РОЗДІЛ «ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ»

УДК 519.71:330.142

ЧУПІЛКОТ.А., к.т.н., доцент
ЧУПІЛКО С.І., ст. викладач

Університет митної справи та фінансів, м. Дніпропетровськ

МАТЕМАТИЧНА МОДЕЛЬ ОПТИМІЗАЦІЇ ОБОРОТНОГО КАПІТАЛУ ТОРГОВОГО ПІДПРИЄМСТВА

Вступ. Оцінка ефективності управління фінансовими ресурсами підприємства відбувається за двома напрямами. З одного боку, результати виробничо-фінансової діяльності підприємства оцінюються на основі даних фінансової звітності: бухгалтерський баланс, звіт про прибутки та збитки, про пересування грошових коштів і т.ін. На основі цих документів розраховуються фінансові показники, динаміка зміни і значення яких дозволяють оцінити ефективність управління підприємством. Але така оцінка не має рекомендацій щодо поліпшення результатів діяльності підприємства. Інший підхід в управлінні діяльністю підприємства полягає в розробці та аналізі економіко-математичних моделей з метою дослідження різних можливостей використання потенціалу і ресурсів підприємства, в тому числі фінансових. В цій роботі пропонується такий підхід використовувати для оптимального управління оборотним капіталом торгового підприємства.

Багато авторів на сьогодні розглядають статичні задачі, демонструючи при цьому абсолютно різні підходи і методи. Часто застосовуються методи економетричного або факторного аналізу. Деякі сучасні роботи присвячені розвитку динамічних моделей, що можуть бути застосовані в різних напрямах діяльності підприємства. Зокрема, застосовуючи економетричні методи, розглядають зміну показників в часі. В основному такі методи аналізу пов’язані з задачами ефективності, фінансових результатів і обсягів виробництва на підприємствах або в регіоні. Є моделі оптимізації, пов’язані з формуванням матеріальних ресурсів підприємства при мінімізації витрат або капіталовкладень, які базуються на диференціальних рівняннях, а також моделі типу міжгалузевого балансу В.Леонтьєва, що відображають максимізацію прибутку в умовах обмежених фінансових потоків.

Моделюванню та оптимізації капіталу підприємства присвячено роботу [1]. Управління фінансовою діяльністю підприємства оптової торгівлі розглянуто в [2].

Ця робота є продовженням публікацій авторів [3], [4], присвячених питанням, пов’язаним з діяльністю малого підприємства і оптимізацією та управлінням оборотного капіталу.

Постановка задачі. Кожне підприємство має свої особливості і, відповідно, стратегію розвитку та управління. Оборотний капітал є основним фактором утворення прибутку. Ефективне управління підприємством має спиратися на науково обґрунтовані розрахунки і висновки, що залежать від багатьох зовнішніх та внутрішніх факторів. Для успішної роботи торгового підприємства необхідні інструменти для всебічного моніторингу і прогнозування основних параметрів стану та діяльності. При цьому потрібно мати можливість аналізу зміни параметрів в залежності від управлінських рішень. Для наукового обґрунтування застосовуються методи економіко-математичного моделювання. Економічні залежності, що виникають при формалізації задачі управління оборотним капіталом підприємства, можуть бути виражені у формі математичних рівнянь і нерівностей з певними обмеженнями.

Актуальною є розробка математичних моделей, які враховують якомога більше складових, що впливають на оборотний капітал підприємства, в тому числі зміни в часі.

Моделі мають бути реалізовані методами чисельного та імітаційного моделювання, що буде науковою підставою для обрання стратегії розвитку підприємства для збільшення оборотного капіталу.

Метою роботи є побудова динамічної економіко-математичної моделі оптимізації доходу підприємства в умовах обмеженого оборотного капіталу з урахуванням параметрів торгівельної діяльності підприємства, що змінюються у часі, та реалізація задачі числовими методами імітаційного моделювання. При моделюванні враховані обсяги оптових закупівель, ціна та інтенсивність роздрібних продажів за обмежень на оборотний капітал, інтенсивність попиту, обсяги товарів, діапазон зміни ціни.

Результати роботи. Розглянемо ситуацію, коли торгове підприємство закуповує оптом деякий набір товарів, який реалізується протягом заданого періоду часу. В умовах обмеженого оборотного капіталу необхідно максимізувати маржинальний дохід, отриманий від реалізації цих товарів в роздрібній мережі. Математична оптимізаційна модель цієї задачі може бути записана таким чином:

$$\sum_{i=1}^{n} x_i \int_0^T (c_i(t) * v_i(t, c_i(t)))dt - \sum_{i=1}^{n} c_i^{(0)} * x_i * p_{i\min} \to \max; \quad (1)$$

$$\int_0^T (v_i(t, c_i(t)))dt \le x_i * p_i^{\min}; \quad (2)$$

$$0 \le x_i \le K_i;\ K_i = \frac{V_i}{p_i^{\min}};\ x_i \in Z^t;\ i = 1,2,...,n; \quad (3)$$

$$v_i(t, c_i(t)) \le d_i(t, c_i(t));\ \ i = 1,2,...,n;\quad \forall t \in [0,T]; \quad (4)$$

$$\sum_{i=1}^{n} c_i^{(0)} * x_i * p_i^{\min} \le S; \quad (5)$$

$$c_i^{\min} \le c_i(t) \le c_i^{\max}. \quad (6)$$

В наведеній оптимізаційній задачі використовуються позначення:

$c_i(t)$ – роздрібна ціна і-го товару, що продається у момент t;

$v_i(t, c_i(t))$ – інтенсивність продажу і-го товару в момент t при роздрібній ціні продажу $c_i(t)$;

$c_i^{(0)}$ – оптова ціна продажу одиниці і-го товару на момент оптових закупок;

інтервал [0,T] – це час, протягом якого повинен бути реалізований в роздробі весь закуплений оптом товар;

$p_i^{\min}$ – мінімально можлива партія оптових закупівель і-го товару (i = 1,2,…,n);

x_i – шукана величина, що задає кількість мінімально можливих партій закупівель і-го товару;

V_i – загальний обсяг і-го товару, який в момент оптових закупівель є на складі;

K_i – число мінімально можливих партій і-го товару, яке є на складі у момент оптових закупівель;

$d_i(t, c_i(t))$ – інтенсивність попиту на і-ий товар у момент часу t при ціні на товар $c_i^{(1)}(t)$;

S – обсяг оборотного капіталу підприємства, який може бути використаний для здійснення оптових закупівель;

c_i^{min}, c_i^{max} – відповідно нижня і верхня межі ціни реалізації і-го товару в роздрібній мережі.

У задачі (1)-(6) необхідно визначити обсяги оптових закупівель x_i (i=1,2,…,n), ціну роздрібних продажів $c_i(t)$ і інтенсивність продажів $v_i(t, c_i(t))$, які б максимізували функціонал (1), що задає маржинальний дохід від реалізованого в роздрібній мережі товару, при обмеженнях на оборотний капітал, інтенсивність попиту по кожному товару, обсягу товарів кожного виду на складі в момент здійснення оптових закупівель та обмеження на діапазон цін при продажу товарів у роздрібній мережі. У загальному випадку задача (1)-(6) є нелінійною задачею оптимального управління, розв'язок якої визначається вибором вектора закупівель $x = (x_1, ..., x_n)$ та вибором вектор-функцій часу $c(t) = (c_1(t), ..., c_n))$ та $v(t, c_i(t)) = (v_1(t, c_1(t)), ..., v_n(t, c_n(t)))$, які відповідно задають роздрібні ціни на товари і інтенсивність реалізації товарів з урахуванням обмеження на попит.

Для аналітичного розв'язку задача (1)-(6) має бути спрощена. Можна використовувати методи, пов'язані з імітаційним моделюванням або припущеннями про стаціонарність деяких вхідних параметрів задачі. Наприклад, можна припустити, що якщо інтервал часу (0,T) не занадто тривалий, а інтенсивність попиту на товари лінійно змінюється в залежності від ціни, то задача (1)-(6) може бути розглянута як задача цілочислової оптимізації, в якій інтенсивність продажу товару і ціна товару не залежать від часу і є незмінними на всьому інтервалі (0,T). Час можна враховувати покроковим розв'язком, де початковими є параметри, отримані на попередній ітерації.

Далі будемо вважати, що інтенсивність попиту на і-й товар буде залежати тільки від його ціни і буде змінюватися за наступним лінійним законом:

$$d_i(c_i(t)) = d_i(c_i^{min}(t)) - k_i(c_i(t) - c_i^{min}), \qquad (7)$$

де $d_i(c_i(t))$ – інтенсивність попиту на і-й товар на інтервалі часу (0,T) при ціні $c_i \in [c_i^{min}, c_i^{max}]$;

$d_i(c_i^{min})$ – інтенсивність попиту на і-й товар при мінімальній роздрібній ціні c_i^{min};

k_i – коефіцієнт, що відображає падіння попиту на і-й товар при переході від мінімальної ціни до ціни в момент часу *t* $c_i(t)$.

З формули (7) видно, що попит на і-й товар лінійно падає при збільшенні роздрібної ціни.

Далі для кожного обсягу закупівель і-го товару може бути розрахована оптимальна ціна продажу, яка максимізує дохід від реалізації і-го товару в обсязі $x_i * p_i^{min}$ в роздрібній мережі.

$$\sum_{i=1}^{n} x_i \int_0^T (c_i(t) * d_i(c_i(t)))dt - \sum_{i=1}^{n} c_i^{(0)} * x_i * p_i^{min} \to \max; \qquad (8)$$

$$\int_0^T (d_i(c_i(t)))dt = x_i * p_i^{min}, \qquad x_i = 1,2,...,K_i \qquad i=1,2,3,\ldots,n. \tag{9}$$

З урахуванням співвідношення (7) задачу (8)-(9) можна переписати в наступному вигляді:

$$T(d(c_i^{min}) - k_i(c_i - c_i^{min}))c_i - \sum_{i=1}^{n} c_i^{(0)} x_i p_i^{min} \to \max; \tag{10}$$

$$T(d(c_i^{min}) - k_i(c_i - c_i^{min})) = x_i p_i^{min}; \tag{11}$$

$$x_i = 1,2,...,K_i, \qquad i=1,2,3,\ldots,n. \tag{12}$$

З урахуванням обмеження (12) ціна c_i при будь-якому обсязі закупівель визначається за формулою:

$$c_i = c_i^{min} + \frac{-x_i p_i^{min} + Td(c_i^{min})}{Tk_i}, \qquad i=1,2,3,\ldots,n, \ \ 0 \le x_i \le K_i. \tag{13}$$

В [4] описано метод гілок і меж для розв'язання задачі (1)-(6) в умовах, коли інтенсивність попиту на товари залежить тільки від ціни.

Розглянемо приклад, у якому порівняємо розв'язки, отримані за допомогою процедури «Поиск решения» в Excel і методом гілок і меж (на першому етапі).

Торгівельна фірма використовує для закупівель оборотний капітал у розмірі 5000 у.о. Вихідні дані наведено в табл.1. Отримуючи дохід від продажу, його вкладають у закупівлю на наступному етапі (місяці) до отримання максимального прибутку.

Таблиця 1 – Вихідні дані та результат розрахунку для оптимального доходу

Вид продукції	1	2	3	4	5
Обсяг товарів на складі, шт	100	150	80	70	60
Мінімальна партія, шт.	10	5	20	10	15
Інтенсивність продажів за день, шт.	4	5	2	3	2
Обсяг продажів за місяць, шт.	120	150	60	90	60
Ціна оптова, у.о.	10	15	20	25	30
Ціна роздрібна, у.о.	12	18	25	28	36
Показник доходності	1,2	1,2	1,25	1,12	1,2
Кількість партій	*2*	*24*	*3*	*0*	*4*
Всього товарів	*20*	*120*	*60*	*0*	*60*

На першому етапі дохід від реалізації складає 6060 у.о. Прибуток, відповідно, – 1060 у.о., і оборотний капітал складе 5060 у.о.

За методом гілок і меж верхню границю цільової функції 5450 у.о. отримаємо при закупівлі, орієнтуючись на товари з найбільшим показником дохідності: третього, другого, першого, п'ятого видів.

При визначенні нижньої оцінки враховуємо цілочислений розв'язок. Нижня оцінка: 4750 у.о. при закупівлі першого, другого, третього і четвертого товарів з мінімальними оптовими цінами у максимально можливому обсязі, виходячи з розміру оборотного капіталу. Далі, перебираючи різні набори товарів, наближаємося до значення доступного оборотного капіталу і отримаємо оптимальний набір товарів, що формує дохід від реалізації і, відповідно, прибуток. Очевидно, оптимальний розв'язок буде отримано при значеннях, указаних у табл.1.

За допомогою метода гілок і меж проблематично отримати розв'язок з урахуванням часу. Процедура «Поиск решения» дає наступні результати: за чотири ітерації (місяці) отримаємо максимальний дохід від реалізації 8000 у.о. Максимальний прибуток – 1520 у.о. Оборотний капітал збільшується на максимальну величину і складає 6520 у.о. У цьому випадку реалізується максимальна кількість товарів і партій, що обумовлена попитом (у табл.1 – обсягом продажу за місяць). Збільшення прибутку стає неможливим при наведених вихідних даних: цінах роздрібних та оптових, обсягу та інтенсивності продажів, запасів відповідних товарів. Для збільшення прибутку потрібно вводити новий асортимент, збільшувати різницю між оптовими та роздрібними цінами і т.ін.

Найчастіше ціна реалізації товару в роздрібній мережі призначається менеджментом компанії, виходячи з власного досвіду та оцінки макроекономічних параметрів в регіоні, таких як попит на товари, рівень зайнятості населення, динаміка зміни середньої або мінімальної зарплати, рівень інфляції і т.д. В цьому випадку може бути розглянута модифікація задачі (1)-(6).

Висновки. Розроблена математична модель дозволяє оцінювати у динаміці тенденції розвитку економічної ситуації на підприємстві в залежності від зміни основних параметрів, що впливають на неї. Модель дозволяє визначити маржинальний дохід підприємства, у різних тенденціях, при значеннях всіх врахованих у моделі показників, які впливають на його економічний стан, за початкових умов, що також є змінними. Запропоновані модель і метод розв'язування проблеми дають змогу кількісно оцінити та обґрунтувати вибір стратегії і, зокрема, параметрів, що впливають на оборотний капітал та маржинальний дохід підприємства.

Побудована модель у подальшому буде розвинена з урахуванням чинників, що впливають на оборотний капітал торгового підприємства і не враховані у цій моделі: ризик, управлінські витрати, кредити, інвестиції тощо.

ЛІТЕРАТУРА

1. Нгуен К.Н. Моделирование и оптимизация управления структурой капитала фирмы / К.Н.Нгуен // Економіка: проблеми теорії та практики. – Дн-вськ: Наука і освіта. – 2000. –№ 29. – С.62-73.
2. Мирская С.Ю. Динамическая модель управления предприятием оптовой торговли / С.Ю.Мирская, В.И.Сидельников, А.К. бликян // Научная мысль Кавказа. – Северо-Кавказский научный центр высшей школы. – 2006. – № 13. – С.85-88.
3. Чупілко Т.А. Економетричне моделювання розвитку малого підприємництва в Дніпропетровському регіоні / Т.А.Чупілко, С.І.Чупілко // Вісник Дніпропетровської державної фінансової академії. – 2010. – №2(24)'10. – С.200-205.
4. Чупілко Т.А. Динамічна модель управління оборотним капіталом торгового підприємства / Т.А.Чупілко, С.І.Чупілко // Вісник Дніпропетровської державної фінансової академії. – 2015. – №1(33). – С.135-147.

Надійшла до редколегії 28.03.2016.

УДК 005.2:004.9

КАРІМОВ Г.І., к.е.н., доцент

Дніпродзержинський державний технічний університет

УПРАВЛІННЯ ЯКІСТЮ НА БАЗІ СТАНДАРТНИХ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Вступ. Задачі управління і контролю якості, трудомісткі і великі за обсягом, найбільш просто автоматизуються у випадку застосування загальної інформаційної системи підприємства ERP / ERP II класу. Така система здатна реалізувати принципи концепції TQM ("загального менеджменту якості"); в більшості програмних продуктів, що реалізують ERP / ERP II - систему, наявні відокремлені модулі "Управління якістю" та/або "Управління бізнес-процесами" [1]. Але у деяких випадках їх використання є недоцільним або неможливим. В першу чергу це стосується невеликих малих та середніх підприємств, що не мають розгалуженої управлінської структури.

Основні за трудомісткістю задачі підсистеми управління та контролю якості пов'язані з накопиченням числової інформації та її опрацюванням за певними алгоритмами. Для вирішення таких задач дуже добре зарекомендували себе стандартні інформаційні технології на основі табличних процесорів та систем керування базами даних [2-5]. В той же час питання інтеграції стандартних інформаційних технологій в загальну автоматизовану інформаційну систему управління потребують подальшого розвитку, зокрема, в напрямку уточнення місця стандартних інформаційних технологій в автоматизованих системах управління якістю.

Постановка задачі. Найпростіший варіант організації автоматизованого управління та контролю якості полягає в використанні загальнодоступних систем опрацювання різноманітної інформації в поєднанні з засобами забезпечення роботи локальної комп'ютерної мережі. Так, текстовий редактор MS Word з інтегрованого пакету ділових застосувань Microsoft Office дає можливість не тільки створювати, редагувати і оформлювати текстові документи у відповідності до певних вимог, а й використовувати при цьому шаблони типових документів, макроси для автоматизації роботи з документами, засоби спільної роботи над документом кількох користувачів з розмежуванням їх прав і т.п. Великі можливості для автоматизації розрахункових задач надаються табличними процесорами, зокрема, Microsoft Excel. Наявні в ньому засоби дозволяють вирішувати і задачі прогнозування впливу якості на конкурентноздатність продукції, і задачі статистичного контролю якості за певними критеріями, і багато інших задач.

Додержуючись орієнтації на використання загальнодоступних програмних систем, можна також запропонувати застосування системи управління базами даних Microsoft Access для створення та експлуатації автоматизованої бази даних АІС "Управління якістю". Особливу увагу при цьому слід приділити детальній проробці системи запитів та форм для забезпечення керівництва організації оперативною, повною та достовірною інформацією для прийняття управлінських рішень [6].

При цьому в міжнародному стандарті з термінології (ISO 9000) виділені два аспекти управління якістю: "загальне" управління якістю (quality management) і управління якістю як оперативна діяльність (quality control). Виходячи з логіки стандарту, такі функції, як політика і планування якості, організація роботи з якості, навчання і мотивація персоналу, прийняття стратегічних рішень і взаємодія з зовнішнім середовищем, повинні бути віднесені до "загального" управління якістю. Контроль якості, інформація про якість, розробка заходів, прийняття оперативних рішень і їх реалізація повинні входити до складу "оперативного" управління якістю [7].

За допомогою загальнодоступного програмного забезпечення можливо автоматизувати, в першу чергу, оперативне управління якістю, яке підтримує виробничий процес в потрібному режимі. Тобто, з десятьох функцій, що складають "петлю якості", в спрощеному вигляді підлягають автоматизації лише функції оперативного управління якістю: контроль якості, інформація про якість, розробка заходів, прийняття оперативних рішень.

У даній роботі розглядаються можливі підходи до автоматизації вказаних функцій при створенні інформаційної системи оперативного управління якістю на базі використання стандартних інформаційних технологій.

Результати роботи. Аналіз задач функцій оперативного управління якістю з врахуванням сучасного стану програмного забезпечення дозволяє розподілити останні так, як показано в табл.1.

Таблиця 1 – Програмне забезпечення ІС «Управління якістю»

Функції		**Задачі підсистеми**	**Стандартне ПЗ**
1	Контроль якості	Аналіз статистичних даних	MS Excel, MS Access
2	Інформація про якість	Пошук, підготовка, оброблення, надання текстової інформації	MS Word, MS Excel, MS Access, ІАС “Парус-Консультант”, Mozilla Firefox / Internet Explorer
3	Розробка заходів	Кількісне та аналогове обґрунтування альтернативних варіантів заходів та рішень, які пропонуються	MS Word, MS Excel, ІАС “Парус-Консультант”, Mozilla Firefox / Internet Explorer
4	Прийняття оперативних рішень		

Наведене в табл.1 програмне забезпечення можна вважати «стандартним», оскільки воно встановлено майже на кожному особистому або офісному ПК (за винятком, мабуть, ІАС “Парус-Консультант”), робота з ним не викликає особливих складнощів та не потребує вузькоспеціалізованих навичок, комплекс програмних засобів є взаємосумісним. Звісно, перелік може бути орієнтованим на інші операційні системи, включати дещо інші програми та таке інше. Але обов’язковими елементами є текстовий редактор, табличний процесор, СУБД, засоби роботи в мережі, засоби доступу до інформаційної бази з законодавства України. Зауважимо, що існує широке коло організацій, орієнтованих на максимальне використання саме стандартних технологій в структурі своєї інформаційної системи управління.

