m=0, m \neq j}^n \frac{1}{(x_j - x_m)} & \text{if } k = j \\ \frac{\pi_j}{\pi_k} \frac{1}{x_j - x_k} & \text{if } k \neq j \end{cases}, \quad (1)$$

where $\pi_k = \prod_{m=0, m \neq k}^n (x_k - x_m)$ for every $k \in \{1, 2, \dots, n\}$.

The finite dimensional representation X of the operator x of multiplication by the variable x is the square matrix of dimension $(n+1)$ with the diagonal $(x_0, x_1, \dots, x_n)$:

$$X = \text{Diag}(x_0, x_1, \dots, x_n). \quad (2)$$

The finite dimensional representation of the identity operator 1 is the $(n+1)$ - dimensional identity matrix denoted by I_{n+1} .

Having constructed the finite dimensional representations of the operators $1, x, d/dx$, we can approximate every operator A in the enveloping algebra of $\{1, x, d/dx\}$.

Example 1.1. We approximate the operator $A = \frac{d^2}{dx^2} + 1$ by $Z^2 + 1$.

Using this approximation, we reduce the boundary value problem

$$\begin{cases} u'' + u = 0 \\ u(0) = 2, \quad u\left(\frac{\pi}{2}\right) = 1 \end{cases} \quad (3)$$

for a C^2 function $u : [a, b] \rightarrow \mathbb{R}$, where $\alpha, \beta \in \mathbb{R}$, to the problem of finding a vector $u_\sigma \in \mathbb{R}^n$ that satisfies

$$\begin{cases} Z^2 u_\sigma + u_\sigma = 0 \\ [u_\sigma]_0 = 2, \quad [u_\sigma]_n = 1 \end{cases} \quad (4)$$

It is natural to ask when the finite dimensional approximation of the original problem has a unique solution in $\mathbb{R}^n$. We address this question in the following sections.

2. Rank of the Finite Dimensional Representation of the differential operator $\sum_{p=k}^m a_p \frac{d^p}{dx^p}$

Let $1 \leq k \leq m \leq n$ and let $a_k, a_{k+1}, \dots, a_m$ be nonzero polynomials. In this section we prove that the finite dimensional representation

$$a_k Z^k + a_{k+1} Z^{k+1} + \dots + a_m Z^m$$

of the differential operator

$$a_k \frac{d^k}{dx^k} + a_{k+1} \frac{d^{k+1}}{dx^{k+1}} + \dots + a_m \frac{d^m}{dx^m}$$

has rank $n+1-k$.

Lemma 2.1. Let $\sigma = \{x_0, x_1, \dots, x_n\}$ be a partition of the interval $[a, b]$ and let Z , given by formula (1), be the finite dimensional

representation of d/dx with respect to this partition. Then $\text{rank } Z = n$ and $Z^{n+1} = 0$.

Proof. First, let us show that $Z^{n+1} = 0$. It is easy to verify that

$$\frac{d^k l_j}{dx^k}(x) = \sum_{m=0}^n [Z^k]_{jm} l_m(x)$$

for every $k \in \mathbb{N}$ and $j \in \{0, 1, \dots, n\}$. Because $\frac{d^{n+1} l_j}{dx^{n+1}}(x) = 0$ and the polynomials $l_0(x), \dots, l_n(x)$ are linearly independent, we have $Z^{n+1} = 0$.

Let us now prove that $\text{rank } Z = n$.

There exists a nonsingular matrix $B = \{B_{ij}\}_{i,j=0}^n$ of dimension $n+1$ such that

$$l_j(x) = \sum_{p=0}^n B_{jp} x^p$$

for every integer j between 0 and n . Then

$$\begin{aligned} \frac{dl_j}{dx}(x) &= \sum_{m=0}^n Z_{jm} l_m(x) = \\ &= \sum_{m=0}^n \sum_{p=0}^n Z_{jm} B_{mp} x^p = \\ &= \sum_{p=0}^n [ZB]_{jp} x^p \end{aligned}$$

Because the polynomial $\frac{dl_j}{dx}(x)$ has degree $n-1$, the last column of the matrix ZB must be zero. Therefore, $\text{rank} Z = \text{rank} ZB \leq n$.

It remains to show that $\text{rank } Z$ cannot be strictly smaller than n . Suppose, on the contrary, that $\text{rank } Z < n$. Then there exist n linearly dependent rows in Z . Without loss of generality we assume that the first n rows of Z are linearly dependent; denote these rows by $A_0, A_1, \dots, A_{n-1}$. There exist constants $\alpha_0, \alpha_1, \dots, \alpha_{n-1}$, not all zeros, such that

$$\sum_{j=0}^{n-1} \alpha_j A_j = 0$$

But then

$$\begin{aligned}\sum_{j=0}^{n-1} \alpha_j \frac{dl_j}{dx}(x) &= \sum_{j=0}^{n-1} \sum_{m=0}^n \alpha_j [A_j]_m l_m(x) = \\ &= \sum_{m=0}^n \left[\sum_{j=0}^{n-1} \alpha_j A_j \right]_m l_m(x) = 0,\end{aligned}$$

which, after integration, gives us

$$\sum_{j=0}^{n-1} \alpha_j l_j(x) = C,$$

where C is a constant.

By setting $x = x_j$ in the previous equality, where $j \in \{0, 1, \dots, n-1\}$, we obtain $\alpha_0 = \alpha_1 = \dots = \alpha_{n-1} = C \neq 0$. On the other hand, if we set $x = x_n$ in the very same equality, we obtain $C = 0$, a contradiction. We have thus proved that $\text{rank } Z$ cannot be smaller than n .

Lemma 2.2. *Let H be a square matrix of dimension $n+1$ such that $\text{rank } H = n$ and $H^{n+1} = 0$. Then $\text{rank } H^k = n+1-k$. Moreover, the Jordan form of H is*

$$J = \begin{pmatrix} 0 & I_n \\ 0 & 0 \end{pmatrix}, \quad (5)$$

where I_n is the identity matrix of dimension n .

Proof. Let J be the Jordan form of H so that $H = B^{-1}JB$ for some invertible matrix B and $\text{diag}(J) = (\lambda_0, \lambda_1, \dots, \lambda_n)$ where λ_i are the eigenvalues of H .

Then $0 = H^{n+1} = B^{-1}J^{n+1}B$ and $J^{n+1} = 0$. Therefore, the matrix J has only one eigenvalue 0 of multiplicity $n+1$ and is given by formula (5). It is now easy to verify that

$$J^k = \begin{pmatrix} 0 & I_{n+1-k} \\ 0 & 0 \end{pmatrix}, \quad (6)$$

and $\text{rank } J^k = n+1-k$. Thus, $\text{rank } H^k = \text{rank } J^k = n+1-k$ as required.

Theorem 2.3. Let $P(z) = a_k z^k + a_{k+1} z^{k+1} + \dots + a_m z^m$ be a polynomial of degree m with real coefficients, where $k \in \{0, 1, \dots, m\}$ and $a_k \neq 0$. If a square matrix B of size $(n+1)$ satisfies $\text{rank } B = n$ and $B^{n+1} = 0$, then $\text{rank } P(B) = n+1-k$.

Proof. Let us rewrite $P(B)$ as

$$P(B) = a_k B^k \left(I + \frac{a_{k+1}}{a_k} B + \frac{a_{k+2}}{a_k} B^2 + \dots + \frac{a_m}{a_k} B^{m-k} \right).$$

$$\text{Define } D = \frac{a_{k+1}}{a_k} B + \frac{a_{k+2}}{a_k} B^2 + \dots + \frac{a_m}{a_k} B^{m-k}.$$

Because the matrix B is nilpotent, D is nilpotent. But then the matrix $I + D$ is invertible. Indeed, the series $\sum_{p=0}^{\infty} (-D)^p$ is a finite sum and equals $(I + D)^{-1}$.

In summary, $P(B) = B^k A$, where A is an invertible matrix. Therefore, $\text{rank } P(B) = \text{rank } B^k = n+1-k$ by lemma 2.2.

The latter theorem can be easily generalized to the following statement.

Theorem 2.4.

Let $P(x, z) = a_k(x) z^k + a_{k+1}(x) z^{k+1} + \dots + a_m(x) z^m$ be a polynomial of degree m , with respect to variables x and z , with real coefficients, where $k \in \{0, 1, \dots, m\}$ and $a_k(x) \neq 0$. If a square matrix B of size $n+1$ satisfies $\text{rank } B = n$ and $B^{n+1} = 0$, and X is a diagonal matrix with nonzero diagonal, then $\text{rank}(X, B) = n+1-k$.

Theorem 2.5. Let $\sigma = \{x_0, x_1, \dots, x_n\}$ be a partition of the interval $[a, b]$ and let Z and X , be the finite dimensional representations of the operators d/dx and x , respectively, corresponding to this partition. Let $P(x, z)$ be a polynomial defined in the hypothesis of theorem 2.4. Then

$\text{rank } Z = n$, $Z^{n+1} = 0$, and $\text{rank}(X, Z) = n+1-k$.

Let us revise example 1.1 in the view of what we have proved. The finite-dimensional approximation of the differential equation $u'' + u = 0$ on the interval $[0, \pi/2]$ is given by

$$Z^2 u_\sigma + u_\sigma = 0, \quad (7)$$

where $u_\sigma \in \mathbb{R}^n$ is a vector whose coordinates approximate the values of the solution u at the nodes of a partition a of $[0, \pi/2]$.

By the last theorem, the rank of the matrix $Z^2 + I$ is $n+1$, thus the system of linear equations (4) has a *unique* solution. On the other hand, we know that the space of solutions to the equation $u'' + u = 0$ is two-dimensional. Therefore, such direct application of the Lie-algebraic method does not help us approximate all the solutions of the differential equation, only some of them. It is possible, however, to still approximate solutions of initial and boundary value problems using the Lie-algebraic method. We simply need to make an appropriate substitution so that to ensure that the solution of the differential equation automatically satisfies initial or boundary conditions. For example, to solve the boundary value problem

$$\begin{cases} u'' + u = 0, & x \in \left[0, \frac{\pi}{2}\right], \\ u(0) = 2, & u\left(\frac{\pi}{2}\right) = 1 \end{cases} \quad (8)$$

we make a substitution

$$u = (2 - 2/\pi x)(x(x - \pi/2)v + 1)$$

and obtain the equation

$$p(x)v'' + q(x)v' + r(x)v = s(x),$$

where

$$p(x) = -\frac{2}{\pi}x^3 + 3x^2 - \pi x, \quad (9)$$

$$q(x) = 2\left(-\frac{6}{\pi}x^2 + 6x - \pi\right), \quad (10)$$

$$r(x) = -\frac{12}{\pi}x + 6 + x\left(2 - \frac{2}{\pi}x\right)\left(x - \frac{\pi}{2}\right), \quad (11)$$

$$s(x) = \frac{2}{\pi}x - 2, \quad (12)$$

whose discrete approximation is

$$(p(X)Z^2 + q(X)Z + r(X))v_\sigma = s(X),$$

where a is a partition of $[0, \pi/2]$. By theorem 2.5, the latter system has a unique solution as long as the matrix X does not contain zeros on its diagonal.

We have chosen the partition of the interval $\left[0.001, \frac{\pi}{2}\right]$ with equally distributed n nodes $x_i = 0.001 + \frac{i}{n}\left(\frac{\pi}{2} - 0.001\right)$. We compared the values $(u(x_0), u(x_1), \dots, u(x_n))$ of the vector v_σ with the values of the exact solution $u(x) = \sin x + 2\cos x$ to the boundary value aforesaid problem. The values of the error $E_n = \sum_{k=0}^n \left| [v_\sigma]_k - u(x_k) \right|$ are given in the following table.

Table 1

The error E_n of the CMPS scheme for boundary value problem, where n is the number of subintervals in the partition of $\left[0.001, \frac{\pi}{2}\right]$

n	4	8	12	16
E_n	2.2788e-004	9.5522e-007	3.9033e-009	1.5205e-011

The latter table demonstrates fast convergence of the CMPS scheme.

АВТОМАТИЗАЦІЯ ДОКУМЕНТООБІГУ В ОРГАНІЗАЦІЇ

М.Б. Вітер

*к.ф.-м.н., доцент, начальник відділу організації обміну та аналізу
інформації Міністерства фінансів України*

Здійснено аналіз ефективності впровадження електронного документообігу в організації, його складових, а також відповідного нормативно-правового забезпечення.

Документообіг – це рух документів в організації з моменту їх отримання або створення до завершення виконання чи відсилання. Правильна організація документообігу сприяє оперативному проходженню документів в апараті управління, врівноваженню навантаження на підрозділи та на посадових осіб, позитивно сприяє на процес управління в цілому.

За даними обстеження, що проводилося організацією Siemens Business Services у ряді компаній:

- 30% часу робочих груп витрачається на пошуки та погодження документів;
- 6% документів безповоротно губляться;
- кожний внутрішній документ копіюється до 20 разів;
- на 20-25% зростає продуктивність праці персоналу при використанні електронного документообігу;
- вартість архівного збереження електронних документів на 80% нижче в порівнянні з їхніми паперовими копіями.

На даний час на Україні уніфікована система документації включає 16 уніфікованих підсистем документації з 4500 уніфікованих форм документів (планової, звітно-статистичної, первинно-облікової, організаційно-розпорядчої, розрахунково-грошової документації та ін.).

Докорінна зміна самої технології документування і документообігу відбувається завдяки застосуванню нових комп'ютерних технологій, що базуються на використанні спільної інформаційної бази, створенні автоматизованих робочих місць, використанні мережевих технологій обміну даними.

Задачі, які вирішуються в процесі автоматизації документообігу установи, багато в чому залежать від характеру її діяльності, інфраструктури, адміністративної організації та багатьох інших факторів.

Внаслідок автоматизації документообігу повинен бути забезпечений контроль над проходженням документів, а також введення до системи документів з різних джерел (паперова документація, електронна пошта та ін.).

Електронний документообіг – це сукупність процесів створення, оброблення, відправлення, передавання, одержання, зберігання, використання та знищення електронних документів, які виконуються із застосуванням перевірки цілісності та у разі необхідності з підтвердженням факту одержання таких документів.

Основною одиницею обліку в системі є реєстраційна картка документа, повнота реквізитів якої забезпечує можливість формування статистичних і аналітичних звітів по різних інформаційних зрізах, легкість під час пошуку.

Електронний документ – документ, інформація в якому зафіксована у вигляді електронних даних, включаючи обов'язкові реквізити документа. Електронний документ може бути створений, переданий, збережений і перетворений у візуальну форму (на екран чи на папір). Обов'язковим реквізитом електронного документа є електронний підпис.

Автоматизація документування практично знімає проблему можливої невідповідності даних первинних документів і введеної в комп'ютер інформації. Кодування реквізитів-ознак здійснюється автоматизовано з використанням інформації класифікаторів.

При одержанні електронного вхідного документа його реєстрація здійснюється в автоматизованому режимі. Для документа, отриманого в результаті обміну між двома системами електронного документообігу, більша частина реквізитів реєстраційної картки формується автоматично. Електронний образ документа прикріплюється до реєстраційної картки. При цьому засвідчується ідентичність документа, що має електронний цифровий підпис і криптозахист.

Сучасні інформаційні системи працюють з інформацією двох видів – структурованою (з баз даних) і неструктурованою (електронні документи). Процентне співвідношення між цими

видами інформації характеризує рівень автоматизації документообігу на підприємстві.

Використання комп'ютерної мережі створює умови для децентралізації виконання різних ділових функцій при збереженні централізованого контролю за документообігом. Можна, наприклад, дозволити підрозділам і навіть окремим працівникам здійснювати самостійну реєстрацію документів і при цьому централізовано контролювати їхнє проходження.

Відносини, пов'язані з електронним документообігом та використанням електронних документів, регулюються Конституцією України, Цивільним кодексом України, законами України “Про інформацію”, “Про захист інформації в автоматизованих системах”, “Про державну таємницю”, “Про телекомунікації”, “Про обов'язковий примірник документів”, “Про електронний цифровий підпис”, “Про електронні документи та електронний документообіг”, “Про Національний архівний фонд та архівні установи”.

В органах державної влади електронний документообіг додатково регулюється постановами Кабінету Міністрів України від 28.10.2004 N 1452 “Про затвердження Порядку застосування електронного цифрового підпису органами державної влади, органами місцевого самоврядування, підприємствами, установами та організаціями державної форми власності” та від 28.10.2004 N 1453 “Про затвердження Типового порядку здійснення електронного документообігу в органах виконавчої влади”, наказом Держкомінформатизації від 16.03.2010 №16 “Про затвердження вимог до форматів електронного документообігу в органах виконавчої влади”, а також іншими нормативно-правовими актами.

Електронний документообіг дозволяє створити в організації єдиний інформаційний простір, інтегруючи в інформаційний вузол усі документальні системи. Інтеграція здійснюється без втрати якості роботи з документами.

Основа подібної інтеграції – надійне сховище документів і взаємодіючі з ним системи документообігу. Всі оброблювані документи зберігаються в єдиному сховищі, що дозволяє забезпечити оптимальний пошук і вибірку інформації при підготовці матеріалів.

ДОЦІЛЬНІСТЬ ЗАСТОСУВАННЯ АВТОМАТИЗОВАНОГО ТЕСТУВАННЯ З ТОЧКИ ЗОРУ ЕКОНОМІЧНИХ ЗАТРАТ

¹А. Батюк, ²Л. Мандзюк

¹к.т.н., доцент, доцент кафедри АСУ НУЛП

²кафедра АСУ НУЛП

Process of the project to develop information systems aimed at the optimal solution of customer business objectives expressed in the form of requirements for software development. Project managers follow different development methodologies, identify strengths and weaknesses of any approach in planning processes and documenting procedures. One of the main issues such a test. The process solves the problem of verification testing requirements specified in the software implementation, providing information about the product. To ensure proper quality of developed software tool, to use modern high-efficiency methods, technologies and design tools, development, maintenance and software development. It is advisable to use automated testing in terms of savings and factors that reduce the risk of damage.

Вступ. Будь-яка інформаційна система (ІС) повинна вирішувати певні завдання замовника або клієнта, задовольняти вимоги до функціональності, які виходять безпосередньо від замовника, або формалізуються бізнес-аналітиками, які беруть участь у проекті з розробки ІС. Процеси реалізації ІТ-проекту з розробки ІС спрямовані на оптимальне вирішення бізнес-завдань замовника, виражених у вигляді вимог до розробки програмного забезпечення.

Основні процеси створення ІС. Сьогодні досить багато публікацій присвячено тому, як створюються інформаційні системи і реалізуються проекти [1,2]. Автори дотримуються різних методологій розробки, визначають переваги та недоліки того чи іншого підходу в плануванні процесів або документуванні процедур, проте загальна схема створення інформаційних систем досить проста і, як правило, складається з таких основних процесів [3]:

1. Постановка задачі у вигляді визначення вимог і подальших робіт з ними;
2. Управління змінами в проекті, при чому такі зміни можуть

- стосуватися як самих вимог до системи, так і організаційної схеми процесу;
3. Розробка програмного забезпечення як безпосереднє написання коду програмної реалізації функціональних вимог;
 4. Тестування програмного забезпечення як процес, що вирішує задачу верифікації відповідності вимог, висунутих до інформаційної системи;
 5. Експлуатація програмного забезпечення, включаючи і необхідне системне адміністрування.

Процес тестування. Розглянемо детальніше процес тестування як складову загального процесу забезпечення якості розробки програмного забезпечення. Вхідною інформацією для процесу тестування є, по-перше, дані про завдання, що вирішуються за допомогою програмної реалізації (ці дані утворюються в результаті формулювання вимог), а по-друге, повідомлення про зміни (ці повідомлення створюються в результаті процесу управління змінами). Процес тестування вирішує задачу верифікації зазначених вимог в програмній реалізації, надаючи інформацію про стан продукту, на основі якої приймається рішення про критерії завершення етапу розробки. Таким чином, тестування, як процес тісно пов'язане з іншими процесами проекту, використовує інформацію, отриману від них, і, у свою чергу, є джерелом інформації для процесу управління проектом.

Поширена помилка, що тестування – це заключна частина розробки програмного забезпечення. Однак, коли продукт вже готовий і в ньому виявлені такі серйозні помилки, як невідповідність початковим вимогам до функціональності, то в більшості випадків виправити їх можна тільки одним способом – значно змінити або навіть переробити вже закінчений програмний комплекс. Це вимагає невинновдано високих тимчасових, грошових і трудових затрат. Чим пізніше буде виявлено дефект або логічну невідповідність у функціональності ІС, тим більший обсяг реалізованої функціональності і логіки спільної роботи модулів потребують переробок. Тому процес тестування розпочинається ще на перших етапах виконання проекту. Зверніть увагу, що процес тестування починається навіть дещо раніше, ніж процес розробки, що характерно для правильного підходу. На рис.1 представлені фази

розробки програмного забезпечення та процеси, що відбуваються на цих фазах.

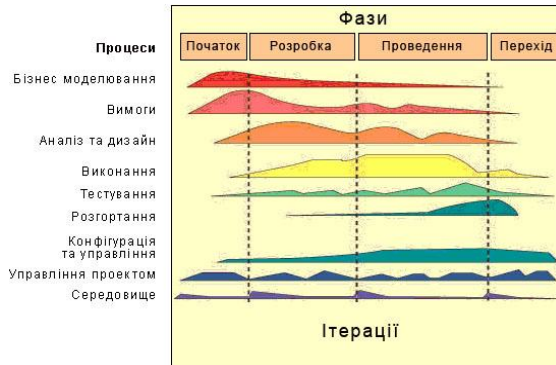


Рис.1. Процеси та стадії життєвого циклу розробки системи

Особливе значення при розробці великих систем набуває технологія автоматизованого тестування, оскільки вона дозволяє автоматизувати великий обсяг робіт з тестування і забезпечити необхідний рівень якості розроблюваного програмного продукту. Проекти з розробки складних програмних систем характеризуються більшим обсягом змін, що вносяться до структури програмного забезпечення, які породжуються роботами з додаткового тестування раніше перевіреної функціональності. Регресійне тестування, як процес, що забезпечує перевірку існуючої функціональності продукту після внесення змін, в системах з тривалим терміном розробки і більшим обсягом функціональності, характеризується ресурсоемістю, витратністю і тривалістю, але разом з тим воно є технологічно необхідним видом робіт і пріоритетним для автоматизації. Автоматизація рутинних, часто повторюваних операцій, які важко виконувати вручну – одне з найбільш доцільних вкладень при комплексній автоматизації тестування.

Прикладами вищевказаних операцій можуть служити:

1. Написання текстів сценаріїв тестування, які при автоматизованому тестуванні створюються автоматично в процесі спостережень за роботою користувача або шляхом діалогу з користувачем;
2. Запам'ятовування результатів тестування, так як при автоматизованому тестуванні вони зберігаються в базах

- даних (із зазначенням умов тестування, параметрів операційного середовища і т.д.) і тим самим стають доступними для подальшого аналізу;
3. Підготовка тестових даних, які автоматично формуються при автоматизованому тестуванні або в ході тестування, або перед його початком; при цьому користувач може описати кілька законів, на основі яких такі дані будуть створюватися;
 4. Запуск тестів у певній послідовності з певними тестовими даними, адже в разі автоматизації виконання тестів може відбуватися в пакетному режимі в заздалегідь визначений час;
 5. Одночасна робота декількох користувачів, яка при автоматизованому тестуванні імітується на програмному рівні; для управління процесом потрібний всього один, а не десятки чи сотні користувачів, як при ручному тестуванні).

Для того щоб забезпечити належний рівень якості розроблюваного програмного засобу, необхідно застосовувати сучасні високоефективні методи, технології та засоби проектування, розробки, супроводу та розвитку програмного забезпечення. Крім аспектів доцільності тестування в процесі розробки ІС і ризиків ІТ-проектів, розглянемо результати розрахунків інвестування в автоматизоване тестування.

Почнемо з питання про повернення інвестицій. Досвід свідчить, що впровадження практик і засобів навантажувального тестування дозволяє отримувати, крім безпосередніх прямих вигод, пов'язаних з економією коштів на підтримку та супровід розроблених систем, а також і вигоди за рахунок покритих цим видом тестування ризиків недоотримання прибутку.

Які ж фактори впливають на обсяги коштів автоматизованого навантажувального тестування? Всі чинники можна розділити на дві групи – це чинники, що викликають пряму економію коштів, і фактори, що знижують ризик нанесення збитків.

Основними пунктами економії є:

1. Економія коштів на реалізацію зайвої надійності продукту.

Витрати пояснюються витратами на реалізацію зайвої функціональності, розрахованої на виконання явно завищених (часом навіть у кілька разів) вимог до

продуктивності, або на зайву оптимізацію архітектури та коду, що збільшує терміни розробки та тестування, а отже, підвищує вартість проекту в цілому.

2. *Економія на вкладеннях в реалізацію зайвої надійності інфраструктури.*

Ці витрати за своїм характером та за причинами виникнення схожі з описаними вище і виникають через брак інформації про можливості застосовуваних технологій. Наслідки цього виражаються у витратах на аналіз і реалізацію зайвої надійності інфраструктури, механізмів дублювання логіки і дорогих рішень з агрегації і міграції даних.

3. *Економія часу і коштів на виявлення і усунення несправностей під час експлуатації.*

Сюди відносяться витрати служб експлуатації та контролю, а також системного адміністрування на локалізацію та виявлення причин збоїв, на відновлення систем до працездатного стану, а також на проведення робіт з відновлення даних і стану інформаційних систем.

Зазначені витрати можуть перетворюватися на істотні витрати у випадку аутсорсингу ІТ-завдань і сервісів за рамки компанії, яка використовує інформаційну систему для вирішення своїх бізнес-завдань.

Основні **ризики**, які покриваються за рахунок проведення навантажувального тестування, такі:

1. *Недоотримання прибутку внаслідок простою або низької продуктивності продукту.*

Цей ризик особливо актуальний для сучасних систем онлайнової торгівлі, обробки замовлень, формування пакетів доставки та транспортної логістики.

2. *Незворотні логічні збої в структурах зберігання даних у результаті несвоєчасних оновлень даних або ускладнених запитів на отримання актуальної інформації.*

Особливу актуальність даний ризик набуває в умовах великого обсягу оброблюваних сучасними системами даних та великої кількості виконуваних за одиницю часу операцій.

Критичний цей ризик і для систем оперативного контролю

та моніторингу, систем, що працюють в режимі реального часу, систем автоматизованого управління агрегатами і апаратами, а також для систем життєзабезпечення.

3. *Незворотні зміни або втрата операційних даних через програмні збої, викликані надмірними навантаженнями.*

Ризик актуальний у сфері зберігання та обробки фінансової інформації, яка носить унікальний характер, не може бути відновлена на основі емпіричних припущень і несе пряму загрозу фінансових збитків.

4. *Незворотні зміни або втрата операційних даних внаслідок апаратного збою, викликаного навантаженнями.*

Цей ризик актуальний не стільки через можливість реальних втрат даних, скільки за рахунок витрат часу, потрібного на відновлення інформації. Якщо інформація не може бути відновлена або зареєстрована в системі в результаті збою, ризик стає критичним.

А тепер наведемо результати практичних розрахунків [3] повернення інвестицій вкладених в автоматизоване функціональне тестування, які враховують витрати на придбання інструменту, навчання персоналу, а також реальну специфіку сучасних проектів з розробки інформаційних систем, що характеризуються великою кількістю уточнених та мінливих змін.

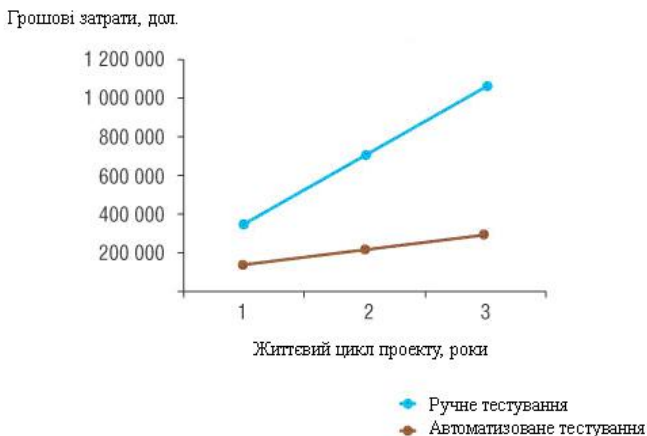


Рис.2. Порівняння витрат коштів на ручне і автоматизоване тестування

Таким чином, тут враховано витрати не лише на створення необхідних автоматизованих сценаріїв, але також на підтримку набору тестів в актуальному стані.