Основна частина обробки цифрової інформації "оперативного" управління якістю відбувається в межах реалізації функції контролю якості (аналіз статистичних даних). При автоматизації заходів управління якістю за допомогою стандартних інформаційних технологій для вирішення цих питань пропонується використовувати табличний процесор (наприклад, MS Excel).

Інструменти контролю якості, що реалізовані на базі математико-статистичних методів обробки результатів вибіркового контролю якості, достатньо просто реалізуються засобами табличного процесора MS Excel, зокрема, за допомогою надбудови «Анализ данных». Перелік відповідних вбудованих засобів наведено в табл.2. Їх використання детально описано в багатьох підручниках та довідниках (наприклад, [8-9]).

Незалежно від засобів реалізації використання статистичних методів базується на великому масиві вхідних даних з якості, які формуються на місцях проведення конт-

Таблиця 2 – Відповідність вбудованих засобів MS Excel інструментам контролю якості

Інструменти (статистичні методи) контролю якості	Вбудовані засоби MS Excel	Узагальнений опис
Контрольний листок	Анализ данных / Выборка, Анализ данных / Описательная статистика	на бланку заздалегідь друкують контрольовані параметри, відповідно до яких можна вносити дані за допомогою позначок або простих символів
Стратифікація (розшарування)	Анализ данных / Описательная статистика, Анализ данных / «Выборка»	здійснюється групування даних залежно від умов побудови й кожної групи даних, тобто розподіл на шари
Гістограма	Анализ данных / Гистограмма	графік, на якому у вигляді стовпчиків показано розподіл даних окремих вимірів або контролю одного й того ж або декількох параметрів, згрупованих за частотою попадання в певний, заздалегідь встановлений той чи інший інтервал значень
Аналіз Парето	Анализ данных / Гистограмма, Анализ данных / Описательная статистика”	стовпчикова діаграма даних, отриманих за кожною перевірною ознакою, дані розташовують у порядку значущості й будують кумулятивну криву
Причинно-наслідкова діаграма (діаграма Ісікаві)	Анализ данных / Регрессия	проблема умовно зображується у вигляді прямої горизонтальної лінії – зображуються основні (першого порядку) фактори, що впливають на проблему – на них впливають причини другого порядку – вони виявляються під впливом факторів третього порядку і так далі
Діаграма розсіювання (розкидання)	Анализ данных / Корреляция, Анализ данных / Ковариация	крапковий графік, де *У* - показник якості, *Х* - фактор, що впливає на якість. На рисунку візуально простежується вид та ступінь кореляції (залежності)
Контрольна карта	Анализ данных / Регрессия	складається звичайно з трьох ліній (центральна лінія – середнє значення контрольованого параметра, вища від центральної – верхня контрольна межа, третя лінія (нижча) - нижня контрольна межа), на осі ординат відкладається значення контрольованого параметра, а по осі абсцис – час вибірки
Графіки	Мастер диаграмм	графічні зображення статистичного матеріалу – стовпчикові, лінійні, кругові, стрічкові та інші графіки.

ролю (виробів, матеріалів, інструменту тощо). Проведення ж обробки первинної інформації кожним контролером на місці її виникнення не має сенсу. При цьому передача всієї інформації, що виникає в межах оперативної діяльності, як на рівень "загального"

управління якістю (quality management), так і для прийняття оперативних рішень також малоефективна. Більш доцільним є застосування інструментів контролю якості, що реалізовані на базі математико-статистичних методів обробки даних, в інформаційному просторі управління якістю підприємства так, як вказано на рис.1.

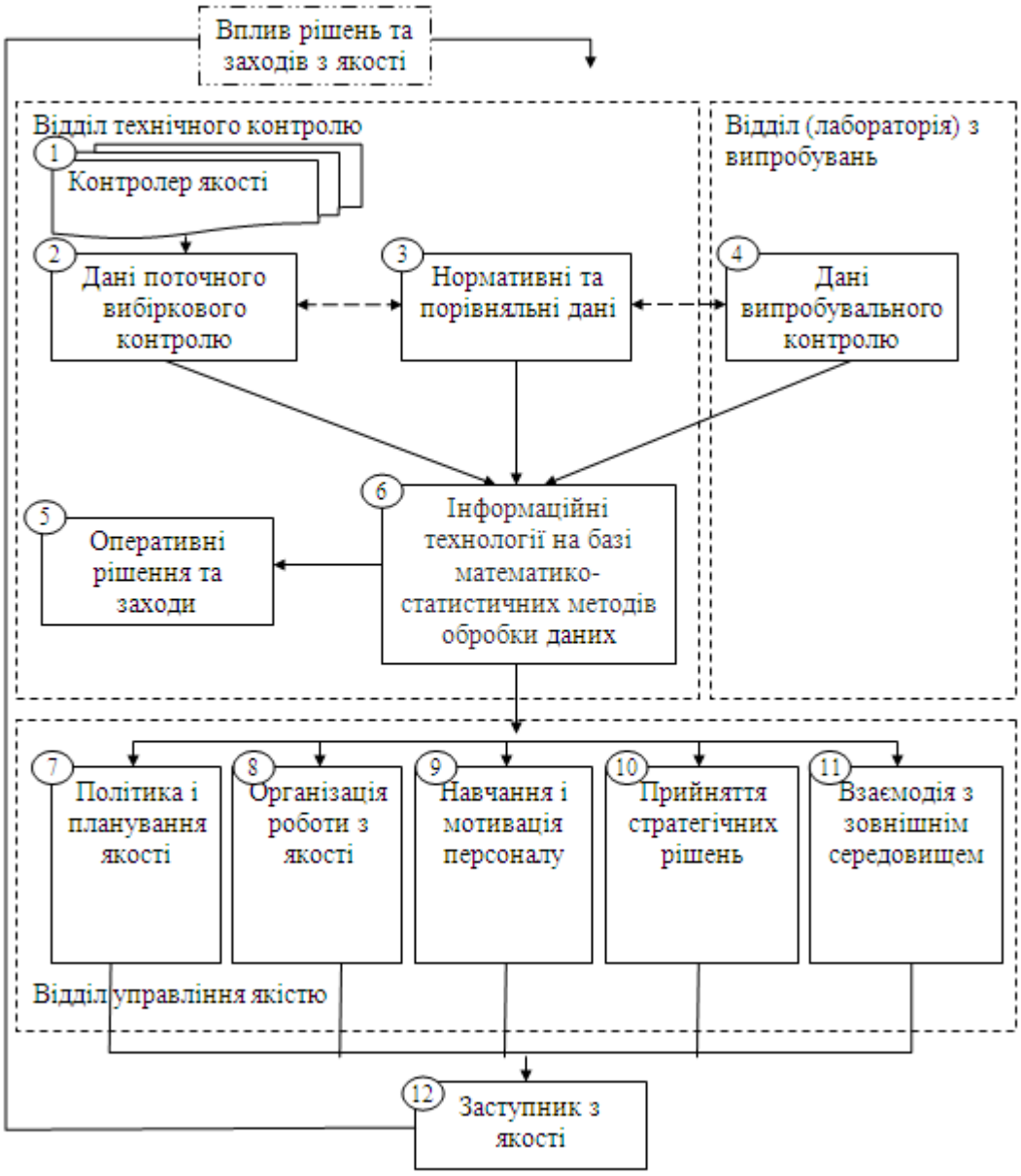


Рисунок 1 – Місце математико-статистичних методів в інформаційному просторі управління якістю

Як видно з рис.1, в загальну систему управління якістю включено три відділи (відділ технічного контролю, відділ (лабораторія) з випробувань та відділ з управління якістю), які підпорядковані заступнику директора з якості. Назви відділів та поділ функцій між ними дещо узагальнені і в кожному конкретному випадку залежать від структури та розміру підприємства.

Блоки 1-6 відповідають рівню оперативного контролю якості, блоки 7-11 – рівню "загального" управління якістю, заступник директора з якості (12) – відповідальний за управління якістю в цілому на підприємстві.

Якщо розглядати наведену на рис.1 схему в якості узагальненого зображення

інформаційної підсистеми «Управління якістю», логічним є створення окремого центру комп'ютерно-математичної обробки статистичних даних. Програмно такий центр може бути реалізовано як у вигляді інтегрованої підсистеми, так і окремим спеціалізованим робочим місцем (АРМом) [6, 10]. Останній варіант є більш прийнятним при орієнтації на використання стандартних інформаційних технологій. Такий АРМ розташовується на місці блока 6.

В організаційному аспекті робота зі створюваним АРМом покладається на "інженера з якості – аналітика" або "аналітичне бюро з якості" (в разі великого обсягу інформації).

Функціонально, з урахуванням задач інформаційної системи «Управління якістю», що наведені в табл.1, АРМ аналітика з якості вирішує наступні задачі:

- аналіз статистичних даних (контроль якості);
- оброблення інформації (частково – інформація про якість);
- кількісне обґрунтування альтернативних варіантів заходів, які пропонуються (розробка заходів);
- кількісне обґрунтування рішень, що пропонуються (прийняття оперативних рішень).

Інші задачі інформаційної системи «Управління якістю» (пошук, підготовка, попереднє оброблення, надання текстової інформації – інформація про якість, аналогове обґрунтування альтернативних варіантів заходів та рішень, які пропонуються – розробка заходів та прийняття оперативних рішень) реалізуються в межах блоків 3 та 5 відповідно.

Таким чином, комп'ютеризоване робоче місце аналітика з якості повинно стати обчислювальним ядром інформаційної підсистеми з оперативного управління якістю. В свою чергу, вірне, своєчасне та доцільно-виважене представлення інформації оперативного рівня позитивно впливає на підсумок діяльності на рівні "загального" управління якістю.

Висновки. На підставі аналізу функцій системи менеджменту якості та можливостей стандартних інформаційних технологій встановлено, що основна частина обробки цифрової інформації "оперативного" управління якістю відбувається в межах реалізації функцій контролю якості; інструменти контролю якості на основі математико-статистичних методів достатньо просто реалізуються засобами табличного процесора MS Excel. Запропоновано організаційний підхід до формування інформаційного простору управління якістю підприємства з визначенням місць застосування інструментів контролю якості, що реалізовані на базі математико-статистичних методів обробки даних.

ЛІТЕРАТУРА

1. Карімов Г.І. Сучасні інформаційні технології у сфері управління якістю / Г.І.Карімов // Збірник наукових праць Дніпродзержинського державного університету: (технічні науки). – Дніпродзержинськ: ДДТУ. – 2015. – Випуск 2 (27). – С.169-173.
2. Каримов Г.И. Комплекс программ для управления материальными ресурсами / Г.И.Каримов, Н.Г.Ревенко, М.В.Дроботова // Комп'ютерне моделювання: міждерж. наук.-метод. конф., 24-26 червня 1998 р.: тези допов. – Дніпродзержинськ, 1998. – С.112.
3. Карімов Г.І. Розробка системи підтримки прийняття рішень по управлінню матеріальними ресурсами підприємства / Г.І. Карімов // Математичне та програмне забезпечення інтелектуальних систем: всеукр. наук.-практ. конф., 17-19 листопада 2003 р.: тези допов. – Дніпропетровськ, 2003. – С.26.
4. Карімов І.К. Економіко-математичне моделювання в управлінні матеріальними ресурсами промислового підприємства / І.К.Карімов, Д.А.Козлов // Математичне та

програмне забезпечення інтелектуальних систем: всеукр. наук.-практ. конф. 15-17 листопада 2006 р.: тези допов. – Дніпропетровськ: ДНУ. – 2006. – С.78.
5. Карімов Г.І. Оптимізація управління запасами: програмне забезпечення для навчального процесу / Г.І.Карімов, І.К.Карімов, Д.А.Козлов // Проблеми математичного моделювання: міждерж. наук.-метод. конф. 23-25 травня 2007 р.: тези допов. – Дніпродзержинськ, 2007. – С.227.
6. Карімов Г.І. Інформаційні системи і технології в управлінні організаціями: монографія / Г.І.Карімов, І.К.Карімов. – Дніпродзержинськ: ДДТУ, 2014. – 143с.
7. Капінос Г.І. Операційний менеджмент: навч. посіб. / Г.І.Капінос, І.В.Бабій. – К.: «Центр учбової літератури», 2013. – 352с.
8. Авраменко В.І. Теорія ймовірностей і математична статистика: навч. посіб. / В.І.Авраменко, І.К.Карімов. – 2-ге вид., перероб. і доп. – Дніпродзержинськ: ДДТУ, 2013. – 247с.
9. Долженков В.А. Microsoft Excel 2000 / В.А.Долженков, Ю.В.Колесников. – СПб.: БХВ – Петербург, 2000. – 1088с.
10. Пономаренко В.С. Проектування автоматизованих економічних інформаційних систем: навч. посіб. / В.С.Пономаренко, О.І.Пушкар, Ю.І.Коваленко. – К.: ІЗМН, 1996. – 312с.

Надійшла до редколегії 25.04.2016.

УДК 004.031.43:338.5

БАБЕНКО М.В., к.т.н., доцент
БРАТУТА М.Ю., студент

Дніпродзержинський державний технічний університет

ВИКОРИСТАННЯ МЕТОДИК ВИЗНАЧЕННЯ ВАРТОСТІ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ПРИ АНАЛІЗІ БЕЗЗБИТКОВОСТІ СТВОРЕННЯ ІНФОРМАЦІЙНИХ СИСТЕМ

Вступ. Історія успіху будь-якої компанії вимірюється величиною отриманого прибутку й зміною її динаміки. Для успішного ведення бізнесу необхідно не тільки прораховувати, скільки компанія заробить при досягненні запланованого обсягу продажів, але й чітко представляти, який мінімальний обсяг продажів необхідний для забезпечення беззбиткової роботи.

Задача полягає у визначенні того обсягу продажів, нижче якого підприємство буде втрачати гроші, вище якого – заробляти. Цей мінімально припустимий обсяг продажів, що покриває всі витрати на виготовлення продукції, не приносячи при цьому ні прибутку, ні збитків, одержав назву *точка беззбитковості* (вона ж – точка рівноваги, вона ж – break-event point).

Таким чином, поняття точки беззбитковості є одночасно і якимсь критерієм ефективності діяльності фірми. Фірма, що не досягає точки беззбитковості, діє неефективно з погляду сформованої ринкової кон'юнктури.

Постановка задачі. Метою роботи є використання методик визначення вартості розробки програмного забезпечення при аналізі беззбитковості створення інформаційних систем для збільшення точності прогнозованих оцінок вартості й строків проектів, зниження ризиків по залученню коштів у проекти, підвищення якості процесу управління програмними проектами в цілому.

Мета аналізу беззбитковості в класичному економічному аналізі – встановити, що станеться з фінансовими результатами підприємства, якщо зміниться певний рівень продуктивності підприємства або обсягу виробництва. В основі аналізу беззбитковості – порівнянність таких показників, як доходи від продажів, витрати, прибуток протягом короткого періоду, тобто періоду, протягом якого вихід продукції обмежений рівнем наявних в розпорядженні підприємства діючих виробничих потужностей.

Результати роботи. Результатом аналізу беззбитковості є пошук точки беззбитковості. Точка беззбитковості або точка рентабельності, прибутку – це рівень продажів в натуральних одиницях (наприклад, в штуках), при якому виручка від продажів дорівнює витратам на виробництво і реалізацію продукції, тобто прибуток дорівнює нулю.

Основним методом визначення точки беззбитковості є CVP-аналіз (*Cast Value Profit* – витрати, обсяг, прибуток) [1], заснований на аналізі співвідношень витрат і обсягів випуску й прибутку.

Графічна інтерпретація визначення й аналізу точки беззбитковості представлена на рис.1.

Під *техніко-економічним обґрунтуванням вартості* (договірної ціни) програмної системи будемо розуміти методику оцінювання трудових, часових і фінансових ресурсів по створенню програмної системи, що відповідає вимогам замовника.

В основу визначення необхідних обсягів ресурсів повинні бути покладені: сукупність бізнес-процесів, реалізованих у майбутній програмній системі і їхній відносній важливості (пріоритет) для замовника; вимоги до функціональної повноти і якості реалізації кожного бізнесу-процесу.

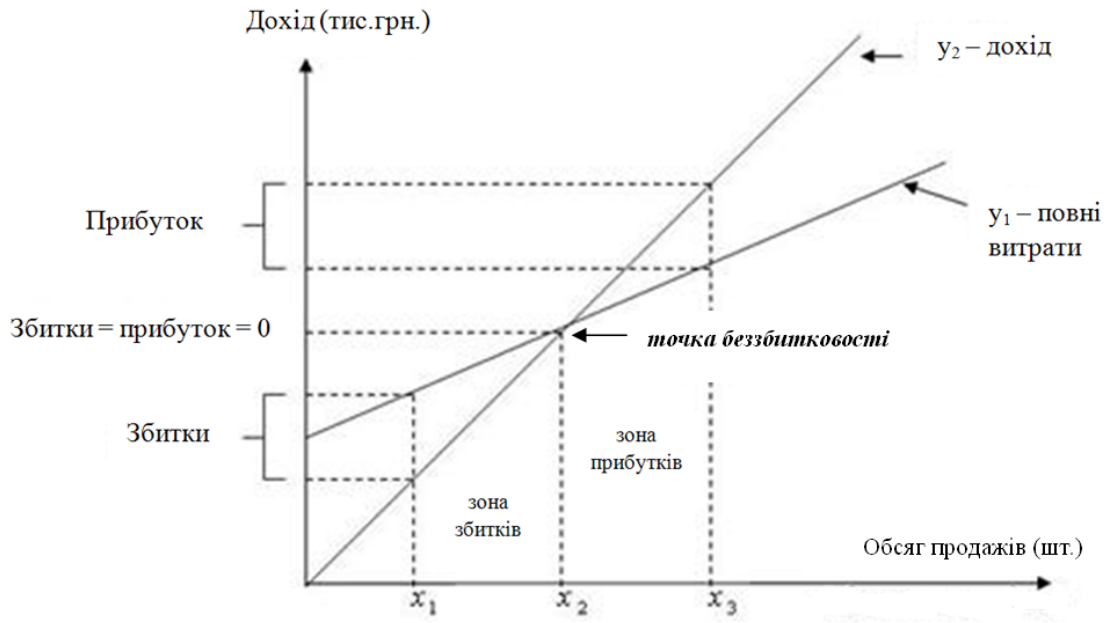


Рисунок 1 – Графічна інтерпретація визначення й аналізу точки беззбитковості

Як основні показники оцінки вартості програмної системи використовуються: складність (розміри) програмної системи; трудовитрати на розробку; тривалість розробки програмної системи в цілому і її окремих етапів; чисельність і кваліфікація фахівців, що залучаються до створення програмної системи; фонд оплати праці фахівців на створення програмної системи в цілому й по конкретному етапу життєвого циклу; інші прямі витрати й накладні витрати, пов'язані зі створенням програмної системи.

Всі нормативи й інші статистичні дані, використовувані в методиках техніко-економічного обґрунтування вартості ґрунтуються на статистичних даних, що узагальнюють закордонний і вітчизняний досвід розробки програмних систем [2].

За рівнем складності вся безліч програмних систем (ПС) ділиться на три типи.

До першого типу відносяться: комплексні програмні системи (КПС) і технології, окремі частини яких реалізовані на різних платформах; територіально-розподілені програмні системи й технології; системи автоматизованого або автоматичного керування, що функціонують у режимі реального часу.

До другого типу відносяться програмні інформаційно-довідкові системи (ІДС), що забезпечують інформаційну підтримку основних бізнес-процесів організації з великою кількістю типів вихідної інформації.

До третього типу відносяться інженерні й науково-технічні пакети прикладних програм (ППП) і технологій, що характеризуються чітко заданим алгоритмом обробки й малими обсягами вихідних даних.

У даній роботі використовуються три методи визначення розмірів програмної системи: прямий метод, метод функціональних точок, метод на основі розмірності бази даних програмної системи.

Прямий метод визначення техніко-економічних параметрів програмної системи заснований на використанні *методу експертних оцінок.*

Майбутню програмну систему треба декомпозувати до рівня елементарних компонентів, а для оцінки розмірів кожного з компонентів використовувати або зовнішніх експертів, що мають досвід розробки подібних систем і готових прототипів, або в якості експертів можуть виступати фахівці розроблювача й замовника [3].

Метод функціональних точок (Function point – FP) ґрунтується на тім, що розміри програмної системи оцінюються в термінах кількості й складності бізнес-процесів (функцій), реалізованих у даному програмному коді [2].

Функціональна точка – це комбінація властивостей програмного забезпечення: інтенсивності використання введення й виведення зовнішніх даних; взаємодії системи з користувачем; зовнішніх інтерфейсів; файлів, використовуваних системою.

Майбутня система з використанням методології структурного аналізу й проектування описується у вигляді багаторівневої графічної моделі, представленої як сукупність взаємозалежних функціональних діаграм (користувальницьких бізнес-процесів).

Метод на основі розмірності бази даних програмної системи. Розмірність програмної системи визначається кількістю об'єктів, атрибутів і їхніх взаємозв'язків на об'єктних діаграмах бізнес-процесів [4].

Атрибут – найпростіший елемент бази даних інформаційної моделі, що містить одну з характеристик предметної області й вводиться або безпосередньо користувачем, або заноситься в базу з довідників і класифікаторів.

Об'єкт – елемент бази даних, сформований з атрибутів і містить інформацію про реальний процес, явище, предмет.

Розмірність програмного забезпечення визначається за наступною формулою:

$$R = 2N \cdot 5K_1 \cdot 10M\,, \qquad (1)$$

де N – кількість об'єктів (таблиць) предметної області, кількість зв'язків між таблицями необмежена й визначається структурою бази даних; K_1 – сумарна кількість взаємозв'язків між об'єктами; M – сумарна кількість атрибутів предметної області, що *припадають на один об'єкт.*

Кількість зв’язків між атрибутами визначається кількістю джерел формування атрибутивної інформації.

Нормалізованою величиною при створенні програмної системи є кількість формованих атрибутів, що входять в електронні таблиці за допомогою встановлених зв’язків.

На кафедрі "Програмне забезпечення систем" Дніпродзержинського державного технічного університету розроблено програмний засіб, що реалізує вищеназвані методики визначення вартості розробки програмного забезпечення для автоматизації процесу аналізу беззбитковості створення інформаційних систем. На головній формі вікна програмного засобу розташовані елементи для введення початкових даних, а також панель вкладок з результатами розрахунків основних етапів аналізу беззбитковості (рис.2).

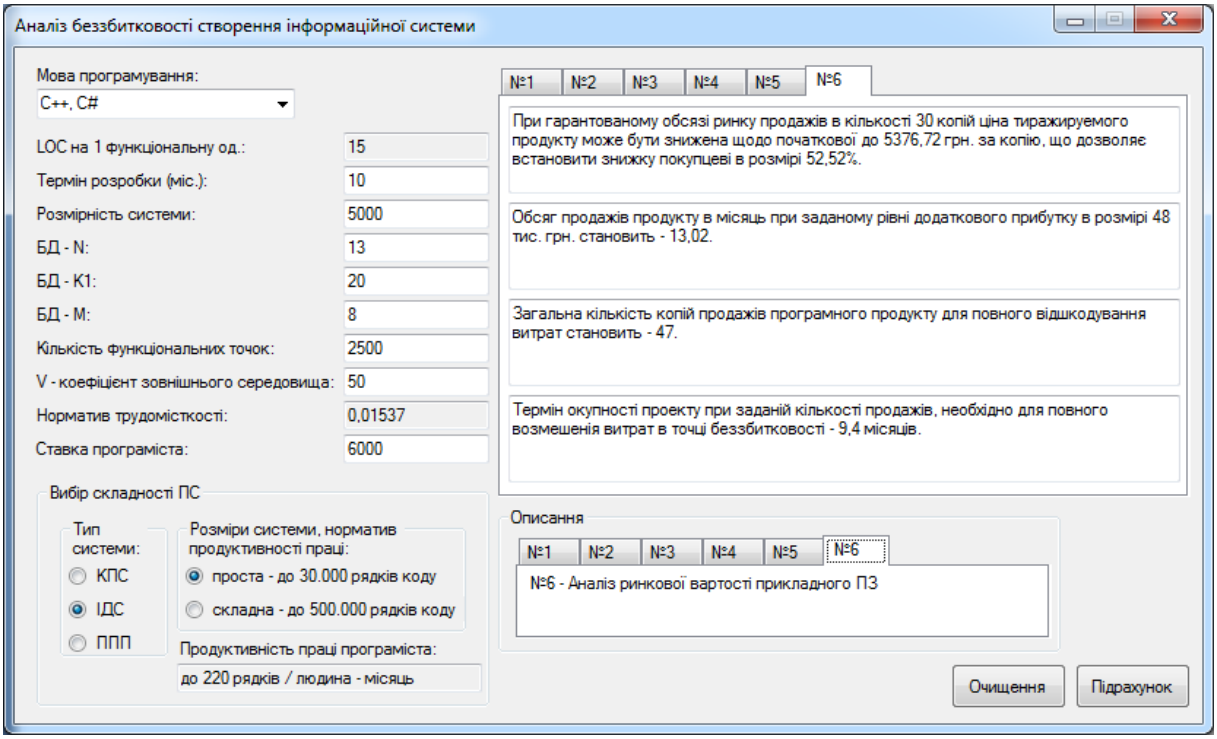


Рисунок 2 – Головна форма вікна програмного засобу

Маємо шість вкладок-етапів: 1 – використання прямого методу; 2 – методу функціональних точок; 3 – методу на основі розмірності бази даних програмної системи; 4 – визначення вартості створення програмної системи при обґрунтуванні договірної ціни на розробку програмного забезпечення; 5 – визначення ринкової вартості програмного забезпечення; 6 – аналіз ринкової вартості програмного забезпечення.

На відповідному етапі генерується графік точки беззбитковості (рис.3).

Висновки. В епоху інформаційних технологій, що швидко розвиваються, зростаючого числа високо бюджетних проектів в області розробки програмного забезпечення дуже важливим стає вміння оцінити на ранніх етапах можливі вигоди й збитки від проекту, проаналізувати можливі сценарії розвитку подій. Тому програмний засіб для автоматизації процесу аналізу беззбитковості створення інформаційних систем, описаний в даній роботі, може стати у нагоді менеджерові з розробки програмного забезпечення для планування етапів життєвого циклу розробки програмного продукту.

ЛІТЕРАТУРА

1. Брігхем Є.Ф. Основи фінансового менеджменту: підручник / Є.Ф. Брігхем; пер. В.Біленький [та ін.]. – Київ: Молодь, 1997. – 1000с.

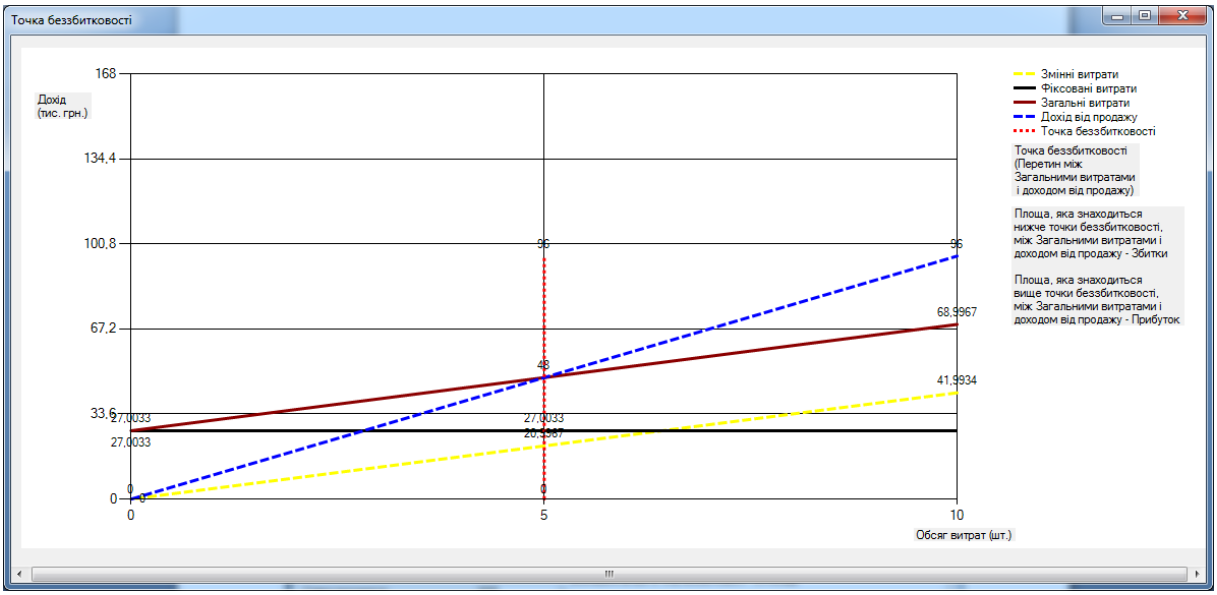


Рисунок 3 – Графік точки беззбитковості

2. Роберт Т. Фатрелл. Управление программными проектами. Достижение оптимального качества при минимуме затрат: Персона / Роберт Т. Фатрелл, Дональд Ф. Шафер, Линда И. Шафер. – М.: Издательский дом «Вильямс», 2004. – 1136с.
3. Липаев В.В. Технико-экономическое обоснование проектов сложных программных систем / Липаев В.В. – М.: СИНТЕГ, 2004. – 284с.
4. Информационные технологии. Виды испытаний автоматизированных систем: ГОСТ 34.603-92. – Дата введения 01.01.93. – М.: Госстандарт. – 1992. – 6с.

Надійшла до редколегії 25.04.2016.

УДК 004.7

БАБЕНКО М.В., к.т.н., доцент
ЖУЛЬКОВСЬКИЙ О.О., к.т.н., доцент
БАБЕНКО Ю.М.*, студент

Дніпродзержинський державний технічний університет
*Національний авіаційний університет

ОСОБЛИВОСТІ ВИКОРИСТАННЯ ІМІТАЦІЙНОГО МОДЕЛЮВАННЯ КОМП'ЮТЕРНИХ МЕРЕЖ У ДОСЛІДНИЦЬКИХ І НАВЧАЛЬНИХ ЦІЛЯХ

Вступ. Будь-яка задача проектування мережі зводиться до забезпечення заданих показників якості при мінімальних витратах. Для дослідження реальних мереж використовуються аналізатори протоколів, які незамінні для дослідження реальних мереж, але вони не дозволяють одержувати кількісні оцінки характеристик для ще неіснуючих мереж, що перебувають у стадії проектування. У цих випадках проектувальники можуть використовувати засоби проектування, за допомогою яких розробляються моделі, що відтворюють інформаційні процеси у мережах.

Такі моделі являють собою комп’ютерну програму, що крок за кроком відтворює події, які відбуваються в реальній системі. Стосовно до обчислювальних мереж їх імітаційні моделі відтворюють процеси генерації повідомлень додатками, розбиття повідомлень на пакети і кадри певних протоколів, затримки, пов’язані з обробкою по-

відомлень, пакетів і кадрів всередині операційної системи, процес отримання доступу комп'ютером до розділюваного мережевого середовища, процес обробки маршрутизатором пакетів, що надходять і т.д. При імітаційному моделюванні мережі не потрібно здобувати дороге устаткування – його робота імітується програмами, що досить точно відтворюють всі основні особливості й параметри такого встаткування.

Існують спеціальні, орієнтовані на моделювання обчислювальних мереж, програмні системи, у яких процес створення моделі спрощений. Такі програмні системи самі генерують модель мережі на основі вихідних даних про її топологію й використовувані протоколи, про інтенсивності потоків запитів між комп'ютерами мережі, довжини ліній зв'язку, про типи використовуваного устаткування й додатків. Програмні системи моделювання можуть бути вузько спеціалізованими й досить універсальними, що дозволяють імітувати мережі всіляких типів. Якість результатів моделювання в значній мірі залежить від точності вихідних даних про мережу, переданих у систему імітаційного моделювання [1, 2].

Такі програмні системи дозволяють мінімізувати витрати на розробку мереж і підготовку проектної документації, провести експерименти, результати яких можуть бути використані для обґрунтування вибору типу мережі, а також скоротити витрати, пов'язані з помилковими рішеннями.

Програмні системи проектування мереж – інструмент, що може придатися будь-якому адміністраторові комп'ютерної мережі, особливо при проектуванні нової мережі або внесенні кардинальних змін у вже існуючу. Продукти даної категорії дозволяють перевірити наслідки впровадження тих або інших рішень. Програми проектування мережі використовують у своїй роботі інформацію про просторове розташування мережі, число вузлів, конфігурації зв'язків, швидкості передачі даних, використовувані протоколи і типи устаткування, виконувані в мережі додатки та включають набір засобів для підготовки вихідних даних про досліджувану мережу і засоби для статистичної обробки отриманих результатів проектування. Також в таких система можна реалізувати імітаційне моделювання Найбільш популярні програмні системи проектування мереж, а також приклад моделювання сегменту мережі Ethernet у системі NetCracker компанії NetCracker Technology, USA більш детально розглянуто у [3].

Постановка задачі. Метою даної роботи є проведення імітаційного моделювання роботи комп'ютерної мережі деякого підрозділу на прикладі кафедри «Програмного забезпечення систем» Дніпродзержинського державного технічного університету. На кафедрі існує 2 класи та викладацька. У кожному класі по 10 комп'ютерів, у викладацькій – 5 комп'ютерів. У кожному класі є сервер з програмами, необхідними для роботи у цьому класі. Також на кафедрі є сервер баз даних та проксі-сервер.

Виходячи з передбаченої інтенсивності обміну даними між робочими станціями та серверами, а також вимог додатків прикладного рівня до параметрів якості обслуговування, приходимо до висновку, що в цій мережі доцільно буде використовувати технологію *Fast Ethernet*.

Комп'ютери у викладацькій будуть підключені до комутатора, який, в свою чергу, буде підключений до центрального комутатора. У класі буде встановлений концентратор або комутатор, до якого буде підключено робочі станції та сервер класу.

При виборі мережного устаткування необхідно врахувати можливість збільшення кількості комп'ютерів у класах та викладацькій.

Для перевірки правильності проектування мережі треба провести імітаційне моделювання і визначити приблизне навантаження на мережеві пристрої та канали передачі.

Результати роботи. Система NetCracker компанії NetCracker Technology, USA являє собою CASE-засіб для автоматизованого проектування, моделювання та аналізу

комп'ютерних мереж. Для проектування структури мережі програма надає можливість вибору необхідного устаткування з вбудованої бази даних, а також додавання в базу даних і конфігурування нового обладнання різних типів. Користувач розміщує обрані компоненти на складальному полі, задає структуру і тип зв'язків між ними, визначає тип програмного забезпечення і характер трафіка між вузлами мережі. Надалі користувач має можливість указати перелік аналізованих характеристик, вигляд відображення статистичної інформації і виконати імітаційне моделювання проектованої мережі.

Для побудови мережі в програмі NetCracker оберемо те обладнання, яке збираємось використовувати при побудові реальної мережі, або максимально наближене до нього за характеристиками. Якщо у класах та викладацькій будуть встановлені комутатори для робочих груп, потрібно обрати для моделювання комутатор приблизно такого ж типу. Центральний комутатор, до якого підключаються інші комутатори та сервери, потрібно обрати з урахуванням більш високих вимог щодо надійності та продуктивності. Моделі мереж викладацької та комп'ютерного класу зображені на рис.2, 3. Модель загальної мережі кафедри зображена на рис.4.

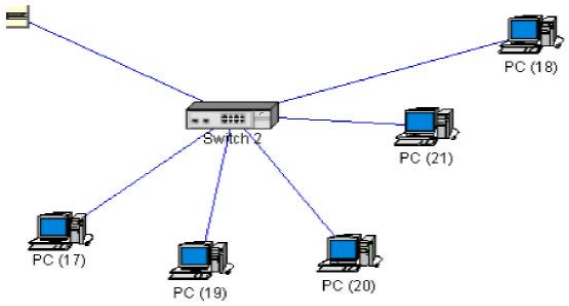


Рисунок 2 – Модель мережі викладацької

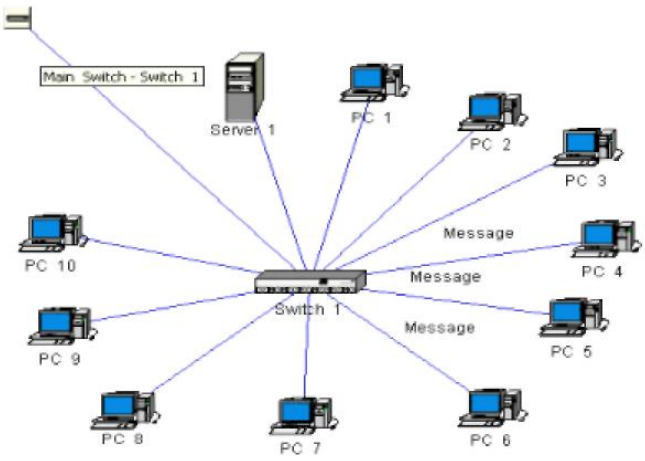


Рисунок 3 – Модель мережі класу

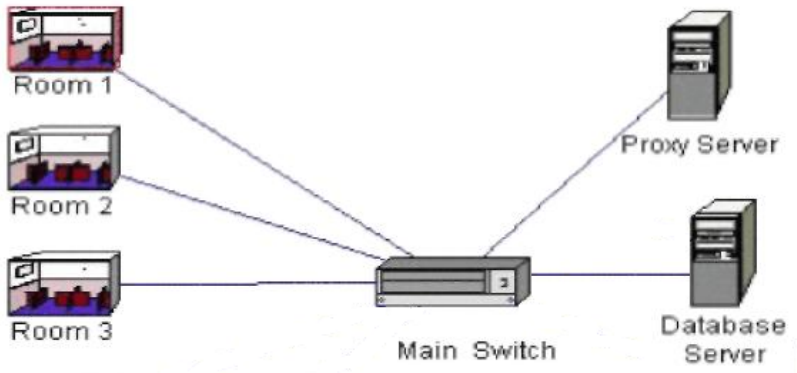


Рисунок 4 – Модель загальної мережі кафедри

Крім того, доцільно врахувати можливість подальшого розвитку мережі і додавання нових робочих станцій та серверів, зарезервувавши для них деяку кількість вільних портів.

Спочатку треба визначити потоки даних, що створюються прикладними програмами. Наприклад, для потоку даних між кожним комп'ютером класу та сервером класу встановимо такі дані: розмір пакету (transactions size) від 500 до 1000 байт, інтервал між запитами (time between transactions) 0,0005с (рис.5).

Далі потрібно підібрати моделі концентраторів або комутаторів, а також адаптерів для серверів та робочих станцій.

Визначаємо потоки даних між комп'ютерами у класах та викладацькій і сервером баз даних та проксі-сервером. Для потоку даних між комп'ютерами та проксі-сервером оберемо параметри, що вказані на рис.6, а для потоку між комп'ютерами та сервером БД – стандартний тип трафіка "SQL server's client".

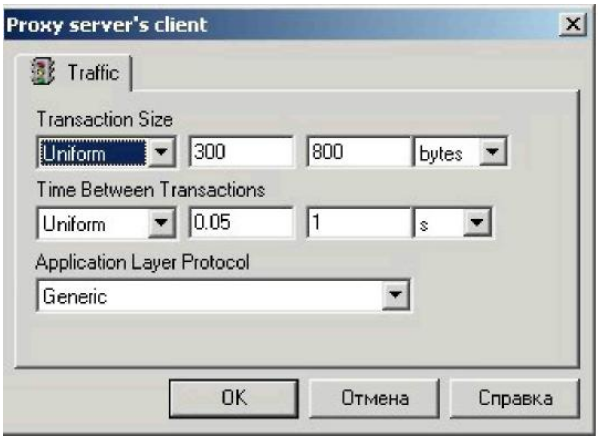


Рисунок 5 – Параметри потоку даних між сервером та комп'ютерами у класі

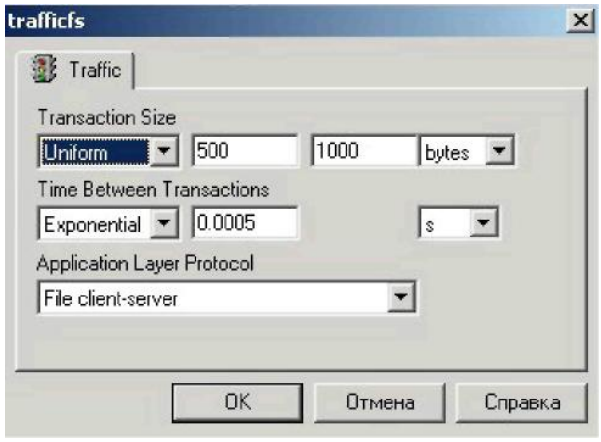


Рисунок 6 – Параметри потоку даних між комп'ютерами та проксі-сервером

Провівши аналіз побудованої мережі бачимо, що використаний концентратор завантажено на 100% та у мережі з'являються колізії (рис.7). Необхідно підібрати комутатор, при якому колізії відсутні.