Розрахунки проводилися для проекту з автоматизації функціонального тестування. Тут обрано проект, обсяг робіт по тестуванню в якому вимірюється десятками тисяч тестових прогонів за час розробки і становить 65 тис. тестових операцій на рік. Вартість години роботи спеціаліста з тестування встановлена в розмірі 30 дол. на годину, а витрати на ліцензії до програмного забезпечення автоматизації тестування та навчання персоналу встановлені в 75 тис. дол..

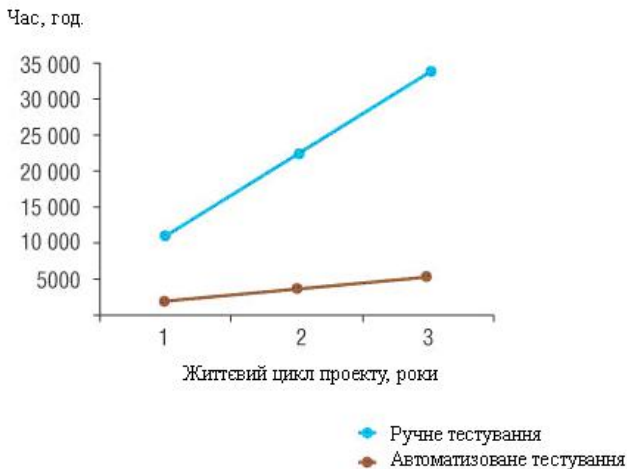


Рис. 3. Порівняння часових витрат при ручному і автоматизованому тестуванні

На рис. 2 і 3 представлені відповідно грошові та часові витрати при ручному і автоматизованому тестуванні. Як видно, витрати коштів на ручне тестування ростуть значно швидше, ніж витрати на автоматизацію тестування із зростанням функціональності системи. Зростання витрат часу в даному прикладі порівнюємо з зростанням грошових витрат. Хоча ситуація може трохи змінюватися від проекту до проекту, проте загальна тенденція зберігається.

Практика впровадження засобів автоматизованого тестування у великі софтверні проекти показує, що це дозволяє домогтися в

десятки разів повнішого покриття функцій ІС, ніж ручне тестування. Базуючись на цих розрахунках можна сказати, що час, витрачений на автоматизоване тестування, в сім разів менше сумарного часу, що витрачається на ручне тестування. А включаючи початкові витрати часу на навчання, отримуємо зростання показників продуктивності в сім разів, що підтверджується наведеним графіком розрахунків часових витрат.

Висновок. Отже, неодноразово доведено, що автоматизоване тестування має багато переваг над ручним, є більш ефективне, економне та доцільне до застосування під час розробки великих ІТ-проектів.

Література

1. Алистер Коуберн “Варианты использования, десять лет спустя” Журнал STQE Март/Апрель 2002 г.
2. Askar Rahimberdiev “Современные процессы разработки программного обеспечения” RSDN Magazine #4-2006
3. Александр Новичков, Вячеслав Панкратов “Автоматизированное тестирование: оценка возврата инвестиций и сопутствующие риски”

УПРАВЛІННЯ ЯКІСТЮ ПРОДУКЦІЇ (ПОСЛУГ) ЗА ДОПОМОГОЮ ЕКСПЕРТНИХ СИСТЕМ

¹А.В. Гунькало, ²О.В. Бойко, ³Н.В. Дорош

¹к.т.н., ст.викл. каф. метрології, стандартизації та сертифікації НУ ЛПІ

²к.т.н., ст.викл. каф. медичної інформатики ЛНМУ ім. Д.Галицького

³к.т.н., доцент кафедри “Електронні прилади” НУ ЛПІ

Розглянуто можливості використання інтелектуальних технологій в управлінні якістю продукції, що сприятиме підвищенню її якості та дозволить врахувати вимоги та очікування споживачів.

Ця робота виконувалась за рахунок бюджетних коштів Міністерства освіти і науки України, наданих як грант Президента України для підтримки наукових досліджень молодих учених.

Управління якістю продукції охоплює встановлення,

забезпечення і дотримання необхідного рівня якості продукції при її розробці, виготовленні та експлуатації, що досягається шляхом систематичного контролю за якістю і цілеспрямованого впливу на умови й фактори, від яких вона залежить. Комплексне управління якістю забезпечує конкурентоздатність випуску продукції як на національному, так і на міжнародному ринках. Використання інтелектуальних технологій, зокрема експертних систем (ЕС) в управлінні якістю продукції, сприятиме поліпшенню її якості і показників діяльності підприємства загалом, прийняттю ефективних управлінських рішень на основі врахування вимог та очікувань споживачів.

Пропонується впровадити у систему управління якістю, яка базується на процесному підході, експертну систему (рис.1).

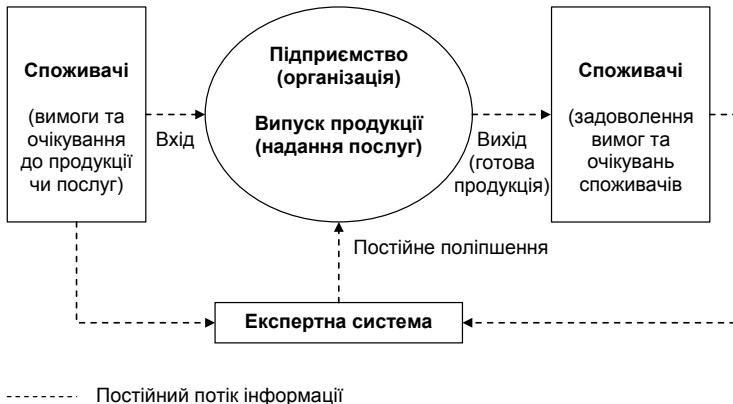


Рис.1. Система управління якістю (СУЯ) продукції (послуг), що базується на процесному підході із застосуванням експертної системи

Призначенням ЕС є обробка (аналіз) інформації, отриманої від споживачів шляхом опитування (анкетування) та забезпечення зворотного зв'язку між споживачами (ринком) та виробниками продукції. База знань ЕС включає ієрархічну сукупність понять предметної області та відношень між ними, що відображають уявлення експертів та споживачів про об'єкт.

Механізм виведення рішення є процедурою згортки поточних характеристик об'єктів по ієрархії бази знань (дерево виведення),

що дозволяє оцінити рівень якості конкретної продукції.

Експертна система в СУЯ є інструментом управління процесом випуску продукції (надання послуг) та удосконалення її якості.

ПРИЙНЯТТЯ РІШЕНЬ ЩОДО ФІНАНСУВАННЯ НАУКОВИХ РОЗРОБОК ЗА ДОПОМОГОЮ “ТЕХНОЛОГІЧНОГО АУДИТУ”

М.О. Цибинога

аспірант, Харківський державний технічний університет будівництва та архітектури

Характерною ознакою сучасної глобальної постіндустріальної революції є вагоме значення науки в розвитку виробничих сил, широке застосування наукових результатів для розробки та освоєння нових технологій, випуску конкурентоздатної продукції нового покоління, оновлення технологічних основ виробництва. В таких умовах постійного розвитку наукоємних виробництв інтелектуальна власність є одним з головних ресурсів подальшого зростання потужностей підприємств, в зв'язку з чим проблема оцінки результативності інноваційної діяльності набуває державного значення. Одним з можливих ефективних методів оцінки результативності наукових розробок є процедура технологічного аудиту, яка не лише формує універсальний звіт щодо проекту, а й виявляє перспективи комерціалізації отриманих результатів.

В ринкових умовах накопичена та знову створена інтелектуальна власність визначають ті ресурси, які виграють ключову роль в технологічному та соціально-економічному розвитку країни. В результаті науково-технічної діяльності дослідних установ створюються винаходи, корисні моделі, промислові зразки, товарні знаки, знаки обслуговування, найменування місць походження товарів, бази даних, топології інтегральних мікросхем, know-how, твори науки, літератури та інше що має перспективи бути комерціалізованим. Всі ці витвори при наявності комерційного потенціалу, який обумовлений світовою новизною, можливостями правового захисту та доведенням результатів до реальних вимог ринку спроможні бути об'єктами

ліцензування та основою для створення малих інноваційних підприємств. Такі результати науково-дослідницької діяльності можуть зацікавити інвесторів інноваційних проектів.

Одним з можливих ефективних методів оцінки результативності наукових розробок є процедура технологічного аудиту, яка не лише формує універсальний звіт щодо проекту, а й виявляє перспективи комерціалізації отриманих результатів.

Поняття “Технологічний аудит” містить в собі експертну оцінку діючих технологічних рішень або технологічних рішень, що проектуються, розробку рекомендацій з комплексу організаційно-технічних засобів, необхідних для забезпечення конкурентоспроможності ідей, розробок, технологій та оцінку їх економічної ефективності. Технологічний аудит наукової установи передбачає аналіз результатів НТП та визначення методів, прийомів та процедур щодо забезпечення виводу результатів НТП на ринок.

Задача аудиту – оцінити можливості та комерційний потенціал результатів НТП. Мірою ефективності може служити конкурентоспроможність продукту. В результаті проведення технологічного аудиту отримуємо висновки про комерційну перспективність проекту (ідеї, розробки, технології) і реальну картину того, як використовується технологічний, виробничий, фінансовий та інші потенціали установи для просування результатів НТП на ринок.

Процедура технологічного аудиту реалізована у вигляді комп’ютерної програми яка автоматизує проведення процедури технологічного аудиту та включає в себе блоки інтерактивного заповнення електронних форм, ухвалення рішень щодо висновків про ефективність НТП, автоматичне формування звіту про виконання проекту.

Програмний комплекс складається з таких інформаційних блоків:

- інформація про наукову установу;
- основні відомості про проект;
- детальний опис проекту;
- відомості про умови виконання проекту;
- науково-технічне підґрунтя проекту;
- маркетинговий аналіз проекту;
- конкурентне середовище результатів проекту;
- опис виробничого процесу проекту;

- план робіт за проектом;
- фінансові аспекти проекту;
- ризики проекту;
- права інтелектуальної власності на результати проекту;
- можливі напрямки комерціалізації;
- підготовка звітного документу.

На основі отриманої інформації здійснюється: 1) формування звіту про результати технологічного аудиту, та 2) необхідних інформаційних матеріалів для внесення в державний реєстр.

Форма звітного документу має таку структуровану форму:

- загальні дані відносно установи;
- профіль установи;
- дані про учасників технологічного аудиту;
- інноваційні характеристики проекту;
- маркетинговий аналіз;
- висновок експертів щодо перспективності комерціалізації проекту;
- резюме керівництва щодо підтримки комерціалізації результатів дослідження.

Дана процедура включає методичні та організаційні рекомендації для визначенням ефективності проектів, що розробляються та впроваджуються; сформульовані єдині вимоги до системи показників і методів їх визначення; загальну характеристику інновації та її розробників; рекомендації щодо визначенням рівня якості і конкурентоспроможності продукту, врахування ризиків і перспективи отримання прибутку.

Література

1. Програмний комплекс “Технологічний аудит” / Колодяжний В.М., Лісін Д.О., Гурко О.Г., Селищев В.С., Сандуленко К.Є. - №31992; 08.02.2010. Київ: Держ. підпр. реєстрації авторських прав, 2010.
2. Эффективность научно-технических проектов и программ: Монография. Пилипенко О.В., Переверзев Е.С., Алпатов А.П., Марченко В.Т., Печеневская О.К., Хоральский П.П. – Днепропетровск: Пороги, 2008. – 509 с.
3. Остапюк С.Ф., Филин С.А. Формирование и оценка эффективности научно-технических и инновационных программ. – М.: Изд-во ООО “Благовест-В”, 2004г.-320 с.

ОСОБЛИВОСТІ ФУНКЦІОНАЛЬНОГО АНАЛІЗУ ДАНИХ ПІД УПРАВЛІННЯМ ОПЕРАЦІЙНИХ СИСТЕМ LINUX

¹В.І. Дубук, ²Т.З. Цюра

¹к.т.н, доцент, доцент кафедри МКД ЛФСУ

²лаборант кафедри МКД ЛФСУ

Досліджено особливості функціонального аналізу даних засобами інформаційних систем під управлінням операційних систем сімейства Linux на основі програмного забезпечення Gnumeric. Розроблена програмна реалізація для деяких функцій, які можуть вбудовуватися у програмне забезпечення Gnumeric. Наведені висновки щодо можливостей практичного використання результатів роботи.

Для розв'язання задач аналізу даних на практиці можуть використовуватися комп'ютерно-орієнтовані методи, побудовані на основі застосування табличних процесорів [1,2,3].

Успішне розв'язання задач функціонального аналізу даних на основі заданих статистичних числових даних може здійснюватися засобами табличного процесора Gnumeric [3], що працює під управлінням ОС з сімейства Linux та Microsoft Windows і має вбудовану групу потужних інструментів [2]. Програмний засіб Gnumeric володіє вільною ліцензією на використання, тому його інсталяція і практичне використання є економічно ефективними.

Функціональний аналіз даних може ґрунтуватись на застосуванні вбудованих функцій табличного процесора Gnumeric [7]: обчислення трендів TREND, розрахунку прогнозних значень FORE-CAST, визначення коефіцієнтів апроксимації функціональних залежностей шляхом лінеаризації на основі методу найменших квадратів LINEST.

Розроблений метод полягає у покроковому визначенні прогнозних значень функціональної залежності для досліджуваного часового ряду даних. **Практичне застосування** методу на основі використання функції прогнозних значень FORECAST уможливорює виявити тенденції зміни фактичних значень параметра в часі і визначити його прогнозні значення в майбутні часові періоди.

Побудовані математичні моделі можна ефективно застосовувати для функціонального аналізу даних, коли для значень

прогнозованого параметра спостерігається стійка тенденція у часі процесу зміни значення функціональної залежності.

Висновки.

1. Для визначення прогнозного значення часового ряду і формулювання прогнозу про тенденції зміни витрат підприємства найприйнятнішою є модель на основі комбінованого застосування алгоритмів прогнозування з використанням вбудованих у програмний засіб Gnumeric функцій FORECAST, TREND та LINEST.
2. Розроблена модель може успішно використовуватись на проміжках монотонності досліджуваної функціональної залежності, забезпечує незначні відносні похибки прогнозування і уможливорює визначити прогнозне значення функціональної залежності на черговий наступний часовий відлік.
3. В роботі досліджено особливості функціонального аналізу даних засобами інформаційних систем під управлінням операційних систем сімейства Linux на основі програмного забезпечення Gnumeric. Розроблена програмна реалізація деяких функцій, які можуть вбудовуватися у програмне забезпечення Gnumeric, що підвищує ефективність функціонального аналізу даних.

Література

1. Гавриленко В.В., Пархоненко А.М. Решение задач аппроксимации средствами Excel // Компьютеры + программы, 2002.-№12.- С.42-47.
2. Дубук В.І., Цюра Т.З. Особливості аналізу економіко-статистичних даних під управлінням операційних систем Linux // Інформаційні технології в економіці, менеджменті і бізнесі. Проблеми науки, практики і освіти: Мат. XV Міжн.наук.-практ. конф., Київ, 25-26.02.2010 р./ Редкол.: І.І. Тимошенко (голова) та ін.-К.: Вид-во Єв-роп. університету, 2010, С.33-34.
3. Baudais E., Breit K., Custer A., Canty T., Dassen R., Goldberg J., Guelzow A.J., Hellan J.K., De Icaza M., Livonen J-P., Kirillov A., Klost S., Leblanc G., Luangkerson L., Miesbouer T., Schuller W., Tigelaar A.S., Twardy Ch., Weber A., Welinder M. The Gnumeric Manual. Ver.1.9.16 - The Gnumeric Team, Gnome Doc. Project, 1998-2009 // [http:// www.gnumeric.org](http://www.gnumeric.org)

СУЧАСНІ ПІДХОДИ ДО РОЗРОБКИ ІНВЕСТИЦІЙНИХ ПРОЕКТІВ

О.Р. Зав'ялова

старший викладач кафедри економічної теорії ЛКА

Сьогодні багато промислових підприємств перебувають в кризовому становищі, яке посилюється можливістю введення зовнішнього управління або конкурсного виробництва. Єдиним способом уникнути цього є об'єктивний аналіз допущених помилок, розробка антикризових заходів і плану стратегічного розвитку в середньостроковій і довгостроковій перспективі. Залучення зовнішнього фінансування для розвитку підприємств стає сьогодні найбільш актуальним завданням. Аналіз економічної ситуації в країні показує, що, незважаючи на кризовий стан економіки, створюються певні передумови для прямих інвестицій в промисловість. Очевидно, що основою інвестиційного процесу, швидше за все, буде розвиток малих і середніх підприємств, оскільки ризик вкладення засобів в цей сектор інвестицій значно нижче, ніж фінансування великих промислових підприємств.

Для оцінки **інвестиційної привабливості** підприємств і ефективності передбачуваних інвестицій, як правило, потрібна група експертів, що володіють спеціальними знаннями в різних областях економіки (маркетинг, фінансовий і управлінський облік, інвестиції, податкове планування і так далі). Такими кадрами володіють консалтингові фірми, які надають такого роду послуги підприємствам. Однак вартість цих послуг сьогодні багатьом підприємствам недоступна.

Тому єдиною можливістю, самостійно, розрахувати і проаналізувати інвестиційний проект, провести аналіз фінансово-господарській діяльності для оцінки інвестиційної привабливості свого підприємства, розробити стратегічний план розвитку, за допомогою власних фахівців, є їх навчання сучасним методам інвестиційного проектування і використання спеціальних комп'ютерних програм, які в тій або іншій мірі допомагають вирішити поставлені завдання. **Основні вимоги**, які пред'являються до комп'ютерних програм такого класу:

- проводити ретроспективний аналіз фінансово-

господарській діяльності з метою визначення найбільш слабких місць в діяльності різних підрозділів підприємства;

- проводити розрахунок і всебічний аналіз бізнес-плану інвестиційного проекту;
- готувати техніко-економічне обґрунтування кредиту, у разі залучення зовнішніх джерел фінансування;
- оцінювати вплив зовнішніх факторів і внутрішніх параметрів на загальну ефективність проекту;
- проводити порівняльну оцінку для відбору найбільш перспективного варіанту проекту;
- швидко виконувати всі рутинні обчислювальні операції;
- на підставі розрахунку і аналізу готувати документацію за проектом для представлення її потенційному інвесторові або кредиторіві.

Сьогодні на ринку існує декілька десятків комп'ютерних програм для розрахунку і порівняльного аналізу інвестиційних проектів, як вітчизняних, так і зарубіжних, яких можна назвати "Project Expert" фірми "ПРО-ІНВЕСТ КОНСАЛТИНГ", "Інвестор" фірми "ІНЕК", "Альт-інвест" фірми "Альт" (Санкт-Петербург), FOCCAL фірми "Центрінвестсофт", "ТЕО-ІНВЕСТ" Інституту проблем управління РАН, COMFAR (Computer Model for Feasibility Analysis and Reporting) і PROSPIN (Project Profile Screening and Pre-appraisal Information system), створені в UNIDO, – Організації Об'єднаних Націй по промисловому розвитку.

Ми виділили три основні російськомовні програми в тій ієрархічній послідовності, яку вони, на нашу думку, займають на сьогоднішній день – "PROJECT EXPERT 7.0 Professional", "ІНВЕСТОР" і "АЛЬТ-ІНВЕСТ". Для порівняння цих програм із західними аналогами вибрана програма "COMFAR 3.0".

Перш за все, слід зазначити, що сьогодні ці програми, крім їх основного призначення, з успіхом використовуються для розробки фінансової моделі і стратегічного плану розвитку промислових підприємств, що особливо актуально в умовах економічної кризи.

У основі цих комп'ютерних програм лежать методичні підходи UNIDO по проведенню промислових техніко-економічних досліджень, а також вітчизняні "Методичні рекомендації за оцінкою ефективності інвестиційних проектів і їх відбору для фінансування",

затверджені Держбудом, Мінекономіки, Мінфіном в 1994 р.

Основними критеріями для порівняння програмних продуктів були вибрані:

- функціональні можливості;
- використання сучасної методики розрахунку, заснованої на імітаційній моделі грошових потоків;
- комплексний підхід до вирішення різних аспектів інвестиційного проектування (ретроспективний аналіз фінансово-господарської діяльності, планування інвестиційної, операційної і фінансової діяльності, порівняльний аналіз, тощо);
- детальний опис параметрів проекту (крок і горизонт розрахунку, номенклатура продукції т.п.);
- можливість опису економічного середовища (інфляція, курс валют, податки і тощо);
- аналітичні можливості (повнота набору показників ефективності інвестицій, фінансових і виробничих показників);
- оформлення результатів.
- експлуатаційні і технічні характеристики
- сумісність з операційними системами;
- можливість передачі даних в поширені застосування MS Excel і MS Word;
- мова програмування, вимоги до апаратного забезпечення.
- інтерфейс
- простота і швидкість введення даних;
- показчик помилкових дій користувача при введенні даних;
- багатовіконний режим введення даних і відображення результатів;
- наочність результатів;
- графічне відображення результатів.

Крім вказаних критеріїв були також оцінені параметри “закритість” пакету та “вартісний” зміст програм.

“Закритість” пакету означає неможливість зміни користувачем формул і алгоритмів, по яких відбуваються обчислення в програмному пакеті. До “закритих” слід віднести пакети, написані на якій-небудь мові програмування, до “відкритих”

– пакети, написані на базі електронних таблиць, де користувач має можливість змінити алгоритм розрахунку. “Закритість” пакету може бути як перевагою, так і недоліком залежно від цілей, які стоять перед інвестором або реципієнтом. Кваліфіковані користувачі можуть створювати досить складні і різноманітні сценарії реалізації проекту, використовуючи гнучкі можливості відкритої системи. В той же час такі можливості пакету несуть в собі загрозу помилок. Відзначимо, що в “закритих” пакетах можливість помилок часто виникає через невірне розуміння користувачем можливостей пакету і економічного змісту дій, що проводяться (відсутня можливість прослідкувати по яких формулах проводиться розрахунок, відсутня контекстна і смислова допомога по різних таблицях або показниках).

Під “Вартісним” змістом програм слід розуміти наявність всіх перерахованих вище вимоги, які входять в ціну однієї програми згідно прайс-листу фірми. Якщо для отримання потрібних характеристик необхідно доповнювати програму додатковими модулями, які природно вимагають додаткових затрат, то це оговорюється окремо, щоб поставити всі програми в однакові умови. З цієї причини проводиться оцінка по показнику “функціональні характеристики/ціна”.

Висновки. Серед найбільш поширених на українському ринку комп’ютерних програм для розрахунку і аналізу інвестиційних проектів, слід вказати, що всі вітчизняні програми по своїх функціональних можливостях і цінових показниках перевершують зарубіжні аналоги. Вибір конкретної програми має бути визначений користувачем залежно від поставлених задач.

1. Програму “Project Expert” можна рекомендувати тим, хто працює, в основному, із західними інвесторами, бо для західних інвесторів найбільш звичний тип документів, що готується даною програмою.
2. Програма “ІНВЕТОР” може бути рекомендована тим, хто свою діяльність пов’язує з вітчизняними інвесторами або передбачає брати участь в інвестиційних конкурсах, що проводяться різними державними структурами.
3. Програму “Альт-Інвест”, швидше за все, можна рекомендувати консалтинговим фірмам як основу для розробки індивідуальної моделі функціонування підприємства.

4. Програми “COMFAR” залишається певним загальноновизнаним міжнародним еталоном, але її ціна і функціональні характеристики, навряд чи привернуть до неї увагу наших користувачів.

Література

1. Басовский Л.Е., Басовская Е.Н. Экономическая оценка инвестиций. – М.: “Инфра-М”, 2007. – 242 с.
2. Волков А. Инвестиционные проекты: от моделирования до реализации. – М.: “Дело”, 2008. – 688 с.
3. Попов В.Л. Управление инновационными проектами. – М.: “Инфра-М”, 2009. – 336 с.
4. <http://cfin.ru/management>.

БАГАТОВИМІРНІ СЕРЕДНІ В ОЦІНЦІ ЧАСОВИХ РЯДІВ ВТРАТ ЛЮДСЬКОГО КАПІТАЛУ УКРАЇНИ

Л.І. Гальків

к.е.н., доцент, доцент кафедри економетрії та статистики ЛКА

Проведено оцінку втрат людського капіталу України в динаміці методом багатовимірних середніх.

В останні десятиліття у цілому світі активізувалися дослідження людського потенціалу, в тому числі чинників, які зумовлюють втрати і погіршення характеристик наявного людського капіталу.

Серед вітчизняних дослідників людського капіталу відомі імена О.Грішньої, В.Денесюка, Я.Жаліло, І.Каленюка, В. Куценко, П.Крайнева, Е.Лібанової, А.Мокія, С.Пирожкова, П.Цибулева, А.Чухна, Л.Семів та ін. Науковці сходяться на тому, що обсяг людського капіталу можна оцінювати не лише кількісно, але й якісно.

Для кількісної оцінки людського капіталу і визначення величини його втрат пропонуємо використати **багатомірний порівняльний аналіз**. При цьому аналіз можна здійснювати в

статистиці (для одного, декількох чи усіх регіонів України) чи у динаміці (за кілька періодів часу). Чим більше показників буде використано для комплексної оцінки, тим достовірнішими будуть висновки аналізу. Проте надто велике збільшення кількості показників ускладнює методику дослідження.

Для оцінки людського капіталу України за період 2005-2009рр. нами обрано три показники (табл.1):

- 1) частка зайнятого населення працездатного віку;
- 2) коефіцієнт природного приросту населення;
- 3) міграційний приріст населення.

Динаміку втрат людського капіталу оцінено також на основі трьох показників:

- 1) частка незайнятого населення;
- 2) коефіцієнт смертності;
- 3) обсяг зовнішньої міграції.

Оскільки в кожну із систем входять неспіввимірні показники, тому необхідно здійснити їх стандартизацію.

Таблиця 1

Вихідні дані оцінювання обсягу людського капіталу та його втрат в Україні за 2005-2009 рр*.

Роки	Характеристики					
	людського капіталу			втрат людського капіталу		
	частка зайнятого населення працездатн. віку, %	коефіцієнт природного приросту населення, проміле	сальдо міграційного приросту населення, тис. осіб	частка незайнят. населення, %	коефіцієнт смертності населення, проміле	кількість вибулих за кордон, осіб
2005	65,4	-7,6	4,6	56,10	16,6	34997
2006	65,9	-6,4	14,2	55,69	16,2	29982
2007	66,7	-6,2	16,8	55,05	16,4	29669
2008	67,3	-5,3	14,9	54,66	16,3	22402
2009	64,7	-4,2	13,4	56,16	15,3	19470

* Джерело: дані Держкомстату України

Стандартизовану систему показників (табл.2 і табл.3) отримано за наступним алгоритмом:

1. усі показники поділено на стимулятори та дестимулятори;
2. у кожній графі табл.1 визначено максимальний елемент, що приймається за одиницю;
3. для стимуляторів усі елементи графи (a_{ij}) поділено на максимальний елемент ($\max a_{ij}$):

$$X_{ij} = \frac{a_{ij}}{\max a_{ij}} \quad (1)$$

4. для дестимуляторів максимальний елемент ($\max a_{ij}$) поділено на усі елементи графи (a_{ij}).

Зауважимо, що при оцінці обсягу вітчизняного людського капіталу в 2005-2009 рр. коефіцієнт природного приросту приймає від'ємне значення, тому збільшення його абсолютної величини виявляє дестимулюючий вплив. Питома вага зайнятого населення та міграційний приріст стимулюють обсяг людського капіталу. На обсяг втрат людського капіталу усі наведені в табл.3 показники виявляють стимулюючий вплив.

Таблиця 2

Матриця стандартизованих коефіцієнтів оцінки обсягу людського капіталу України за 2005-2009 рр.

Роки	Частка зайнятого населення працездатного віку	Коефіцієнт природного приросту населення	Сальдо міграційного приросту населення
2005	0,971768	0,552632	0,27381
2006	0,979198	0,65625	0,845238
2007	0,991085	0,677419	1
2008	1	0,792453	0,886905
2009	0,961367	1	0,797619

Таблиця 3

Матриця стандартизованих коефіцієнтів оцінки обсягу втрат людського капіталу України за 2005-2009 рр.

Роки	Частка незайнятого населення	Коефіцієнт смертності населення	Кількість вибухів за кордон
2005	0,998932	1	1
2006	0,991631	0,975904	0,856702
2007	0,980235	0,987952	0,847758
2008	0,973291	0,981928	0,640112
2009	1	0,921687	0,556333

Вважаючи показники системи рівновагомими, багатовимірну середню, яка виступає їх інтегральною оцінкою, у кожному році аналізованого часового ряду обчислено за арифметичною простою формулою:

$$\overline{X_i} = \frac{\sum_{j=1}^n X_{ij}}{n}, \quad (2)$$

де i – номер року; j – номер показника; n – кількість показників.