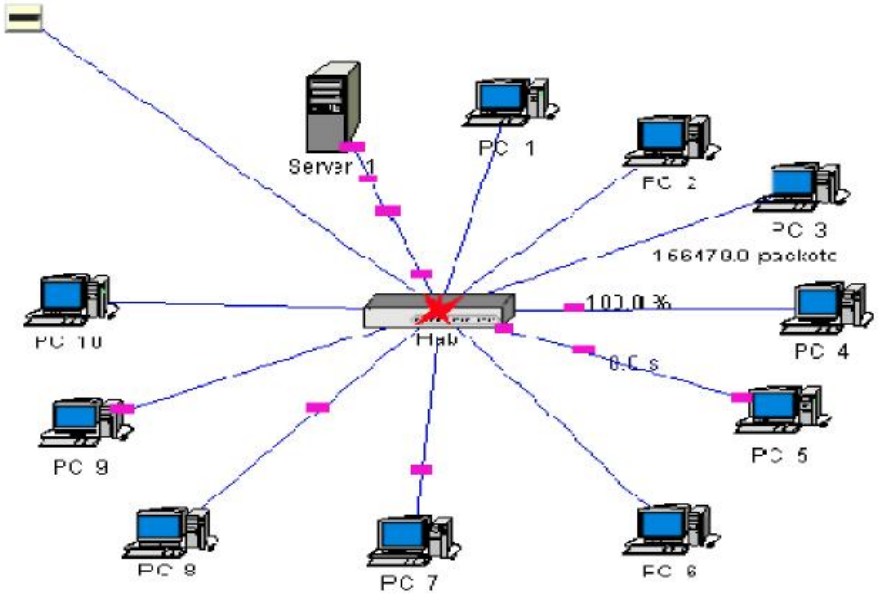


Рисунок 7 – Модель мережі класу з використанням концентратора

Висновки. NetCracker представляє собою CASE-засіб автоматизованого проектування, моделювання і аналізу комп'ютерних мереж з метою мінімізації витрат на розробку мереж і підготовку проектної документації. Дозволяє провести експерименти за допомогою імітаційного моделювання роботи комп'ютерної мережі, результати яких можуть бути використані для обґрунтування вибору типу мережі, середовища передачі, мережевих компонент обладнання та програмно-математичного забезпечення. Програмні засоби NetCracker дозволяють виконати збір відповідних даних про існуючі мережі

без зупинки їх роботи, створити проект цієї мережі та виконати необхідні експерименти для визначення граничних характеристик, можливості розширення, зміни топології і модифікації мережного обладнання з метою подальшого його вдосконалення і розвитку. Для реалізацій функцій імітаційного моделювання у складі NetCracker передбачені засоби завдання характеристик трафіків різних протоколів; засоби візуального контролю заданих параметрів; засоби накопичення статистичної інформації та формування звітної документації про проведені експерименти. Все це говорить про можливість використання системи для дослідницьких і навчальних задач проектування, моделювання та аналізу комп'ютерних мереж.

ЛІТЕРАТУРА

1. Проектирование и диагностика компьютерных систем и сетей: уч. пособ. / [Бондаренко М.Ф., Кривуля Г.Ф., Рябцев В.Г. и др.]. – Киев: НМЦ ВО, 2000. – 306с.
2. Пономаренко Л.А. Инструментальные средства проектирования, имитационного моделирования и анализа компьютерных сетей: уч. пособ. для студ. вузов / Л.А.Пономаренко, В.И.Щелкунов, А.Я.Скляров. – 2-е изд., испр. и доп. – Харьков: Компания СМИТ, 2006. – 488с.
3. Бабенко М.В. Моделювання комп'ютерних мереж як засіб вивчання, проектування та оптимізації роботи мережі / Бабенко М.В., Алексеєва Ю.О. // Збірник наукових праць Дніпродзержинського державного технічного університету (технічні науки). – 2012. – Випуск 3 (20). – С.168-174.

Надійшла до редколегії 25.04.2016.

УДК 004.9

БОЖУХА Л.М., к.ф.-м. н., доцент
ЗІНЬКОВСЬКИЙ Д.В., студент

Дніпродзержинський державний технічний університет

ПРО НОВІТНІ ТЕХНОЛОГІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ТА ЇХ НЕДОЛІКИ ПРИ РОЗРОБЦІ ДОДАТКУ ТАЙМ-МЕНЕДЖЕРА

Вступ. Час є найбільш цінним ресурсом. Точність і правильність цього твердження особливо легко відчути в нашу соціально-онлайнову епоху, коли банальне бажання подивитися прогноз погоди непомітно перетворюється на присід за комп'ютерний стіл тривалістю кілька десятків хвилин. Дана проблема поширена настільки, що навіть встигла отримати спеціальну назву – прокрастинація. Прокрастинація описує процес, коли людина усвідомлює всю важливість та необхідність виконання певної справи, але натомість цього витрачає дорогоцінний час даремно, відволікаючись на різні дрібниці та розваги.

«Тайм-менеджмент» – управління своїм часом, досягнення поставлених цілей та вірна їх мотивація. На винахід цього терміну претендує компанія Time Management International. Її засновник, данець Клаус Меллер, в 70-ті роки винайшов Time Manager – складно влаштований блокнот-щоденник, який можна вважати прабатьком сучасного органайзера. Тайм-менеджмент – це технологія, яка дозволяє керувати часом у реальних ситуаціях повсякденного життя. Вирішення цієї проблеми викликало бажання написати time-manager під Android ОС, який включав би в себе засоби для продуктивного керування власними справами.

Аналізуючи ринок подібних пропозицій, можна дійти висновку, що вони всі або платні, або з малим функціоналом, або частково платні. Це й стало мотивом для створення власного тайм-менеджеру. Метою розробки додатку є створення механізму будильника для нагадувань про діяльність. Нагадування має оповіщати про діяльність звуко-

вим сигналом та вібрацією, якщо вона увімкнена у телефоні. При роботі з повідомленнями додатку може надаватися можливість вибору відстеження діяльності, видалення/ редагування. Створення механізму відстеження часу і таймерів реалізує вибір типу таймеру за обмеженістю. Робота кожного таймера повинна бути виділена в окремі потоки.

Постановка задачі. Мобільні програмні рішення дуже відрізняються від десктопних програм тим, що потрібно слідкувати за рівнем заряду батареї, відстежувати орієнтацію екрану, перезавантаження відбувається частіше, ніж у персональних комп'ютерів. Це дуже часто спонукає розробників проектувати варіанти роботи додатку для подальшого продовження роботи додатку після перезавантаження. Усі ці проблеми плануються бути вирішеними при розробці програмного додатку «Тайм менеджер під управлінням Android ОС» [1].

Ключовим моментом створення програмного забезпечення є аналіз функціональної взаємодії об'єктів автоматизації [2]. В основі функціонального аналізу лежить принцип декомпозиції дій. Результатом аналізу в цьому контексті є функціональна модель, що дає уявлення про предметну область у термінах функцій і груп даних, що супроводжують виконання цих функцій. У найбільш узагальненому вигляді функціональна модель – це опис предметної області, заснований на аналізі семантики об'єктів і явищ, виконаний без орієнтації на використання програмних або технічних комп'ютерних засобів, у якому акцентується функціональний аспект моделювання предметної області [3].

Результати роботи. В якості засобу розробки функціональної моделі предметної області обрана методологія IDEF0, яка передбачає представлення результатів аналізу предметної області у вигляді діаграм: контекстної діаграми (рис.1) та діаграми декомпозиції (рис.2).

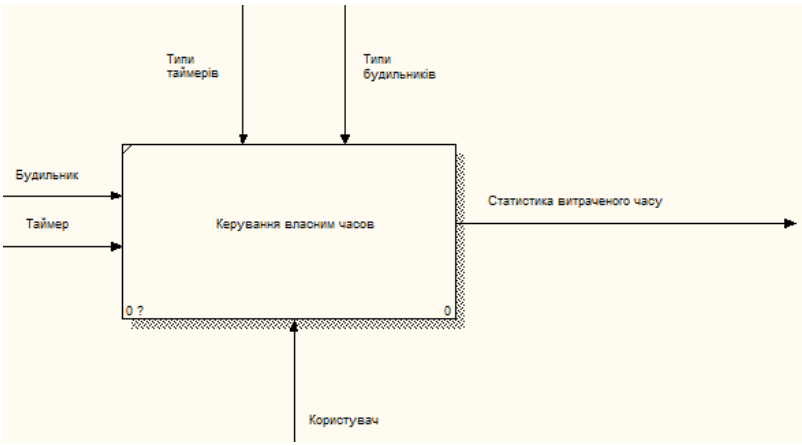


Рисунок 1 – Контекстна діаграма предметної області

Основним призначенням інформаційної системи (ІС) є оперативне забезпечення користувача інформацією про зовнішній світ шляхом реалізації питально-відповідного відношення. Питально-відповідні відношення дозволяють виділити для ІС певний її фрагмент – предметну область (ПрО), яка буде втілена в автоматизованій ІС. Інформація про зовнішній світ подається в ІС у формі даних, що обмежує можливості змістовної інтерпретації інформації й конкретизує семантику її подання в ІС.

Сукупність цих виділених для ІС даних, зв'язків між ними й операцій над ними утворить інформаційну й функціональну моделі ПрО, що описують її стан із певною точністю. Інформаційна й функціональна моделі ПрО є вхідними даними для процесу проектування БД [2].

Сутність ПрО є результатом абстрагування реального об'єкта шляхом виділення й фіксації набору його властивостей. Об'єкти взаємодіють між собою через свої властивості, що породжує ситуації. Ситуації – це взаємозв'язки, які виражають взаємини між об'єктами. Ситуації у предметній області описуються за допомогою висловлювань про ПрО з використанням виразами і обчисленнями предикатів, тобто формальної, ма-

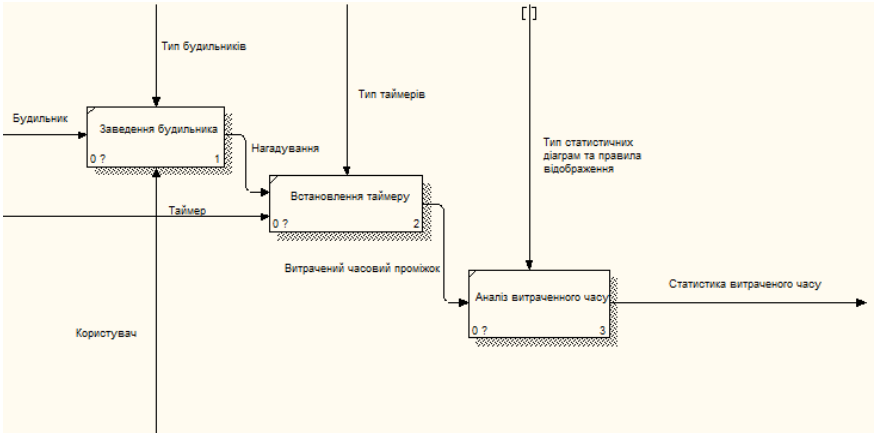


Рисунок 2 – Діаграма декомпозиції першого рівня предметної області

тематичної логіки. Розрізняють статичні й динамічні ситуації. Динамічні ситуації використовуються в одному циклі відтворення, наприклад, надходження нагадування на смартфон; статичні ситуації використовуються в багатьох циклах відтворення, наприклад, множинний відлік таймерів [4].

В якості інструментарію відображення об'єктної моделі предметної області обрано діаграму сутність-зв'язок. Враховуючи вимоги розробки діаграми сутність-зв'язок, описана специфікація об'єктів предметної області та специфікація атрибутів сутностей. Множинність зв'язку між сутностями «Будильник» та «Повідомлення будильника» становить 1:1.

Загальний вигляд діаграми «сутність-зв'язок» для описаної предметної області показано на рис.3.

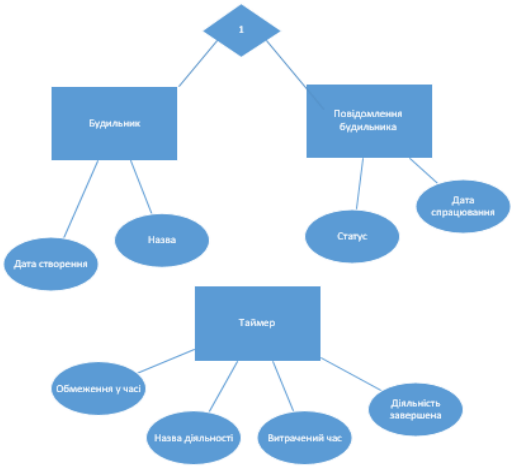


Рисунок 3 – Загальний вигляд діаграми «сутність-зв'язок»

Проблеми ринку Android-додатків такої вузької спеціалізації, як time management територіальні та пов'язані з купівельною спроможністю власників смартфонів на території України. Ці проблеми взаємопов'язані, бо чималий відсоток від західних додатків є платними чи частково безкоштовними, які виражаються у обмеженому функціоналі, без якого або важко або взагалі обійтись неможливо.

Перевантаження функціями – є актуальною проблемою будь-якого програмного забезпечення. Н а сьогодні середній екран смартфона – це 4 дюйми; простота та мінімалістичність не в збиток продуктивності додатку є основними важелями з завоювання уваги клієнта. Але в той же час багато з розробників нехтують цим знанням.

Архітектура Android є фреймворк-орієнтованою (framework-based) у противагу вільній (free-style) архітектурі. Вільні додатки, написані на Java, починаються з класу, що має метод main (), і розробляються в повній відповідності до настрою розробника. На противагу цьому фреймворк-орієнтовані додатки ґрунтуються на існуючому фреймворку. Їх розробка зводиться до розширення деяких класів або реалізації інтерфейсів, наданих фрейморками. Такий додаток не може бути запущений поза фреймворком або без нього. Прикладом можуть бути веб-додатки на Java, в яких розробники реалізують інтерфейс Servlet або розширюють одну з його реалізацій, або програми Eclipse RCP, в якому розробник розширює один з класів Editor або View. Фреймворк-орієнтована архітектура обмежує свободу розробників, наказуючи їм, що і як слід робити. Але, в під-

сумку, зникають повторювані ділянки коду (boilerplate code) і розробникам доводиться ретельно слідувати шаблонами проектування.

Для Java існує безліч фреймворків. Тим не менш, команда Android вирішила створити свій власний. Можливо, однією з причин, по якій вони так вчинили, була необхідність підтримувати унікальну систему управління пам'яттю. В «звичайній» Java об'єкти знаходяться в пам'яті до тих пір, поки збирач сміття НЕ добереться до них. Відбувається це тільки тоді, коли на об'єкт немає жодного посилання від «живих» об'єктів. В Android це не так. Якщо якийсь інтерфейс користувача ховається (тобто його більше не бачимо на екрані), то немає ніякої гарантії, що він знаходиться в пам'яті, навіть є додаток, який надалі збирається використовувати цей інтерфейс. Якщо у ОС Android є достатньо вільної пам'яті, ці об'єкти можуть зберігатися в ній, але збирач сміття може знищити їх у будь-який момент, коли ОС вирішить, що пам'яті залишилося занадто мало. Те ж вірно і для процесів. Процес, який в даний момент часу не вказує користувачу ніякого графічного інтерфейсу, може бути знищений ОС Android на абсолютно законних підставах.

Розглянемо приклад. Нехай наш додаток має екрани A і B. Користувач спочатку відкриває екран A, а потім екран B; з цього моменту екран A став невидимий. Це означає, що екран A і вся логіка, що міститься в ньому, може перебувати в пам'яті, а може і не перебувати. Так, немає гарантій, що об'єкти, пов'язані з екраном A знаходяться в пам'яті, поки видно екран B. Логіка екрану B не повинна зав'язуватися на знаходження об'єктів екрану A в пам'яті. Побічний ефект полягає в тому, що архітектура Android примушує до використання архітектурного стилю «shared nothing». Це означає, що різні частини Android програми можуть викликати один одного і взаємодіяти між собою тільки формально; жоден з них не може звернутися до іншого безпосередньо. А що трапиться, якщо користувач вирішить повернутися на екран A? Напевно, він захоче побачити його в тому ж стані, в якому він залишив його, вірно? Фреймворк Android вирішує цю проблему: для кожного стану життєвого циклу існують методи, кожен з яких може бути перевизначений розробником. Ці методи викликаються фреймворком в заздалегідь певні ключові моменти, наприклад, коли користувальницький інтерфейс показується на екрані, ховається і т.п. У цих методах розробник може реалізувати логіку для зберігання і відновлення стану об'єктів.

Подібна обробка приховування користувальницьких інтерфейсів і існування кнопки «назад» на Android-пристроях призводять до необхідності наявності стека користувальницьких інтерфейсів, в якому поточний видимий інтерфейс поміщається на вершину, а всі інші зсуваються вниз (стекова операція «push»). Подібний стек існує в Android. У документації він називається «activity stack» або іноді «back stack».

У плані обробки взаємодії між інтерфейсом і його логікою Android можливе використання архітектурного шаблону «Model-View-ViewModel» (MVVM – найкраща архітектура GUI-додатків на даний момент).

Архітектура MVVM була створена з метою поділу праці дизайнера і програміста, неможливого, коли Java-розробник намагається побудувати GUI в Swing або розробник на Visual C ++ намагається створити користувальницький інтерфейс в MFC. Розробники мають безліч навичок, але створення зручних і привабливих інтерфейсів вимагає абсолютно інших талантів, ніж ті, якими володіють розробники. Ця робота більше підходить для дизайнерів інтерфейсів. Хороші дизайнери інтерфейсів краще знають, чого хочуть користувачі, ніж експерти в галузі проектування та написання коду. Зрозуміло, буде краще, якщо дизайнер інтерфейсів створить інтерфейс, а розробник напише код, який реалізує логіку цього інтерфейсу, але технології типу Swing або MFC не дозволяють діяти таким чином.

Архітектура MVVM вирішує цю проблему ясним поділом відповідальності:

- розробка користувальницького інтерфейсу здійснюється дизайнером інтерфейсів за допомогою технології, більш-менш природної для такої роботи (XML);

- логіка користувача інтерфейсу реалізується розробником як компонент ViewModel;
- функціональні зв'язки між інтерфейсом і ViewModel реалізуються через Біндінг (bindings), які є правилами типу «якщо кнопка А була натиснута, повинен бути викликаний метод onButtonAClick () з ViewModel». Біндінг можуть бути написані в коді або визначені декларативним шляхом (Android використовує обидва типи).

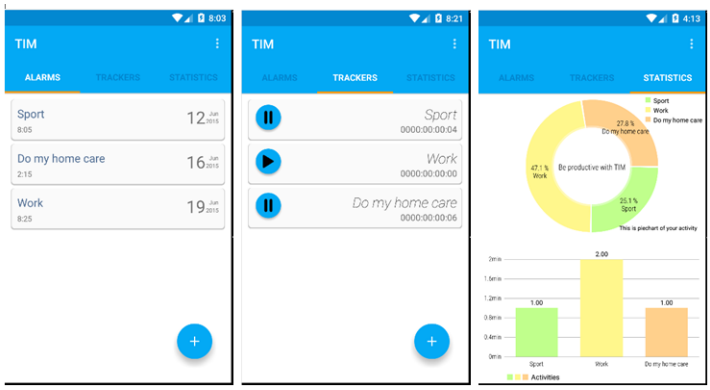


Рисунок 4 – Екранні форми (AlarmsFragment.java – Екранна форма будильників, TrackersFragment.java – Екранна форма таймерів, StatisticsFragment.java – Екранна форма графічного представлення)

Архітектура MVVM використовується в тому чи іншому вигляді усіма сучасними технологіями, наприклад, Microsoft WPF і Silverlight, Oracle JavaFX, Adobe Flex, AJAX.

Структура інтерфейсної частини даної АІС (рис. 4) як сукупність засобів для обробки та відображення інформації, максимально пристосованих для зручності користувача, складається із програмної назви сторінки та опису сторінки. У графічних системах інтерфейс користувача реалізовується багатовіконним режимом, змінами кольору, розміру, видимості вікон, їхнім розташуванням, сортуванням елементів вікон, доступністю багатокористувацьких налаштувань.

Функції додатку представлені на діаграмі варіантів використання (рис.5).

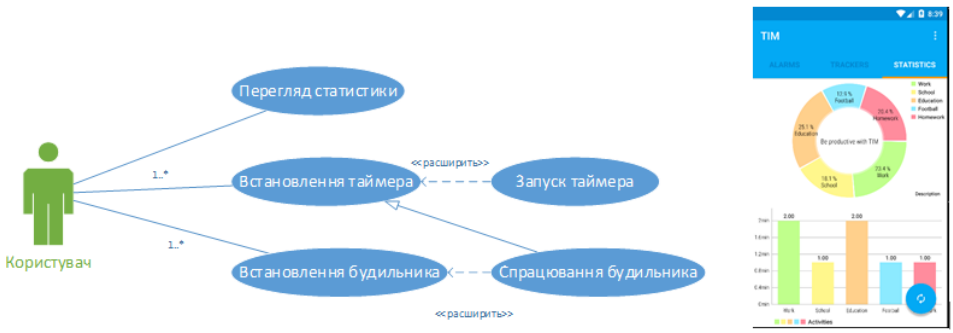


Рисунок 5 – Діаграма варіантів використання та статистка діяльності

Після додавання декількох таймерів можна відстежувати їх роботу (рис.5). Користувач може скористуватися таймерами для відстеження діяльності без використання будильника. Будильник у даному додатку розроблений лише для нагадування про ту роботу, яку потрібно відстежити.

Як було описано вище, операційна система Android використовує компоненти BroadcastReceivers, Service, Intent для обміну повідомленнями між системою.

Це мабуть було б і все, якби з додатку не виходили, і пристрій ніколи не перезавантажувався і був завжди ввімкнений. Для вирішення цих проблем у маніфесті додат-

ку треба вказати дозвіл відстежувати системні повідомлення, які розсилає система, а саме потрібна інформація перезавантаження девайсу. При виході з додатку та вказавши потрібний дозвіл, програмне забезпечення підписано на повідомлення перезавантаження. Після отримання повідомлення перезавантаження запускається сервіс, який перевіряє статус будильників та встановлює їх в початковий рівень з обчисленою різницею часу.

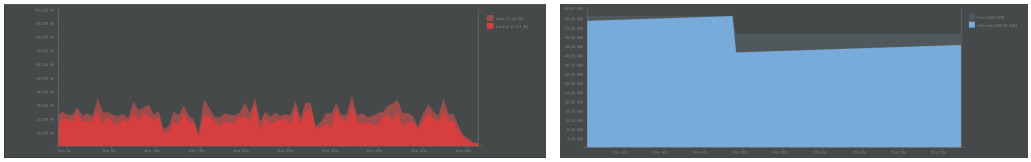
У випадку, коли девайс вимкнений, для роботи будильника потрібний ще один дозвіл на пробудження телефону зі сну on/off режиму (навіть при розрядці батареї).

Опираючись на механізм потоків Java, робота кожного таймера виноситься в окремий потік Java(Thread). Android звичайно надає свої засоби для вирішення таких проблем, як винесення важкої роботи в інші потоки, це AsyncTask, Loader, IntentSevice та ін. Але жоден з них не призначався з ціллю постійної роботи протягом кількох годин або днів, якщо користувач не зупинив додаток (пауза у роботі).

AsyncTask виконує роботу в іншому потоці і має дуже зручний механізм доступу до головного потоку, але більше призначений для відносно неважких операцій (завантаження зображення з сервера та іншої діяльності, яка має логічний кінець). Поганою практикою вважається мультизапуск AsyncTask. Loader не можна використовувати ідеологічно, бо він лише конфліктував би з іншими Loader'ами, які використовуються для завантаження таймерів у список на екрані. IntentService – окремий потік – теж не найкраще рішення, бо багато IntentSevice в системі спровокує Android звільнити ресурси та повбивати екземпляри інтент-сервісів, та ж проблема і з AsyncTask. Жодному з цих компонентів не можна задати пріоритет виконання. Thread підходить для ролі потоку, який буде працювати невідомий проміжок часу, до перезавантаження системи. Є можливим запуск багатьох екземплярів класу Thread.

Було зроблено тестування розробленого додатку в звичайних та екстремальних ситуаціях (рис.6, 7):

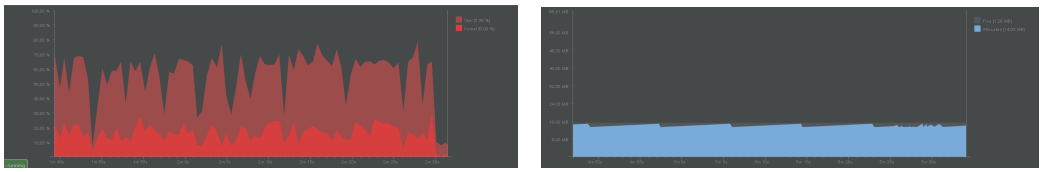
- навантаження на процесор смартфона з 50 сторонніми потоками та постійною прокруткою списку ListView з максимальним API 22 та з мінімальним API 17;
- об'єм оперативної пам'яті, яка використана під час виконання 50-ма потоками роботи на відлік часу API 22 та API 17.



а) б)

а) навантаження процесора з 50 запущеними сторонніми потоками та прокручення списку;
б) об'єм оперативної пам'яті, яка використана 50-ма потоками

Рисунок 6 – API 22



а) б)

а) навантаження процесора з 50 запущеними сторонніми потоками та прокручення списку;
б) об'єм оперативної пам'яті, яка використана 50-ма потоками

Рисунок 7 – API 17

Висновки. Проведено аналіз новітніх технологій та розглянуто проблеми їх використання у межах предметної області розробки програмного забезпечення додатку «Тайм-менеджер» до операційної системи Android. За результатами проведених досліджень виконано аналіз структури та методів побудови програмних додатків для операційної системи Android.

За допомогою Intellij IDEA та Android Studio створено програмний додаток, який містить у собі наступний функціонал: механізм будильника для нагадувань про діяльність; механізм відстеження часу та таймерів з вказівкою типу обмеження; графічний механізм відображення витраченого часу.

При створенні програмного продукту вирішено проблеми перезавантаження пристрою, спрацювання таймеру під час сну телефону, відновлення роботи будильників після випадкового перезавантаження, багатопоточного запуску таймерів, обробки зміни орієнтації екрану, роботи у фоновому режимі.

ЛІТЕРАТУРА

1. Sovelluksen Valmistaja Create an elegantly designed Reminder/Alarm clock application [Электронный ресурс] / Learn Android Development | Download Free Apps. – 2014. – Режим доступу: http://www.appsrox.com/android/tutorials/remindme/.
2. Отзывчивое Android-приложение или 1001 способ загрузить картинку [Электронный ресурс] / Блог компании EastBanc Technologies, Разработка под Android. – 2013. – Режим доступу: http://habrahabr.ru/company/eastbanctech/blog/ 192998.
3. Сухоруков И. Многопоточность в Java / Программирование, Параллельное программирование, JAVA. – 2012. – Режим доступу: http://habrahabr.ru/post/164487.
4. Material Design: на Луну и обратно [Электронный ресурс] / Блог компании REDMADROBOT. – 2015. – Режим доступу: http://habrahabr.ru/company/redmadrobot/blog/252773/.

Надійшла до редколегії 26.04.2016.

004.9:004.912

ДРАНИШНИКОВ Л.В., д.т.н., професор
ШКУРКО О.А., магістр

Дніпродзержинський державний технічний університет

АНАЛІЗ ТА ВИЛУЧЕННЯ ІНФОРМАЦІЇ З ТЕКСТІВ

Вступ. Комп’ютерна лінгвістика (також відоме поняття "обробка природної мови" від дослівного перекладу з англійської naturallanguageprocessing) – це надзвичайно важлива область комп’ютерних наук яка ґрунтується на знаннях інформатики, лінгвістики і, все частіше, на теорії ймовірності та статистики.

На високому рівні абстракції комп’ютерна лінгвістика використовує комп’ютери при обробці людської (природної) мови (рис.1). З одного боку цієї проблеми ми маємо те, що часто називають розумінням природної мови, де ми беремо текст в якості вхідних даних для комп’ютера, а потім він обробляє текст і робить щось корисне. З іншого боку, ми маємо те, що часто називається naturallanguagegeneration, де комп’ютер, у деякому сенсі, виробляє мову у спілкуванні з людиною (користувачем системи).

Одна з найстаріших програм і проблема великої важливості – це машинний переклад. Це проблема зіставлення речень однією мовою з реченнями іншою мовою. І це дуже-дуже складне завдання. Помітний швидкий прогрес в останні 10-20 років у цій області. Наприклад, подивимось на результат перекладу фрази, яка багатьом знайома, Googletranslate з української на англійську. Хоча переклад далекий від ідеалу, ви все одно зможете зрозуміти багато з того, що було сказано в оригіналі.

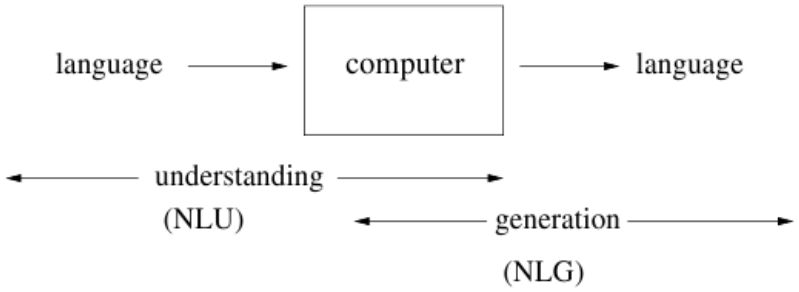


Рисунок 1 – Комп'ютерна лінгвістика як абстрактна система вводу/виводу

Другий приклад застосування – те, що часто називають вилученням інформації (information-extraction). Мета в даному випадку така: взяти якийсь текст у якості вхідних даних і створювати деякі структуровані, тобто представлені у вигляді бази даних, представлення деяких ключових частин тексту. Розглянемо це застосування на прикладі пошуку вакансій. На вході ми маємо вакансію. Вихід охоплює різні важливі аспекти цієї вакансії, наприклад, галузь, посаду, місто, компанію, зарплату, і т.і. Виходить, що ця інформація є витягом з цього документа. Ці важливі для витягу аспекти представлені у вигляді частини природної мови. У цьому і полягає проблема важливості розуміння природної мови. В якомусь сенсі це просто перетворення неструктурованого тексту на вході до структурованого представлення інформації у вигляді бази даних. Тут можна чітко сформулювати мету вирішення цієї конкретної проблеми – вилучення інформації. Після того, як ми виконали цей крок, ми можемо, наприклад, виконувати складний пошук. Наприклад, необхідно знайти всі робочі місця в рекламній індустрії з конкретним розміром заробітної платні. Такий пошук дуже важко сформулювати за допомогою звичайної пошукової системи, але при першому запуску система отримання інформації через веб-сайти видає всі вакансії, які знаходяться в інтернеті. Потім можна виконати запит до бази даних і виконувати набагато більш складні запити, ніж цей. Крім того, можна б було виконати і статистичні запити. Можливо було б запитати, яка кількість робочих місць у бухгалтерії змінилась за ці роки або кількість робочих місць в області розробки програмного забезпечення в Києві за останній рік.

Інше ключове застосування комп'ютерної лінгвістики – це анотування текстів. У такому випадку вирішується задача витягу фактів для створення анотації з одного або декількох документів. Один з прикладів реалізації алгоритмів анотування текстів – система NewsBlaster. В якості вхідної інформації вона сприймає досить багато документів про певну новину, а на виході видає анотацію до всіх новин певної тематики. Використання такої системи корисне для обробки великих масивів даних з метою вилучення ключової інформації.

Великий обсяг накопиченої інформації і висока швидкість надходження нової пред'являють все більш жорсткі вимоги до сучасних інформаційних порталів. По-перше, за умови, що масиви даних постійно розростаються, стає важко (практично неможливо) знайти потрібну інформацію; по-друге, дані часто дублюються і суперечать одне одному. Для вирішення цих проблем необхідний перехід на новий якісний рівень при обробці інформації – необхідно вести обробку на семантичному рівні, тобто враховувати зміст документів, що надходять. Така обробка тексту природною мовою забезпечується системами автоматичного аналізу, що використовують лінгвістичний підхід.

Постановка задачі. Окрім описаних вище застосувань комп'ютерної лінгвістики розглянемо деякі базові проблеми цієї області науки, які залежать від цих застосувань.

Перша проблема – це те, що називається проблемою маркування. На абстрактному рівні проблема виглядає так: в якості вхідної інформації маємо деяку послідовність символів, на виході ми повинні отримати таку ж послідовність, але кожен символ має бути співвіднесений з певною категорією. Розглянемо декілька прикладів маркування текстів. Перший – розпізнання частин мови. Як вхідну інформацію система

сприймає речення, на виході кожне слово речення промарковане залежно від частини мови. Це одна з основних задач комп'ютерної лінгвістики. Якщо вирішувати її з достатньо великою точністю, вона буде дуже корисною для застосування в широкому колі інших застосувань. Другий приклад – розпізнання іменованих сутностей. Основа прикладу дуже подібна до попереднього прикладу, на вході є певне речення, на виході система маркує певні сутності, якщо вони присутні у реченні. Найчастіше маркують такі сутності: назви компаній, географічні назви, імена людей та посади. Основною метою роботи є розробка методу й алгоритму отримання даних з текстів новин у галузі вилучення даних.

Результати роботи. *Первинна обробка тексту.* Отже, на вході у нас текст природною мовою. Аналізувати його необхідно відразу на всіх лінгвістичних рівнях: графематичному, лексичному, морфологічному, синтаксичному, семантичному.

Текст ділиться на абзаци, речення, слова. Потім слова нормалізуються – виділяється їх початкова форма. Далі проводиться повний або частковий синтаксичний розбір, визначаються залежності і зв'язки між словами в реченнях.

На перший погляд здається, що розбити текст на речення не складає ніяких труднощів. Потрібно просто орієнтуватися на знаки пунктуації, що маркують кінець речення. Але працює цей метод далеко не завжди. Адже, наприклад, крапка може означати і скорочення, використовуватися в дробових числах або URL посиланнях. Будь-який знак може використовуватися у назвах компаній або сервісів, наприклад, Yahoo! або Yandex.Maps.

З виділенням початкової форми теж не все так просто. Звичайно, в більшості випадків її можна отримати за допомогою морфологічного словника. Але його можна застосовувати далеко не завжди. Наприклад, деякі слова можуть бути як іменником, так і дієсловом. І перш, ніж звертатися до морфологічного словника, потрібно зняти омонімію. Вирішується ця проблема за допомогою корпусу мови, у якому всі слова розмічені за частинами мови, і омонімія знята. Таким чином, ґрунтуючись на контексті і статистиці вживання слова в корпусі, можна прийняти рішення, до якої ж частини мови належить те чи інше слово.

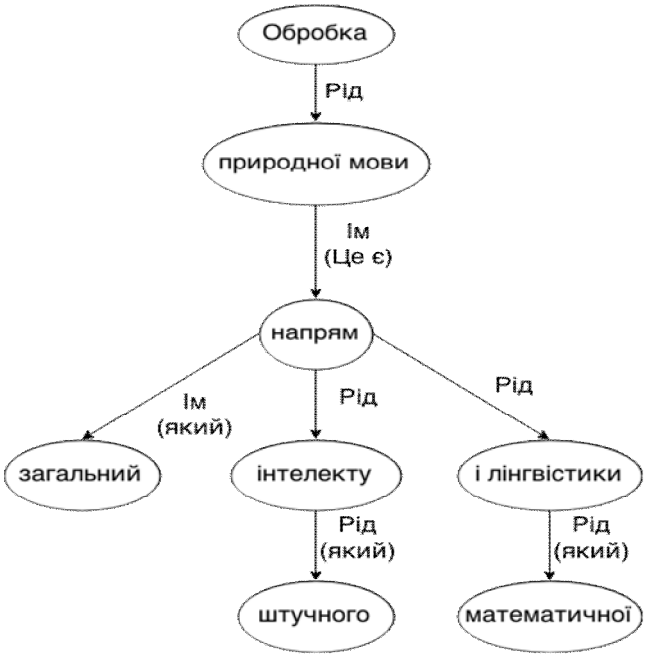


Рисунок 2 – Синтаксичне дерево як граф залежностей та відношень між словами речення, побудований за допомогою синтаксичного парсера

Наступний етап – повний або частковий синтаксичний розбір. Вибудовується граф залежностей і відносин між словами в межах речення. Ось приклад синтаксичного дерева, яке можна побудувати за допомогою синтаксичного парсера (рис.2).

Алгоритмів, які в будь-яких умовах можуть побудувати повний синтаксичний граф без помилок не існує. Однак для більшості прикладних задач TextMining достатньо і часткового розбору.

Крім морфологічної омонімії, про яку ми говорили вище, буває також синтаксична і «об'єктна» омонімія. В якості прикладу синтаксичної омонімії наведемо: «Тінь яблуні не заважає». Таке речення може бути інтерпретоване як «Тінь від яблуні не заважає» або як «Тінь

не заважає яблуні», і обидва варіанти прочитання будуть синтаксично правильні. Для вирішення такої проблеми потрібно залучати вже семантичний аналіз.

«Об'єктна» омонімія передбачає, що у двох різних реальних об'єктів можуть бути однакові найменування. Наприклад, в Україні є одразу декілька відомих людей з прізвищем Шевченко: письменник, футболіст та політик. Якщо не навчити систему розрізняти цих людей, при спробі вилучення фактів про них можуть виникати різноманітні казуси.

Витяг фактів. Коли всі ці кроки пройдені, можна переходити безпосередньо до отримання фактів. За допомогою спеціальних алгоритмів ми хочемо отримати з неструктурованого уривка тексту, в якому всі потрібні нам об'єкти та факти будуть розмічені і категоровані. Наочно уявити це можна наступним чином (рис.3).

POST	***Міністр економічного розвитку й торгівлі Айварас***	
FIO	***Абромавичус*** прогнозує приплив інвестицій в ***українську***	GEO
	економіку після закінчення війни на ***Донбасі***.	GEO
	Про це він заявив під час телеефіру на ***ТРК "Лтава"***.	COMP
	Він закликав позитивний приклад приходу як інвестор	
GEO COMP	***американської*** компанії ***Monsanto***, що вже вклала ***250 млн***	NUMBER
	доларів у будівництво заводу з виробництва насіння	
GEO	кукурудзи в ***Житомирі***. Серед негативних факторів міністр	
	зазначив падіння гривні.	
	З його слів, при сприятливих умовах поліпшення в ***українській***	GEO
	економіці можуть з'явитися вже в ***2016 році***.	DATE

Рисунок 3 – Текст з розміченими та категорованими сутностями

Умовно можна виділити три основні підходи, що застосовуються у витягу фактів: за онтологіями; спираючись на правила (Rule-based); спираючись на машинне навчання (ML – MachineLearning).

У нашому випадку онтології – це «концептуальні словники», що представляють собою структури, в яких описуються деякі поняття та/або об'єкти, відносини між ними, а також їх характеристики (рис.4).

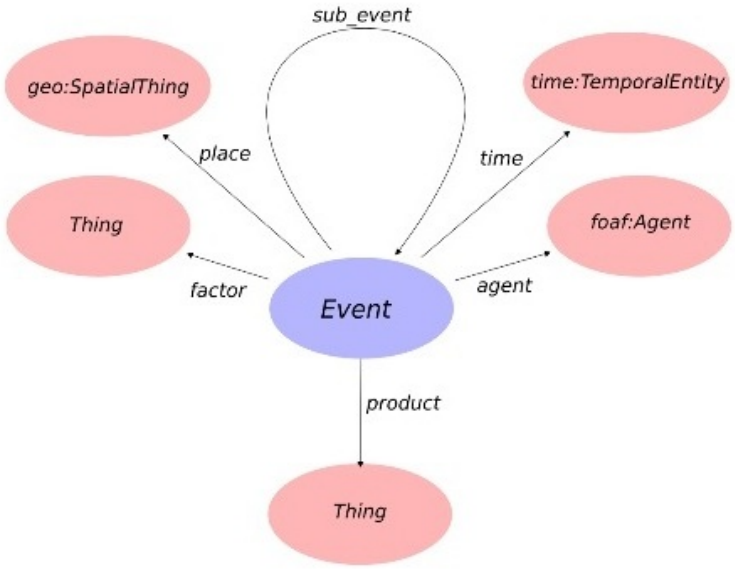


Рисунок 4 – Онтологія як концептуальний словник, що описує деякі поняття, об'єкти, відношення між ними та їх характеристики

Онтології можуть бути універсальними (в них робиться спроба описати максимально широкий набір об'єктів), галузеві (з інформацією за предметними областями) і вузькоспеціалізовані (призначені для вирішення конкретної задачі). Також можуть застосовуватися онтології об'єктів (бази знань). Найбільш яскравий приклад бази знань – це Вікіпедія.