За величиною багатовимірних середніх здійснено ранжування часових рядів (табл.4).

Дані табл. 4 вказують на те, що інтегральна оцінка обсягу вітчизняного людського капіталу найменше значення приймала у 2005 році (59,94%). Щорічно цей показник зростав і досяг у 2009 році максимального значення – 91,97%. При аналізі втрат людського капіталу тенденція протилежна. Так, найбільше значення багатовимірної середньої втрат людського капіталу в Україні за досліджуваними показниками визначено для 2005 року (99,96%), а найменше – для 2009 року (82,60%).

Таблиця 4

**Багатовимірні оцінки обсягу людського капіталу та його втрат
в Україні за 2005-2009 рр.**

Роки	Людський капітал		Втрати людського капіталу	
	багатовимірна середня	ранг року	багатовимірна середня	ранг року
2005	0,599403	1	0,999644	5
2006	0,826895	2	0,941412	4
2007	0,889501	3	0,938648	3
2008	0,893119	4	0,865110	2
2009	0,919662	5	0,826007	1

Для характеристики динаміки рівня втрат людського капіталу обчислимо показник, що характеризує співвідношення багатовимірних середніх втрат людського капіталу та його обсягу (рис.1).

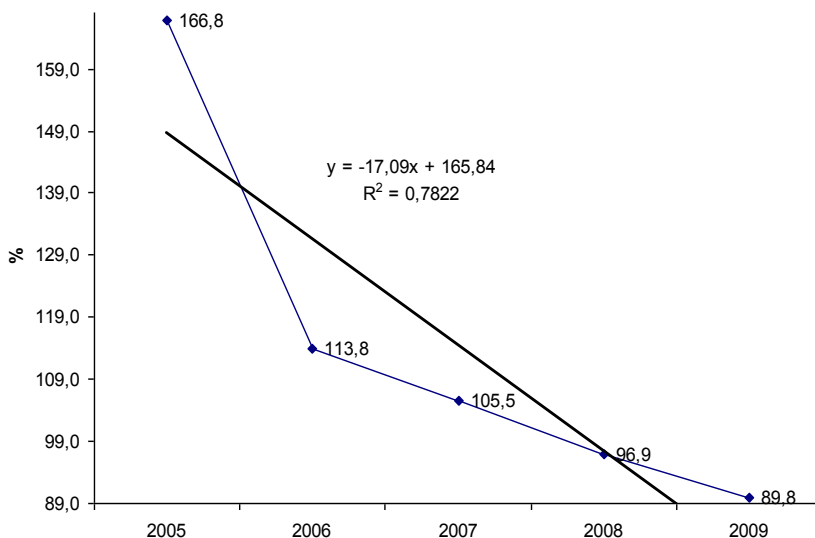


Рис.1. Динаміка та лінійний тренд випередження багатовимірної оцінки втрат людського капіталу щодо його обсягу в Україні за 2005-2009 рр.

Отже, у 2005 році інтегральна оцінка втрат людського капіталу перевищувала оцінку обсягу людського капіталу на 66,8%. За наступний рік цей показник зменшився на 52,9 в.п. і склав 13,8%. Далі темп зміни сповільнився і абсолютне скорочення відносно попереднього року склало відповідно 8,3 в.п. (2007 р.), 8,7 в.п. (2008 р.), 7,0 в.п. (2009 р.). Визначене лінійне рівняння тренду вказує на те, що щорічно у середньому оцінка втрат людського капіталу відставала від його обсягу на 17,09 в.п., а значення коефіцієнта детермінації ($R^2 = 0,7822$) дозволяє прогнозувати подальше зниження відставання.

Література

1. Сайт Державного комітету статистики України: [Електронний ресурс]. – Режим доступу: <http://ukrstat.gov.ua>
2. Гальків Л.І. Статистика: Навч. посібник / С.О. Матковський, Л.І. Гальків, О.С. Гринькевич, О.З. Сорочак – Львів: “Новий світ-2000”, 2009. – 460с.
3. Семів Л.К. Регіональна політика: людський вимір. Монографія. – Львів: ІРД НАН України, 2004. – 392с.

ОБГРУНТУВАННЯ ПЛАНУВАННЯ НА ПІДПРИЄМСТВІ ЯК ФАКТОР ПІДВИЩЕННЯ ЕФЕКТИВНОСТІ ЙОГО ДІЯЛЬНОСТІ

¹І.А. Маринич, ²Л.О. Гелей

¹к.е.н., доцент кафедри економіки підприємства ЛКА

²аспірант кафедри економіки підприємства ЛКА

Формування в Україні економіки ринкового типу зумовлює переосмислення ролі торговельних підприємств, переоцінки існуючих форм господарювання. Виникає необхідність формування економічного механізму торговельних підприємств в ринкових умовах під впливом регулятивних дій держави.

Розглядаючи позиції вчених-економістів щодо визначення суті та змісту економічного механізму, робимо висновок, що економічний механізм торговельного підприємства – це методи

господарювання, інструменти і важелі управління підприємством у визначеній формі власності та обраній системі оподаткування, які забезпечують функціонування підприємства в ринкових умовах та під впливом регулятивних дій держави.

Дотримуючись позиції науковців, які економічний механізм ототожнюють з управлінням економічним процесом на підприємстві, вважаємо, що останній – це свідома діяльність людей, яка полягає у впливі на об'єкти, процеси, і участь в них, здійснювана з метою надати чітко визначеного напрямку господарській діяльності і досягти бажаного результату. Відтак до основних елементів економічного механізму відносимо:

- прогнозування і планування господарської діяльності;
- контроль та оцінку діяльності;
- встановлення матеріальної відповідальності;
- мотиваційний механізм діяльності підприємства[1].

Всі функції органічно поєднані і тільки за їх повного виконання можна досягнути ефективного функціонування торговельного підприємства. Проте одним з найважливіших складників є планування.

Питання планування діяльності торговельних підприємств, зокрема, на сучасному етапі розвитку економічної науки висвітлено в працях як зарубіжних, так і вітчизняних авторів: Алексєєвої М.М., Басовського Л.Е., Раїцького К.А., Мазаракі А.А., Ушакової Н.М., Лігоненко Л.О. та інші.

Трансформаційні процеси в економіці України спричинили зміну умов функціонування підприємств, розширили самостійність суб'єктів господарювання та створили передумови до їхньої повної відповідальності за результати своєї діяльності. Прискорення темпів економічного розвитку, динамічність чинників зовнішнього оточення суб'єктів господарювання та пов'язана з цим зростаюча невизначеність викликали необхідність структурної перебудови системи планування діяльності вітчизняних підприємств. У зв'язку з цим, виникла необхідність ґрунтовнішого розгляду проблеми формування взаємопов'язаної системи планів на сучасних підприємствах. Створення *надійної та гнучкої системи планування*, адаптованої до вітчизняних умов господарювання, спрямованої на вирішення проблем щодо побудови та реалізації стратегії, бюджетування та

взаємоузгодженості різних видів планів, дозволить підвищити ефективність діяльності підприємства, удосконалити процес управління ним та зробить його більш конкурентоспроможним.

Мета дослідження полягає в обґрунтуванні теоретичних положень формування системи планів на підприємстві як однієї з необхідних умов його ефективного розвитку. На сьогодні існує чимало проблем, пов'язаних з організацією системи планування на сучасних українських підприємствах: нереальність планів; використання некоректної інформації та оперативність складання планів; недоступність планів для членів трудового колективу, їхня непрозорість та складність побудови; порядок запровадження планів; роз'єднаність між стратегічним, поточним та оперативним плануванням; відсутність комплексності та системності.

Необхідність формування системи планування на підприємстві визначається його значним впливом на його розвиток, гармонізацію відносин із зовнішнім середовищем, забезпеченням добробуту власників, найманих працівників підприємства та суспільства в цілому.

Організація системи планування направлена на створення найбільш сприятливих умов для удосконалення процесів виробництва та управління підприємством. Структура системи планування будується на підставі наступних параметрів: організаційно-правової форми суб'єкта господарювання; ступеня масовості виробництва; взаємозв'язків між окремими виробничими одиницями, що входять до складу суб'єкта господарювання; структури суб'єкта господарювання; розмірів суб'єкта господарювання та його виробничих одиниць (обсяг виробництва, чисельність працівників, ціна основних виробничих фондів); зовнішніх виробничо-господарських зв'язків; складу та структури кадрів; рівня якості та конкурентоспроможності продукції, інноваційної політики підприємства; ефективності використання ресурсів; методів організації виробництва, праці та управління.

Суб'єкти господарювання обирають різні підходи до вибору організаційної структури системи планування, але прийнято виділяти три групи підприємств з найбільш типовими схемами організаційних структур планування: галузеві, міжгалузеві, багатогалузеві. В залежності від термінів та ступеню деталізації елементів систему планування підприємства можна розподілити на

такі плани: стратегічний, перспективний та поточний плани; програма; проект; бізнес-план.

Метою системи планування є розробка направлення перспективної та поточної діяльності підприємства, а також забезпечення узгодженої роботи всіх його підрозділів для реалізації розроблених цілей. Щоб процес виробництва і реалізації продукції здійснювався безперервно, слід скоординувати всі його стадії у часі та просторі, що може бути досягнуто тільки на основі єдиного плану. Саме через це побудова системи планування охоплює розробку системи збалансованих показників, що характеризують взаємозв'язки процесів виробництва і реалізації продукції на вказаний період як в цілому по підприємству, так і по його структурним підрозділам.

Отже, плани підприємств оперативно коригуються при зміні ринкової кон'юнктури. Вони охоплюють всі сторони діяльності підприємства: від виникнення ідеї по задоволенню потреб споживачів до отримання та використання прибутку. Ступінь обґрунтованості будь-яких планів залежить від достовірності вхідних показників. Потрібно мати *якомога більше* коректних та точних показників для здійснення ефективної планової діяльності, оскільки кожне підприємство є лише невеликою частиною загальної ринкової системи та економіки держави зокрема.

Плануючи свою майбутню діяльність, підприємство має можливість: використовувати майбутні сприятливі умови; вчасно виявляти проблеми, які виникають; бути готовим до раптових змін у зовнішньому середовищі; покращувати координацію дій; покращувати забезпечення необхідною інформацією; раціональніше розподіляти ресурси; формалізувати обов'язки і відповідальність працівників; удосконалювати процедуру контролю та ін. [1].

Основними завданнями побудови системи планів на підприємстві є: виявлення потреб ринку у певному виді продукції, оцінка напрямків і величини зміни цих потреб у майбутньому; встановлення цілей майбутнього економічного розвитку підприємства; визначення тенденцій розвитку підприємства; виявлення альтернативних шляхів розвитку підприємства; визначення потреби й обмежень у ресурсах, необхідних для досягнення поставлених цілей; усунення перешкод на шляху їх досягнення; розробка рекомендацій щодо подальшого розвитку.

Таким чином, побудова системи планування позитивно впливає на роботу підприємства, створює умови для його сталої роботи, надає можливість в більшій мірі використовувати всі ресурси, а таким чином сприяє покращенню його роботи, підвищенню ефективності та конкурентоспроможності.

Література

1. Грещак М.Г. та інш. Внутрішній економічний механізм підприємства: Навч. Посібник/ М.Г. Гребешкова, О.С. Коцюба; за ред.. М.Г. Грещака.- К.: КНЕУ, 2001.-228с.
2. Орлов О. О. Планування діяльності промислового підприємства. Підруч. – К.: Скарби, 2002. – 336с.

DECISION-MAKING IN WASTEWATER TREATMENT

¹A. Dichko, ²Y. Sadova

¹assistant Professor NTUU “KPI”, Kyiv

²graduate NTUU “KPI”, Kyiv

В умовах невизначеності при коливаннях складу і концентрації забруднюючих речовин в стічних водах, стану активного мулу в спорудах з біологічного очищення стоків актуальним постає питання застосування теорії прийняття оптимальних рішень в технології біотрансформації забруднень стічних вод.

Human role in modern management systems is determinative. Decision making person creates goals of system and alternative ways of its development. He determines real structure of system and forms its behavior. The most difficult and responsible stage of human's activity in management systems and the major factor of any direction is decision making. In process of looking for the best decision it is necessary to use maximum relevant information (data and knowledge bases). But decision making person often doesn't have much time, enough information and knowledge about object and situation in which it functions. The most important moments in decision making is goals determination, forming decision making problems and choice of adequate alternative (variant of decision which satisfies problem's limitation and may achieve the goal). The choice is made on base of advantages of decision making person.

The task of decision making may be formulated in next way: there is a lot of decisions; realization of each alternative results in certain consequences; analysis and estimation of results by effectiveness (criteria) characterizes alternatives. Having considered advantages of decision making person, it is necessary to build model of choice the best alternative.

The logical choice of criteria at decision-making means selection of such base variable value in which function of belonging achieves maximal value. Thus there is a problem of determination of belonging of those or other objects to the noted unclear sets.

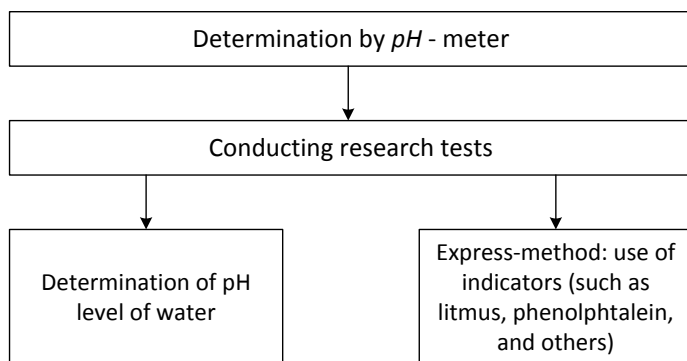


Figure 1. Example 1

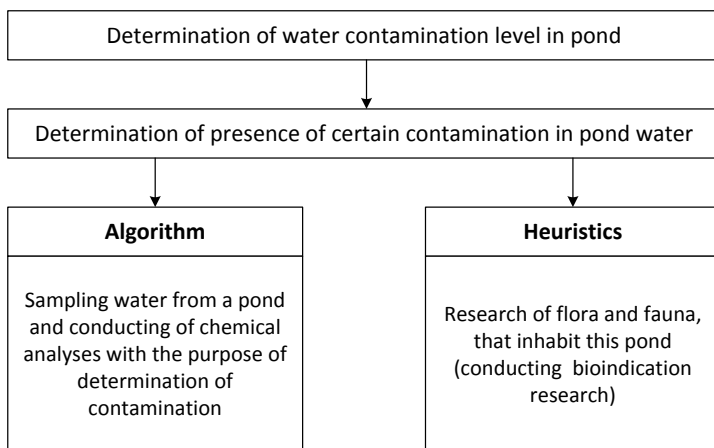


Figure 2. Example 2

For example:

1. Certain value of BOD is normal for water dumping into pond, but at the same time it does not allow to use such water for drinking.
2. Radiation level which is normal for the district of location AES is exceeded for other locality.
3. Air temperature in summer in Ukraine for Ukrainians is high, and for habitants of the African countries is temperate.
4. Full garbage bucket in apartment – is too much garbage. The same amount of garbage on dump – is little.

However, a base variable can achieve the maximal value of belonging in a few points of the top, this criterion does not guarantee the only decision. To show the problem of transformation of unclear set in scalar, it is important to consider a few graphs of functions of belonging.

A function of belonging – is the dependence of degree of element belonging to unclear set from a base variable and from used rules.

Heuristic and traditional algorithmic methods of decision-making are used at presence of unclear, incomplete information. *Heuristic* methods are based on empiric rules or simplifications that limit search of decisions effectively.

Comparison of heuristic and traditional algorithmic methods are in the followings examples 1, 2 (Figures 1&2).

ФУНКЦІОНАЛЬНІ АСПЕКТИ НАЦІОНАЛЬНОЇ ПОЛІТИКИ ЩОДО ВИКОРИСТАННЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

С. Голубка

к.е.н., доцент університету банківської справи НБУ

При формуванні сприятливого інформаційно-технологічного іміджу національної економіки на перспективу та її розвитку потрібна, насамперед, державна підтримка розвитку інформаційних технологій. Адже серед країн – розробників ІТ-рішень для компаній та організацій, Україна у цьому списку посідає лише 75 місце з 122. Хоч

у галузі розробки програмного забезпечення, безумовно, вона знаходиться у першій десятці.

Серед основних напрямів національної політики щодо розвитку інформаційного суспільного простору та інформаційних технологій в Україні слід виділити:

1. інформаційне забезпечення діяльності вищих органів влади;
2. інформаційна інфраструктура прозорого державного діловодства, спеціалізовані відомчі інформаційні системи, в тому числі, податкової, міграційної, системи довідкової та юридичної інформації щодо внутрішньої та зовнішньої торгівлі, тощо.

Основною стратегічною метою визначення конкретних дій з реалізації завдань щодо розвитку інформаційного суспільного простору в Україні можуть бути наступні:

- забезпечення комп'ютерної та інформаційної грамотності населення;
- розвиток національної інформаційної інфраструктури та її інтеграція зі світовою просторово-інформаційною інфраструктурою;
- збереження культурної спадщини України шляхом її електронного документування;
- використання ІКТ для вдосконалення державного управління тощо.

В Україні є вдало реалізовані проекти державного масштабу. Зокрема, побудована Національна система конфіденційного зв'язку, яка об'єднує центральні органи влади та гарантує безпеку передачі інформації. У центральних органах виконавчої влади України також успішно впроваджується загальнодержавна інтегрована система документообігу на основі використання спеціальної інформаційно-телекомунікаційної системи органів державної влади. Реалізуються окремі елементи системи "Електронний Уряд".

На окрему увагу заслуговує проект Ukrainian Research and Academic Network (URAN). Результатом співпраці НТУУ "Київський Політехнічний інститут" з провідними ІТ-компаніями – CISCO, Intel, Microsoft та інші – стало створення національної науково-освітньої телекомунікаційної мережі.

У відповідному законодавстві слід надати субвенції і малим та

середнім компаніям, адже на сьогоднішній день тільки великі вітчизняні компанії мають відповідні фінансові можливості і можуть дозволити собі проводити самостійну маркетингову роботу, а також працювати з західними клієнтами від свого імені. Рішення решти українських ІТ-компаній продається західними консультантами під своїми торговими марками. У більшості випадків розроблювач навіть не має права згадувати про ці рішення у своєму інформаційному листі. Внаслідок цього 80–90% українських розробок не ідентифікуються західними замовниками, як “Зроблено в Україні”.

Варто також згадати про створення потужних Центрів Обробки Даних і надання відповідних сервісів для віддалених користувачів, зокрема – збереження великих обсягів даних та виконання ресурсоємних прикладних програм. Цей напрямок наразі активно розвивається у світі, тому в Україні теж можна запровадити надання такого виду сервісів не тільки для вітчизняних, але й для закордонних користувачів. Це, безумовно, ефективно вплине на розвиток національної економіки та імідж України у світі.

Література

1. Закон України “Про національну програму інформатизації” від 04.02.98, №74/98 – ВР. – К., 1998.
2. Інформаційні технології та тенденції розвитку міжнародної інформації // Вісник книжкової палати – 2000 р. – №6.
3. [http://www. // iu. org. ua](http://www.iu.org.ua)
4. [http://www. // ua xxi. com](http://www.ua.xxi.com)

СЕКЦІЯ 3

Моделювання інформаційних, економічних і соціальних процесів, інформаційні технології в економіці

ДО ПОБУДОВИ ОПТИМАЛЬНИХ СТРАТЕГІЙ ПОШУКУ ЗАПИСІВ У ПОСЛІДОВНИХ ФАЙЛАХ БАЗ ДАНИХ ЗА ВИКОРИСТАННЯ БЛОЧНОГО ПОШУКУ

¹А.В. Мельничин, ²Г.Г. Цегелик

¹к.т.н., доцент кафедри ТОП ЛНУ ім. І.Франка

²д.ф.-м.н., професор, професор кафедри МКД ЛФСУ

Для різних законів розподілу ймовірностей звертання до записів досліджено ефективність пошуку записів у послідовних файлах за використання методу блочного пошуку в блоці записів, який попередньо локалізований шляхом читання і перегляду останніх записів кожного блоку.

У [1,4] для різних законів розподілу ймовірностей звертання до записів [3] (рівномірного, “бінарного”, Зіпфа, узагальненого, частковим випадком якого є розподіл, що наближено задовольняє правило “80-20”) досліджена ефективність таких двох варіантів пошуку записів у послідовних файлах баз даних з використанням методу послідовного перегляду, як послідовне читання блоків записів в основну пам’ять і їх послідовний перегляд та послідовний перегляд блоку записів, який попередньо локалізований шляхом читання блоків записів в основну пам’ять і перегляду їх останніх записів. У [2] досліджена ефективність використання методу блочного пошуку в блоці записів, який попередньо локалізований шляхом читання блоків записів в основну пам’ять і перегляду їх останніх записів.

Оскільки методу блочного пошуку є одним із найуживаніших методів, то постає задача дослідження ефективності використання

цього методу для пошуку записів у блоці, який попередньо локалізований шляхом читання і перегляду останніх записів кожного блоку.

Припустимо, що файл, який містить N записів, розбитий на n блоків по ms записів у кожному і процес пошуку запису відбувається так. Спочатку локалізується блок, який містить шуканий запис, шляхом послідовного читання і перегляду останніх записів кожного блоку. Після цього пошук потрібного запису продовжується в локалізованому блоці за допомогою методу блочного пошуку. При цьому локалізований блок записів умовно розбивається на s підблоків по m записів в кожному. Представимо математичне сподівання загального часу, необхідного для пошуку запису у файлі, у вигляді суми математичного сподівання часу, необхідного для локалізації блока записів, математичного сподівання часу, необхідного для локалізації підблока записів, і математичного сподівання часу, необхідного для пошуку запису в локалізованому підблоці. Тоді математичне сподівання загального часу, необхідного для пошуку запису у файлі, виразиться формулою:

$$E = a + \sum_{i=1}^n it_1 \sum_{k=1}^s \sum_{j=1}^m p_{(i-1)ms+(k-1)m+j} + \\ + \sum_{i=1}^n \sum_{k=1}^s kt \sum_{j=1}^m p_{(i-1)ms+(k-1)m+j} + \sum_{i=1}^n \sum_{k=1}^s \sum_{j=1}^m jtp_{(i-1)ms+(k-1)m+j},$$

або

$$E = \sum_{i=1}^n \sum_{k=1}^s \sum_{j=1}^m (a + it_1 + (k+j)t) p_{(i-1)ms+(k-1)m+j},$$

де $a = b + dms$ – час читання блоку в основну пам'ять, $t_1 = b + d + t$ – час читання запису в основну пам'ять і його перегляд, t – час перегляду запису в основній пам'яті.

Знайдемо явний вираз для E у випадку різних законів розподілу ймовірностей звертання до записів. Для цього введемо позначення

$$E_0 = \sum_{i=1}^n \sum_{k=1}^s \sum_{j=1}^m jp_{(i-1)ms+(k-1)m+j}, \quad E_1 = \sum_{i=1}^n \sum_{k=1}^s \sum_{j=1}^m kp_{(i-1)ms+(k-1)m+j},$$

$$E_2 = \sum_{i=1}^n \sum_{k=1}^s \sum_{j=1}^m ip_{(i-1)ms+(k-1)m+j}.$$

Якщо розподіл імовірностей звертання до записів є рівномірним, то

$$E_0 = \frac{1}{2}(m+1), \quad E_1 = \frac{1}{2}(s+1), \quad E_2 = \frac{1}{2}(n+1).$$

Тоді

$$E = a + \frac{1}{2}(n+1)t_1 + \frac{1}{2}(s+m+2)t,$$

або

$$E = b + dms + \frac{1}{2} \left(\left(\frac{N}{ms} + 1 \right) (b+d) + \left(\frac{N}{ms} + s + m + 3 \right) t \right).$$

Нехай ймовірності звертання до записів задовольняють “бінарний” розподіл. Оскільки [4]

$$E_0 = \frac{m}{2^N} + \left(2 - \frac{m+2}{2^m} \right) \frac{2^m}{2^m - 1} (1 - 2^{-N}),$$

$$E_1 = \frac{s}{2^N} + \left(\frac{2^m}{2^m - 1} - \frac{s}{2^{ms} - 1} \right) (1 - 2^{-N}), \quad E_2 = \frac{2^{ms}}{2^{ms} - 1} (1 - 2^{-N}),$$

то, нехтуючи величиною 2^{-N} , з достатньо високою точністю можемо прийняти

$$E = a + \frac{2^{ms}}{2^{ms} - 1} t_1 + \left(\frac{2^m}{2^m - 1} \left(2 - \frac{m+2}{2^m} \right) + \frac{2^m}{2^m - 1} - \frac{s}{2^{ms} - 1} \right) t,$$

або

$$E = b + dms + \frac{2^{ms}}{2^{ms} - 1} (b+d) + \left(4 + \frac{s-1}{2^{ms} - 1} - \frac{m-1}{2^m - 1} \right) t.$$

Припустимо, що ймовірності звертання до записів розподілені за законом Зіпфа. Оскільки [4]

$$E_0 = \frac{1}{H_N} (m \cdot S_m(sn) - (H_N - 1)N),$$

$$E_1 = \frac{1}{H_N} (H_N + s \cdot S_{ms}(n) - S_m(sn)), \quad E_2 = \frac{1}{H_N} ((n+1)H_N - S_{ms}(n)),$$

то

$$E = a + \frac{1}{H_N} ((n+1)H_N - S_{ms}(n))t_1 + \frac{1}{H_N} (H_N + s \cdot S_{ms}(n) - S_m(sn) + m \cdot S_m(sn) - (H_N - 1)N)t),$$

де

$$S_{ms}(n) = \sum_{k=1}^n H_{kms}, \quad S_m(sn) = \sum_{k=1}^{sn} H_{km}$$

Використовуючи апроксимацію $S_{ms}(n)$ і $S_m(sn)$ відповідно виразами [3]

$$\bar{S}_{ms}(n) = n(H_N - 1) + \frac{1}{2} \ln n + C_1, \quad \bar{S}_m(sn) = sn(H_N - 1) + \frac{1}{2} \ln sn + C_1,$$

де $C_1 = 0,5 \ln 2\pi$, з достатньо високою точністю можемо прийняти

$$E = a + \frac{1}{H_N} \left(\left(H_N + n - \frac{1}{2} \ln n - C_1 \right) t_1 + \left(H_N + \frac{1}{2} ((s+m-1)(\ln n + 2C_1) + (m-1) \ln s) \right) t \right),$$

або

$$E = b + dms + \frac{1}{H_N} \left(\left(H_N + \frac{N}{ms} - \frac{1}{2} \ln \frac{N}{ms} - C_1 \right) (b+d) + \left(2H_N + \frac{N}{ms} + \frac{1}{2} \left((s+m-2) \left(\ln \frac{N}{ms} + 2C_1 \right) + (m-1) \ln s \right) \right) t \right).$$

Якщо ймовірності звертання до записів задовольняють узагальнений закон розподілу, то [4]

$$E_0 = \frac{1}{H_N^{(c)}} (m \cdot S_m^{(c)}(sn) + H_N^{(c-1)} - N H_N^{(c)}),$$

$$E_1 = \frac{1}{H_N^{(c)}} (H_N^{(c)} + s \cdot S_{ms}^{(c)}(n) - S_m^{(c)}(sn)), \quad E_2 = \frac{1}{H_N^{(c)}} ((n+1)H_N^{(c)} - S_{ms}^{(c)}(n)).$$

Тоді

$$E = a + \frac{1}{H_N^{(c)}} \left(((n+1)H_N^{(c)} - S_{ms}^{(c)}(n))t_1 + (H_N^{(c-1)} + (1-N)H_N^{(c)} + s \cdot S_{ms}^{(c)}(n) + (m-1)S_m^{(c)}(sn))t \right),$$

де

$$S_{ms}^{(c)}(n) = \sum_{k=1}^n H_{kms}^{(c)}, \quad S_m^{(c)}(sn) = \sum_{k=1}^{sn} H_{km}^{(c)}.$$

Використовуючи апроксимації $S_{ms}^{(c)}(n)$ і $S_m^{(c)}(sn)$ відповідно виразами [3]

$$\bar{S}_{ms}^{(c)}(n) = nH_N^{(c)} + \frac{N^{1-c}}{1-c} \left(\frac{c-1}{2-c} n + \frac{\alpha^{(c)}(n)}{n^{1-c}} \right),$$

$$\bar{S}_m^{(c)}(sn) = snH_N^{(c)} + \frac{N^{1-c}}{1-c} \left(\frac{c-1}{2-c} sn + \frac{\alpha^{(c)}(sn)}{(sn)^{1-c}} \right),$$

де

$$\alpha^{(c)}(n) = H_n^{(c-1)} - \frac{1}{2-c} n^{2-c}, \quad \alpha^{(c)}(sn) = H_{sn}^{(c-1)} - \frac{1}{2-c} (sn)^{2-c},$$

з достатньо високою точністю можемо прийняти

$$E = a + \frac{1}{H_N^{(c)}} \left(\left(H_N^{(c)} - \frac{N^{1-c}}{1-c} \left(\frac{c-1}{2-c} n + \frac{\alpha^{(c)}(n)}{n^{1-c}} \right) \right) t_1 + \right. \\ \left. + \left(H_N^{(c-1)} + H_N^{(c)} + \frac{N^{1-c}}{1-c} \left(\frac{c-1}{2-c} N + \frac{s \cdot \alpha^{(c)}(n)}{n^{1-c}} + \frac{(m-1)\alpha^{(c)}(sn)}{(sn)^{1-c}} \right) \right) t \right),$$

або

$$E = b + d \frac{N}{n} + \frac{1}{H_N^{(c)}} \left(\left(H_N^{(c)} - \frac{N^{1-c}}{1-c} \left(\frac{c-1}{2-c} n + \frac{\alpha^{(c)}(n)}{n^{1-c}} \right) \right) (b+d) + \right. \\ \left. + \left(H_N^{(c-1)} + 2H_N^{(c)} + \frac{N^{1-c}}{1-c} \left(\frac{c-1}{2-c} (N-n) + \frac{(s-1)\alpha^{(c)}(n)}{n^{1-c}} + \left(\frac{N}{ns} - 1 \right) \frac{\alpha^{(c)}(ns)}{(ns)^{1-c}} \right) \right) t \right).$$

У табл.1 (з точністю до 0,1) наведені оптимальні значення величини E/d для різних законів розподілу ймовірностей звертання до записів для деяких b/d , $t/d = 0,1$ і деяких N .