Отже, у нас є певна онтологія. Спираючись на контексти і вже наявні списки об'єктів, можна будувати гіпотези по відношенню до об'єктів і фактів у тексті, а

далі верифікувати або відхиляти ці гіпотези. Зліва на рис.4 наведено текст, в якому кольором розмічені об'єкти, про які хотілося б отримати якусь інформацію. Як онтології застосуємо Вікіпедію (рис.5). Відправляючи туди запити за ключовими словами з нашого тексту, ми отримаємо список статей, розташований праворуч на рис.5. Червоним на рис 4 в ньому позначені статті, що відносяться відразу до кількох об'єктів.

Канцлер Німеччини	http://ua.wikipedia.org/wiki/Канцлер
Ангела Меркель	.../wiki/Канцлер_Німеччини
вважає, що суть	.../wiki/Німеччина
Мінської угоди	.../wiki/Ангела_Меркель
полягає у проведенні	.../wiki/суть
в Донецькій та	.../wiki/Мінська_угода
Луганській областях	.../wiki/Мінськ
виборів, згідно з	.../wiki/угода
українською	.../wiki/Донецька_область
Конституцією.	.../wiki/Луганська_область
	.../wiki/вибори
	.../wiki/Конституція
	.../wiki/Конституція_України

Рисунок 5 – Вікіпедія як онтологія

Тепер наша мета – відсіяти невірні гіпотези. Зробити це можна різними способами. Найчастіше застосовується машинне навчання, різні контекстні та синтаксичні фактори.

Витяг інформації за допомогою онтологій дозволяє отримати досить високу точність NER і відсутність випадкових спрацьовувань. Зняття омонімії також відбувається з високою точністю. До недоліків цього підходу можна віднести низьку повноту, адже отримати можна тільки те, що вже є в онтології. А в онтологію потрібно або додавати об'єкти руками, або вибудовувати процедуру автоматичного додавання.

Інший підхід – машинне навчання – вимагає великого обсягу ввідних даних. Потрібно максимально покрити лінгвістичною інформацією навчальну вибірку текстів: розмітити всю морфологію, синтаксис, семантику, онтологічні зв'язки. Плюси цього підходу в тому, що він не вимагає ручної праці, крім створення розміченого корпусу. Не потрібно складати правила або онтології. За необхідності така система легко перенастроюється і перенавчається. Правила виходять більш абстрактними. Однак є і мінуси. Інструменти для автоматичної розмітки українськомовних текстів поки не дуже розвинені, а існуючі – не завжди легко доступні. Корпуси повинні бути досить об'ємними, розмічені правильно, послідовно та повністю. А це досить трудомісткий процес. Крім того, якщо щось пішло не так, складно відстежити, де саме виникла помилка, і точково її виправити.

Третій підхід – підхід, заснований на правилах, тобто написання шаблонів уручну. Аналітик складає описи типів інформації, які потрібно отримати. Підхід зручний тим, що якщо в результатах аналізу виявляються помилки, дуже просто знайти їх причину і ввести необхідні зміни в правила. Найпростіше складаються правила для відносно стандартизованих об'єктів: імен, дат, назв компаній і т.і.

Вибір оптимального підходу визначається конкретною задачею. Зараз найчастіше застосовуються онтології і машинне навчання, однак майбутнє – за гібридними системами.

Для програмної реалізації була обрана задача оцінки тональності відгуків. Мета реалізації програмного засобу – створення інструменту, який буде з високою точністю

автоматично класифікувати відгуки, залишені у соціальній мережі Twitter за тональністю. В якості вхідної інформації система повинна сприймати слово або словосполучення. Під час роботи система відстежує та аналізує всі повідомлення з вхідним запитом у режимі реального часу. Як вихідну інформацію система видає знайдені повідомлення, результат сентиментального аналізу та мету даних повідомлень. Програмний засіб реалізовано мовою програмування Python з використанням API соціальної мережі Twitter та платформи NLTK. Власне алгоритм обробки відгуків базується на наївному баєсівському класифікаторі.

Класифікатор повинен пройти навчання, щоб вміти відрізняти позитивні, негативні та нейтральні відгуки. Для цього зберігається та маркується деякий набір речень (повідомлень соціальної мережі). Після цього система навчається, маркуючи тестові повідомлення.

Послідовність роботи системи та взаємозв'язки описаних вище функцій зображені на рис.6.

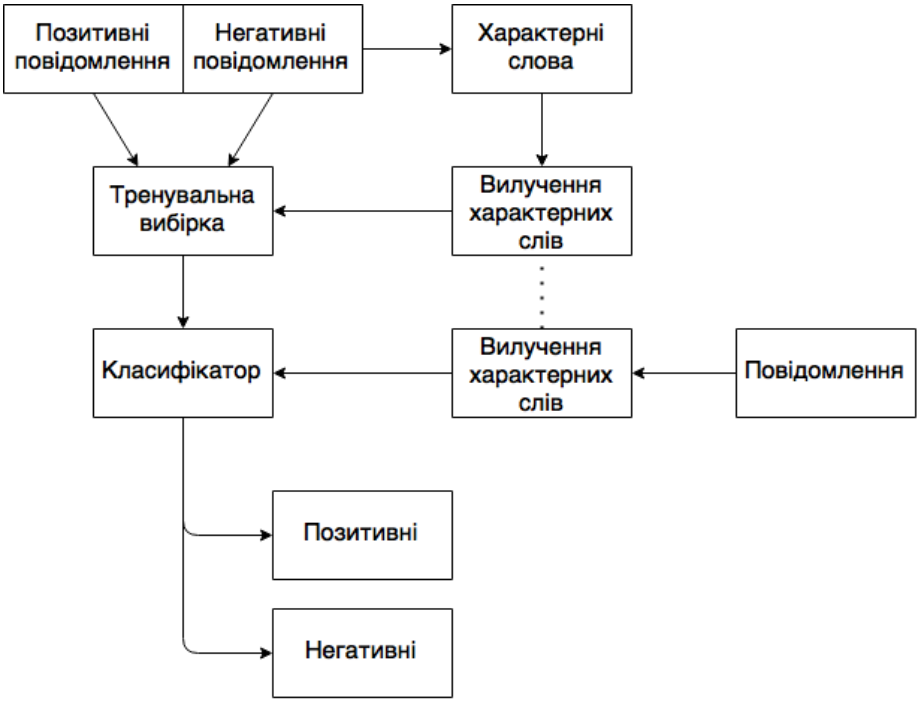


Рисунок 6 – Послідовність роботи та взаємозв'язки компонентів системи

Висновки. Розроблений програмний засіб може бути використаний для загального аналізу тональності відгуків. Але його можна адаптувати під конкретні потреби певної предметної області, наприклад, реалізувати вилучення та аналіз інформації про такі сутності, як ціна, якість, зовнішній вигляд або технічні характеристики. Також можна додати функцію вилучення іменованих сутностей, наприклад, таких як імена та прізвища, посади, назви організацій, дати або географічні назви.

ЛІТЕРАТУРА

1. Bird S. Natural Language Processing with Python. Analyzing Text with the Natural Language Toolkit / S.Bird, E.Klein, E.Loper. – O'ReillyMedia, 2009. – 504c.
2. Шумейко А.А. Интеллектуальный анализ данных (Введение в DataMining) / А.А.Шумейко, С.Л.Сотник. – Дн-вск: Белая Е.А., 2012. – 210с.

Надійшла до редколегії 26.04.2016.

УДК 004.896:347132.15

ДРАНИШНИКОВ Л.В., д.т.н., професор
ДЕНИСЕНКО В.М.*, інженер-програміст
НАЙВЕРТ О.В.*, к,т.н.

Дніпродзержинський державний технічний університет
*Наукове виробниче підприємство «УКРАГРОТЕХ», м. Дніпродзержинськ

АВТОМАТИЗОВАНА СИСТЕМА РАННЬОГО ВИЯВЛЕННЯ ЗАГРОЗИ ВИНИКНЕННЯ НАДЗВИЧАЙНИХ СИТУАЦІЙ ТА ОПОВІЩЕННЯ

Вступ. Протягом останніх 10 років в Україні виникло близько 3000 надзвичайних ситуацій, в яких більше 4000 осіб загинуло та більше 13500 осіб постраждало. Відповідно до національного класифікатора НС більше половини відносяться до НС техногенного характеру, які щорічно забирають сотні життів, завдають багатомільйонних збитків державі та населенню. Найефективнішим методом досягнення високого рівня безпеки, що відповідає світовим стандартам і вимогам діючого законодавства України, є впровадження автоматизованих систем раннього виявлення надзвичайних ситуацій та оповіщення населення (АСРВНСО).

Завдання системи АСРВНСО: відслідковувати стан джерел небезпеки на об'єктах, виявляти на початковій стадії загрозу виникнення НС (викиди газу, аміаку, витоки горючих та вибухонебезпечних рідин, руйнування будівель й обвалення покрівель тощо) та оповіщати персонал і населення, яке перебуває в зонах можливого ураження, а також сили реагування для своєчасного запобігання виникненню НС або їх ліквідації.

Уникнення великих матеріальних збитків, зниження кількості техногенних катастроф, підвищення рівня техногенної безпеки на кожному підприємстві з небезпечними об'єктами – це, насамперед, першочерговий крок у важливій спільній діяльності, метою якої є найголовніше – збереження людського життя. Для цього використовують етапність, що визначає послідовність розробки та впровадження елементів інформаційної системи.

Постановка задачі. Об'єкт дослідження – процедури розробки Автоматизованої системи раннього виявлення загрози виникнення надзвичайних ситуацій та оповіщення на території небезпечного об'єкта підприємства та використання АСРВНСО в роботі диспетчерами за допомогою програмно-апаратного комплексу Автоматизованого робочого місця (АРМ) диспетчера-оператора АСРВНСО.

Мета роботи – вивчення, аналіз галузі робіт по проектуванню та створенню АСРВНСО, розробка типового проекту АСРВНСО та розробка структури й програмного забезпечення АРМ АСРВНСО і пультів централізованого спостереження АСРВНСО в інтегрованому програмному середовищі з використанням об'єктно-орієнтованих технологій Eclipse SDK Version: 3.7.2 Build id: M20120208-0800 (c) Copyright Eclipse contributors and others 2000, 2012. сучасною мовою програмування Java.

Результати роботи. В основі роботи систем АСРВНСО лежить процес постійного (цілодобового) контролю потенційно небезпечних параметрів технічного процесу. Автоматичні датчики безперервно фіксують параметри протікання тих чи інших виробничих процесів і порівнюють їх із заданими нормами.

Якщо показники датчиків перевищують встановлені норми і наближаються до докритичних (тривога) або критичних (аварія) значень, система оповіщає про це персонал і передає сигнал на пульт оператора. Оповіщення здійснюється світловим і звуковим способом.

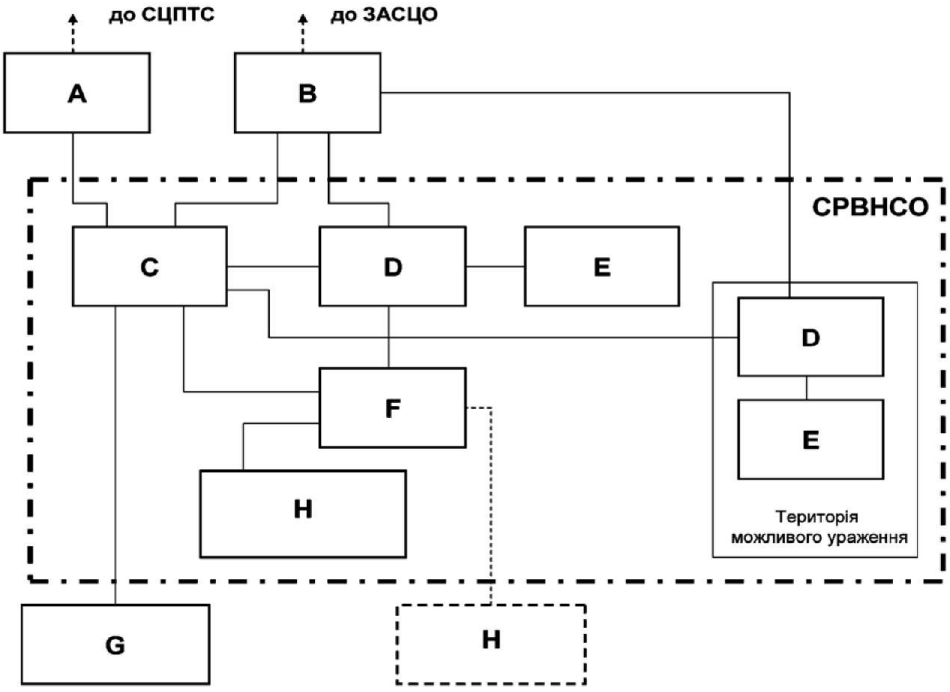
При досягненні докритичних або критичних показників система АСРВНСО

може подавати сигнал для автоматичного включення/виключення того чи іншого обладнання. Своєчасне виявлення небезпеки і точна локалізація джерела тривоги дозволяє оперативно відреагувати на ситуацію, що склалася, і запобігти аварії, яка могла б досить швидко перерости в надзвичайну ситуацію.

Системи раннього виявлення надзвичайних ситуацій, як правило, є модульними, отже, можливо підібрати оптимальний набір обладнання для кожного об'єкта. Для великих виробництв передбачено додаткові блоки розширення, за допомогою яких будується розподілена система. Системи АСРВНСО застосовуються також і в місцях громадського користування, таких як супермаркети, торгово-розважальні комплекси, бізнес-центри і т.і. Принцип роботи систем той же – контроль за допомогою спеціальних датчиків над певними параметрами. Тільки якщо у виробничих приміщеннях перевіряються параметри технічного процесу, то в місцях громадського користування контролюється стан самого будинку, його несучих частин та елементів. Датчики вловлюють найменші зміщення несучих конструкцій і повідомляють, якщо ці зсуви перевищують допустимі параметри.

Підприємства-розробники мають всі необхідні дозволи на виконання проектних та пусконалагоджувальних робіт і пропонують тільки сертифіковане обладнання з отриманням гарантованого якісного результату.

Структурну схему автоматизованої системи раннього виявлення загрози виникнення надзвичайних ситуацій та оповіщення населення й управління евакуацією людей згідно з ДБН В.2.5-76:2014 наведено на рис.1.



СЦПТС – система централізованого пожежного та техногенного спостереження;
ЗАСЦО – загальнодержавна автоматизована система централізованого оповіщення;
A – ПЦС – пульт централізованого спостереження;
B – ТАСЦО – територіальна автоматизована система централізованого оповіщення;
C – ПК – пульт керування АСРВНСО ; D – ПО – пристрій оповіщення;
E – КТЗІО – кінцеві технічні засоби інформування та оповіщення;
F – КП – комунікаційний пристрій; H – джерела первинної інформації;
G – суміжні системи забезпечення безпеки

Рисунок 1 – Структурна схема АСРВНСО

Головний критерій адекватності структурної моделі предметної області полягає у функціональній повноті розроблювального ПЗ АРМ АСРВНСО.

В основі різних методологій моделювання предметної області ПЗ АРМ АСРВНСО лежать принципи послідовної деталізації абстрактних категорій. Зазвичай, моделі будуються на трьох рівнях: на зовнішньому рівні (визначенні вимог), на концептуальному рівні (специфікації вимог) і внутрішньому рівні (реалізації вимог). Так, на зовнішньому рівні модель відповідає на запитання, що повинна робити система, тобто визначається склад основних компонентів системи: об'єктів, функцій, подій, організаційних одиниць, технічних засобів. На концептуальному рівні модель відповідає на запитання: „Як повинна функціонувати система?" Інакше кажучи, визначається характер взаємодії компонентів системи одного і різних типів. На внутрішньому рівні модель відповідає на запитання: „За допомогою яких програмно-технічних засобів реалізуються вимоги до системи?" З позиції життєвого циклу ПЗ АРМ АСРВНСО описані рівні моделей відповідно будуються на етапах аналізу вимог, логічного (технічного) і фізичного (робітника) проектування.

Для проведення етапу аналізу предметної області був обраний об'єктно-орієнтований метод. Об'єктно-орієнтований підхід використовує об'єктну декомпозицію, при цьому статична структура описується в термінах об'єктів і зв'язків між ними, а поведінка системи описується в термінах обміну повідомленнями між об'єктами. Метою методики є побудова моделі організації, що дозволяє перейти від моделі використання до моделі, що визначає окремі об'єкти, які беруть участь у реалізації функцій.

Розробка та впровадження ПЗ АРМ АСРВНСО дозволить:

- прискорити процес детального та глобального спостереження інформації від обладнання дільниць об'єктів АСРВНСО;
- забезпечити точність, достовірність, швидкість і зручність роботи при розв'язанні задач спостереження та управління АРМ АСРВНСО;
- виключити непродуктивні трудовитрати ручної технології обробки інформації;
- підвищити ефективність аналізу за станом усіх параметрів контролю, стану обладнання дільниць об'єктів АСРВНСО в цілому.

Визначення вимог до програмного засобу. Призначення програми. Програма призначена для використання на об'єктах з підвищеною небезпекою в рамках АРМ АСРВНСО диспетчерами, які проводять цілодобове спостереження за станом техногенної безпеки. *Місце функціонування.* Центральна диспетчерська підприємства, диспетчерська дільниці або об'єкта, диспетчерські пульти центрального спостереження за технічними засобами ПЦСТЗ, регіональної автоматизованої системи централізованого оповіщення РАСЦО, системи централізованого моніторингу СЦМ.

Назва програми. Програмний додаток повинен мати унікальну назву, яка б чітко й однозначно ідентифікувала програмний додаток, була інформативною та одночасно короткою. Цим умовам відповідає така назва програмного додатку – УАТ АРМ АСРВНСО, яке розшифровується як «УАТ АРМ АСРВНСО – Універсальний Аналітичний Термінал Автоматизоване робоче місце диспетчера-оператора Автоматизованої системи раннього виявлення надзвичайних ситуацій та оповіщення людей у разі їх виникнення».

Автономність. Виходячи із аналізу умов майбутнього використання програмного додатку УАТ АРМ АСРВНСО, програма не повинна бути прив'язаною до одного якогось комп'ютера, повинна бути мобільною, мати змогу працювати в перспективі з мобільного інформаційного носія. Таким чином, програмний додаток УАТ АРМ АСРВНСО повинен являти собою виконуваний комплекс, який потребує інсталяції на

комп'ютері, оскільки існують вимоги безпеки при використанні, але окремі модулі можна використовувати без інсталяції.

Адаптованість. Розробка програмного додатку не повинна викликати необхідність застосування нового комп'ютерного обладнання або програмного забезпечення. Розроблюваний програмний додаток повинен бути пристосований для роботи на існуючому обчислювальному обладнанні.

Вимоги до інтерфейсу. Аналіз складу потенційних користувачів програми показав:

1. Програма не повинна пред'являти до користувача будь-яких спеціальних знань або навичок, управління програмою має здійснюватися за допомогою стандартних команд ОС Windows з використанням клавіатури і маніпулятора «миша».
2. Термінологія програми повинна відповідати термінології МУ 6/113-30-19-83.
3. Інтерфейс повинен бути інтерактивним, користувач повинен мати можливість у ході використання змінювати типові налаштування, коригувати форми візуалізації, а сама програма повинна реагувати на помилки користувача шляхом недопущення збоїв та виведення на дисплей відповідних повідомлень.
4. Програма повинна мати інтуїтивно зрозумілий графічний інтерфейс та надавати можливість самостійно опанувати користуванням цим програмним додатком.
5. Шрифт, графічне і колірне оформлення інтерфейсу не повинні викликати стомлюваності користувача і не завдавати шкоди його здоров'ю.

Збереження результатів. Розроблюваний програмний додаток повинен мати можливість зберігати результати в електронному або друкованому вигляді.

Можливість вдосконалення. Код програми повинен бути відкритим і мати достатню кількість докладних коментарів для того, щоб за необхідності будь-який програміст міг працювати з кодом програми.

Визначення розміру екранної форми. Мова програмування Java дозволяє створювати програмний продукт як у вигляді консольного, так й у вигляді віконного додатку. Для забезпечення діалогу користувача з програмою і ведення оптимального діалогу розроблюваний програмний додаток повинен мати масштабований віконний інтерфейс.

Розміри вікон (форм) мовою програмування Java визначаються в кількості пікселів і тому було поставлено питання, який розмір повинна мати віконна форма додатку з розбиттям на допоміжні зони з уніфікованою інформацією, щоб вона цілком помістилася на екрані. Аналіз стану комп'ютеризації диспетчерських показав, що монітори комп'ютерів середніх розмірів і мають роздільну здатність 1024 на 768 точок та вище, але чим кращий монітор, тим краще сприймається інформація. До програми буде закладена можливість роздрукування результатів на принтері, тому розмір форми адаптуємо також до розмірів стандартного друкованого аркуша А4.

Ураховуючи розміри екрану, паперу для друку та виходячи з аналізу обсягів інформації, що відображається на екрані, робимо висновок, що оптимальним розміром форми програмного додатку буде повноекранна версія. Можливість змінювати розміри форми не виключаємо.

Визначення кількості екранних форм. УАТ АРМ АСРВНСО є програмно-апаратним засобом автоматизованого комплексу раннього виявлення загрози виникнення надзвичайних ситуацій та системи оповіщення АРМ АСРВНСО.

Система раннього виявлення надзвичайних ситуацій на об'єктах підвищеної небезпеки та програмно-апаратний комплекс з автоматизованих робочих місць диспетчера-оператора призначені для оперативного виявлення критичних ситуацій та запобігання їх розвитку на об'єктах підвищеної небезпеки (АЗС, АГЗС, НБ, ГБ, підприємств різних галузей), а також видачі сигналу тривоги персоналу об'єкта, оточуючим і службі ДСНС.

Комплект програмного забезпечення (ПЗ) УАТ АРМ АСРВНСО призначений для автоматизації робіт диспетчера при виявленні загрози НС або при виникненні НС, перевірки взаємодії АРМ АСРВНСО з приладами, датчиками й апаратурою комплексу АСРВНСО, забезпечує контроль технічного стану та поведінки датчиків, а також забезпечує ряд статистичних та аналітичних функцій, надає можливість віддаленого доступу (через локальну комп'ютерну мережу або модемну) до інформації, отриманої комплексом АРМ АСРВНСО через Сервер БД.

УАТ АРМ АСРВНСО відрізняється сучасною конструкцією взаємозв'язків елементів і передовим дизайном обробки та візуалізації інформації. Розробляються і використовуються інноваційні ідеї Інтерактивного інтерфейсу користувача з адаптивними смисловими підходами.

Пропонується замінити АРМ АСРВНСО новим технічно більш сучасним програмно-апаратним комплексом з удосконаленим інноваційним візуальним представленням інформації.

Проект названо УАТ АРМ АСРВНСО. Це спроба розробки перспективної моделі АРМ АСРВНСО на зміну існуючим.

Цей проект передбачає застосування підходів з чітко вираженими трьома вимогами: замість пердів і вікон – повна одноекранна форма візуалізації та спливаючі онлайнові підказки, допоміжні підказки кнопок і т.і., з додатковими незалежними вікнами відпрацювання виявленої загрози НС або відстеження ліквідації при виникненні НС; зміни в зовнішньому оформленні, інтерактивні спливаючі інформаційні дані; озвучування інформаційних даних, режимів візуалізації даних; окремо виділене модульне прискорене опитування; окремо виділений модульний аналітичний блок і занесення інформації на Сервер БД.

АРМ АСРВНСО повинне забезпечувати: настройку параметрів робочого місця; налаштування конфігурації обладнання; відображення на екрані службової та оперативної інформації; виконання команд оператора.

АРМ АСРВНСО також повинне забезпечувати: 1) ведення бази даних з технічних засобів спостережуваних об'єктів (ТСНО); 2) збільшення інформативності ТСНО за рахунок перевизначення (переопитування) подій; 3) ведення бази даних по об'єктах; 4) ведення бази даних по спрацюваннях; 5) опис складних конфігурацій систем спостережуваних об'єктів; 6) простоту зміни конфігурацій систем спостережуваних об'єктів; 7) створення сценаріїв дій щодо подій у системі спостережуваних об'єктів; 8) ведення графічної багаторівневої і багатошарової бази даних; 9) імпорт даних з БД формату АРМ АСРВНСО і АРМ ПЦСТС (пульт централізованого спостереження та моніторингу технічного стану) АСРВНСО.

Комплекс призначений для роботи в середовищі операційної системи> (ОС) Windows XP ServicePack 2 і вище і СУБД (> = 5.1 MySQLServer). Комплекс повинен забезпечувати інформаційну та мережеву сумісність з АРМ ПЦСТС АСРВНСО і АРМ ДСНС з урахуванням номерів версій і протоколів.

Взаємодія користувача і програмного додатку. Передній план загального екрана АРМ АСРВНСО запропоновано розбити на сім основних роздільних його зон, що збільшує інформативно корисний простір і покращує проведення моніторингу. Сама екранна форма має поліпшений підхід до оформлення і відображення всієї необхідної інформації, онлайн підказки, поліпшену інформативність для відпрацювання надзвичайних подій диспетчером-оператором.

Варіант оформлення екранної форми включає в себе 7 зон екрана (рис.2).

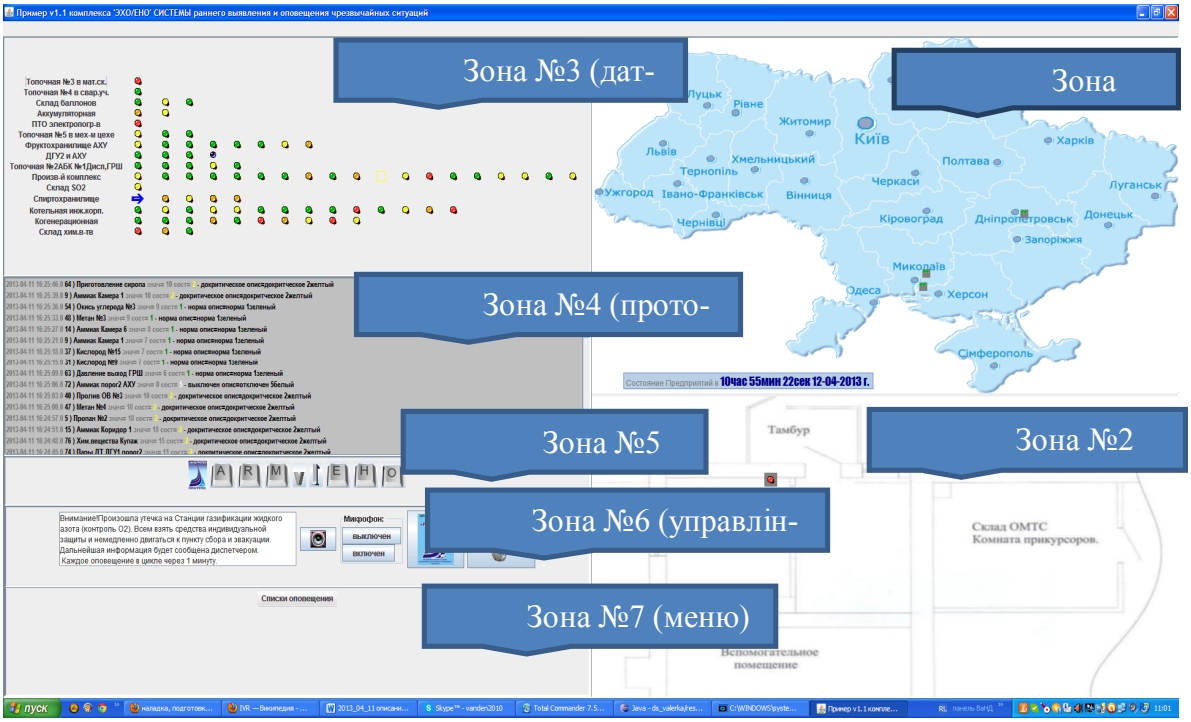


Рисунок 2 – Зони основного екрана АРМ АСРВНСО в режимі АРМ ТО, АРМ ПЦСТС АСРВНСО Диспетчера (АРМ ПЦСТС АСРВНСО) по модулю VizuAll_ЕНО

Інформація про стан датчиків на схемі ділянки (цеху) статична, а не динамічна відповідає стану на час переходу – це зроблено спеціально, щоб була можливість оцінювати ситуацію на певний час, робити копії образу екрана (скріншоти, PrintScreen екрану) для звіту про ситуацію.

Висновки. Таким чином, УАТ АРМ АСРВНСО – це програмно-апаратний комплекс для вирішення завдань виявлення, аналізу загрози НС, звукового попередження, телефонного оповіщення, розсилки факсів та SMS з підтримкою WEB-інтерфейсу (додаткова опція). Комплекс має можливість при додаванні опції в проекті додатково включати проведення професійної розсилки повідомлень різного формату (голос, факс, SMS, TTS, IVR).

Програма АРМ АСРВНСО може становити практичний інтерес для будь-яких підприємств по контролю вимог техногенної безпеки на об’єктах підвищеної небезпеки, дільницях об’єктів АСРВНСО, які мають задачі спостереження та управління АСРВНСО.

ЛІТЕРАТУРА

1. Ідентифікація потенційно-небезпечного об’єкта: Наказ Міністерства України з питань НС від 23.02.2006 р. № 98 «Про затвердження Методики ідентифікації потенційно небезпечних об’єктів». – 2006. – 100с.
2. Ідентифікація об’єкта підвищеної небезпеки: Закон України від 18.01.2001 р. № 2245 «Про об’єкти підвищеної небезпеки», ст.9; Постанова КМУ від 11.07.2002р № 956. «Про ідентифікацію і декларування безпеки об’єктів підвищеної небезпеки», зі змінами, внесеними згідно з Постановою КМУ № 990 від 21.09.2011р. – 100с.

Надійшла до редколегії 26.04.2016.

УДК 004

ЗАВГОРОДНІЙ В.В., к.т.н., доцент
ЯЛОВА К.М., к.т.н., доцент

Дніпродзержинський державний технічний університет

ТЕХНОЛОГІЇ ТА ЗАСОБИ ПРОГРАМНОЇ РЕАЛІЗАЦІЇ СИСТЕМ ЕЛЕКТРОННОГО НАВЧАННЯ

Вступ. Швидкий розвиток інформаційних технологій (ІТ) сприяє їх впровадженню в усі сфери життя, не виключаючи й освіту. Інтенсивна комп'ютеризація навчальних закладів та популяризація самоосвіти стали підґрунтям для застосування інформаційних комп'ютерних технологій в галузі електронного навчання (ЕН).

Високі вимоги ринку праці та розрізненість методик навчання і контенту навчальних програм вищих навчальних закладів (ВНЗ) України призвели до необхідності застосування компетентнісного підходу у вищій освіті, який передбачає набуття студентами загально встановлених базових та професійних компетентностей для кожної кваліфікації [1]. Водночас зменшення аудиторного навантаження робить вкрай актуальною необхідність інформаційної підтримки навчання, наприклад, засобами систем електронного навчання СЕН (e-learning system), а їх ключовий принцип «освіта через усе життя» (life-long education [2]) дозволяє вдосконалювати та розвивати свої знання постійно.

Постановка задачі. Побудувати уніфіковану модель архітектури СЕН, визначити способи програмної реалізації її прошарків, провести порівняльний аналіз безкоштовних платформ для ЕН, визначити можливі проблеми впровадження СЕН в умовах сьогодення та шляхи їх вирішення.

Результати роботи. Розробка СЕН в даний час є актуальним напрямком у розвитку ІТ, які спрямовані на допомогу викладачу і студенту в навчальному процесі [3]. Для забезпечення ефективного надання навчальних послуг на відстані СЕН повинна бути реалізована, враховуючи функціональні та якісні вимоги щодо програмного коду. Враховуючи вимоги веб-орієнтованої програмної реалізації, необхідності багатокористувального режиму та наявності сховища навчальних матеріалів, була розроблена уніфікована архітектура СЕН та визначені програмні засоби і технології її реалізації:

1) User Access – рівень користувального доступу до даних, що включає графічний інтерфейс системи, який передається через браузер. Класичним підходом до створення веб-інтерфейсу є використання мови гіпертекстової розмітки HTML при застосуванні CSS і JavaScript для створення єдиного дизайну та візуальних ефектів сторінок. Переваги цього підходу полягають у простоті та високій швидкості розробки вебдизайну, але різна реалізації програмних специфікацій коду в різних браузерах викликає проблеми підтримки такої розробки. Альтернативою цьому підходу є застосування Silverlight, Adobe Flash або Java-аплетів, оскільки більшість браузерів підтримують ці технології за рахунок використання плагінів. Недоліками цього підходу є високий поріг входження до технологій та підвищені вимоги до використання ресурсів. На фоні цих недоліків набуває популярності застосування асинхронних запитів AJAX, використання яких дає змогу виділити частини веб-інтерфейсу, які не будуть перезавантажуватися через веб-сервер;

2) Common Services – рівень сервісів, що забезпечує збереження ідентифікуючих даних користувачів та взаємодії між всіма суб'єктами навчання. Також до цього рівня відносяться програмні механізми побудови розкладів, графіків навчання, формування календарів здавання завдань, тощо. Концепція веб-сервісів може бути реалізована за допомогою різноманітних технологій, але загальний стандарт реалізації веб-сервісів встановлено World Wide Web Consortium. Найпоширенішим способом реалізації веб-

сервісів є створення XML-сервісів, взаємодія з якими реалізується засобами Simple Object Access Protocol (SOAP). Веб-сервіси також можуть бути реалізовані із застосуванням Java-технологій, наприклад, AXIS (Apache eXtensible Interaction System);

3) Learning services which provide core functionality – рівень створення, збереження та модифікації інформаційний навчальних ресурсів. На цьому рівні реалізовуються програмні механізми логіки використання та керування навчальними матеріалами з боку кожної ролі користувачів. Якщо веб-додаток реалізовується із використанням архітектурного партерна MVC, то реалізація цього рівня архітектури системи буде представлена у вигляді набору контролерів систем. В якості мови програмної реалізації контролерів може бути застосована будь-яка мова програмування: C#, Java, PHP, Phyton;

4) Database – рівень збереження даних СЕН. На цьому рівні передбачається програмна реалізація сховища даних. Засобами реалізації рівня сховища даних можуть бути системи управління реляційними базами даних (БД) MSSQLServer, MySQL тощо. Швидкими темпами набувають популярності технології NoSQL-databases з різними видами представлення сховищ даних: сховище «ключ-значення», сховище типу BigTable, документо-орієнтовані СУБД, БД на основі графів. Окрім цього в якості сховища даних можуть застосовуватися дані, які збережені в форматі XML;

5) Infrastructure – рівень, що містить програмні механізми клієнт-серверної обробки даних, обробки повідомлень мережі та керування протоколами мережі Інтернет. Найпоширенішим транспортним протоколом, що застосовується для передачі даних в системі «клієнт-сервер» є протокол HTTP. Для передачі пошти може застосовуватися протокол SMTP (Simple Mail Transfer Protocol). Для передачі файлів в текстовому та бінарному форматі зі спеціального файлового серверу на комп'ютер користувача застосовується протокол FTP (File Transfer Protocol) та інші.

Ринок IT в сфері надання освітніх послуг представлений багатьма безкоштовними навчальними платформами, серед яких найвідомішими є: Moodle, Sakai, ATutor, Claroline, Dokeos, ILIAS, LAMS, OpenACS, WRC e-Education System, Прометей, eLearning Server 4G, Blackboard Learn, REDCLASS Pro та TrainingWare. Розглянемо їх функціональні і технічні характеристики, встановимо ряд суттєвих критеріїв якості та наведемо результати їх порівняльного аналізу.

1. Moodle – це навчальна платформа, яка призначена для об'єднання студентів та викладачів в єдину інтегровану систему, що вільно розповсюджується за ліцензією GNU (General Public License) як веб-додаток та надає можливість створювати сайти для on-line-навчання [4]. Функціональні можливості платформи Moodle:

- забезпечує роботу з БД MySQL, MSSQL, Oracle, PostgreSQL, Interbase, Foxpro, Access, ADO, Sybase та ODBC;
- просто встановлюється на будь-яку PHP-платформу (Linux, Windows, Unix, MacOS);
- має вбудований редактор, який дозволяє викладачу створювати web-сторінки дисциплін, не володіючи мовою гіпертекстової розмітки HTML;
- підтримує функцію колективного редагування текстів;
- має широкі можливості для обміну файлами різних форматів, розсилки пошти, форуму та чату, on-line опитувань та дискусій;
- має можливість зберігати повну та достовірну інформацію про роботу студентів: їх активність, відвідуваність, оцінки, коментарі, повідомлення в чатах та форумах;
- інтерфейс платформи дає можливість роботи людям з різним рівнем підготовки в галузі IT та різними фізичними можливостями.

2. Sakai – це безкоштовний продукт з повністю відкритим вихідним кодом, який займає лідируючі позиції серед систем підтримки освіти. Найбільші і найбільш авторитетні університети світу обирають його за низьку вартість використання, надійність і можливість тонкого настроювання системи під їх конкретні вимоги. Серед ВНЗ, що ви-

користовують Sakai, можна назвати Стенфордський та Єльський університети, Каліфорнійський університет в Берклі, університети Оксфорда та Кембриджа [5]. Можливості платформи Sakai:

- має високий ступінь захисту (заснований на технології Java);
- складається з окремих модулів, необхідний набір яких обирається кожним студентом самостійно;
- розповсюджує навчальні матеріали в локальній мережі навчального закладу або через Інтернет;
- має внутрішній інструмент, який дозволяє проводити on-line лекції або семінари з можливістю супроводу показом презентації;
- має синхронізовані з календарем інструменти, які дозволяють видавати та приймати завдання в певні часові інтервали;
- забезпечує роботу з БД MySQL та Oracle;
- підтримка PHP-платформ (Linux, Windows, Unix, MacOS).

3. ATutor – це вільно поширювана web-орієнтована система управління навчальним контентом, розроблена з урахуванням ідей доступності й адаптованості [4]. Викладачі можуть швидко збирати, структурувати зміст навчального матеріалу для проведення занять on-line. Можливості платформи Atutor:

- забезпечує роботу з БД MySQL;
- підтримка PHP-платформ (Linux, Windows, Unix, MacOS);
- дає можливість адміністраторам керувати курсами (навігація, пошук, засоби комунікації), даними користувачів (їх права) та загальними параметрами системи (використання додаткових модулів, розробка власних шаблонів інтерфейсу);
- дає можливість студентам редагувати свої персональні дані, переглядати курси, проходити тестування та опитування, брати активну участь у спілкуванні (чат, форум, конференції, електронна пошта, коментарі), завантажувати та обмінюватися файлами;
- дає можливість викладачам створювати курси та визначати права доступу до них, завантажувати у різних форматах навчальні матеріали, створювати та керувати тестами.

4. Claroline LMS (Classroom Online Learning Management System) – це платформа для електронного навчання (eLearning) та електронної діяльності (eWorking) з відкритим кодом, яка дозволяє викладачам створювати ефективні on-line-курси та керувати процесом навчання і спільними діями через мережу. За допомогою Claroline можна створювати, редагувати та управляти всіма навчальними матеріалами. Також присутня функція розмежування доступу, система контролю досягнень студентів та модуль авторизації. Можливості платформи Claroline [6]:

- забезпечує роботу з БД MySQL з функцією розмежування прав доступу;
- підтримка PHP-платформ (Linux, Windows, Unix, MacOS);
- дозволяє викладачам створювати ефективні on-line-курси;
- дозволяє керувати процесом навчання і спільними діями через мережу Інтернет;
- дозволяє створювати навчальні матеріали в різноманітних форматах (текст, PDF, HTML, відео), редагувати та управляти ними;
- має можливість переглядання статистики активності користувачів;
- наявність системи контролю активності та прогресу навчання студентів.

5. Dokeos – це платформа побудови сайтів електронного навчання, яка заснована на гілці Claroline LMS та є клоном вільно поширюваного програмного продукту, розробленого деякими членами первісної команди розробників Claroline LMS, які задумали змінити орієнтацію додатка. Зараз Dokeos буде потрібен скоріше організаціям, ніж університетам, тому що більше орієнтований на професійну клієнтуру, наприклад, на персонал підприємства. Dokeos безкоштовний і залишиться таким надалі, оскільки ліцензія

Claroline LMS (GNU/GPL) визначає, що гілки підпадають під ту ж ліцензію [6]. Функціональні та апаратні характеристики Dokeos:

- забезпечує роботу з БД MySQL;
- підтримує PHP-платформи (Linux, Windows, Unix, MacOS);
- дозволяє імпортувати офісні документи (Word, PowerPoint, Excel);
- дозволяє створювати веб-сторінки за допомогою шаблонів;
- дозволяє вести облік та контроль успішності студентів;
- дозволяє створювати on-line-курси;
- дає можливість створювати відеоконференції.