У табл.1 значенню $c=0$ відповідає рівномірний розподіл, а значенню $c=1$ – закон Зіпфа.

На рис.1 показана залежність оптимального значення величини E/d від зміни закону розподілу ймовірностей звертання до записів для деяких b/d , $t/d = 0.1$ і $N = 10^6$.

Таблиця 1

Оптимальні значення величини E/d

N	b/d	Закони розподілу						
		c=0	c=0.2	c=0.4	c=0.6	c=0.8	c=1	"Бінарний"
104	10	488,4	461,5	425,8	377,1	311,2	231,3	25,9
	100	1575,3	1495,0	1389,0	1245,1	1051,6	817,9	209,0
	1000	5980,0	5745,4	5439,4	5028,7	4479,7	3813,9	2012,3
105	10	1508,4	1423,1	1309,0	1149,0	919,6	625,0	25,9
	100	4652,1	4395,3	4052,1	3571,5	2884,9	2004,8	209,0
	1000	15660,0	14859,3	13795,9	12317,3	10218,8	7536,3	2012,3
106	10	4732,2	4462,6	4101,0	3587,2	2819,6	1776,3	25,9
	100	14378,8	13565,5	12474,9	10926,5	8615,6	5478,0	209,0
	1000	46261,6	43706,3	40284,1	35436,9	28225,9	18456,3	2012,3

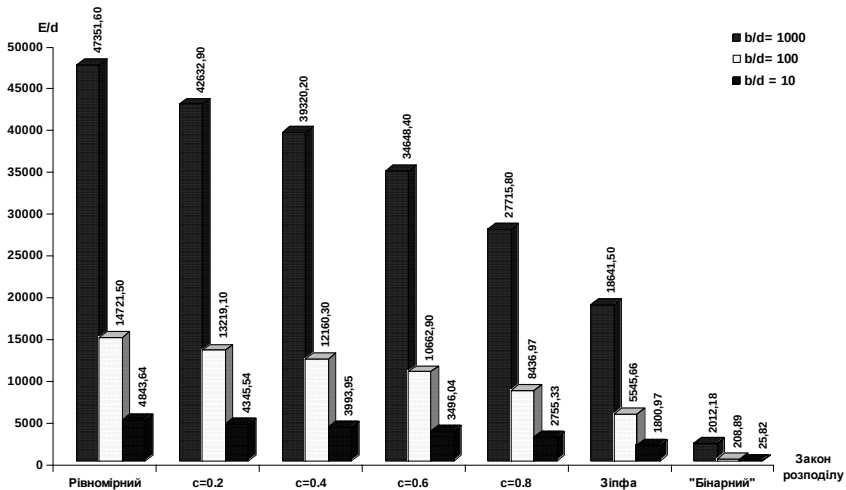


Рис.1. Оптимальне значення величини E/d для різних законів розподілу ймовірностей звертання до записів, деяких b/d , $t/d = 0.1$ і $N = 10^6$

Із рис.1 видно, що зі зміною закону розподілу ймовірностей звертання до записів суттєво змінюється оптимальне значення

величини E/d .

Література

1. Кудеравець Х. С. Побудова та аналіз оптимальних стратегій пошуку записів в послідовних файлах для різних законів розподілу ймовірностей звертання до записів / Кудеравець Х. С., Цегелик Г. Г. – Львів: 1997. – 70с. (Препринт / Львівський державний університет ім. І. Франка; №1 – 97).
2. Мельничин А. В. Оптимальні стратегії пошуку записів в послідовних файлах баз даних при використанні методу блочного пошуку в локалізованому блоці записів / А. В. Мельничин, Г. Г. Цегелик // Наук. вісник Ужгород. ун-ту. Сер. матем. і інформ. – 2009. – Вип. 18. – С.92-98.
3. Цегелик Г. Г. Организация и поиск информации в базах данных / Г. Г. Цегелик. – Львов: Вища шк., 1987. – 176 с.
4. Цегелик Г. Г. Системы распределенных баз данных. / Г. Г. Цегелик. – Львов: Свит, 1990. – 168 с.

AN ALGORITHM FOR SIMULATION OF FOREST MANAGEMENT RESPONSE TO CARBON PRICE INCENTIVE

M. Gusti

*Associate Professor at the International Information Department of the Lviv National Polytechnic University,
Research Scholar at the International Institute for Applied Systems Analysis,
Laxenburg, Austria*

Представлено алгоритм прийняття рішень щодо лісокористування при наявності заохочень накопичувати вуглець у біомасі лісів у вигляді податків та доплат. Продемонстровано адекватність алгоритму на прикладі країн Європейського Союзу.

In current work we extend the algorithm of forest management decisions in the Global Forest Model (G4M¹), which was presented at the

¹G4M has been developed in the International Institute for Applied Systems Analysis – IIASA (Austria).

CITEM'2009 [1], with a module that adjusts forest management as a response to carbon price incentive. G4M is a geographically explicit (operates on a regular grid of 0.5X0.5 deg) agent-based model that simulates decisions made by virtual land/forest owners on deforestation and afforestation taking into account profitability of forestry and agriculture [2], and forest management decisions aimed at satisfying wood demand at country level [1]. The virtual land owner operates within one grid cell.

The carbon price incentive means that the forest owner is paid for the carbon stored in forest living biomass above a baseline or pays a tax, if the carbon in forest living biomass is below the baseline. The baseline is estimated assuming forest management without the carbon price incentive – business-as-usual (BAU) case. The forest owner faces a dilemma – get larger revenue for harvesting and selling more wood but pay the carbon tax, or get smaller revenue for harvesting and selling less wood but get paid for the biomass accumulated over the baseline.

Forest is a **complex dynamic system**. It is modelled with a number of age cohorts starting from zero (just planted trees) till the oldest tree². The forest management in the model is represented (as in [1]) with rotation length (determining rate of forest harvesting and planting, and age structure) and thinning (keeping a certain stocking degree of forest). Current forest management determines characteristics of the forest for many years. Forest owner chooses forest management to get revenue not only today but for many years, therefore we use net present value (NPV) as an indicator of long-term economic activity. For each cell every year we estimate NPV for the next 50 years and biomass during the model run with zero carbon price (BAU). If the carbon price is zero the forest management algorithm, which provides wood harvest satisfying wood demand at the state level, is applied [1].

There is a hierarchy of interests: the state must provide wood amount matching the demand and create conditions for carbon sequestration (both on the state level) by adjusting forest management; forest owners every year try to get NPV not smaller than the BAU NPV (NPV_{baU}). Wood production satisfying wood demand at the state level is of the highest priority.

We use a two step procedure. STEP1. Every year, starting from

² Forest in each grid cell is modeled using forest management module developed by G.Kindermann from IIASA.

2011, forest management in each cell is disturbed towards increasing the rotation length. For the forest used for wood production, where NPV estimated for the maximum biomass rotation length (NPV_{wc}) (always with thinning) is greater than the BAU NPV (NPV_{bau} , $NPV_{bau} > 0$), current rotation length is increased proportionally to the $(NPV_{wc} - NPV_{bau}) / NPV_{bau}$. If the NPV condition is not satisfied current rotation length is increased by 5 years. In all cases the maximum rotation length is not allowed to be higher than the rotation length maximising biomass. NPV for the new rotation length is estimated (NPV_c) and kept in memory. NPV in all cases is estimated for the next 50 years.

STEP 2. Since production of wood balancing wood demand has higher priority than the carbon accumulation, after Step 1 the forest management of forests within each country is adjusted to harvest as much as the country wood demand.

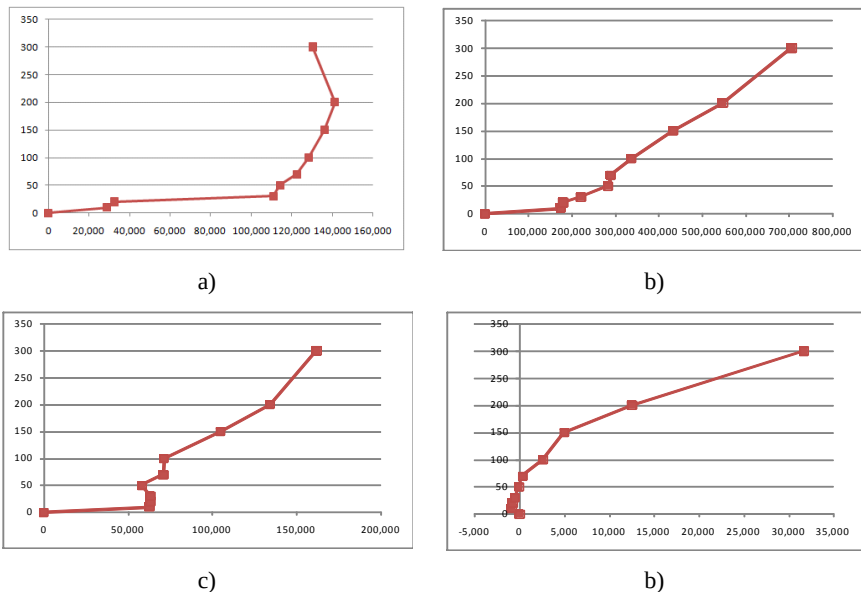


Figure 1. Cost curves for the forest management response to the carbon price incentive for Finland (a), France (b), Germany (c) and Austria (d) in the year 2050. Y-axis is for the carbon price, Euro/tCO₂, and X-axis is for the carbon accumulated above baseline for the period 2011-2050, th.t CO₂.

When adjusting the forest management towards wood demand it is required the new NPV multiplied by an adjustment hurdle coefficient to

be greater or equal NPV_c estimated on Step 1. The adjustment hurdle varies from 1 to 2500 and to -1. The forest management adjustment for the cells within each country starts with the hurdle=1. If the total harvest does not match wood demand, the hurdle is increased by 0.3 and the forest management adjustment is repeated for the forests within the country again. The last hurdle tried is minus one, allowing forest management leading to negative NPV in order to satisfy wood demand. We assume that the forest owners getting NPV smaller than NPV_{bau} are compensated by the government that is not reflected in the algorithm explicitly.

Cost curves (amount of additionally accumulated carbon depending on the carbon price) for Finland (a), France (b), Germany (c) and Austria (d) are shown in Figure 1.

The algorithm shifts wood production to other places in the countries considered allowing harvest of similar amount of wood and the same time accumulate carbon in living biomass.

References

1. Gusti M. An Algorithm for Decision Making on Forest Management in the Global Forest Model (G4M)/ Proceedings of the conference CITEM'2009, Lviv, 2009, p. 110-114.
2. Kindermann G., Obersteiner M., Rametsteiner E. and McCallum I: Predicting the Deforestation-Trend under Different Carbon-Prices// Carbon Balance and Management.- 2006.- P.1-15; doi:10.1186/1750-0680-1-15

МОНІТОРИНГ АКТИВНОСТІ КОНКУРЕНТІВ ТА ЗАХИСТ ВІД АГРЕСИВНИХ ДІЙ В ОНЛАЙН- СПІЛЬНОТАХ

О.П. Пелещишин

менеджер служби підтримки ERP-систем, ВАТ ПБК “Славутич”

Розглянуто основні аспекти моніторингу активності конкурентів у віртуальних спільнотах. Виділено особливості компонент системи моніторингу та інструменти їхньої реалізації. Запропоновано методи захисту та протидії агресивним діям конкурентів в онлайн-спільнотах.

Маркетингова діяльність підприємства у мережі Інтернет та глобальній системі World Wide Web на сьогодні є обов'язковою складовою загальної системи маркетингу.

Одними із базових стратегій використання онлайн-спільнот у процесі маркетингу є моніторингова та аналітична стратегії. Моніторингова стратегія забезпечує підприємству збір відгуків і критичних зауважень користувачів про підприємство, продукцію, ефективність роботи з клієнтами. Аналітична стратегія є розширенням моніторингової стратегії за рахунок поглибленого обліку та аналізу маркетингової інформації, що наявна у віртуальних спільнотах. Її реалізація забезпечує зокрема:

- збір та аналіз інформаційних матеріалів конкурентів;
- виявлення додаткових факторів зацікавлення у продукції;
- виявлення потенційних загроз для підприємства у сфері маркетингу.

Задача **моніторингу активності** конкурентів в онлайн-спільнотах має два основні аспекти:

- відстежування загальної активності конкурентів, виявлення нових напрямів діяльності, прогнозування конкурентної ситуації;
- протидія агресивним діям конкурентів у онлайн-спільнотах, зокрема з компрометації підприємства чи його продукції, реагування на такі дії.

Особливостями реалізації першого напрямку з точки зору інформаційних технологій є:

- значне збільшення множини маркетингових термінів за рахунок термінів, що характеризують діяльність конкурентів;
- значне збільшення бази даних дискусій та певне збільшення бази даних спільнот;
- відсутність достовірної інформації про маркетинговий характер дописів (хто і з якою метою їх створив).
- ускладнення комп'ютерно-лінгвістичних алгоритмів опрацювання текстів;
- збільшення частки ручної праці операторів з опрацювання

матеріалів із онлайн-спільнот.

Протидія агресивним діям конкурентів у онлайн-спільнотах є вужчою задачею у порівнянні з відстежуванням загальної активності конкурентів. Вона має ряд принципових особливостей:

- моніторинг повинен вестися в режимі реального часу, повинна функціонувати система оперативного оповіщення про агресивні дії щодо підприємства;
- окрім складової моніторингу та аналізу необхідно також реалізовувати складову оперативного реагування на агресивні дії;
- повинен здійснюватися моніторинг результатів захисту проти агресії.

Засоби моніторингу в реальному часі повинні включати наступні компоненти:

- підсистему оперативного отримання повідомлень про дописи в існуючих темах;
- підсистему моніторингу динаміки дискусій;
- підсистему лінгвістичного аналізу повідомлень;
- підсистема моніторингу журналів доступу до Веб-сервера.

Підсистема оперативного отримання повідомлень про дописи в існуючих темах реалізується за допомогою збору інформації, що поступає від сайтів онлайн-спільнот. Збір інформації здійснюється через отримання повідомлень електронною поштою або з RSS-каналів окремих дискусій. Після отримання інформації здійснюється автоматизований розбір повідомлення та занесення його в базу даних.

Підсистема моніторингу динаміки дискусій реалізує збір узагальнених даних щодо динаміки зміни кількості та обсягів дискусій. Основними інструментами моніторингу є:

- систематичний пошук кількості тематичних дискусій за допомогою пошукових систем та розширених запитів з використанням контрольних слів негативного характеру (“скарга на...”, “проблеми з...” тощо);
- систематичний контроль за динамікою росту існуючих

дискусій за допомогою автоматизованих засобів отримання повідомлення про нові повідомлення в дискусіях та за допомогою ручного або автоматичного аналізу сторінок дискусій.

Підсистема лінгвістичного аналізу повідомлень повинна здійснювати доволі простий автоматизований лінгвістичний аналіз на предмет виявлення критичних та образливих слів в адресу підприємства та продукції, вести облік динаміки появи таких висловів.

Первинними ознаками інформаційної атаки при аналізі журналів доступу до Веб-сервера є наявність за адресами, з яких здійснено перехід:

- переважно критичних дописів;
- дописів з однаковим або схожим текстом у різних дискусіях та спільнотах;
- дописів з закликами щодо поширення подібної інформації;
- дописів, що випадають з контексту дискусії, мають ознаки спаму або автоматизованої публікації.

Основними методами захисту від агресивних дій конкурентів та протидії є:

- інформування онлайн-спільнот про замовний характер негативних дописів;
- апеляція до адміністрації онлайн-спільнот щодо порушення етики та правил спільнот у негативних дописах;
- зміна акцентів у дискусіях з негативних на позитивні.

Моніторинг результатів завершує процес протидії інформаційній атаці та включає:

- оцінку реальних наслідків атаки, ефективність протидії, загрози повторної атаки
- ідентифікацію основних “больових точок” для критики;
- ідентифікацію основних учасників онлайн-спільнот, що брали участь у атаці;
- остаточну оцінку створених у процесі атаки дискусій в онлайн-спільнотах.

ДО ОПТИМІЗАЦІЇ РОЗПОДІЛУ ФІНАНСОВИХ РЕСУРСІВ ВУЗУ З ПРИВАТНОЮ ФОРМОЮ ВЛАСНОСТІ СВОЇМ ФІЛІЯМ

¹О.Я. Прядко, ²Г.Г. Цегелик, ³Л.М. Шевчук

¹асистент кафедри ДАІС ЛНУ ім. І.Франка, здобувач

²д.ф.-м.н., професор, професор кафедри МКД ЛФСУ

³старший викладач кафедри СГД ЛФСУ

Розглянуто використання методу динамічного програмування для розв'язування задачі про оптимальний розподіл коштів вузу з приватною формою власності своїм філіям.

В [1] для оптимального розподілу фінансових ресурсів вищого навчального закладу з приватною формою власності своїм філіям запропоновано використовувати оптимізаційну модель. В роботі цю ж задачу пропонуємо розв'язувати, використовуючи метод динамічного програмування [2].

Нехай для розподілу своїм філіям вуз може виділити m одиниць коштів, обсяг кожної з яких становить S умовних одиниць. Відомо, що кошти бажано розподілити серед n філій $P_1, P_2, \dots, P_n$. При цьому одній філії вуз може виділити не більше k грошових одиниць коштів, де $k = 1, 2, \dots, m$. В залежності від розміру коштів, виділених для тої чи іншої філії, вуз отримує певний дохід. Задача полягає в такому розподілі коштів серед філій, за якого сумарна величина доходу вузу була б найбільшою.

Нехай $g_i(x_j)$, $i = 1, 2, \dots, n$, - дохід від розподілу $x_j = j$, $j = 0, 1, \dots, m$, грошових одиниць коштів філії P_i . Позначимо через $F_i(x_j)$, $i = 1, 2, \dots, n$, дохід від розподілу $x_j = j$, $j = 0, 1, \dots, m$, одиниць коштів серед перших i філій $P_1, P_2, \dots, P_i$, а через $F_i^*(x_j)$, $i = 1, 2, \dots, n$, - дохід від оптимального розподілу $x_j = j$, $j = 0, 1, \dots, m$, одиниць коштів серед перших i філій $P_1, P_2, \dots, P_i$ (максимальний дохід).

Процес розв'язування задачі розіб'ємо на n кроків. На

$F_n^*(m)$, де

$$F_n(m) = \begin{cases} g_n(0) + F_{n-1}^*(m), \\ g_n(1) + F_{n-1}^*(m-1), \\ \\ g_n(m) + F_{n-1}^*(0) \end{cases}$$

i

$$F_n^*(m) = \max_{0 \leq k \leq m} \{g_n(k) + F_{n-1}^*(x_j - k)\}.$$

Оптимальний план розподілу коштів серед філій $P_1, P_2, \dots, P_n$ визначаємо так.

Нехай $F_n^*(m)$ приймає найбільше значення для $k = l_1$. Тоді l_1 одиниць коштів треба виділити філії P_n . Далі необхідно розподілити $m - l_1$ одиниць коштів серед філій $P_1, P_2, \dots, P_{n-1}$. Припустимо, що $F_{n-1}^*(m - l_1)$ приймає найбільше значення для $k = l_2$. Це означає, що l_2 одиниць коштів треба виділити філії P_{n-1} . Якщо $F_{n-2}^*(m - (l_1 + l_2))$ приймає найбільше значення для $k = l_3$, то l_3 одиниць коштів треба виділити для філії P_{n-2} . І т.д. Нехай $F_2^*(m - (l_1 + l_2 + \dots + l_{n-2}))$ приймає найбільше значення для $k = l_{n-1}$. Тоді l_{n-1} одиниць коштів треба виділити філії P_2 . Нарешті, $l_n = (m - (l_1 + l_2 + \dots + l_{n-1}))$ одиниць коштів треба виділити філії P_1 .

Максимальний дохід від розподілу коштів серед філій становить $F_n^*(m)$ одиниць.

Література

1. Вітер М. Б. Оптимізація розподілу фінансових ресурсів між філіями вищого навчального закладу з приватною формою власності / М.Б. Вітер, Г.Г. Цегелик, Л.М. Шевчук // Наук. журнал. “Економіка і управління”. – Київ: Вид. Європ. ун-ту,

2009. – №2. – С. 94-99.

2. Кузнецов Ю.Н. Математическое программирование/ Ю.Н. Кузнецов, В.И. Кузубов, А. Б. Волощенко. – М.: Висш. шк., 1980. – 300с.

МЕТОДИКИ МОДЕЛЮВАННЯ ЛОГІСТИЧНОЇ WEB-СИСТЕМИ

Н.І. Бойко

к.е.н., старший викладач Львівської комерційної академії

У роботі розглянуто сутність та значення імітаційного моделювання як процесу, що включає конструювання моделі та аналітичне її застосування для вивчення певних проблем. Запропоновано методології моделювання імітаційної моделі логістичної web-системи та способи її програмування.

Останнім часом все частіше ставиться завдання щодо інтеграції систем діловодства зі вже існуючими в організаціях автоматизованими інформаційними системами [1]. Для прикладу приведемо програму, яка побудована повністю на Web-технологіях, що забезпечують роботу зі системою з будь-якого комп'ютера через звичайний браузер і можливість інтеграції даних з різних корпоративних додатків. Інтернет-програма підтримує можливість організації роботи з документами засобами електронної пошти.

Програма повністю побудована на Інтернет-технологіях. Розвинені вбудовані засоби захисту даних забезпечують безпеку її роботи.

Особливістю системи є розвинена інтеграція з документами MS Office – MS Word і MS Excel. Робота програми будується на використанні шаблонів документів, в які автоматично вставляються необхідні відомості з баз даних. Так, наприклад, реєстраційна картка документа може бути видимою в стандартному для програми вікні Web-браузера, так і бути одержаною в звичному для користувача вигляді у форматі файлу MS Word або MS Excel [2].

Цінну інформацію щодо теоретичних основ проектування імітаційних моделей та використання систем масового обслуговування надає Інтернет-ресурс [4]. Зокрема у ньому

розглядається процес імітаційного моделювання мереж систем масового обслуговування – один з можливих варіантів моделювання web-системи.

Але наявність великої кількості невирішених проблем, дискусійність багатьох теоретичних положень, а також їх велике практичне значення обумовлюють необхідність подальшого дослідження web-систем.

Актуальність означених завдань та необхідність проведення подальших досліджень з метою їх автоматизації не тільки суто діловодних функцій, процесів обслуговування, інформаційно-документаційної сфери управління, але і багато інших сфер роботи зумовили вибір теми і мети дослідження. Метою даної публікації є дослідження імітаційної моделі логістичної web-системи, яка могла б адекватно моделювати роботу реальної системи, яка пов'язана з прийманням та обслуговуванням заявок із глобальної мережі та забезпечувала б наочною статистикою роботу реальної системи.

Термін web-система (web-сервер) має декілька трактувань. З одного боку, це набір документів, пов'язаних гіперпосиланнями (при цьому у web-системи є основна сторінка, через яку за один або декілька кроків доступні всі інші). З іншого боку, термін web-система може означати комп'ютер або локальну мережу, на якій розміщений набір документів, доступний через локальну або глобальну мережу, і який забезпечує доступ користувачам. Нарешті, останнє значення цього терміну – програмне забезпечення, призначене для доступу до набору документів через локальну або глобальну мережу.

У нашому випадку web-система представлена у вигляді мережі з великої кількості комп'ютерів – серверів, з'єднаних між собою певним чином. Ззовні на вхід web-системи поступає потік заявок (вимог), котрі треба обробити за певний період часу, залучивши для цього певну кількість ресурсів внутрішньої мережі web-системи. Природа даних заявок не відіграє особливої ролі, важливими є лише закон розподілу їх поступлення та сукупність умов необхідних для їх обслуговування.

Документи та дані в такій системі зберігаються централізовано на сервері, де, власне, і встановлюється сама програма. В ній активно використовуються впливаючі підказки, що прискорює знайомство зі програмою і її використання на

початковому етапі.

Web-інтерфейс системи значно спрощує її налаштування, яка обов'язково потрібна. Для організації руху документів (повідомлення про надходження документа на узгодження (підтвердження)) може використовуватися електронна пошта.

Можливість вивантаження опису документа у форматі XML у файл дозволяє забезпечити інтеграцію (обмін даними і документами) з іншими організаціями (територіально віддаленими підрозділами), а використання формату XML максимально полегшує взаємодію з програмним забезпеченням інших виробників відповідно до Стандарту взаємодії систем автоматизації забезпечення, документації управління [5].

В цілому програма, безумовно, цікава і заслуговує уваги при вирішенні питання автоматизації діловодства і роботи з документами у великих та середніх організаціях. Особливо корисної вона є для територіально розподілених підприємств [3].

Робота будь-якої такої системи полягає в обслуговуванні потоку заявок (вимог), що поступає на її вхід. Заявки поступають в систему одна за одною в деякі, взагалі кажучи, випадкові моменти часу. Обслуговування заявки, що поступила, продовжується якийсь час, після чого система звільняється для обслуговування наступної заявки.

Відзначимо випадковий характер надходження заявок і випадковий характер проміжків часу, необхідного для їх опрацювання. В цілому маємо випадковий процес, при якому можливі як перевантаження, так і простой.

Практичне використання моделей локальних мереж у багатьох випадках припускає наявність інформації про реальні характеристики обчислювального процесу. Така інформація може бути отримана емпіричними методами, на основі яких в даний час створюються засоби для дослідження апаратно-програмних компонентів локальної мережі.

Головна і найбільш очевидна мета імітаційного моделювання (зокрема моделювання роботи логістичної web-систем) – з'ясувати, як вплинуть на продуктивність окремі зміни конфігурації системи або збільшення навантаження на неї. Процес моделювання включає три фази [2-4]. На фазі валідації будується базова модель існуючої системи, перевіряються і обґрунтовуються припущення, що лежать

в її основі. На фазі проектування модель використовується в прогностичних цілях, для прогнозу впливу різних модифікацій на продуктивність. На фазі верифікації реальна продуктивність модифікованої системи порівнюється з результатами моделювання. Взяті разом, ці три фази утворюють модельний цикл.

Результат імітаційного експерименту за своєю природою носить статистичний характер, і тому для його інтерпретації потрібний певний статистичний аналіз. Це найбільш критична ділянка в розробці моделі, тому що часто, дивлячись на вихідні дані, важко їх узагальнити і зробити висновки. Ми можемо тільки відмітити, що відбулося у цьому конкретному випадку.

При реалізації моделюючих алгоритмів на комп'ютері виробляється інформація про стани досліджуваних систем. Ця інформація є вихідною для визначення наближених значень або оцінки шуканих величин.

При наявності великої кількості однорідної інформації традиційним в програмуванні є використання масивів. Для зберігання об'єктів, кількість яких наперед невідома, а саме така ситуація є характерною для процесу імітаційного моделювання, – можна використовувати динамічні масиви. При відсутності в обраній мові програмуванні згаданих конструкцій, їх майже завжди можна замінити на зв'язні списки, що, правда, з деяким зниженням продуктивності.

Тому необхідно таким чином організувати фіксацію і обробку результатів моделювання, щоб оцінки для шуканих величин формувалися поступово по ходу моделювання, без спеціального запам'ятовування всієї інформації про стани системи.

Якщо при моделюванні даної системи враховуються випадкові чинники, то як оцінки для шуканих величин використовуються середні значення, дисперсії та інші ймовірнісні характеристики випадкових величин, отримані при багатократному моделюванні. Ця обставина дозволяє формувати оцінки таким чином, що для запам'ятовування самої оцінки використовується лише одна (іноді декілька) змінна, а всі інші величини, які вже використовувались при даній оцінці, не запам'ятовуються.

Вся робота логістичної web-системи організована на основі механізму подій: виконуючи вибір подій з черги починаючи з тої, час настання якої найменший, ми синхронізуємо роботу системи в

цілому в часі та гарантуємо обробку всіх подій в системі своєчасно і в заданій послідовності.

До основних переваг такої web-системи можна віднести:

1. Зручність та простота у використанні.
2. Висока швидкодія в процесі моделювання та опрацювання статистики
3. Можливість взаємодії з іншими програмними продуктами, завдяки використанню універсальної мови представлення даних XML.

Основними недоліками системи є:

1. Прив'язаність до конкретної платформи.
2. Відсутність можливості перегляду інформації протягом усього періоду моделювання.
3. Представлення вихідної інформації такою як вона є – без встановлення причинно-наслідкових зв'язків між вхідними та вихідними даними.

Концептуальною основою системи є представлення логістичної web-системи у вигляді мережі обчислювальних систем. При створенні імітаційної моделі вона в свою чергу замінюється множиною систем масового обслуговування (СМО), які відображають конфігурацію мережі, кожен елемент якої є окремою системою, що функціонує по певному алгоритму. На множині однієї обчислювальної системи задаються джерела інформаційних потоків, обслуговування яких елементи множини виконують у відповідності із заданими алгоритмами.

Література

1. <http://study.utmn.ru/~ykaryakin/3curs/index.htm>
(Основы моделирования)
2. Компьютерное моделирование. Учебный курс
www.tula.net/tgpu/resources/comp_model/index.htm
3. Лоу А. М., Кельтон В. Д. Имитационное моделирование. Классика CS. 3-е изд., СПб: изд. дом „Питер”, 2004. – 848 с.
4. Менаске Д., Алмейда В. Производительность Web-служб. Анализ, оценка и планирование. – СПб: ООО “ДиаСофтЮП”, 2003. – 480 с.
5. Труб И. Объектно-ориентированное моделирование на C++ – СПб.: Питер, 2006. – 411 с.

СИСТЕМА АВТОМАТИЗОВАНОЇ ПОБУДОВИ ПРОГРАМНИХ ПРОЕКТІВ MAVEN

А. Батюк, І. Хомишин, С. Пилипчук

*кафедра автоматизованих систем управління, Національний університет
“Львівська політехніка”*

This article contains nutshell review of project development process and it automation via build system Maven.

Вступ. Ця стаття містить опис процесу розробки програмних продуктів на мові Java та про можливість автоматизації цього процесу за допомогою Maven. Терміном “програмний продукт” може описуватися будь-яка програма (консольна, десктопна програма з графічним інтерфейсом, веб-сайт або веб-сервіс), але в цій статті використовується цей термін використовується як синонім до терміну веб-ужиток, веб-аплікація, веб-проект.

Для того, щоб розробити веб-проект, необхідно визначитися з технологіями, що будуть використані, продумати архітектуру, написати код, компілювати та протестувати його. Зазвичай, проекти пишуться з використанням не тільки стандартних бібліотек мови, а й з використанням бібліотек третіх осіб. Тому потрібно завантажити необхідні бібліотеки в певну папку (зазвичай це папка lib, що знаходиться в кореновому каталозі проекту) і вказати її компілятору. При цьому слід уважно стежити за версіями бібліотек і залежностями між ними тому, що часто різні версії бібліотек є несумісними або вимагають присутності інших бібліотек – такі відношення бібліотек називаються залежностями проекту. Неправильно побудовані залежності в кращому випадку, не дозволять скомпілювати проект, в гіршому – аплікація буде некоректно себе поводити, і на з’ясування причин такої поведінки може піти багато часу і зусиль.

Написання проекту – це створення текстових файлів, в яких містяться описи класів, їх полів і методів, а також конфігураційні файли. Для того, щоб ці файли зміг виконувати комп’ютер їх треба скомпілювати, тобто перетворити у байт-код, який розуміє віртуальна машина Java. Зазвичай для цього з проекту будується виконуваний файл (з розширенням jar або war). Процес побудови такого файлу і називається збором проекту (Building of the Project).

Теоретично компіляцію і створення jar-файлу можна здійснити командами з консолі використовуючи засоби віртуальної машини Java (JVM). Але реальні проекти складаються з сотень, а то й тисяч файлів, і компіляція та збір їх вручну є неможливим. Скомпілювати java-файли в class-файли можна за допомогою IDE (інтегрованих середовищ розробки, таких як NetBeans, Eclipse, MyEclipse), але це не вирішує проблему збору виконавчих файлів, які можна було б запускати без IDE. Хоча деякі IDE дають можливість імпортувати виконуваний jar-файл, але такий підхід у більшості випадків є неприйнятним. Для вирішення таких проблем були розроблені різноманітні засоби автоматизації збору проектів.

Перш ніж перейти до розгляду Maven варто ознайомитися із Ant, на основі якого побудований Maven. Ant – це так званий “build script”, який:

- повністю зроблений на Java, а, отже, не залежить від платформи;
- використовує XML формат для файла-сценарію, що значно спрощує опис збірки;
- легко розширюється;
- інтегрований з багатьма IDE (IDEA, NetBeans, Eclipse).

Ant надає можливість компілювати, копіювати, видаляти файли, створювати zip/tar архіви чи директорії, пакувати jar-файли, практично не задумуючись над тим, які процеси реально запускаються при цьому в операційній системі.

Наведемо приклад Ant-скрипта:

```
<project name="example" default="build" basedir=".">
  <!--Описуємо структуру
        нашого проекту.-->
  <property name="src.home" value="${basedir}/src"/>
  <property name="src.main.java" value="${src.home}/main/java"/>
  <property name="output.home" value="${basedir}/output"/>
  <property name="output.classes" value="${output.home}/classes"/>

  <!-- Компілюємо код на Java. -->
  <target name="compile">
    <javac source="1.6" target="1.6"
      srcdir="${src.main.java}" destdir="${output.classes}"/>
  </target>
```

```

<target name="build" depends="compile">
<!-- Пакуємо jar-файл. -->
<jar destfile="${output.home}/${project.name}.jar">
<fileset dir="${output.classes}"/>
</jar>
</project>

```

Проте, Ant не передбачає механізму управління залежностями проекту. Розробник повинен сам завантажувати потрібні бібліотеки, слідкувати за взаємозалежностями між ними. Саме тому доцільно розглянути таку систему як Maven.

Ці системи мають різні підходи до побудови проектів. Ідеї в Ant полягають в тому, що розробник обдумує, які операції з вихідними файлами йому буде зручно автоматизувати, та описує їх. Maven пропонує модель – POM (Project Object Model). [3]

Структура проекту набуває досить чітко окреслених форм, а сам процес збірки проходить через набір визначених етапів – фаз. Об'єкт проекту містить інформацію про різні деталі: опис, списки розробників та учасників, шлях до репозиторію системи контролю версій і ще багато чого корисного. Фази збірки мають стандартний сценарій та можуть доповнюватися плагінами. [3]

Maven накладає досить жорсткі (але й досить раціональні) обмеження стосовно структури, а головою його перевагою є управління залежностями проекту. Розглянемо структури проектів:

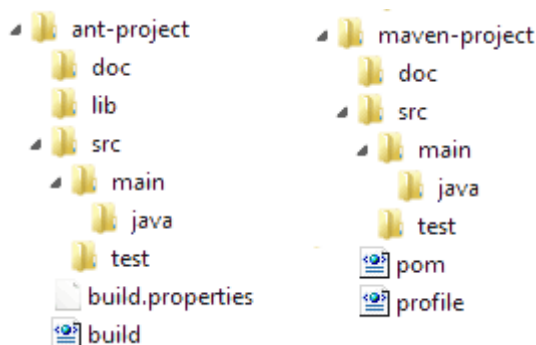


Рис.1 Структура проектів які використовують Ant та Maven

З рис.1 видно, що в Maven-проекті відсутня папка lib. Це

тому, що всі необхідні бібліотеки зберігаються окремо від проекту, в так званому локальному репозиторії, автоматично включаються в проект (для цього потрібно зробити декларацію в файлі `pom.xml`) і можуть використовуватися в багатьох проектах одночасно.

Ще однією корисною можливістю Maven є архетипи – шаблони, за допомогою яких можна легко розпочинати розробку нових проектів, автоматично отримуючи початкові конфігураційні файли для зв'язки, наприклад, Spring та Hibernate.

До **головних особливостей Maven** належать [2]:

- простий старт проектів, що будуються на основі найкращих практик програмування – новий проект чи модуль створюється за секунди;
- уніфіковане використання між проектами означає, що не потрібно пояснювати структуру проекту для нового члена команди розробників;
- управління залежностями проекту, включаючи автоматичне оновлення та керування транзитивними залежностями;
- можливість одночасно працювати з кількома проектами;
- великий репозиторій останніх версій бібліотек і метаданих, що швидко збільшується;
- розширюваність, з можливістю написання плагінів на Java чи скриптових мовах;
- легкий доступ до нових функцій без значних змін конфігурації;
- можливість виконання Ant-скриптів для управління залежностями і розгортання проектів поза межами Maven;
- можливість пакувати будь-яку кількість проектів у JAR - або WAR-файли.

POM. Файл `pom.xml` є ключовою конфігурацією Maven для проекту. В ньому міститься основна інформація, що необхідна для побудови проекту. Зазвичай POM є досить великим файлом. Але не обов'язково знати всі деталі про складові частини POM, для того щоб використовувати їх. Найпростіший приклад POM:

```
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://
www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://
```

```

maven.apache.org/xsd/maven-4.0.0.xsd">
<groupId>com.mycompany.app</groupId>
  <artifactId>my-app</artifactId>
  <packaging>jar</packaging>
  <version>1.0-SNAPSHOT</version>
  <name>Maven Quick Start Archetype</name>
  <url>http://maven.apache.org</url>
  <properties>
    <spring.version>3.0.2.RELEASE</spring.version>
  </properties>
  <dependencies>
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>3.8.1</version>
      <scope>test</scope>
    </dependency>
    <dependency>
      <groupId>org.springframework</groupId>
      <artifactId>spring-core</artifactId>
      <version>${spring.version}</version>
    </dependency>
  </dependencies>
  <plugins>
    <plugin>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-compiler-plugin</artifactId>
      <version>2.1</version>
      <configuration>
        <source>1.6</source>
        <target>1.6</target>
      </configuration>
    </plugin>
  </plugins>
</project>

```

Складові елементи pom.xml

- groupId та artifactId – ідентифікатор групи та артефакту відповідно, використовуються ідентифікації проекту. Артефактом прийнято називати будь-який проект (jar-

файл), що міститься в репозиторії. Для споріднених проектів groupId може бути однаковим. Ці параметри вибираються розробниками довільно;

- **packaging** – параметр вказує тип вихідного файлу проекту. Значення вибирається з переліку дозволених, найчастіше використовуються значення jar, war, ear;
- **version** – позначає версію артефакту. При збірці проекту в файл, назва файлу формується за принципом поєднання значень параметрів artifactId та version;
- **name** – ім'я проекту. Параметр не є обов'язковим;
- **url** – посилання на сайт проекту;
- **properties** – розділ, який описує змінні, що можуть використовуватись в файлі. Наприклад, змінна, оголошена як `<spring.version>3.0.2.RELEASE</spring.version>` пізніше використовується в іншому місці файлу для опису версії артефакту `<version>${spring.version}</version>`. Слід пам'ятати, що значення змінних в pom.xml міняти не можна, тому точніше було б назвати такі конструкції константами;
- **dependencies** – розділ, що описує залежності проекту. Тільки в його межах можна використовувати параметри `dependencie`, кожен з яких і описує необхідну бібліотеку. Опис потрібних бібліотек збігається з обов'язковими параметрами, що описують поточний проект;
- **scope** – вказує на якій фазі побудови проекту використовувати артефакт.

Фази Maven. Розглянемо типові фази, які може проходити Maven проект (назви наводяться англійською, оскільки вони є одночасно командами, які використовуються в командній консолі):

- **validate** – перевіряється чи проект є правильним і чи є доступна вся необхідна інформація;
- **compile** – відбувається компіляція вихідного коду проекту;
- **test** – за допомогою модульних тестів (unit tests) тестується скомпільований код. Ці тести не повинні вимагати пакування чи розгортання проекту;

- **package** – компільований код пакується в дистрибутивний формат, наприклад в JAR;
- **integration-test** – якщо необхідно, то пакет обробляється і розгортається на необхідне апаратне середовище, де запускаються інтеграційні тести;
- **verify** – запускаються будь-які перевірки на відповідність пакету якісним критеріям
- **install** – пакет інсталюється в локальний репозиторій для того, щоб локально використовуватися як залежність (dependency) в інших проектах;
- **deploy** – виконується на тестовому або кінцевому апаратному забезпеченні, копіюється кінцевий пакет на віддалений репозиторій для спільного використання з іншими проектами та розробниками.

Виконання будь-якої фази включає виконання всіх попередніх фаз, тобто команда

```
mvn test
```

призведе до виконання фаз **validate, compile, test**.

Є ще дві додаткові фази, які не включені до списку типових:

- **clean** – прибирає після попередніх побудов артефактів (знищує старі файли);
- **site** – генерує сайт з документацією для проекту.

Інсталяція. Maven є побудований на основі Java, тому його використання необхідно інсталювати віртуальну машину Java. В залежності від налаштувань мережі, можливо доведеться зробити додаткові налаштування (зокрема параметрів проксі-сервера). Детальнішу інформацію можна знайти в документації та інструкції з інсталяції.

Висновок. Розробка багатьох проектів містить частину роботи, яка є рутинною. Для пришвидшення розробки цю роботу можна автоматизувати. На сьогодні існує кілька систем, які дозволяють це зробити. В цій статті було розглянуто одну з них. Використання Maven робить процес розробки програмних продуктів швидким, гнучким і зручним.

Література

1. Iverson, W. (1 March 2007 p.). Building web applications with maven 2. Отримано з today.java.net: <http://today.java.net/building-web-applications-with-maven-2.html>
2. The Apache Software Foundation. (16 October 2009 p.). Apache Maven.
3. Мазур, Р. (2 Лютого 2009 р.). Від Ant до Maven, а потім назад. Отримано з stanfy.com.ua: <http://stanfy.com.ua/>

МОДЕЛЮВАННЯ СОЦІАЛЬНОГО ЗАБЕЗПЕЧЕННЯ НАСЕЛЕННЯ УКРАЇНИ ДЛЯ ПОШУКУ ШЛЯХІВ ПІДВИЩЕННЯ ЙОГО СОЦІАЛЬНОГО ЗАХИСТУ

Т.М. Степура

к.е.н., зав.кафедри економіки та менеджменту, ЛФ ЄУ

Розглянуто місце соціального страхування у соціальній політиці країни, наведено основні передумови та результати кореляційно-регресійного аналізу впливу факторів на розміри пенсійних виплат. На основі проведеного аналізу запропоновано основні шляхи підвищення соціального захисту населення.

Основоположними поняттями, які визначають соціальну спрямованість економічної національної системи є розвиток соціальної інфраструктури, соціальна політика, соціальний захист, соціальне забезпечення, соціальне страхування. Усі вони взаємопов'язані та мають підпорядкований зміст.

Соціальний захист – це система правових, організаційних, економічних заходів щодо підтримки рівня життя та задоволення соціальних потреб населення у конкретних соціально-економічних умовах.

Соціальний захист реалізовується через соціальне забезпечення, соціальну підтримку та соціальні гарантії. У ракурсі вищенаведеного визначення соціального захисту соціальне забезпечення можна охарактеризувати як механізм реалізації цього захисту через правові, організаційні та економічні важелі. Соціальне забезпечення по суті є процесом перерозподілу економічних благ,

створених у державі, на користь тих, хто більшою мірою потребує соціальної підтримки.

Соціальне забезпечення на практиці реалізується через фінансові потоки та трансферти, хоча може мати як матеріальну (фінансову чи уречевлену) форму, так і нематеріальну (послуга, інформація тощо).

Спільною рисою для усіх форм реалізації соціального захисту є те, що вони покликані надати населенню певний рівень безпеки, тобто запобігти ризику. Поняття соціального ризику є недостатньо дослідженим у науковій літературі і здебільшого трактується у вузькому розумінні.

Можливості соціальної страхової системи держави насамперед мали б залежати від її економічної спроможності, тобто від стану розвитку економіки, що виражається у макропоказниках (ВВП, інфляція, безробіття, оплата праці, ставки кредитування, інноваційність економіки тощо).

За проведенням авторським аналітичним дослідженням було відібрано основні фактори, які впливають на розміри соціальних виплат. Факторними ознаками обрано:

- 1) ВВП у порівнянних цінах (отриманий шляхом коригування фактичного ВВП на індекс-дефлятор);
- 2) інфляцію (індекси споживчих цін);
- 3) питому вагу населення працездатного віку у його загальній чисельності. Вибір показника зумовлений тим, що саме працездатне населення виступає платником страхових внесків. З цього погляду логічніше було б обрати у ролі ознаки зайняте населення, однак тоді виникло б явище мультиколінеарності із показником оплати праці в економіці країни;
- 4) середню для обох статей очікувану тривалість життя при народженні, оскільки цей показник прямим чином обумовлює вікову структуру населення у майбутньому, що виключило його тісний кореляційний зв'язок із попередньою факторною ознакою ($R=0,32$), разом з тим, достатньо корелює із результируючою ознакою – середнім розміром пенсій ($R=0,63$);
- 5) суму оплати праці найманих працівників в економіці країни, яка слугує базисом нарахувань та утримань у страхові фонди та основою їх фінансового забезпечення;

б) сукупні відрахування та утримання у соціальні страхові фонди.

В останньому показнику було враховано три фонди соціального страхування, окрім Фонду загальнообов'язкового державного соціального страхування від нещасного випадку на виробництві та професійного захворювання, які спричинили втрату працездатності, оскільки цим видом страхування передбачено лише нарахування на фонд заробітної плати, що сплачує роботодавець. Разом з тим, обрахування цього показника у загальному навантаженні соціальними фондами на оплату праці складає певні труднощі, оскільки законодавством передбачено диференціацію ставки нарахувань у залежності від класу ризику виробництва чи галузі економіки.

Результуючим показником обрано середній розмір призначеної місячної пенсії пенсіонерам, які перебувають на обліку в органах Пенсійного фонду (грн.). Варто зазначити, що середній розмір пенсії обчислюється, виходячи із сум пенсій за віком, за інвалідністю та у разі втрати годувальника. Виплати останніх двох видів пенсій здійснюються із Фонду загальнообов'язкового державного соціального страхування від нещасного випадку на виробництві та професійного захворювання, які спричинили втрату працездатності. Тому дослідження впливу сукупного розміру страхових внесків у зазначені вище соціальні фонди на суму пенсійних виплат вважаємо коректним, тим більше, що відсотки відрахувань у Пенсійний фонд серед усіх інших є суттєво вищими.

До обчислення моделі факторні ознаки були перевірені на предмет наявності мультиколінеарності шляхом обчислення парних коефіцієнтів кореляції.

Параметри рівняння було обчислено способом найменших квадратів. Підвищення надійності результатів досягнуто шляхом включення у модель 12 періодів спостережень (з 1998 до 2009 років).

У результаті отримано рівняння регресії, яке має вигляд:

$$\bar{y} = 6330,593 + 0,000302x_1 - 0,26513x_2 - 24,9425x_3 - 82,7893x_4 + 0,001397x_5 + 17,64648x_6$$

Множинний коефіцієнт кореляції становив 0,998, а коефіцієнт детермінації 0,996, що свідчить про наявність тісного зв'язку між

дібраними факторними та результуючою ознаками. Отже, фактор, які включено у модель, формують 95,6% величини результату (y). За критерієм Фішера встановлено, що побудована модель адекватна експериментальним даним.

Також для виявлення порівняльної сили впливу окремих факторів на результат було обчислено часткові коефіцієнти еластичності.

Розраховані часткові коефіцієнти еластичності показують, що за абсолютним приростом найбільший вплив на розмір середньої пенсії має четвертий фактор (очікувана тривалість життя при народженні) зі збільшенням якого на 1% розмір середньої пенсії знижується на 19,06%. Наступні за силою впливу фактори – питома вага працездатного населення у всій чисельності населення та розмір ставок відрахувань у соціальні фонди, при зростанні якого на 1% пенсія зростає на 2,36%.

Отже, проведений кореляційно-регресійний аналіз уможливив узагальнюючий висновок про те, що першочерговий вплив на пенсійне забезпечення мешканців України чинять не макроекономічні показники, а демографічні та похідні від них показники економічного та соціального навантаження на працюючих осіб. Тому вирішення питань підвищення соціального захисту населення лежить у площині соціально-демографічної політики, яка, безсумнівно, вже у свою чергу, залежить від економічної спроможності держави та політичної волі ефективно її провадити. І у цьому керунку вважаємо за необхідне вжити наступних кроків:

- Виважена демографічна політика держави, – завдання довгострокового характеру, для вирішення якого необхідно вживати стратегічних та послідовних кроків. Останнє в умовах політичної нестабільності та відсутності єдиного підходу до реформування соціально-економічних засад суспільного життя видається особливо складним. Окрім підтримки молодих сімей та стимулювання народжуваності ця політика повинна включати низку заходів щодо зміцнення здоров'я населення та дітей зокрема, забезпечення доступу до якісної освіти, в тому числі й професійно-технічної, підвищення мобільності працівників, вирішення проблеми першого робочого місця для молоді, розвиток соціальної інфраструктури, зокрема на селі тощо.

- Поступове підвищення оплати праці в економіці країни та збільшення її частки у сукупних доходах населення. Реалізація цього можлива лише при умові економічного зростання на засадах інноваційного розвитку, сприяння піднесенню галузей економіки на основі визначення регіональної пріоритетності (наприклад, туристична та рекреаційна сфера Львівщини).
- Проведення комплексної та цілісної інституційної реформи системи соціального страхування населення, що уже передбачено прийняттям законопроекту “Про збір та облік єдиного внеску на загальнообов’язкове державне соціальне страхування” №2464-VI від 08.07.2010р., який має набрати чинності з 01.01.2011р. Разом з тим, законопроектом передбачено спрощення процедури справляння єдиного внеску на соціальне страхування, однак не усунуто усе дублювання функцій робочих органів фондів соціального страхування, не передбачено їх підпорядкування, що суперечить принципам ефективного менеджменту і призводить лише до половинної економії витрат на адміністрування соціальних внесків.
- Розширення бази сплати соціальних внесків. Зважаючи на те, що податкове та інше навантаження на фонд оплати праці й так надмірне, можливості збільшувати їх немає. Тому для вирішення проблеми наповнення соціальних фондів, насамперед Пенсійного, коштами необхідно виводити заробітну плату з “тіні” збільшуючи таким чином базу сплати внесків. Зреалізувати це можливо завдяки поступовому зростанню продуктивності праці і економіці країни, що може бути досягнуто завдяки низці заходів: оновлення техніко-технологічної бази виробництва, оптимізація управлінських процесів, покращення організації праці, матеріальне і моральне стимулювання праці тощо.

Література

1. Соціальні індикатори рівня життя населення: стат. збірник. – Л.: Гол-не упр. статистики у Льв. обл., 2010. – 170с.

М'ЯКІ ОБЧИСЛЕННЯ В ЕКОНОМІЦІ ТА ФІНАНСАХ

¹К.Ф. Базилюк, ²С.Р. Ладик

¹к.т.н., доцент, доцент кафедри ПМ і КІС УАД

²к.е.н., доцент, доцент кафедри ПЗІ ЛКА

У статті досліджено можливості застосування м'яких обчислень при вирішенні задач економіки. Останнім часом різко підсилювся інтерес до прийняття рішень в умовах неповної (недостовірної, суперечливої й ін.) інформації, коли єдиним визначенням умови ухвалення рішення є термін “нечіткі умови”, що розуміються досить широко і багатоманітно. Природно, що у таких ситуаціях мова не може йти про точність обчислень. Вважається нормальним при прийнятті рішень використовувати не оптимальні рішення, а практично припустимі. Терпимість до низької точності обчислень (перетворень) вважається альфа й омегаю методології обчислень, які називають “м'якими”.

Переважна більшість сучасних економічних моделей, що побудовані на базі економетричного апарату та теорії нечіткої логіки, не враховують нетривіальність моделювання економічних процесів, яка обумовлена наступними причинами: недосконалістю статистичної інформації, її нечіткістю, іноді навіть суперечливістю і недоступністю, перевагою якісних факторів над кількісними, істотною нелінійною залежністю рівня основних показників від цих факторів. Крім того, в економетричних моделях попередньо передбачається, що вид шуканої залежності вихідної змінної від вхідних параметрів може бути лінійним, логарифмічним або у вигляді багаточлена. Усе, що не відповідає цій ідеалізації, відкидається.

Нетривіальність же моделювання таких складних процесів, як економічні і фінансові, обумовлена низкою факторів. Це:

- недосконалість статистичної інформації для економіки України, її нечіткість, іноді навіть суперечливість, недоступність;
- необхідність враховувати не тільки кількісну інформацію, але й якісну;
- короткий часовий інтервал, за яким робиться прогноз;
- істотно нелінійна (невідома заздалегідь) залежність вихідної змінної від вхідних параметрів;

- велика кількість як макроекономічних і мікроекономічних, так і політичних факторів, що впливають на економічні і фінансові процеси;
- високий рівень вимог до рішень, що приймаються у процесі економічного планування на основі побудованих моделей.

Усе це не дозволяє забезпечити високий рівень адекватності моделі, побудованої на основі економетричного підходу.

У таких умовах ефективним засобом є використання інформаційних технологій на базі м'яких обчислень, для яких характерна "терпимість до неточності, невизначеності або часткової істинності для досягнення зручності маніпулювання, низької вартості рішення і кращої відповідності реальності" [1].

М'які обчислення включають дослідження методами нечіткої логіки, штучних нейронних мереж, генетичних алгоритмів та ймовірного моделювання. Кожна зі складових методології має багато можливостей для її використання в рамках м'яких обчислень. Нечітка логіка лежить в основі методів роботи з неточністю, зернистою структурою інформації, наближеним міркуванням, обчислень зі словами (computing with words). Використання нейромереж дає можливість навчання, адаптації та ідентифікації. Генетичні алгоритми забезпечують можливість систематизувати випадковий пошук і досягати оптимального значення характеристик.

Використання нечіткої логіки дозволяє найбільш адекватно здійснити перехід від словесного якісного опису об'єкта, що є характерним для міркування людини, до числових кількісних оцінок та характеристик його стану і сформулювати на основі цього прості та ефективні алгоритми розв'язання задачі, тобто дозволяє моделювати людське мислення. Подібно до того, як *нечіткі множини* розширили рамки класичної математичної теорії множин, *нечітка логіка* увійшла практично у більшість методів Data Mining, наділивши їх новою функціональністю.

В останні роки нейронні мережі увійшли в практику скрізь, де потрібно вирішувати завдання прогнозування, класифікації, керування. Такий успіх визначається декількома причинами:

- нейронні мережі – потужний метод моделювання, що дозволяє відтворювати надзвичайно складні нелінійні залежності;

- справляються з “прокляттям розмірності”, який не дозволяє моделювати лінійні залежності у випадку великої кількості змінних;
- вчать на прикладах (експериментальних та статистичних даних);
- прості у використанні.

Рівень знань, необхідний для успішного застосування нейронних мереж, набагато менший, ніж, наприклад, при використанні традиційних методів статистики. Однак, як і у випадку ймовірно-статистичних методів, для навчання нейронних мереж потрібна велика вибірка експериментальних даних.

Нечіткі нейронні мережі (fuzzy-neural networks) здійснюють виводи на основі апарату нечіткої логіки, проте параметри функцій належності налагоджуються з використанням алгоритмів навчання нейромереж.

На відміну від нейронних мереж, де за неструктурованими числовими даними здійснюється пошук розв'язку задачі шляхом навчання і тренування, нечіткі технології (fuzzy-системи) використовують експертну інформацію про закономірності, виявлені у наявних експериментальних даних, у вигляді природно-мовних правил “ЯКЩО – ТО”. Ці правила, які формалізуються за допомогою нечіткої логіки, дозволяють будувати моделі ідентифікації і прогнозу за відносно малими (порівняно з нейронними мережами) вибірками експериментальних даних. Сполучення двох описаних технологій є об'єднанням здатності до навчання нейронних мереж з наочністю і легким інтерпретуванням нечітких правил “ЯКЩО – ТО”.

Таким чином, *головними перевагами нейронечіткої технології* при вирішенні економічних задач є:

- можливість використання при моделюванні не тільки кількісної, чіткої експериментальної інформації про закономірності, як в економетричних моделях, а й нечіткої, якісної, яка подається експертами (спеціалістами) природною мовою;
- можливість застосування для прогнозування будь-яких економічних і фінансових процесів;
- можливість моделювання при відносно малих (порівнянно з економетричними підходами)

- експериментальних вибірках;
- можливість відтворювання складних нелінійних (непередбачених наперед) залежностей змінної від великої кількості вхідних параметрів;
- можливість зниження витрат на навчання, завдяки попередньому структуруванню на базі знань;
- здатність моделей до навчання, властиву нейронним мережам;
- достатньо загальний вигляд, при якому в ролі вхідних змінних можуть бути фактори, що впливають на економічні та фінансові процеси;
- можливість використання в процесі підтримки прийняття рішень там, де доводиться мати справу з інформаційною невизначеністю, з недоліком або недовірою до інформації.

Найбільше поширення в даний час одержали архітектури нечіткої нейромережі виду ANFIS и TSK.

Отже, дослідження застосування нейронечітких технологій дозволяє виділити такі основні особливості:

- нейромережі досить часто використовуються для прогнозування показників досліджуваного процесу, оскільки є ефективним засобом розв'язання задач, для яких важко скласти явний алгоритм. Основними проблемними питаннями при їх використанні є складнощі вибору відповідного типу мережі й необхідність достатньо великої кількості даних та, іноді, часу для їх навчання;
- генетичні алгоритми широко застосовуються як альтернативні засоби для розв'язання різноманітних задач оптимізації. І хоча сама постановка задачі в їх термінах не дає можливості проаналізувати статистичну значимість одержуваного з їх допомогою рішення, сила генетичного алгоритму полягає в його здатності маніпулювати одночасно багатьма параметрами, що для багатьох предметних областей має важливе значення;
- значне поширення застосування нечіткої логіки як методології вирішення задач пов'язане з тим, що на відміну від традиційної математики, що вимагає на

кожному кроці точних і однозначних формулювань закономірностей, нечітка логіка дозволяє маніпулювати в процесі моделювання мінімальним набором закономірностей. При цьому відбувається підключення інтуїції та досвіду експерта.

Для *управління знаннями* досить часто застосовують нейронечіткі інформаційні технології для створення нечітких експертних систем та баз знань.

Нечітка база знань являє собою сукупність лінгвістичних (значеннями яких є слова або речення природної мови) знань-правил типу: ЯКЩО < входи >, ТО < вихід >. У правилах нечіткої бази знань сконцентровані досвід експерта та його розуміння причинно-наслідкових зв'язків “входи... вихід”. Нечітка база знань може бути легко сформована досвідченим експертом, оскільки вона є сукупністю висловлювань природною мовою.

Нечіткі запити до баз даних (fuzzy queries) – перспективний напрям у сучасних системах обробки інформації. Цей інструмент дає можливість формулювати запити на природній мові, що неможливо при використанні стандартного механізму запитів. Для цієї мети розроблена нечітка реляційна алгебра і спеціальні розширення мов SQL для нечітких запитів. Більша частина досліджень у цій області належить західноєвропейським ученим Д.Дюбуа і Г. Праде.

Наявність унікальної структури моделі нечіткої бази знань для кожного класу задач дозволяє не тільки значно скоротити обсяг вибірки для навчання, але і забезпечити певну нечутливість моделі до випадкових викидів початкових даних.

Отже, саме м'які обчислення, які позбавлені названих недоліків економетричних підходів, дозволяють розширити можливості моделювання складних економічних і фінансових процесів, що є дуже актуальним у реальних умовах розвитку економіки України за відсутності достовірних даних, неповної і нечіткої статистичної інформації про об'єкт, складних нелінійних залежностей виходів від входів системи.

Література

1. Заде Л.А., Батыршин И.З. Роль мягких вычислений и нечеткой логики в понимании, конструировании и развитии

- информационных интеллектуальных систем: Пер. с англ. // Новости искусственного интеллекта. 2001. № 2. С. 7-11.
2. Притоманова О.М. Моделювання економічних і фінансових процесів на основі нейронечітких технологій оцінки і аналізу схильності підприємства до банкрутства // Вісник Дніпропетровського ун-ту. Серія: Економіка. – 2008. – С. 105-115.
 3. Масалович А. Нечеткая логика в бизнесе и финансах. www.tora-centre.ru/library/fuzzy/fuzzy-.htm
 4. Рибалка О.В. Визначення схильності підприємства до банкрутства за допомогою нечітких нейронних мереж // Актуальні проблеми економіки. – 2006. – №1 (55). – С. 199-206

ІНВЕСТИВАННЯ В ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ: ПРОБЛЕМИ ТА ПРІОРИТЕТИ РОЗВИТКУ

В. Голубка

здобувач кафедри економіки підприємства ЛНУ ім. І.Франка

Серед пріоритетів загальнодержавного розвитку інформаційно-інноваційні та наукоємні технології повинні займати одну з основних ролей, адже вони є одним з основних шляхів створення інформаційного суспільства в Україні та її інтеграції у світове співтовариство.

Створення новітнього інформаційного суспільства в Україні потребує ряд заходів, які можна поділити на тактичні та стратегічні:

- тактичні – створення сприятливих умов для розвитку всіх конкурентоспроможних сфер ІТ-галузі;
- стратегічні – інвестування у розвиток кадрового потенціалу шляхом підтримання відповідних вітчизняних галузей науки та освіти на сучасному технологічному рівні, а також забезпечення інтеграції України у світові інформаційні простори.

Виконання таких заходів потребує стабільного та у повному обсязі інвестування коштів з ціллю розробки відповідної стратегії та створення нормативної бази, яка б сприяла розбудові інституційної інфраструктури цієї галузі.

У свою чергу, залучення інвестицій в ІТ-індустрію потребує

дієвого механізму вкладання реальних інвестицій, а також створення сприятливих умов для подальшого розвитку цієї галузі. Адаже іноземні інвестори, що вкладають свої кошти у розвиток інформаційних технологій стикаються з труднощами, які потрібно звести до мінімуму.

1. Одна з основних проблем – кадри. Найбільша потреба виникає у спеціально підготовлених менеджерах (маркетологах, менеджерах з продажу, проектних менеджерах та керівниках виробництва), оскільки компанії-розробники, окрім створення програмного продукту, повинні мати можливість забезпечити його маркетингову розробку і подальше просування його на ринки збуту.
2. Необхідно створити ділову атмосферу з добре забезпеченою технологічною дисципліною усередині колективу, тобто створити умови для спроможності команди взяти на себе певні зобов'язання з чітко визначеними показниками і виконати їх у зазначені терміни. З іншого боку, більшість вітчизняних інвесторів чекають на короткий термін повернення своїх інвестиційних вкладень (менше року) та, водночас, великої прибутковості, чого на перших етапах розвитку новітніх технологій, як правило, не буває.
3. На даному етапі розвитку економіки України набувається відсутність побудови чіткої інфраструктури у даній галузі. Яскравим свідченням цього є той факт, що навіть Асоціація розробників програмного забезпечення не в змозі навести точні дані щодо кількості існуючих фірм і самостійних програмістів, які безпосередньо займаються проблемами інформаційних технологій та інвестицій у цій галузі.

Розвиток інформаційних технологій на новітній основі, безумовно, потребує суттєвого вдосконалення української законодавчої бази, зокрема інвестиційного, податкового, авторського законодавств тощо.

Вирішення цих завдань в першу чергу дозволить забезпечити прискорений розвиток сфери інформатизації у частині розбудови в Україні інформаційного простору, а також розробки та впровадження новітніх конкурентоспроможних інформаційно-комунікативних технологій (ІКТ) в усі сфери суспільного життя.

Література:

1. Деверадж С., Кохли Р. Тайны ИТ: измерение отдачи от инвестиций в информационные технологии. – М.: Бук-пресс, 2006. – 198с.
2. Гужва В.М. Інформаційні системи і технології на підприємствах: Навч. посібник. – К.: КНЕУ, 2001. – 400 с.
3. Гринберг А.С., Король И.А. Информационный менеджмент. М.: ЮНИТИ, 2003.
4. Крылов Э.И., Власова В.М., Чеснокова В.В. Основные принципы оценки эффективности инвестиционного проекта / СПбГУАП. – СПб.: 2003, 28с.
5. Пересада А.А., Онікієнко С.В., Коваленко Ю.М. Інвестиційний аналіз: Навч.-метод. посібник для самост. вивч. дисц. – К.: КНЕУ, 2003. – 134 с.

ІНТЕРПРЕТАЦІЯ АЛГОРИТМУ ЛІ ДЛЯ ВИКОНАННЯ НА БАГАТОПОТОЧНІЙ АРХІТЕКТУРІ NVIDIA® CUDA

Т.З. Цюра

лаборант кафедри МКД ЛФСУ

Алгоритм пошуку найкоротшого шляху в графі з ребрами одиначної довжини широко використовується для розв'язання багатьох прикладних завдань у найрізноманітніших програмних продуктах – від трасування друкованих плат та креслень до стратегічних ігор. У всіх цих випадках важливим, якщо не першочерговим, є показник ефективності виконання даного алгоритму.

Метою даної статті є дослідження особливостей застосування багатопотокових архітектур на прикладі NVIDIA® CUDA для максимізації ефективності виконання алгоритму Лі. У ході дослідження спроектовано три варіанти програмної реалізації алгоритму: для виконання на CPU, для виконання на GPU з лінійною моделлю розподілу потоків та для виконання на GPU з щільним заповненням блоків. Проводиться порівняння показників ефективності.

Вершини досліджуваного графа представляють координати на площині чи у просторі, які є вільними у лабіринті. Крім того,

визначені початкові (вхід у лабіринт) та кінцеві (вихід) координати. Лабіринт може містити стіни, які не можуть входити у шлях[1].

Індексація вершин у лабіринті регламентується відповідним предикатом доступності вершин на основі поточної позиції. Наприклад, для діагонального хвильового алгоритму предикат перевірятиме 8 можливих сусідів, для алгоритму Лі – лише 4.

Від початкових позицій, у всі допустимі предикатом сторони, поширюється “хвиля” – вільні вершини отримують значення довжини пройденого шляху. На кожному етапі поширення необхідно переглядати усі вершини графа. Розповсюдження “хвилі” триватиме, поки не буде досягнуто кінцевої вершини (виходу)[2].

Даний граф представлений у вигляді двовимірної матриці цілих чисел, яка зберігає стани лабіринту на протязі виконання ітерацій. Вільній вершині графа відповідатиме значення 0, недоступній вершині (стіні) – 1. Початкова ітерація хвилі присвоюватиме відповідному елементу матриці значення 10, з наступними ітераціями присвоюване значення збільшуватиметься на 1. Різниця кінцевого та початкового значень буде рівна найкоротшому шляху у графі. Предикат передбачатиме розповсюдження хвилі згідно алгоритму Лі [2].

Вхідними даними для тестування слугуватиме велика кількість матриць 20 порядку, об’єднана у масив. Участь у тестуванні ефективності беруть 4 програмні реалізації – 2 на CPU і 2 на GPU.

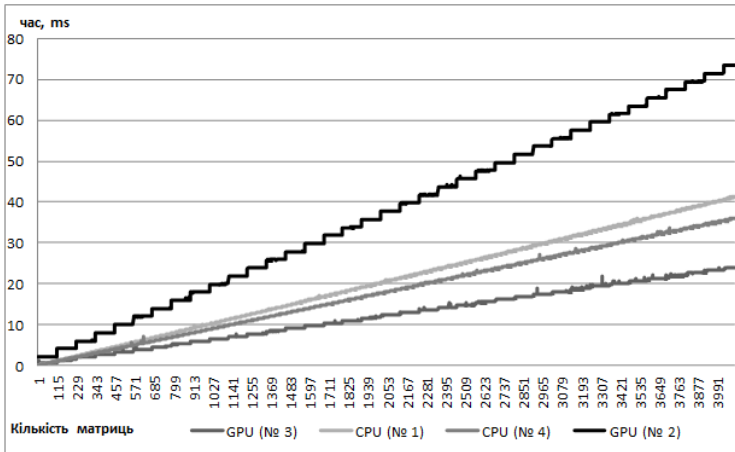
Програмна реалізація № 1 для виконання на центральному процесорі передбачає послідовне обчислення шляху виходу для кожного графу.

Обчислення на графічному процесорі з лінійним принципом розподілу потоків (№ 2) виконуватиметься послідовно у межах блоку.

Програмна реалізація № 3. Для максимізації ККД графічного прискорювача, необхідно заповнити блоки числом потоків, близьким до $T_{\max} = 512$ (для даного GPU)[4]. Порядок матриці розподілу потоків у цьому випадку буде рівний $\lfloor \sqrt{T_{\max}} \rfloor$.

Оскільки, слабким місцем досліджуваного алгоритму є частота звернення до пам’яті, тестування також виконано на CPU з інтегрованим контролером пам’яті (№4).

РЕЗУЛЬТАТИ:



РЕАЛІЗАЦІЯ № 1

Центральний процесор без інтегрованого контролера пам'яті показав хороший результат – 42 мс для 4096 матриць розміром 20x20.

РЕАЛІЗАЦІЯ № 2

Число активних потоків на GPU, згідно Compute Visual Profiler v 3.1.1 [3], не перевищувало 110. Збільшені витрати на ініціалізацію потоків також знизили ККД. Незважаючи на проведену оптимізацію звернень до пам'яті, виконання тесту зайняло 74 мс.

РЕАЛІЗАЦІЯ № 3

Усі 22*22 потоки були задіяні для обчислення. Час виконання тесту – 23 мс.

РЕАЛІЗАЦІЯ № 4

Суттєве прискорення звернень до пам'яті скоротило час виконання на 6 мс (15 %).

ВИСНОВОК

Виконання тесту GPU над числом матриць, меншим за кількість активних потоків, займає приблизно однаковий час. Зі збільшенням кількості матриць на наступній ітерації тесту, GPU ініціалізуватиме новий блок активних потоків, час виконання якого рівний часу виконання попереднього. Ця особливість роботи

багатопоточних систем надає графіку № 2 та № 3 ступінчастості.

В результаті дослідження досягнуто часу виконання тесту на GPU у 1,5-2 рази меншого, ніж на CPU з інтегрованим контролером пам'яті. У програмному коді наявно 4 неповні розгалуження для обробки сусідніх клітинок. Усі активні потоки поділяються на дві групи при виконанні неповного розгалуження: потоки, які виконують блок операторів при виконанні умови і потоки, виконання яких оминає цей блок. Особливістю GPU є послідовне виконання даних груп активних потоків, що знижує ефективність розпаралелювання. Для чотирьох розгалужень число груп становитиме вже $4^2 = 16$. Поширення “хвилі” у одному напрямі відбуватиметься у 16 разів швидше, ніж у 4-ох, що відкриває великий простір для оптимізації внаслідок зміни логіки роботи предиката доступності.

Використання програмної реалізації алгоритму Лі на GPU з щільним заповненням потоків у блоці дає можливість звільнити центральний процесор інформаційної системи для інших обчислень та збільшити у два рази об'єм оброблюваних графів.

Література

1. Lee, C. Y. (1961), “An Algorithm for Path Connections and Its Applications”, IRE Transactions on Electronic Computers EC-10, ст. 346–365
2. <http://users.eecs.northwestern.edu/~haizhou/357/lec6.pdf>
3. http://www.nvidia.com/object/cuda_get.html
4. <http://developer.nvidia.com/object/gpucomputing.html>

СЕКЦІЯ 4

Новітні комп'ютерні технології. Прикладні додатки, засоби захисту та інтелектуальні методи у телекомунікаціях

ВИКОРИСТАННЯ СУЧАСНИХ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ДЛЯ ЗАБЕЗПЕЧЕННЯ ВИБОРЧОГО ПРОЦЕСУ ОРГАНІВ МІСЦЕВОГО САМОВРЯДУВАННЯ

¹О.В. Марковець, ²А.М. Пелешишин

¹асистент кафедри ІСМ НУ ЛП

²д.т.н., професор кафедри ІСМ НУ ЛП

Розглянуто основні функції виборчих комісій та проблеми, які виникають під час їхньої роботи. Описано завдання, які потрібно вирішити, для підвищення ефективності роботи комісій. Подано рекомендації для реалізації електронної системи підтримки проведення виборів.

Головним органом місцевого рівня, який забезпечує організацію та проведення місцевих виборів є територіальна виборча комісія (ТВК). Її завданням є інформаційне та технічне забезпечення дільничних виборчих комісій (ДВК) для проведення виборів. Основними функціями ТВК є:

- формування списків виборців згідно адреси реєстрації;
- формування списків кандидатів в депутати по окремих округах;
- формування дільничних виборчих комісій з претендентів, які подаються представниками місцевих осередків партій та кандидатів в депутати
- призначення голів дільничних виборчих комісій;
- формування списків довірених осіб кандидатів та офіційних спостерігачів на виборах;

- друкування бланків протоколів комісій;
- отримання результатів голосування від дільничних виборчих комісій;
- підрахунок результатів та визначення переможців виборів.

Як показує досвід, ефективність роботи ДВК на теперішній час дуже низька. Це, в першу чергу, зумовлено тим, що більшість роботи члени ДВК проводять вручну, це стосується насамперед заповнення протоколів та складання актів. Крім того, значно сповільнюють роботу комісій технічні помилки і описки при заповненні протоколів, які потрібно виправляти, переписуючи протоколи. Також важливим фактором який впливає на ефективність роботи ДВК є безперервність засідання ДВК при підрахунку голосів, яке може тривати від декількох годин до кількох діб.

Для усунення цих проблем та недоліків необхідно впровадити електронну систему забезпечення проведення місцевих виборів. А оскільки доступ до мережі Інтернет набув широкого розповсюдження і може бути доступним на кожній виборчій дільниці, ми пропонуємо використати Інтернет-технології для реалізації даної системи.

Система забезпечення проведення місцевих виборів повинна складатись з інформаційної та аналітичної підсистем. Завданнями інформаційної підсистеми будуть:

- достовірної інформації про всіх суб'єктів виборчого процесу;
- формування бланків протоколів із введенням інформації про склади ДВК, кандидатів в депутати, номери дільниць та округів;
- формування бланків актів про роботу ДВК;
- введення результатів голосування.
- Основними завданнями аналітичної підсистеми будуть:
- Перевірка правильності внесення даних в бланки протоколів за результатами підрахунку голосів;
- Формування звіту по результатах роботи ДВК

Для реалізації інформаційної підсистеми найкраще вибрати

одну із систем управління контентом. На нашу думку найоптимальніший варіант це – CMS Drupal. Вона проста в застосуванні, має інтуїтивний інтерфейс та достатньо готових модулів для реалізації поставлених завдань. Для реалізації аналітичної підсистеми доцільно скористатись засобами мови програмування Perl. Ця мова дає широкі можливості з опрацювання текстової інформації, а також для забезпечення роботи в мережі Інтернет.

Ця система не повинна замінити працю членів ДБК у виборчому процесі, а призначена допомогти в правильності заповнення виборчої документації, зменшенні кількості помилок та опечаток, тим самим пришвидшити роботу комісії у підведенні результатів виборів.

ОГЛЯД БАГАТОМОДУЛЬНОГО КАРКАСУ РОЗРОБКИ SPRING FRAMEWORK

А. Батюк, С. Гринець, С. Пилипчук

*кафедра автоматизованих систем управління, Національний університет
“Львівська політехніка”*

Overview of popular framework set which includes framework for different purposes: web development, database management, web services and other.

Вступ. Spring Framework це багатомодульний каркас розробки, який забезпечує вирішення багатьох завдань, з якими стикаються Java розробники і організації, які хочуть створити інформаційну систему, яка використовує Java як основну мову програмування. Spring Framework складається з великої кількості компонентів, тому тяжко визначити його основне призначення. Spring Framework має в собі модулі, зв'язані з Java Enterprise платформою, які вирішують багато тривіальних задач зв'язаних з розробкою аплікацій промислового рівня. Spring Framework – найбільш відомий за можливість легкої інтеграції його з великими аплікаціями які історично написані на інших технологіях. В більшості випадків, розробник може легко додавати і інтегрувати модулі Spring Framework не порушуючи структурної цілісності

існуючої системи і не змінюючи технологій які використовуються, оскільки в Spring Framework включена інтеграція практично зі всіма популярними стандартами і технологіями. Все це робить Spring Framework доступним до використання в більшій частині областей використання Java. Spring Framework може бути розглянуто як колекцію менших фреймворків або фреймворків в фреймворку. Більшість цих фреймворків можуть працювати незалежно один від одного, проте, вони забезпечують більш широку функціональність при спільному їх використанні. Ці фреймворки легко діляться на структурні елементи типових комплексних програм:

- Контейнер Inversion of Control: конфігурування компонентів аплікацій та управління життєвим циклом Java об'єктів.
- Фреймворк аспектно-орієнтованого програмування: працює з функціональністю, яка не може бути реалізована можливостями об'єктно-орієнтованого програмування на Java без втрат.
- Фреймворк доступу до даних: працює з системами управління реляційними базами даних на платформі Java використовуючи JDBC і Object-relational mapping які забезпечують вирішення завдань, які повторюються у великому числі Java-based аплікацій.
- Фреймворк управління транзакціями: координація різних API управління транзакціями і інструментарій налаштовуваного управління транзакціями для об'єктів Java.
- Фреймворк Model-view controller: каркас, заснований на HTTP і сервлетах, що надає безліч можливостей для розширення і налаштування поведінки веб-аплікацій.
- Фреймворк віддаленого доступу: налаштовувана передача Java-об'єктів через мережу в стилі RPC , підтримує RMI , CORBA , HTTP-based протоколи, включаючи web-сервіси (SOAP).
- Фреймворк аутентифікації і авторизації: конфігурується інструментарій процесів аутентифікації та авторизації, що підтримує багато популярних протоколів, інструментів, практик, які стали індустріальними стандартами, через дочірній проект Spring Security (раніше відомий як Acegi

Security).

- Фреймворк віддаленого управління: налаштовувані засоби керування Java об'єктами локального або віддаленого середовища за допомогою JMX.
- Фреймворк роботи з повідомленнями: налаштовувана реєстрація об'єктів-слухачів для прозорої обробки повідомлень і черг повідомлень за допомогою JMS , покращене відправлення повідомлень за стандартом JMS API.
- Тестування: каркас, що підтримує класи для написання модульних та інтеграційних тестів.

Розглянемо ці модулі більш детально.

Контейнер Inversion of Control. Центральною частиною Spring Framework є контейнер Inversion of Control, який надає засоби конфігурування та управління об'єктами Java з допомогою зворотніх викликів. Контейнер відповідає за управління життєвим циклом об'єкту: створення об'єктів, виклик методів ініціалізації та конфігурування об'єктів шляхом зв'язування їх між собою. Зазвичай конфігурування контейнера здійснюється шляхом завантаження XML файлів, що містять опис *Bean* 'ів , інформацію необхідну для створення *bean* 'ів і опис їх зв'язків.

Об'єкти можуть бути зв'язані за допомогою *Пошуку залежностей*, або *Впровадження залежності*. *Пошук залежності (autowire)* – шаблон проектування, в якому для зв'язування виконується пошук підходящих об'єктів з визначеним типом, іменем або тих, які втілюють певний інтерфейс. *Впровадження залежності (injection)* – шаблон проектування, в якому контейнер передає екземпляри об'єктів іншим об'єктам або за допомогою конструктора , властивості, фабричного методу.

Фреймворк аспектно-орієнтованого програмування. Аспектно орієнтований фреймворк, дозволяє використовувати переваги аспектного програмування в середовищі Java. Це означає що він дає можливість додавати код в певні місця переходу (Join Point) аплікації. Наприклад виконувати кодл перед викликом певних методів, перехоплювати створення об'єктів, додавати генерований код на етапі компіляції для вирішення певних тривіальних задач (генерації *get* і *set* методів для роjo об'єктів, генерація *CRUD*

методів для контролерів і сервісів) і т.д.

Фреймворк доступу до даних. Основними задачами цього фреймворку є вирішення стандартних проблем з якими стикаються розробники при роботі з базами даних і ORM фреймворками. Підтримуються такі популярні фреймворки доступу до даних як: JDBC, iBatis, Hibernate, JDO, JPA, Oracle TopLink, Apache OJB, і Apache Cayenne. Для всіх цих фреймворків Spring надає підтримку таких можливостей як:

- Управління ресурсами – автоматичне отримання і звільнення ресурсів бази даних.
- Управління винятковими ситуаціями – переведення виняткових ситуацій які мають відношення до доступу до даних до ієрархії доступу до даних Spring фреймворку.
- Управління транзакціями
- Розгортка ресурсів – отримання об'єктів баз даних з обгортки для з'єднань.
- Абстрагування – для роботи з BLOB і CLOB типами даних.

Всі ці можливості доступні при роботі з шаблонними класами Spring фреймворку для кожного з підтримуваних фреймворків. На думку критиків, шаблонні класи не надають переваг перед використанням АПІ інших фреймворків напряду але без їх використання на розробника лягає відповідальність за управління ресурсами баз даних.

Фреймворк управління транзакціями. Управління транзакціями Spring відбувається в рамках механізму абстракції від платформи Java. Its abstraction is capable of: Це дає можливість:

- працювати з місцевими та глобальними транзакціями (локальні транзакції не вимагають наявності серверу аплікацій);
- працювати робота з вкладеними транзакціями;
- працювати з безпечними точками транзакцій;
- працюють практично у всіх середовищах на основаних платформі Java.

Для порівняння, JTA підтримує лише вкладені транзакції і глобальні транзакції а також вимагає сервера додатків (а в деяких

випадках і розгортання додатків на сервері аплікацій). Spring фреймворк надає *PlatformTransactionManager* для ряду стратегій управління транзакціями:

- Транзакції для JDBC з'єднання;
- Транзакції для ORM об'єктів;
- Управління транзакціями через JTA *TransactionManager* і *UserTransaction*;
- Транзакції для інших ресурсів, таких як об'єктні бази даних.

Також механізм фреймворку надає два способи додавання управління транзакціями ваплікації:

- Програмно, використовуючи в *TransactionTemplate*;
- за допомогою метаданих таких як XML і анотації Java 5.

Разом Spring фреймворку доступу до даних, який має свою інтеграцію з фреймворком транзакцій, надається можливість систему транзакцій через конфігурації без необхідності покладатися на JTA або EJB. Фреймворк транзакцій також інтегрується з системами кешування даних.

Фреймворк Model-view-controller. В Spring Framework входить власний MVC фреймворк, який первинно не планувався. Розробники вирішили написати свою версію через те, що вважали популярний Jakarta Struts фреймворк незручним і через відсутність кращих альтернатив. Зокрема, недостатнім був рівень відділення запиту від презентаційного рівня а також рівень відділення моделі від запиту.

Як і Struts, Spring MVC фреймворк базується на запитах. Він описує стратегії для всіх відповідальностей, які мають бути оброблені за допомогою сучасної системи обробки запитів. Головною метою інтерфейсів є простота і доступність, що зроблено для того, щоб користувачі могли легко написати свою реалізацію, якщо реалізація фреймворку їх не влаштовує. Всі інтерфейси тісно пов'язані з Servlet API і це розглядається іноді як невдача зі сторони розробників фреймворк в представленні абстракції вищого рівня для веб аплікацій. Тим не менш, цей зв'язок також дає впевненість що Servlet API є все ще доступним для розробників і може бути використаний для вирішення нетривіальних задач.

Найбільш важливі інтерфейси визначені в Spring MVC, і їх обов'язки, перераховані нижче:

- **HandlerMapping**: викликання об'єктів для обробки запитів базуючись на будь яких внутрішніх або зовнішніх параметрах.
- **HandlerAdapter**: обробка вхідних запитів.
- **Controller**: логічно знаходиться між моделлю і представленням і використовується для управління і пере направлення вхідних запитів на правильну обробку.
- **View**: відповідає за повернення клієнту відповідь (включаючи перехід до представлення повз контролер).
- **ViewResolver**: вибір представлення на основі логічної назви або місцезнаходження.
- **HandlerInterceptor**: перехоплення і додаткова специфічна обробка вхідних запитів.
- **LocaleResolver**: виначення і збереження locale (локалі – мови і приблизного місцезнаходження користувача).
- **MultipartResolver**: полегшує роботу з отриманням файлу.

Абстракції, що надаються цими інтерфейсами, є дуже високими, що дозволяє використовувати їх в безлічі варіацій. Тим не менш, розробники вільні в написанні васних імплементацій.

Легкості тестування реалізацій цих інтерфейсів з високим рівнем абстракції сприяє висока пов'язаність контролерів з Spring Inversion of Control – розробник може легко замінити імплементацію і імітувати обробку запитів аплікацією, що розробляється. Також, *DispatcherServlet* тісно пов'язаний з контейнером Spring Inversion of Control і використовує його для налаштування веб-шарів аплікації.

Висновок. Практичне застосування Spring MVC фреймворк показало, що він є надзвичайно гнучким і дозволяє легко використовувати розроблені на ньому модулі в одній або багатьох аплікаціях і не зв'язує розробника роботою над цілим ланцюжком обробки запиту. Це дає змогу розпаралелити процес розробки аплікації між декількома людьми, визначивши попередньо формат моделі і логіку перенаправлення контролюючого шару.

Література

1. <http://www.springsource.org/documentation>

ПРОБЛЕМИ ЗАХИСТУ ВІД НЕСАНКЦІОНОВАНОГО ДОСТУПУ ДО БАЗ ДАНИХ SQL-СЕРВЕРА FIREBIRD

А.В. Костенко

к.ф.-м.н., доцент, завідувач кафедри ПЗІ ЛКА

Впровадження сучасних інформаційних технологій, призвело, на жаль, до появи нового виду злочинів, таких як комп'ютерна злочинність і комп'ютерний тероризм – незаконне втручання в роботу електронно-обчислювальних машин, систем і комп'ютерних мереж, розкрадання, привласнення комп'ютерної інформації. Кібертероризм – це нова форма тероризму, яка для досягнення своїх терористичних цілей використовує комп'ютери і електронні мережі, сучасні інформаційні технології [2]. По своєму механізму, способом здійснення і приховання комп'ютерні злочини мають певну специфіку, характеризуються високим рівнем латентності і низьким рівнем розкриття.

Одною з важливих проблем є збільшення ймовірності несанкціонованого доступу до систем електронного документообігу. Для їх усунення і забезпечення належного рівня захищеності електронних документів застосовуються різні механізми і засоби забезпечення безпеки. Відповідне налаштування цих засобів залежить від технології обробки інформації, прийнятої в організації, порядку і правил поведінки з інформацією, яку потрібно захистити.

У цій статті розглянуті питання, які пов'язані з безпекою файлів бази даних SQL сервер Firebird.

Щоб отримати доступ до бази даних SQL сервера Firebird, застосуванню (користувачеві) необхідно підключитися до процесу сервера Firebird. При отриманні запиту на з'єднання процес сервера перевіряє користувача по базі даних безпеки (security database). Для відомого на цьому сервері користувача сервер дозволяє застосуванню (користувачеві) провести доступ до будь-якої бази даних, а потім сервер використовує ролі і права користувачів, визначені в самій базі даних, щоб забезпечити тонке налаштування системи доступу до об'єктів бази даних.

При підключенні до бази даних користувачеві немає необхідності мати прямий доступ до, власне, файлу бази даних. Всі

з'єднання і весь доступ відбувається через процес сервера баз даних, який виконує звернення до файлу бази даних, з метою виконати запити користувача. Сервер є тим “пристроєм”, який обмежує або дозволяє доступ зареєстрованим користувачам відповідно до прав цих користувачів, визначених для кожного конкретного користувача в кожній конкретній базі даних.

Кожний встановлений SQL сервера Firebird [1] має наперед визначеного користувача з ім'ям SYSDBA. Цей користувач є адміністратором, і має необмежений доступ до всіх баз даних, доступних даному SQL серверу. Права, визначені в конкретній базі даних, ігноруються, якщо підключення до бази даних здійснюється користувачем SYSDBA. Коли база даних вперше копіює на сервер, можливо використовувати користувача SYSDBA для налаштування привілеїв відповідно до означених на SQL сервері користувачів і вимог сервера. Це, на жаль, також означає, що кожен з користувачів комп'ютерної мережі, який має прямий доступ до файлу бази даних, без зусиль зможе скопіювати цей файл з сервера, хоча пароль користувача SYSDBA йому невідомий. Перемістивши скопійований файл на інший сервер, де пароль буде відомий, цей користувач отримає **необмежений доступ** до вмісту бази даних.

Отже, безпека SQL сервера Firebird виходить з припущення про те, що серверний процес працюватиме в достатньо безпечному оточенні. Сам SQL сервер Firebird не робить жодних заходів для забезпечення зовнішньої безпеки. Як тільки будь-хто дістає фізичний доступ до файлу бази даних, не існує способів запобігти доступу до всіх даних (у тому числі, метаданих) цього файлу.

“Достатність” безпеки визначається рівнем безпеки, необхідним і/або потрібним конкретній інсталяції (установці) SQL сервера. Цей рівень дуже сильно змінюється від інсталяції до інсталяції. Це означає, що для забезпечення певного рівня безпеки файл бази даних має бути захищений на рівні операційної системи. В більшості випадків процес сервера Firebird повинен працювати з правами окремого користувача (обліковому запису) операційної системи, і лише цей обліковий запис (і ще, напевно, адміністратор операційної системи) повинен мати прямий доступ на рівні операційної системи до файлів баз даних. Файли бази даних ніколи не слід розташовувати в папках, до яких відкритий доступ з мережі або доступ будь-яких локальних користувачів, крім спеціально обумовленого персоналу.

Для ефективності заходів безпеки, також дуже бажано, щоб серверне устаткування було встановлене в місці, куди обмежений фізичний доступ сторонніх осіб. Таке обмеження не допустить несанкціонований доступ до жорстких дисків сервера.

Приведені обмеження не дуже корисні розробникам, які створюють бази даних для розповсюдження і установки на віддалених серверах, і які хотіли б захистити свою інтелектуальну власність. Така власність може включати:

1. метадані;
 - структури таблиць;
 - початкові і скопійовані тексти процедур, що зберігаються;
 - початкові і скопійовані тексти тригерів;
2. деякі дані, що містяться в таблицях.

Розробник створює базу даних та відповідне клієнтське програмне забезпечення для установки на віддалених серверах і клієнтах. Звично сервери обслуговуються людьми, деякі з них мають повний доступ до системи, де працює SQL сервер Firebird і де знаходяться бази даних, – ці люди здійснюють резервне копіювання баз даних і інші процедури обслуговування. Як описано в попередньому розділі, прямий доступ до файлу бази даних забезпечує можливість повного і необмеженого доступу до інформації, що міститься в базі даних, – метаданим і даним.

У таких випадках розробник не може мати впевненості у безпеці свої інтелектуальної власності, оскільки:

- користувачі здатні скористатися методом зворотного інжинірингу бази даних, а потім скористатися результатами в своїх цілях;
- середовище роботи сервера (обладнання і програм) не забезпечуватиме потрібний рівень безпеки, що може привести до неавторизованого доступу до бази даних.

Проблеми які найбільше хвилюють розробників:

- захист дизайну бази даних (структури таблиць, процедури, що зберігаються, тригери і так далі) від доступу всіх користувачів бази даних на віддалених, не контрольованих розробником, серверах;
- заборона доступу користувачів віддалених, не

контрольованих розробником, серверах до даних певних таблиць. Ці таблиці містять інформацію, яка використовується виключно для внутрішнього використання застосуванням.

Оскільки SQL сервер Firebird не містить вбудованих механізмів шифрування, то інтуїтивною відповіддю на обидва питання – наявними засобами SQL сервера реалізація побажань неможлива. Користувач, що дістав доступ до файлу бази даних, дістає доступ до всіх даних цього файлу.

У першому випадку, не існує “обхідних шляхів”, тому що серверний процес повинен мати доступ до файлу бази даних і метаданих. У другому випадку, можливо на клієнтській стороні реалізувати механізм шифрування/дешифровки даних, але у такому разі у пропадає можливість додавати індекси і використовувати можливості вбудованого пошуку.

Існує єдиний спосіб вирішити вказані раніше проблеми: розмістити бази даних і SQL сервер на власних серверах, і надайте до них доступ клієнтам, наприклад, через Інтернет і тощо. Можливості термінальних серверів (terminal servers) (Windows або Linux/Unix) так само можна використовувати для реалізації таких вимог.

Таким чином, у розробника є контроль над базою даних, і він може обмежувати доступ до різних функцій і структур баз даних, використовуючи наявні в SQL сервері Firebird методи забезпечення безпеки (ролі, привілеї і так далі).

Різні бібліотеки доступу до бази даних звертаються до метаданих: первинні ключі, домени (domain), що значно полегшує процес розробки клієнтських застосувань. Це підкреслює той факт, що розробник не може запобігти доступу користувачів до метаданих без припинення надання вказаним застосування інформації, яка їм необхідна.

Література

1. Бори Х. Firebird: руководство разработчика баз данных: Пер. с англ. – СПб.: БХВ-Петербург, 2006. – 1104 с.
2. Травников Ю. Преступления в Паутине: Границы без замков /Ю. Травников, <http://www.pl-computers.ru/article>, 2003.

ПРОБЛЕМА ЗБОРУ СТАТИСТИЧНОЇ ІНФОРМАЦІЇ ПРИ МОНІТОРИНГУ ОБЧИСЛЮВАЛЬНОГО КЛАСТЕРА

О.М. Томашевський

к.т.н., доцент, завідувач кафедри МКД ЛФ ЄУ,

Розглянуто специфіку збереження результатів моніторингу завантаженості обчислювального кластера в залежності від часового проміжку засобом Nagios. Запропоновано використання засобів автоматичного розподіленого збереження даних моніторингу шляхом створення додаткового плагіна для Nagios або застосування можливостей сервісу збереження статистичної інформації collectd.

Моніторинг робочих станцій (PC) обчислювального кластера є необхідним для визначення **максимальної пропускної здатності** ресурсів останнього, планування та визначення параметрів навантаження на систему, що обумовлює ефективне використання кластера для вирішення розподілених задач. Якість спостережень за основними характеристиками PC, такими як завантаженість RAM Usage (Random-Access Memory, оперативної пам'яті), центрального процесора CPU (Central Processing Unit) та системи в цілому System Load, вхідного-вихідного трафіка IO (Input-Output), безпосередньо впливає на результати моніторингу. Проте, навіть при рівномірних зчитуваннях показників завантаженості системи, з достатніми для досягнення певної точності часовими інтервалами, аналіз результатів може ускладнитись невідповідними представленням або збереженням отриманих даних.

Сучасні засоби для моніторингу комп'ютерних систем та мереж представлені рядом програмних засобів: як комерційних – Solar Winds, 3COM Network Supervisor, HP OpenView і т.д.; так із відкритим кодом – наприклад Nagios [2], Big Brother, Angel Network Monitor, HiWayS, MARS, Autostatus, NocMonitor, RITW, Zabbix, NetXMS, OpenNMS тощо. Засоби з відкритим кодом (Open Source), на відміну від комерційних додатків, є безкоштовними і компенсують зазвичай меншу функціональність можливістю необмеженого розширення властивостей шляхом створення модулів розширень (скриптів, плагінів). Розглянемо **специфіку збереження результатів** моніторингу завантаженості PC обчислювального кластера в залежності від часового проміжку засобом Nagios.

Програму Nagios, відому раніше як Netsaint, було розроблено Етаном Галстадом (Ethan Galstad), що підтримує та розвиває систему сьогодні, спільно з командою розробників, які випускають офіційні і неофіційні плагіни. Спочатку Nagios призначалась для роботи під GNU/Linux, але на сьогоднішній день вона також добре працює і під іншими операційними системами, такими як Sun Solaris, AIX і HP-UX.

До можливостей Nagios входять [3]:

- моніторинг сіткових служб (SMTP, POP3, HTTP, NNTP, ICMP, SNMP);
- моніторинг стану хостів (завантаження процесора, використання диску, системні логи);
- підтримка віддаленого моніторингу через шифровані тунелі SSH або SSL;
- проста архітектура модулів розширень (плагінів), що дозволяє використовувати будь-яку мову програмування за вибором Shell, C, Perl, Python, PHP, C # та інші;
- паралельна перевірка сервісів;
- можливість визначати ієрархію хостів мережі, виявляти та розрізняти несправні і недоступні;
- надсилання повідомлення у разі виникнення проблем із сервісом або хостом (за допомогою пошти, пейджера, смс, або іншим способом, що визначається користувачем через модуль системи);
- можливість визначення обробника подій, що відбулись із сервісами або хостами для проактивного вирішення проблем;
- автоматична ротація лог-файлів;
- можливість організації спільної роботи декількох систем моніторингу з метою підвищення надійності та створення розподіленої системи спостереження;
- утиліта nagiosstats, яка формує загальний звіт за всіма хостами, по яких проводиться моніторинг.

При створенні модуля розширення моніторингу навантаження РС обчислювального кластера для Nagios вказується, які саме показники потрібно знімати (RAM Usage, CPU, System Load, IO), частоту виконання спостережень, відображення результатів – таблицно або графічно. Графічне відображення результатів

моніторингу представляється найбільш доцільним внаслідок кращого візуального сприйняття у порівнянні з табличними даними (рис.1). Проте, з часом результати спостережень, для оптимального використання комп'ютерної пам'яті, автоматично піддаються усередненню 100:1 [1].

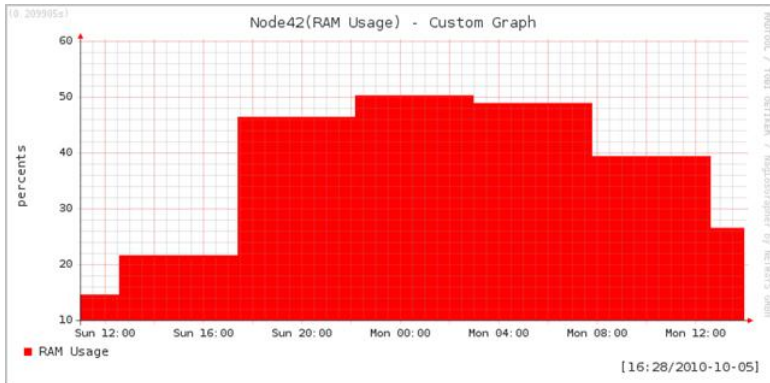


Рис.1. Результат моніторингу завантаженості оперативної пам'яті PC обчислювального кластера з періодом 4 год. (засіб Nagios)

При використанні усереднених значень отриманих результатів моніторингу неможливо прослідкувати реальну динаміку навантаження (рис.2) та зробити висновок про критичну межу запущених у системі задач.

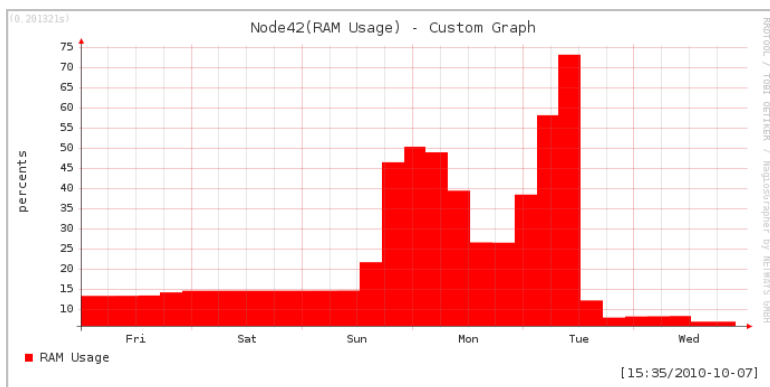


Рис.2. Усереднений результат моніторингу завантаженості оперативної пам'яті PC обчислювального кластера (засіб Nagios)

Рішенням розглянутої проблеми може бути автоматичне розподілене збереження всіх результатів моніторингу – додавання у плагін контролю навантаження процедури копіювання вмісту вихідної таблиці із заданим часовим інтервалом, що виключає процес усереднення, на спеціально призначений жорсткий диск – даної РС або серверу. При цьому необхідно врахувати додаткову завантаженість системи та CPU під час проведення копіювання, що може вплинути на результати моніторингу системи загалом, і внести потрібні модифікації.

Інше вирішення зазначеної вище проблеми полягає у об'єднанні можливостей **Nagios** як системи моніторингу і нотифікації та **collected** як сервісу збору статистичної інформації у комп'ютерній системі [4].

Collected – одна з програм з відкритим кодом, що базується на оперуванні Round Robin Database (RRD), так звану “кільцевою” базою даних (БД), архітектура якої передбачає постійне оновлення даних та їх збереження при сталому обсязі самої БД. Програму написано на С для підвищення продуктивності та портативності, що дозволяє запускати її на системах без мови сценаріїв чи daemon stop, таких як вмонтовані системи. У той же час вона включає в себе можливості оптимізації та функції для обробки сотень тисяч наборів даних. На рис.3 представлено результат роботи **collected**.

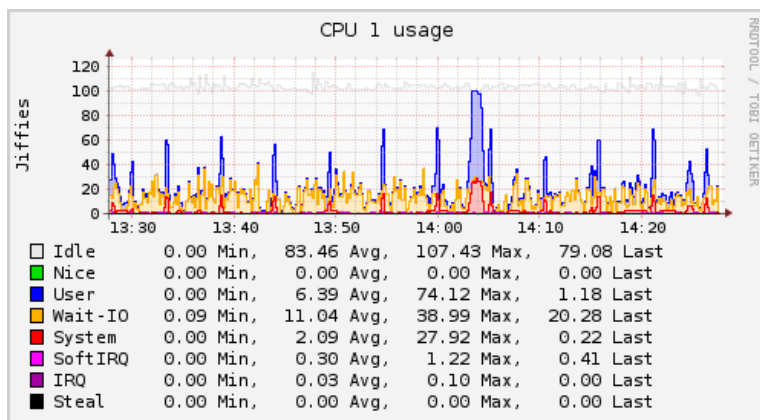


Рис.3. Дані завантаженості оперативної пам'яті РС обчислювального кластера за 60 хв. (засіб collected)

Для роботи із засобами RRD спочатку створюється кільцева БД, в яку будуть записуватись дані про стан заданих елементів обчислювального кластера, наприклад, CPU. Зазначимо, що внаслідок природної обмеженості дискового простору в будь-якій базі може зберігатись лише певна кількість даних. Тобто знову виникає необхідність в усередненні. Проте, механізм RRD дозволяє записувати в одну клітинку даних декілька показників, перетворюючи їх за допомогою статистичної функції [4]. У кожному RRD при такому підході записується декілька рядів клітинок, що беруть показники з одного джерела, але різним способом комбінуючи дані, що надходять, що забезпечує збереження даних за тривалий період моніторингу з мінімально можливим усередненням.

Література

1. Wolfgang Barth. Nagios. System and Network Monitoring. 2nd Edition. – San Francisco: “No Starch Press”, 2008. – 719 с.
2. <http://www.nagios.org/>
3. <http://en.wikipedia.org/wiki/Nagios>
4. <http://collectd.org/>

ОПТИЧНІ ПАРАМЕТРИ СВІТЛОВИПРОМІНЮЮЧОГО ЕЛЕМЕНТА ДЛЯ СТВОРЕННЯ ПРИСТРОЇВ ВІДОБРАЖЕННЯ ІНФОРМАЦІЇ НА ЙОГО ОСНОВІ

¹М.Р. Гладун, ²Д.Ю. Волинюк, ³С. Михайлівський

¹кафедра математики і комп'ютерних дисциплін, ЛФ ЄУ
^{2,3}ІТРЕ, кафедра ЕП НУ “Львівська політехніка”

В роботі наведені умови створення світловипромінюючого елемента на основі органічних матеріалів з випромінюванням в синій області спектру видимого діапазону оптичного випромінювання. Представлений спектр електролюмінесцентного випромінювання та кольорова діаграма даного елемента. Також проведені дослідження напрямленості світлення запропонованого елемента. На основі отриманих оптичних параметрів світловипромінюючого елемента запропоновано його використання в якості комірки синього кольору при створенні дисплеїв.

Вступ. За останні роки з використанням органічних матеріалів

у технології світлодіодів стає можливим створення пристроїв відображення інформації та освітлення різноманітних конфігурацій на гнучких та прозорих основах [1,2]. Основними параметрами органічних світловипромінювальних діодів (OLED) є спектр випромінювання, яскравість, напруга живлення, довговічність, тощо. Однією з базових структур органічного світлодіода є сандвіч-структура, що являє собою два електроди (катод і прозорий анод), між якими міститься активний електролюмінісцентний органічний матеріал. У цей активний органічний шар з відповідних електродів інжектуються дірки та електрони, де рекомбінують з подальшим випромінюванням світла. Для інжекції електронів використовують метали з малою роботою виходу (Ca, Mg, Al), для інжекції дірок – високопровідний ITO ($\text{In}_2\text{O}_3:\text{SnO}_2$). Люмінесцентний шар може бути, наприклад, з низькомолекулярного матеріалу (Alq_3), або зі спряженого полімеру (PPV) [3,4]. У спрощеній структурі OLED міститься лише один органічний шар, що поєднує в собі інжекційну, транспортну та випромінювальну функції. Це можливо, якщо органічний матеріал має збалансовану електронну та діркову рухливість. Однак така структура характеризується високою (понад 10 В) напругою живлення та нестабільністю викликаною взаємодією матеріалу катода з матеріалом органічного шару.

Метою нашої роботи була розробка світлодіодної структури з використанням дірково – транспортного шару йодиду міді (CuI) для створення OLED з максимумом випромінювання в області ~ 400 нм.

Експеримент. Дослідження проводилися на основі створеної багатошарової структури ITO/CuI/карбазол/Ca/Al методом термовакuumного напилення в одному технологічному циклі (рис.1).

На скляній підкладці з електропровідним оптично прозорим покриттям ITO у вакуумі $<10^{-5}$ Торр з молібденового тигля при температурі $\sim 400^\circ\text{C}$ і швидкістю осадження 0.2 нм/с осаджували плівку CuI, товщина якої становила ~ 10

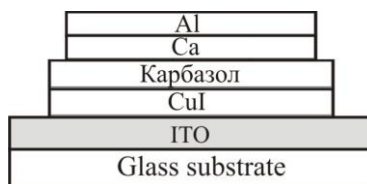


Рис.1. Схема розробленого елементу ITO/CuI/карбазол/Ca/Al

нм. На поверхню плівки CuI при температурі $\sim 250^\circ\text{C}$ зі швидкістю 0.5-1 нм/с наносили плівку карбазолу завтовшки 50 нм. Потім формувалась плівка металічного кальцію, поверх якої формувався

шар алюмінію товщиною ~ 200 нм.

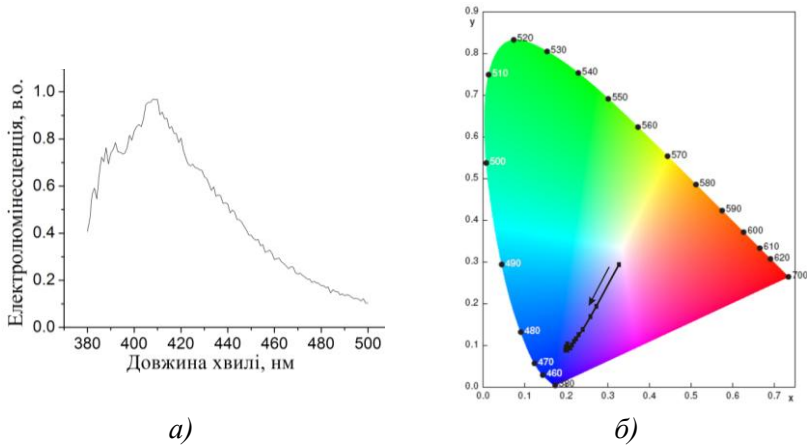


Рис.2. Спектр електролюмінесценції (а) та кольорова діаграма (б) елементу ІТО/CuI/карбазол/Ca/Al

На рис. 2 а зображено спектр електролюмінесценції, з якого видно, що максимум свідчення розробленого елементу рівний 410 нм. Колір свідчення структури ІТО/CuI/карбазол/Ca/Al також відмічений на рис. 2 б, де стрілкою відмічено напрямок зміни кольору свідчення зразка в залежності від росту величини живлення комірки. Координата кольору $x=0.22$ та $y=0.09$ дасть змогу відтворити достатню кількість кольорів пристрою відображення на основі розробленого світловипромінюючого елементу.

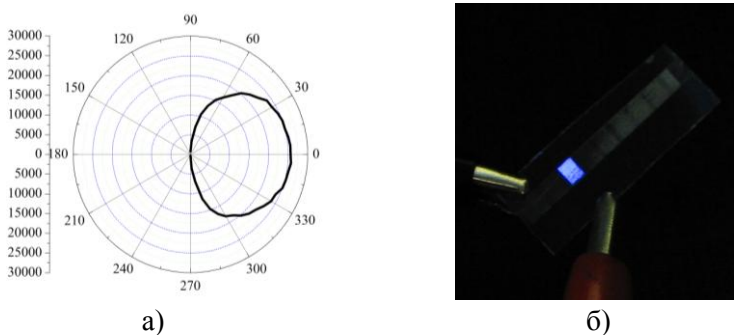


Рис.3. Індикатриса розсіювання (а) та світлина (б) електролюмінісцентного елементу ІТО/CuI/карбазол/Ca/Al

Важливою характеристикою для дисплеїв є також кут огляду, який залежить від індикатриси розсіювання кожного елемента зображення. На рис.3 *a* представлена індикатриса розсіювання елементу *ITO/CuI/карбазол/Ca/Al* з даної залежності можна визначити, що максимальний кут огляду становить 145°. Візуальне зображення досліджуваного зразка зображене на рис. 3 *б*.

Висновок. В роботі представлені оптичні параметри створеного світловипромінюючого елемента на основі органічних матеріалів. Визначений максимум свічення розробленого елемента 410 нм, координати кольору та максимальний кут огляду 145°.

Література

1. J.Kido. Organic displays// Phys.World.- 1999.-Vol.12 (3).-P.27.
2. В. Майская. Органические светодиоды // Электроника: Наука, Технология, Бизнес.- 2007.-№5.
3. J.Shinar. Organic light-emitting devices// Springer.-2004.
4. Кухто А.В. Электролюминесценция тонких пленок органических соединений // Журнал прикладной спектроскопии.-2003.-Т.70, №2.- С.151.

ФІШИНГАТАКИ, ЇХ ВИДИ ТА ХАРАКТЕРИСТИКА

Х.О. Засадна

к.ф.-м.н., доцент, доцент каф. комп'ютерних технологій ЛІБС УБС НБУ

У статті описано види фішинг-атак, які базуються на використанні сучасних засобів зв'язку в поєднанні з методами соціальної інженерії. Розкрито суть механізмів захисту від фішингу та фармінгу. Перелічено профілактичні заходи безпеки при використанні електронної пошти, мережі Інтернет, банківських послуг, послуг електронної комерції.

Фішинг – це різновид соціальної інженерії, спрямований на злодійське отримання конфіденційної інформації особи. Соціальна інженерія – це спосіб управління діями людини без використання технічних засобів. Методи соціальної інженерії базуються на використанні слабкостей людини. Сьогодні соціальну інженерію часто використовують в мережі інтернет для отримання закритої (таємної) інформації або інформації, яка має високу вартість. Такою

інформацією є, наприклад, пароль для входження в закриту комп'ютерну систему. Якщо його дізнатись, можна скопіювати з неї таємну інформацію, при цьому не треба шукати слабкі місця в системі захисту чи вгадувати пароль.

Фішинг – вид інтернет-шахрайства, мета якого – *отримати конфіденційні дані користувача*. До них відносять логічні імена і паролі, номери кредитних карт, банківських рахунків та іншу приватну інформацію. Існує кілька тактик виманювання інформації, які використовують масові повідомлення електронної пошти від імені популярних брендів; банків, платіжних систем, інтернет-сервісів, соціальних мереж, благодійних організацій, організацій, що займаються електронною комерцією.

Мішенню фішерів найчастіше є клієнти банків та електронних платіжних систем.

Класичний фішинг полягає в тому, що зловмисник розповсюджує електронною поштою підроблений під офіційний лист (“закидає вудочку”), в якому вимагається “перевірка” певної інформації або виконання визначених дій. Цей лист, як правило, містить вказівку на фальшиву веб-сторінку, яка імітує справжню (відкрити її – значить “проковтнути наживку”). Фальшива веб-сторінка має такий же зовнішній вигляд, як і справжня, містить логотип банку (чи платіжної системи) і форму, у яку вимагається ввести конфіденційну інформацію – від логічного імені та паролю до домашньої адреси та пін-коду банківської карти. У схемі класичного фішингу все залежить від того, “клієне” користувач “наживку” чи ні, адже фішер переконуватиме жертву добровільно виконати низку дій. Увійшовши на сайт зловмисника, користувач може повідомити іншу цінну інформацію, яка дозволить отримати доступ до аккаунта (облікового запису користувача, який ідентифікує його у деякій комп'ютерній системі).

Крім прямої вказівки на віртуальний сайт у листі зловмисника може міститися вказівка на сайт з перенаправленням (редиректом). Перенаправлення URL-адреси – технологія, яка використовується у всесвітній павутині WWW для того, щоб веб-сторінка була доступна під кількома URL-адресами. Для веб-майстра, який відповідає за зовнішній вигляд і роботу веб-сайта, домена, веб-сервера та його веб-сторінок є кілька причин, за якими він може використовувати перенаправлення URL. Одна з них, коли користувач браузера уводить URL-адресу з помилкою. Провайдери часто реєструють такі

домени “з помилкою” і перенаправляють їх на “правильні” адреси. Механізм прихованого перенаправлення користувачів на фішингові сайти отримав назву фармінгу.

Фармінг – прихований різновид фішингу – важче виявити, ніж фішинг. Автоматичне перенаправлення на фальшивий веб-сайт у той момент, коли користувач пробує увійти на офіційний веб-сайт фінансової чи комерційної організації є непомітним, а участь користувача зводиться до мінімуму – достатньо відвідати сайти, що цікавлять зловмисника. Реалізують фармінг-атаки спеціальні шкідливі програми, які використовують для прихованого перенаправлення зміну інформації DNS на комп’ютері користувача або сервері провайдера.

Листи, які ніби відправлені з банку, можуть повідомляти користувача про необхідність подзвонити за певним номером для вирішення проблем з його банківськими рахунками. Ця техніка називається **вішинг** (голосовий фішинг). Цей різновид фішингу полягає у використанні автонабирача і можливостей інтернет-телефонії. Схема обману є такою: клієнти банку чи платіжної системи отримують електронною поштою повідомлення від її адміністрації або служби безпеки. У повідомленні вказується, що з вашого банківського або карткового рахунку було знято велику суму готівкових коштів або що інформація про ваші рахунки потрапила до шахраїв, і просять негайно подзвонити на певний номер телефону. Користувач прослуховує інструкції автовідповідача, який вказує на необхідність увести на клавіатурі телефону номер свого рахунку і пароль доступу до нього або номер кредитної картки та її PIN-код. Вішери можуть особисто дзвонити жертвам, переконуючи їх, що вони є представниками офіційних організацій.

SMS-фішинг називається **смішинг**. Зловмисники розсилають повідомлення, що містять вказівку на фішинговий сайт. Відкриваючи його і уводячи свої особисті дані, жертва передає їх зловмисникам. У повідомленні також можуть пропонувати негайно подзвонити для вирішення несподіваних проблем.

Більш ніж 70 % фішингових атак є успішними. Використовуючи ефект несподіванки, фішери своїми діями намагаються створити критичну ситуацію, стривожити людину, налякати її, очікують миттєвої необдуманого реакції. Наприклад, електронний лист містить заголовки “Щоб відновити доступ до свого банківського рахунку...”, “Ваш картковий рахунок буде

заблоковано, якщо Ви негайно не підтвердите такі дані...”, “На ваш рахунок в банку було вчинено фішинг-атаку...”, “Терміново! Ваші реквізити можуть бути вкрадені...”, “З метою перевірки Вашого балансу просимо негайно надати таку інформацію...”. Це стимулює до виконання наступних вказівок зловмисників.

Фішинг базується на незнанні користувачем основ безпеки комп’ютерних мереж: банки, платіжні системи, сервісні системи ніколи не розсилають листів з проханням повідомити свої облікові дані, паролі та іншу конфіденційну інформацію.

Існують різні методи для боротьби з фішингом, в тому числі законодавчі міри і спеціальні технології. Банки, веб-служби, ЗМІ часто попереджують про різноманітні способи дій фішєрів з використанням методів соціальної інженерії. У відповідь на лист з проханням підтвердити обліковий запис чи інші конфіденційні дані треба зв’язатися з організацією, від імені якої відправлене повідомлення, для підтвердження його істинності. Не слід використовувати наведені номери телефонів і гіпервказівки у сумнівних повідомленнях, URL-адреси краще вводити самостійно в рядок адрес браузера. Практично всі повідомлення організацій містять якусь конкретну особисту інформацію, недоступну для фішєрів. Наприклад, звертаються до адресатів за іменем і по батькові, а не “Шановний клієнт банку...”. Хоча її наявність не гарантує безпеку повідомлення.

Важливим методом боротьби з фішингом є створення списку фішингових сайтів і порівняння адрес. У 2006 році з’явилася методика використання спеціальних DNS-сервісів, які фільтрують відомі фішингові адреси – цей метод працює у всіх браузерах.

Зменшити кількість фішингових електронних повідомлень можуть спеціальні спам-фільтри для програм електронної пошти.

Є фірми, що пропонують банкам та іншим організаціям послуги круглодобового контролю, аналізу і допомоги у закритті фішингових сайтів

Використання ефективних засобів антивірусного захисту дозволяє попередити проникнення вірусів типу “троянський кінь” на комп’ютер. Фішєри використовують троянські програми не лише для редирекції, але й для збору конфіденційної інформації. *Троянські кейлогери* – спеціальні програмні продукти або апаратні пристрої – реєструють кожне натискання клавіші клавіатури комп’ютера. Кейлогери вбудовують у стандартні повідомлення

формату HTML, впливаючі вікна на веб-сторінках, мультимедійні об'єкти. Тобто фішери використовують у веб-сайтах технологію Flash: зовні віртуальний сайт виглядає як справжній, але текст зловмисника захований в графічних об'єктах, які часто закривають рядок адрес і відкривають іншу – фальшиву – веб-сторінку.

Сьогодні найбільш розповсюдженими методами крадіжки конфіденційної інформації є фармінг і різні шкідливі програми. Спосіб стопроцентного захисту від фармінг-атак не існує, тому необхідно використовувати профілактичні заходи безпеки:

- використовувати і регулярно оновлювати сучасне ліцензійне антивірусне програмне забезпечення, антивірусні компанії вбудовують у свої продукти так звані антифішингові фільтри;
- використовувати захист електронної поштової скриньки, відключивши попередній перегляд повідомлень;
- не відкривати вкладення електронних листів від незнайомих і сумнівних адресатів;
- не відкривати спамові листи і не відповідати на них;
- використовувати відомого, офіційного Інтернет-провайдера;
- перевіряти URL-адресу веб-сайту, який запрошує конфіденційну інформацію;
- перевіряти, чи сеанс зв'язку почався з правильної адреси веб-сайту, чи не додані ніякі зайві символи до адреси URL. Якщо веб-сайт розміщений на захищеному сервері, то адреса повинна починатися з імені протоколу `https://` ("s" від security), а не зі звичайного `http://`. На такому сайті можна перевірити стан іконки з піктограмою "Замок" в рядку стану браузера (поводити курсором миші над цією піктограмою);
- перевіряти сертифікати веб-сайтів. Для цього в меню "Файл" або в меню правої кнопки миші вікна браузера вибирають команду "Властивості", натискають кнопку "Сертифікати" і перевіряють, чи є у веб-сайта діючий сертифікат, виданий офіційною організацією;
- щомісячно перевіряти виписки з банківських рахунків і кредитних карток, а у випадку виявлення підозрілої трансакції негайно зв'язатися з банком;
- використовувати послугу SMS-банкінг;

- контролювати рахунок за допомогою лімітування сум операцій – встановити суму гранично можливого зняття готівки або платежу в торгових точках;
- завести окрему банківську картку для інтернет-платежів, тримати на ній невеликі суми і поповнювати лише перед виконанням транзакцій;
- ознайомитися з інформацією про проведення транзакцій в захищеному режимі на веб-сайті банку;
- використовувати безплатні програми захисту від фішинг-атак, які є в інтернет-середовищі.

Сервіси, спрямовані на боротьбу з фішингом називаються PhishTank і GeoTrust.

Література

1. Tom Jagatic, Nathaniel Johnson, Markus Jakobsson, Filippo Menczer Social Phishing.
2. Количество фишинг-атак возросло на 40%. Astera (17 апреля 2009).
3. Предотвратите попадание нежелательной почты в папку “Входящие”. Microsoft (26 января 2007).
4. <http://ru.wikipedia.org/wiki/>.

ЗАХИСТ ІНТЕЛЕКТУАЛЬНОЇ ВЛАСНОСТІ (ВИНАХОДІВ ТА КОРИСНИХ МОДЕЛЕЙ) В ГОСПОДАРСЬКИХ СУДАХ УКРАЇНИ

Ю.І. Макогон

прокурор відділу Львівської обласної прокуратури

Інтелектуальні системи, інноваційні проекти та нанотехнології в Україні не знаходять правового захисту в господарських судах та стають здобичею іноземних компаній. Господарський кодекс та Господарсько – процесуальний кодекс не захищають власників винаходів та корисних моделей. Економічна система України від цього має мільярдні збитки та ізольована від експорту, хоч вступила в Світову систему торгівлі.

Згідно зі статтею 54 Конституції України право власності на винахід гарантується. П'ятий розділ Цивільного кодексу України

регулює захист на об'єкти промислової власності. Статтями 13 та 41 Закону України “Про власність” об'єкти промислової власності у нашій країні є приватною та належить підприємцеві, якщо він для її створення вклав інвестиції, уклав договір із винахідником (автором) на таку наукову діяльність. Авторство на винахід охороняється безстроково.

Згідно ст. 1 Земельного кодексу України передбачено, що земля є основним національним багатством, яке перебуває під особливим захистом держави. Право власності на землю гарантується.

Слід зазначити, що, згідно з чинним законодавством, авторство на створену інтелектуальну власність є безстроковим та немайновим трудовим інтелектуальним ресурсом і його охороняє закон. Автор промислової власності має право на авторську винагороду, розмір якої не обмежується законом.

Роботодавці, як приватні власники, завжди зацікавлені в тому, щоб виготовляти конкурентноспроможні товари та продовольство та надавати медичні послуги громадянам, послуги побутові, технологічні споживачам та отримувати прибутки. Регулятором між авторами та приватними сільськогосподарськими, промисловими, медичними та продовольчими виробниками є цивільне законодавство, економічний закон попиту та пропозиції, конкуренція між ними за монополіні ціни та високі прибутки, які можна досягти тільки виготовленням виробів, що захищені охоронними документами. В такій конкурентній боротьбі постійно виникають господарські спори, які можливо вирішувати через мирову угоду або через позов до господарського суд. Такі суди створені у обласних центрах, а апеляційні суди діють розташовані в окремих регіонах.

В умовах ринкових відносин створення сучасних технологій та ліцензійного обміну товарами та продовольством, медичними послугами виконуються тільки через регулювання і активізацію об'єктів промислової власності. Це тільки початкова стадія, а, точніше – передумова, оскільки головним фактором у досягненні успіхів та отриманні максимального прибутку є наявність інвестиції. Інвестор завжди бажає одержати прибуток та гарантії, що його фінансові ресурси будуть захищені через судову систему. Інвестора також цікавлять наявні патенти на винаходи та корисні моделі, які дають гарантію на 10-20 років для монополістичної конкуренції.

Це правове та економічне завдання значно ускладнюється в умовах переходу від командно-адміністративної системи до ринкової, в якій малий та середній бізнес, фермерство ще не розвинуті, прибутки незначні та не перетворюються в інвестиції, які можна було залучати у сільськогосподарське та промислове виробництво для масового виготовлення конкурентоспроможних продовольства з використанням національних трудових інтелектуальних ресурсів. Господарські суди та державні органи, неспроможні захистити власників патентів на винаходи, надати правову підтримку не тільки авторам – винахідникам, ученим, конструкторам та венчурним компаніям для початкового створення масового виробництва продуктів та їх повного й всебічного захисту в Україні та за її межами, в Європі [113, 136]. Частина науковців та винахідників припиняє свою інтелектуальну діяльність, емігрує до інших країн, а частина переходить на роботу в торгівлю та домашнє сільське господарство.

Як було зазначено раніше, кількість зареєстрованих винаходів, корисних моделей та промислових зразків порівняно з іншими країнами значно зменшилась. Так, наприклад, Україна в рік подає на реєстрацію до 200 винаходів у Російській Федерації та до двох десятків у європейських країнах, тоді, як Німеччина патентує в інших країнах до 40 тисяч винаходів на рік. В Україні у виробництві сільськогосподарських товарів та продовольства появляється біля десятка модернізованих продуктів в рік, а в Німеччині появляється до двох тисяч на рік, які захищені патентами на винаходи. Якщо українське сільське господарство експортує до інших країн переважно сировину та продукцію землеробства, інколи навіть за демпінговими цінами, то Німеччина експортує 60 відсотків своєї продукції як кінцевий товар.

Як бачимо тепер інтелектуальні ресурси мають прямий вплив на розвиток економіки будь – якої країни, вони є джерелом зростання продуктивності праці, нагромадження чистих прибутків та багатства, яке є базою для капіталовкладення або інвестування. Але є можливі тоді, коли господарські суди спосібні захистити фізичних осіб – підприємців, власників охоронних документів від інших фірм, які не володіють ліцензіями, як це передбачено Законом України “Про охорону прав на винаходи і корисні моделі” Природа інвестицій походить винаходів та корисних моделей технологічного призначення, від діяльності корпорацій, які володіють тисячами

винаходами та від залучення вільного капіталу населення, торгових фірм. Недосконалість господарського законодавства та відсутність досвіду судової системи, в умовах перехідного періоду, а також вільні кошти населення, що їх воно отримало як трансферні платежі, накопичуються у комерційних банках на депозитних рахунках або перебувають у поза банківській системі в очікуванні вигідного та безпечного застосування для отримання прибутків. Ці кошти вкладаються у придбання нерухомості, навчання дітей, поліпшення медичного обслуговування. Але їх рідко вкладають в запатентовані інвестиційні проекти, які мають найвищий ризик невдач. Іноді цей ризик досягає 50-70 відсотків. Уклавши вільні кошти на депозитний рахунок у комерційному банку, цей ризик становить до 20 відсотків. Невеличкі банки завжди більш ризиковані. Вони завжди пропонують клієнтам – населенню – високі процентні ставки за зберігання коштів на депозитних рахунках. Інтелектуальні системи, інноваційні проекти та нанотехнології в Україні не знаходять правового захисту в господарських судах та стають здобиччю іноземних компаній. Господарський кодекс та Господарсько – процесуальний кодекс не захищають власників винаходів та корисних моделей. Економічна система України від цього має мільярдні збитки та ізольована від експорту, хоч вступила в Світову систему торгівлі.

GOOGLE WEB TOOLKIT (GWT) – ПЛАТФОРМА ПРОГРАМУВАННЯ ДЛЯ СТВОРЕННЯ ВЕБ-УЖИТКІВ НА ОСНОВІ ТЕХНОЛОГІЇ AJAX

А. Батюк, В. Голощанов, С. Пилипчук

*кафедра автоматизованих систем управління, Національний університет
“Львівська політехніка”*

This article contains nutshell review of project development using GWT.

Вступ. Ця стаття містить опис процесу розробки веб-ужитків на мові Java, використовуючи фреймворк Google Web Toolkit (GWT).

Google Web Toolkit (GWT) – це безкоштовний Java

фреймворк, який дозволяє веб-розробникам створювати Ajax ужитки на основі Java. Випускається під ліцензією Apache версії 2.0. GWT робить акцент на повторне використання та кросбраузерну сумісність.

Використовуючи GWT, розробники швидко можуть писати і відлагоджувати AJAX додатки на мові Java, використовуючи інструментарій налагодження Java. Компілятор GWT перекладає код Java додатку у відповідний браузеру JavaScript і HTML.

Утиліта командного рядка applicationCreator, що поставляється разом з GWT, автоматично створює всі файли, потрібні для нового GWT-проекту. Вона також дозволяє створювати файли проекту Eclipse.

Існує модуль Google Plugin для IDE Eclipse версій 3.3 – 3.5, що дозволяє спростити процеси створення GWT-проекту і розміщення готових додатків на сервісі Google App Engine.

GWT використовується в багатьох продуктах компанії Google, включаючи Google Wave і Google AdWords. GWT – система з відкритим вихідним кодом, абсолютно безкоштовна, і використовується тисячами розробників зі всього світу. Її мета полягає у забезпеченні продуктивної розробки високошвидкісних веб-ужитків без необхідності, щоб розробник був експертом в особливостях різних браузерів, XMLHttpRequest і JavaScript.

Набір GWT складається з:

- **SDK**

GWT SDK містить бібліотеки Java API, компілятор та сервер для розробки. Це дозволяє писати клієнтські програми на Java і запускати їх на сервері як JavaScript.

- **Плагін для Eclipse**

Плагін для Eclipse забезпечує IDE підтримку Google Web Toolkit і App Engine.

- **Speed Tracer**

Speed Tracer – додаток до Chrome, який дозволяє точно визначити проблеми продуктивності у ваших веб-додатках.

Написання програм за допомогою Google Web Toolkit. GWT SDK надає множину Java API і віджети. Це дозволяє писати AJAX додатки на Java, а потім компілювати код у високооптимізований JavaScript, що працює у всіх браузерах, включаючи браузери для мобільних Android і iPhone.

Побудова програм AJAX таким чином є більш ефективна завдяки високому рівню абстракції над такими загальними поняттями, як DOM маніпуляцій і XHR комунікація

Також ви не обмежені наперед визначеними віджетами. Все, що ви могли зробити з DOM і JavaScript може бути зроблено і в GWT, в тому числі, що взаємодія з рукописним JavaScript.

Відлагодження. Ви можете відлагоджувати програми AJAX у вашому улюбленому IDE, як і настільні додатки, і у вашому улюбленому браузері, так як ви б писали JavaScript. Плагін GWT для розробки заповнює відстань між байт-кодом Java в відлагоджувачі і JavaScript в браузері.

Завдяки плагіну GWT, непотрібною стає компіляція коду у JavaScript, щоб побачити результат роботи програми у вікні браузера. Ви можете використовувати ті ж цикли редагування-оновлення-перегляду, які ви звикли використовувати з JavaScript, в той же час інспектувати змінні, встановлювати точки зупинки, а також використовувати всі інші засоби налагодження доступні вам з Java.

Запуск. Коли ви будете готові до розмістити проект на сервері, GWT компілює ваш код Java в оптимізовані, автономні JavaScript файли, які автоматично запускаються на всіх основних браузерах, а також мобільних браузерах для Android і iPhone.

Комунікація з сервером. Всі GWT додатки запускаються як JavaScript код в браузері кінцевого користувача. Часто, однак, є потреба створити більше, ніж просто окремий додаток на стороні клієнта. Може виникнути потреба комунікації з сервером.

Для цього платформа GWT використовує технологію AJAX. AJAX – це один з компонентів концепції DHTML, за якого веб-сторінка не перезавантажується у фоновому режимі відправляє запити на сервер і сама звідти довантажує потрібні користувачу дані.

GWT надає декілька різних способів спілкування з сервером. Який формат даних ви використовуєте, в кінцевому рахунку залежить від сервера, з яким необхідно взаємодіяти.

- Remote Procedure Calls (GWT RPC). Якщо є можливість запуску Java-сервера і створення інтерфейсу для серверної бізнес-логіки додатка, то GWT RPC буде найкращим вибором. GWT RPC – це механізм передачі об'єктів Java з сервера через стандартний HTTP. GWT RPC використовується для прозорих та зрозумілих запитів на Java сервлети, лишаючи GWT необхідність піклуватися про такі деталі низького рівня, як серіалізація об'єктів.
- Отримання даних за допомогою JSON HTTP. Якщо ваш сервер не може запустити Java сервлети або у вас уже є серверний ужиток, який використовує інші формати даних, наприклад XML або JSON, то GWT забезпечить універсальні класи, які будуть створювати HTTP запит, і JSON та XML класи, які використовуються для обробки відповідей.
- Крос-серверні JSONP запити. Якщо ви створюєте застосування шляхом змішування функціональності різних програмних інтерфейсів та джерел даних (концепція Mashup), вам доведеться працювати з SOP (Same Origin Policy) обмеженнями доступу. GWT використовує JavaScript Native Interface (JSNI) для написання `<script>` тегів, які повертають JSON з padding (JSONP).

Оптимізація. Google Web Toolkit містить два потужні інструменти для створення оптимізованих веб-додатків. GWT компілятор виконує оптимізацію вашого коду – in-lining – видалення “мертвого” коду, оптимізації стрічок, і багато іншого. Встановлюючи спліт-точки в коді, можна також сегментувати завантаження у кілька фрагментів JavaScript, поділяючи великі програми для швидкого запуску.

“Вузькі” місця в продуктивності не обмежуються JavaScript. Код HTML і CSS часто поводять себе дивним чином, що їх важко

дослідити. Speed Tracer є новим розширенням для Chrome у Google Web Toolkit, який дозволяє діагностувати проблеми продуктивності в браузері.

Висновок. Розробка більшості веб-проектів вимагає від програміста глибокого знання багатьох мов програмування та технологій. Платформа GWT значно зменшує вимоги до знання програміста таких технологій та мов, як CSS, HTML, XML, JavaScript.

Завдяки цій платформі Java розробник, який ніколи не писав інтерфейсну частину для веб-проекту, а завжди працював на серверною, може швидко написати веб-проект, оскільки від нього буде вимагатись лише знання мови Java.

Література

1. Robert Hanson, Adam Tacy. GWT in Action.
2. Стаття Google Web Toolkit з <http://code.google.com/webtoolkit/overview.html>.
3. Vipul Gupta. Accelerated GWT

DISCLOSURE OF WEB-MEMBERS PERSONAL INFORMATION IN INTERNET

S.S. Fedushko

*master student of Applied Linguistics Department
Lviv Polytechnic National University*

Визначення рівня розкриття особистої інформації (ім'я, вік, стать, освіта, контактні дані) в облікових записах користувачів Веб-форумів є актуальною проблемою, оскільки з кожним днем зростає кількість користувачів Веб-спільнот. Дані облікових записів користувачів чотирьох Веб-форумів проаналізовано та визначено рівень надання особистої інформації залежно від статі користувача Веб-форумів.

In this research of the level of disclosure personal information in Internet personal data of Web-members is analyzed. Namely, there is nickname, full name, age, education, birth date, location and personal

information (e-mail, link to personal web page and phone number). The personal data of members accounts of Web-communities (*Lviv. Ridne Misto Forum, Ternopil Forum, Kids. Family Forum, ROCK.LVIV.UA: Forum*) is analyzed.

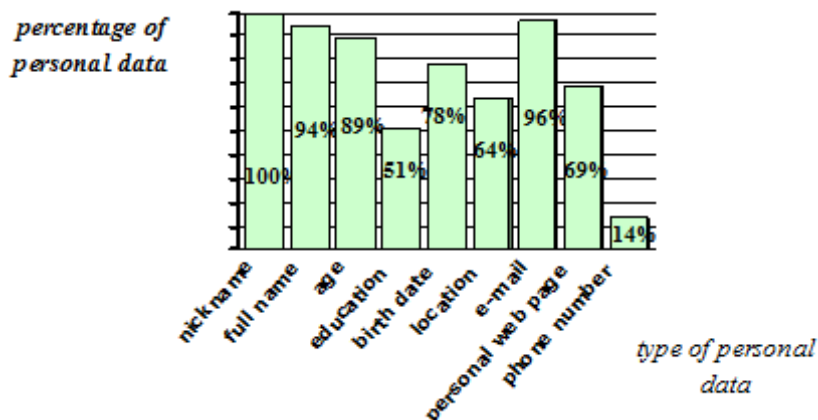


Figure1. Distribution of disclosed personal information by men

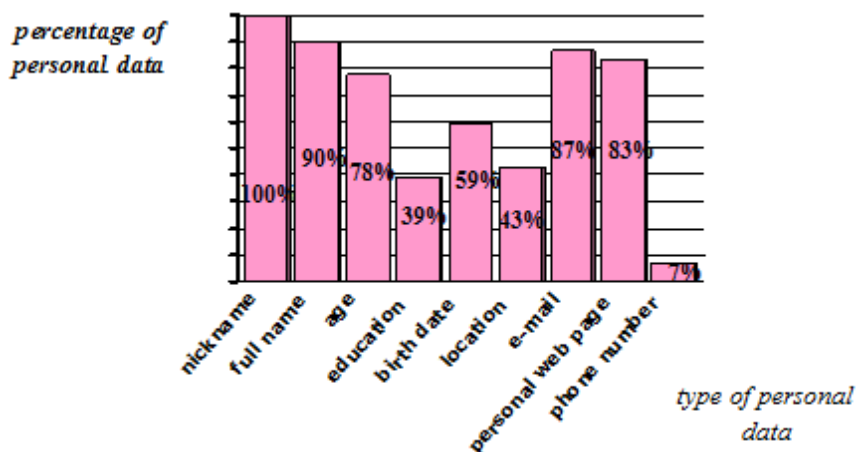


Figure2. Distribution of Disclosed personal information by women

The results of our gender investigation of disclosure of personal information of the Web-members in their accounts are shown in Figure 1 and Figure 2.

As depicted in Figure 2, Web-members reveal a considerable amount of personal information about themselves. The most disclosed personal information, which the men give, includes full name (94%), age (89%) and e-mail (96%). Less disclosed information includes a birth date (78%) or education (51%). Accounting for the different types of contact information: email (96%) link to personal web page (69%) is provided more often than a phone number (14%).

The first results is that men provide more personal information than females, such as name, age, and e-mail, location and birth date. There is a trend for males to provide their location more often than females. However, as seen in Figure 1, females are more likely than males to provide a link to a personal web page (87%).

Thus, there are gender differences for the majority of categories: full name, age, education, birth date, location, e-mail, link to personal web page and phone number. However, as seen in Figure 1 and Figure 2, there are no gender differences for nicknames because to provide nicknames is the demand of administration of Web-communities.

Сучасні інформаційні технології в економіці, менеджменті та освіті (СИТЕМ): Матеріали всеукраїнської науково-практичної конференції, Львів, 12 листопада 2010 р. – Львів: Львівська філія Європейського університету, 2010. – 167 с.

До збірки увійшли тези доповідей учасників міжвузівської науково-практичної конференції *“Сучасні інформаційні технології в економіці, менеджменті та освіті (СИТЕМ)”* (Львів, 12 листопада 2010 р.). Розглядаються питання навчально-методичних основ освіти, обміну досвідом в ІТ-навчанні, математичного моделювання економічних, соціальних і інформаційних процесів. Окрему увагу приділено питанню впровадження сучасних інформаційних технологій в економіці, техніці та бізнесі.

Для науковців, викладачів вищих навчальних закладів, аспірантів та студентів.

Тези доповідей надруковано в авторській редакції без внесення суттєвих змін оргкомітетом

Вимоги до оформлення публікацій

Тези доповідей (до 5 сторінок) українською, російською, англійською мовами подаються окремими файлами у форматі *.doc або *.docx (набрані в редакторі MS Word) і включають в себе:

- заголовок (12 pt., напівжирний, по центру);
- ініціали та прізвище авторів (12 pt, курсив, напівжирний, по центру, відступ 12 pt зверху і 6 pt знизу);
- посаду і організацію, науковий ступінь та звання авторів (10 pt, курсив, по центру);
- телефони, факс, e-mail, адреси веб-сайтів; (10 pt, по центру)
- анотація українською мовою, максимум 5 речень (11 pt, курсив, вирівнювання по ширині, відступ 6 pt зверху і знизу);
- текст (11 pt);

Шрифт: Times New Roman, міжрядковий інтервал – одинарний. Абзацний відступ – 1 см.

Параметри сторінок: **розмір паперу A5**, всі поля по 1,7 см.

Малюнки виконуються, по можливості, векторною графікою; скановані малюнки подаються з роздільною здатністю 300 dpi.

Формули виконуються в Microsoft Equation або MathType.

Підписано до друку 08.11.2010
Формат 60х84 1/16.
Папір друк. №3.
Умовн. друк. арк. 10.5.
Тираж 50 прим.