6. ILIAS (Integriertes Lern Informations und Arbeitskooperations System) – це вільна система управління навчанням. Система особливо поширена у ВНЗ Німеччини. Відповідає стандарту SCORM (Sharable Content Object Reference Model – «модель спільного використання розповсюдженого контенту») – збірка специфікацій та стандартів, яка розроблена для СЕН. Програмне забезпечення публікується під ліцензією GNU і може працювати на будь-якому сервері, який підтримує PHP та MySQL. Можливості платформи ILIAS:

- має велику кількість інструментів для спілкування (чати, форуми, блоги, тощо);
- має свою внутрішню систему для обміну повідомленнями;
- має можливість об'єднувати користувачів в групи;
- має можливість формувати on-line-курси у вигляді HTML-файлів;
- має конструктор тестів, який підтримує різні типи питань.

7. LAMS (Learning Activity Management System) – це система управління послідовністю навчальних дій, яка надає викладачам візуальні засоби для розробки структури навчального процесу з можливістю задавати послідовність видів навчальної діяльності. LAMS надає викладачу інтуїтивно зрозумілий інтерфейс для створення освітнього контенту, який може містити різні індивідуальні завдання, завдання для групової роботи та фронтальну роботу з групою учнів [7]. Можливості платформи LAMS:

- забезпечує роботу з БД MySQL;
- підтримка PHP-платформ (Windows та MacOS);
- має високу ступінь захисту (заснована на технології Java).

8. WRC e-Education System – це СЕН, що організовує всі стадії навчального процесу, підтримуючи його методичне і технічне забезпечення. Можливості платформи WRC e-Education System:

- низький поріг входу використання та інтуїтивно зрозумілий інтерфейс;
- широкі можливості контролю і управління навчальним процесом;
- підтримка форматів SCORM 2004, SCORM 1.2 та LRM;
- можливість підключення додаткових модулів;
- можливість інтеграції з іншим ПЗ;
- високий ступінь надійності і безпеки системи;
- низькі вимоги до апаратної конфігурації сервера та клієнтського терміналу.

9. Прометей – це СЕН, за допомогою якої можна побудувати в Інтернет віртуальний університет і проводити дистанційне навчання великої кількості слухачів, автоматизувавши при цьому весь навчальний цикл – від прийому заявок до видачі сертифікату. Можливості платформи Прометей:

- дружній інтерфейс користувача та простота освоєння і експлуатації;
- відсутність ліцензій на клієнтські місця;
- можливість використання методики on-line-навчання, яка базується на командній роботі;

- висока продуктивність і масштабованість при збільшенні кількості користувачів і навантаження системи;
- можливість використання графіки та мультимедіа в тестах;
- можливість об'єднання декількох систем в єдине освітнє середовище;
- можливість інтеграції з бухгалтерськими, інформаційними та ERP-системами;
- невисокі вимоги до ресурсів сервера і клієнтських місць СЕН.

10. REDCLASS Pro – це комплекс програмно-апаратних засобів, навчальних матеріалів і методик навчання, які дозволяють дистанційно навчатися, підвищувати кваліфікацію, контролювати знання в будь-яких галузях діяльності людини, а також отримувати практичні навички експлуатації та управління програмними продуктами, обладнанням і технологіями. Можливості платформи REDCLASS Pro:

- підтримка управління навчальним процесом, навчання з можливістю контролю знань і збором статистичних результуючих даних;
- підтримує завантаження курсів в міжнародних стандартах подання контенту SCORM 1.2, SCORM 2004 і AICC, що дозволяє використовувати в процесі навчання навчальні матеріали, створені в сторонніх засобах розробки;
- надає можливість емулювати роботу з об'єктом навчання з використанням термінального або графічного інтерфейсу;
- має систему ролей користувачів, яка дозволяє враховувати особливості будь-якого процесу навчання та розмежувати права доступу до навчальних матеріалів;
- має вбудовану систему взаємодії учасників освітнього процесу в режимі реального часу (чати) і в асинхронному режимі (електронна пошта, форуми);
- має вбудовану систему проведення анкетування користувачів.

На рис.1 наведено результати порівняльного аналізу характеристик описаних СЕН, що проводився з метою визначення СЕН на теренах ВНЗ України.

Рисунок 1 – Порівняльний аналіз характеристик СЕН

Виходячи з даних, наведених на рис.1 та при описі кожної СЕН, можна дійти висновку, що навчальна платформа Moodle містить найбільшу кількість позитивних якісних характеристик. Однак, імплементація Moodle в рамках ВНЗ України може зіштовхнутися з існуючими проблемами технічного супроводу системи, необхідності оновлення серверів та мережі, низької вмотивованості студентів та викладачів до застосування

системи, низьким рівнем ІТ-компетенцій учасників навчання, тощо. Можливими шляхами рішення описаних проблем може бути проведення наступних основних заходів.

1. Технічні заходи:

- розгортання серверу та отримання доменного імені для доступу до системи;
- налагодження «клієнт-серверної» обробки даних.

2. Організаційні заходи:

- визначення меж відповідальностей викладачів, студентів та адміністраторів системи;
- створення корпоративних правил, рекомендацій щодо використання системи та створення навчального контенту;
- визначення рівня доступу до операцій керування навчальним контентом: додавання, видалення, редагування матеріалів, тестів, завдань тощо;
- розробка єдиного шаблону електронного представлення навчальних матеріалів ВНЗ;
- навчання студентів та викладачів, заохочення та мотивація для роботи з системою.

3. Заходи програмної підтримки системи:

- налагодження системи керування БД MySQL;
- створення та заповнення БД навчальними матеріалами та допоміжними відомостями;
- розгортання платформи Moodle;
- створення посилання на систему в рамках сайту ВНЗ;
- при необхідності – створення додаткових програмних модулів засобами мови програмування PHP та вбудова їх до стандартної архітектури Moodle.

Висновки. У роботі представлено узагальнену модель архітектури сучасної СЕН. Для кожного рівня архітектури СЕН встановлено засоби і технології його програмної реалізації. З метою визначення найбільш привабливого варіанту СЕН для застосування у ВНЗ наведено результати порівняльного аналізу безкоштовних платформ для ЕН. Визначено та описано комплекс заходів, що можуть мінімізувати негативні явища під час впровадження та використання системи.

ЛІТЕРАТУРА

1. Завгородній В.В. Концепція створення єдиного інформаційного освітнього простору України на прикладі дистанційного навчання ІТ-студентів / В.В.Завгородній, К.М.Ялова // Вісник Кременчуцького національного університету імені Михайла Остроградського. – Кременчук: КрНУ. – 2014. – № 2(85). – С.112-118.
2. Завгородний В.В. Перспективы использования дистанционного образования для обучения ІТ-студентов в Днепродзержинском государственном техническом университете / В.В.Завгородний, Е.Н.Яловая // Вісник Кременчуцького національного університету імені Михайла Остроградського. – Кременчук: КрНУ. – 2015. – № 1(90). – С.154-161.
3. Yalova K. Challenges and prospects in development of e-learning system for IT students / K.Yalova, M.Romanyuha, L.Sorokina // Int. J. Cont. Engineering Education and Life-Long Learning, 2016. – Vol. 26. – No. 1. – P.25-43.
4. Богомолов В. А. Обзор бесплатных систем управления обучением [Електронний ресурс] / В.А.Богомолов // Educational Technology & Society. – 2007. – №10 (3).
5. Electronic resource. Access point: http://lmsware.ru/sakai/.
6. Electronic resource. Access point: http://www.claroline.net/.
7. Electronic resource. Access point: http://www.lamscommunity.org.

Надійшла до редколегії 26.04.2016.

УДК 519.688

АКСЬОНОВ В.С., аспірант
ОЛІЙНИК Л.О., к.ф-м.н, доцент

Дніпродзержинський державний технічний університет

ВИКОРИСТАННЯ ЕЛЕМЕНТІВ ПАРАМЕТРИЧНОГО ПРОГРАМУВАННЯ ПРИ МОДЕЛЮВАННІ ПРОЦЕСУ ФОРМУВАННЯ НАВЧАЛЬНОГО ПЛАНУ

Вступ. В роботах авторів [1, 2] наведено метод моделювання навчального плану, що базується на поєднанні елементів метода Гоморі та сіткового планування. Цей метод дозволяє оптимізувати процес формування навчального плану ВНЗ та спростити роботу відповідальної особи. Складність задачі не дає змогу визначити єдиний оптимальний розв'язок, а лише деякі з припустимих розв'язків, що задовольняють нормативам та містять достатню кількість годин для того, щоб отриманий план був прийнятним для використання в навчальному процесі.

Однак, недоліком вищезгаданого методу є те, що на певних етапах він виконується в ітеративному режимі, шляхом додавання на кожному кроці нових навчальних дисциплін до вже розташованих у плані. Це призводить до того, що навчальні дисципліни, які розподіляються раніше, мають більш широкі межі для розподілення і займають більше годин, тим самим мають великий вплив на наступні навчальні дисципліни і залишають для них менше вільних годин для розподілення. Таким чином, виникають ситуації, коли певній навчальній дисципліні виділяється кількість годин близьку до її максимальної межі. Одночасно з цим іншим навчальним дисциплінам не вистачає вільних годин розподілення. З іншого боку, надавши вищеназваній навчальній дисципліні менше годин, але в межах мінімуму годин, можливо звільнити години для тих навчальних дисциплін, яким не вистачало вільних годин. З іншого боку, мінімізація розподілення годин може призвести до іншої вади: план матиме багато вільних годин, які можливо було використати для виділення навчальним дисциплінам більше годин, дозволяючи покращити якість вивчення навчальних дисциплін.

Постановка задачі. У даній роботі пропонується певне вдосконалення попереднього методу шляхом включення елементів параметричного програмування, а саме шляхом варіації кількості доступних для розподілення годин у кожному семестрі. Цей підхід дозволить отримувати різні плани і обирати серед них такий, що містить максимальну кількість повністю розподілених навчальних дисциплін та максимальну кількість лекційних годин.

Умовні позначення:

S − множина навчальних дисциплін, що необхідно розподілити;

s − навчальна дисципліна, яку необхідно розподілити;

$H_{\min}(s)$ − мінімальна кількість годин, які мають бути відведені для навчальної дисципліни s на аудиторну роботу;

$H_{\max}(s)$ − максимальна кількість годин, які мають бути відведені для навчальної дисципліни s на аудиторну роботу;

$R(s)$ − множина навчальних дисциплін, що мають бути прочитані до навчальної дисципліни s;

Π − масив годин навантаження для навчальних дисциплін;

$x(s,i)$ − кількість годин навчальної дисципліни s у i−му семестрі;

$w(i)$ − кількість тижнів у i−му семестрі;

$t(i)$ − максимальна кількість годин на тиждень у i−му семестрі;

$W(s)$ − вага навчальної дисципліни s;

g − група, множина навчальних дисциплін з рівною вагою;

G − множина груп, що ще не були розподілені;

$index(g)$ − індекс групи g в множині G, відсортованій за зростанням значення ваги предметів, що належать до групи;

$countGroup(G)$ − кількість груп g, що належать до групи G;

$countSubject(g)$ − кількість навчальних дисциплін, що належать до групи g;

$\delta t(i,g)$ − величина, на яку варіюється $t(i)$ для групи g.

Задана множина навчальних дисциплін S. Необхідно розподілити всі навчальні дисципліни $s \in S$ у плані Π так, щоб:

$$\forall s \in S, H_{\min}(s) \le \left(\sum_{i=1}^{8} x(s,i) \times w(i) \right) \le H_{\max}(s); \quad (1)$$

$$\forall i \in [1;8], \sum_{s \in S} x(s,i) \le t(i). \quad (2)$$

У даній постановці задача є класичною задачею цілочисельного програмування.

Крім названих вище умов є ще умова того, що навчальні дисципліни $s_r \in R(s)$ повинні бути прочитані до навчальної дисципліни s.

Результати роботи. *Алгоритм методу.*

1. Для всіх $s \in S$ обчислюємо вагу за формулою:

$$W(s) = 1 + \sum_{s_r \in R(s)} W(\mathrm{s}_r). \quad (3)$$

Такий механізм гарантує, що вага навчальної дисципліни s щонайменше на одиницю більша за вагу навчальної дисципліни s_r, від якої залежить s,

2. Виконується сортування навчальних дисциплін $s \in S$ та розподіл їх на групи з рівною вагою.
3. Серед груп G обирається така група g, що

$$W(s \in g) = \min_{s \in S} W(s). \quad (4)$$

Обирається певне значення $\delta t(i,g)$.

4. Для $\forall i \in [1;8]$ підраховуємо значення

$$t(i) = \left(\mathrm{t}_{\max}(i) - \sum x(s,i) \right) \times \delta t(i,g). \quad (5)$$

5. Для кожного $s \in g$ будуємо комплект рівнянь обмежень, що враховують умови (1)-(2) відповідно до правил побудови рівнянь обмежень лінійного програмування (ЛП).
6. Отримана задача ЛП розв'язується за допомогою симплекс методу.
7. Якщо одна чи декілька цільових змінних має не цілий розв'язок, обирається така, що:

$$\{\mathrm{x(s,i)}\} = \max_{s \in g; i \in [1;8]} \{x(\mathrm{s,i})\}; \quad (6)$$

якщо таких змінних немає, виконується перехід до кроку 11.

8. Для обраного $x(s,i)$ будується додаткове обмеження за формулою

$$(-1) \times \{\mathrm{a}_0\} + \sum_{i=1}^{n} (-1) \times \{\mathrm{a}_i\} \times \begin{cases} \{\mathrm{a}_i\} > 0; 1 \\ \{\mathrm{a}_i\} \le 0; \dfrac{\{\mathrm{a}_0\}}{\{\mathrm{a}_0\} - 1} \end{cases}, \quad (7)$$

де a_0 – значення цільової змінної,

a_i – значення у відповідних стовпцях симплекс таблиці,

$\{a_0\}$ – дробова частина змінної.

9. Виконується перехід до кроку 7.
10. Група g видаляється з множини G. Якщо $G \neq \varnothing$, робота алгоритму закінчена, інакше – виконується перехід до кроку 3.

Для використання цього методу необхідно визначитися з правилами вибору $\delta t(i,g)$ та кількістю різних наборів $\delta t(i,g)$, які перевіряються. Також слід визначити функцію оцінки якості плану після його створення.

Наприклад, для оцінки якості навчального плану створеного алгоритму можливо використати функцію, що обчислює суму кількості розподілених годин навчальних дисциплін, але має штраф за вихід кількості годин за межі максимальної та мінімальної кількості:

$$\forall s \in S, c(s) = \sum_{i=1}^{8} x(s,i) \times w(i); \tag{8}$$

$$H = \sum_{s \in S} \begin{cases} c(s) > H_{\max}; -100 \\ c(s) > H_{\min}; -100 \\ c(s) \end{cases}. \tag{9}$$

Таким чином, обравши план з найбільшим значення H, ми отримаємо план з найбільшою кількістю розподілених годин та найменшою кількістю навчальних дисциплін, що необхідно розподіляти в ручному режимі оператору.

Для вибору $\delta t(i,g)$ можна, наприклад, взяти одну з наведених нижче формул:

1) кожній групі під час розподілу надається лише 5% від загальної кількості вільних годин, що на цьому кроці доступні для розподілу:

$$\delta t(i,g) = 5\%; \tag{10}$$

2) для кожної групи надавати трохи більше простору, ніж для попередньої, таким чином, щоб перша група отримала мінімум годин для розподілення, а остання група мала весь доступний простір для розподілу, дозволяючи предметам, що розподіляються останніми, отримати максимум вільного місця для їх розподілення:

$$\delta t(i,g) = \left(1 - \frac{\left(countGroup(\mathrm{G}) - \mathrm{index(g)}\right)}{countGroup(G)}\right) \times 100\%; \tag{11}$$

3) кожній групі навчальних дисциплін надається відсоток від максимальної кількості годин пропорційний кількості навчальних дисциплін у цій групі:

$$\delta t(i,g) = \left(\frac{countSubject(g)}{\sum_{gg \in G} countSubject(gg)}\right) \times 100\%; \tag{12}$$

4) кількість годин, що надаються кожній групі, пропорційна сумі нормативних годин навчальних дисциплін, що належать групі:

$$\delta t(i,g)=\left(\frac{\sum\limits_{s\in g}\left(\frac{H_{\max}(s)+H_{\min}(s)}{2}\right)}{\sum\limits_{gg\in G}\sum\limits_{sg\in gg}\left(\frac{H_{\max}(sg)+H_{\min}(sg)}{2}\right)}\right)\times 100\%. \qquad (13)$$

Висновки. Наведена модифікація методу генерації навчального плану дозволяє зменшити кількість роботи, що необхідно буде виконати оператору по завершенню виконання автоматизованого розподілення навчальних дисциплін. Якщо буде визначено достатню кількість різних правил для $\delta t(i,g)$, ймовірність необхідності втручання оператора до автоматично розподіленого плану буде мінімальною, і в більшості випадків такі зміни не будуть потрібні.

Але, з іншого боку, додавання нового набору правил призведе до пропорційного зростання часу автоматичного розподілення. Цей недолік компенсується тим, що навіть досить великий набір правил на сучасних обчислювальних системах може бути обраховано за значно менший проміжок часу, ніж час, необхідний оператору для того, щоб ввести всі необхідні зміни самотужки.

ЛІТЕРАТУРА

1. Аксьонов В.С. Автоматизація розробки навчального плану вищого навчального закладу освіти / Аксьонов В.С., Олійник Л.О. // Проблеми інноватизації вищої професійної освіти: І міжнар. наук.-метод. конф., 3-5 червня 2013 р.: матеріали конференції. – Дніпродзержинськ: ДДТУ, 2013. – С.54-55.
2. Аксьонов В.С. Автоматизація оптимального планування навчальної діяльності вищих навчальних закладів / Аксьонов В.С. // Інновації у вищій освіті – комунікація та співпраця у сучасному університетському середовищі за допомогою специфічних цифрових інструментів: міжнар. наук.-метод. конф., 23-24 червня 2015 р.: матеріали конференції. – Дніпродзержинськ: ДДТУ, 2015. – С.117-119.

Надійшла до редколегії 04.05.2016.

УДК 004.031.43

ЯШИНА К.В., к.т.н., доцент

Дніпродзержинський державний технічний університет

АВТОМАТИЗОВАНА СИСТЕМА УПРАВЛІННЯ ВІДДІЛОМ АСПІРАНТУРИ СУЧАСНОГО ВИЩОГО НАВЧАЛЬНОГО ЗАКЛАДУ

Вступ. Сучасний вищий навчальний заклад (ВНЗ) має інформаційні системи, пов'язані з бухгалтерією, обігом товарів та послуг. Інформаційні системи охоплюють сукупність засобів та методів, що дозволяють користувачу збирати, зберігати, передавати і обробляти відібрану інформацію. На сьогоднішній день найбільш ефективними є інформаційні системи, в основу яких покладені бази даних (БД). Такі системи забезпечують надійне збереження інформації та своєчасний доступ до неї.

Доцільно застосувати інформаційну систему, яка базується на сучасній БД, й для автоматизації роботи відділу аспірантури вищого навчального закладу, так як відділ

аспірантури зберігає велику кількість інформації про керівників (посада, вчене звання, кількість аспірантів, наукова спеціальність, місце роботи і т.д.) та аспірантів (кафедра, місце роботи, отримана по закінченню університету та майбутня наукова спеціальності, тема дисертаційної роботи, кількість публікацій, кількість складених кандидатських іспитів і т.д.).

Постановка задачі. Задачею дослідження є аналіз та описання предметної області (роботи відділу аспірантури сучасного університету), проектування та створення бази даних (БД), організація SQL-запитів та розробка клієнтського додатка для роботи з БД.

Результати роботи. Аспірантура є формою підготовки науково-педагогічних та наукових кадрів вищої кваліфікації та відкривається при вищих навчальних закладах третього або четвертого рівнів акредитації і прирівняних до них закладах післядипломної освіти. У ВНЗ аспірантура діє на основі Законів України «Про освіту», «Про вищу освіту», «Про наукову і науково-технічну діяльність» та відповідних положень. Так у Дніпродзержинському держаному технічному університеті розроблені положення «Про підготовку аспірантів» та «Про роботу відділу аспірантури».

Під час аналізу предметної області встановлено зв'язок між об'єктами автоматизованої інформаційної системи (АІС) «Аспірантура», відображений на рис.1.

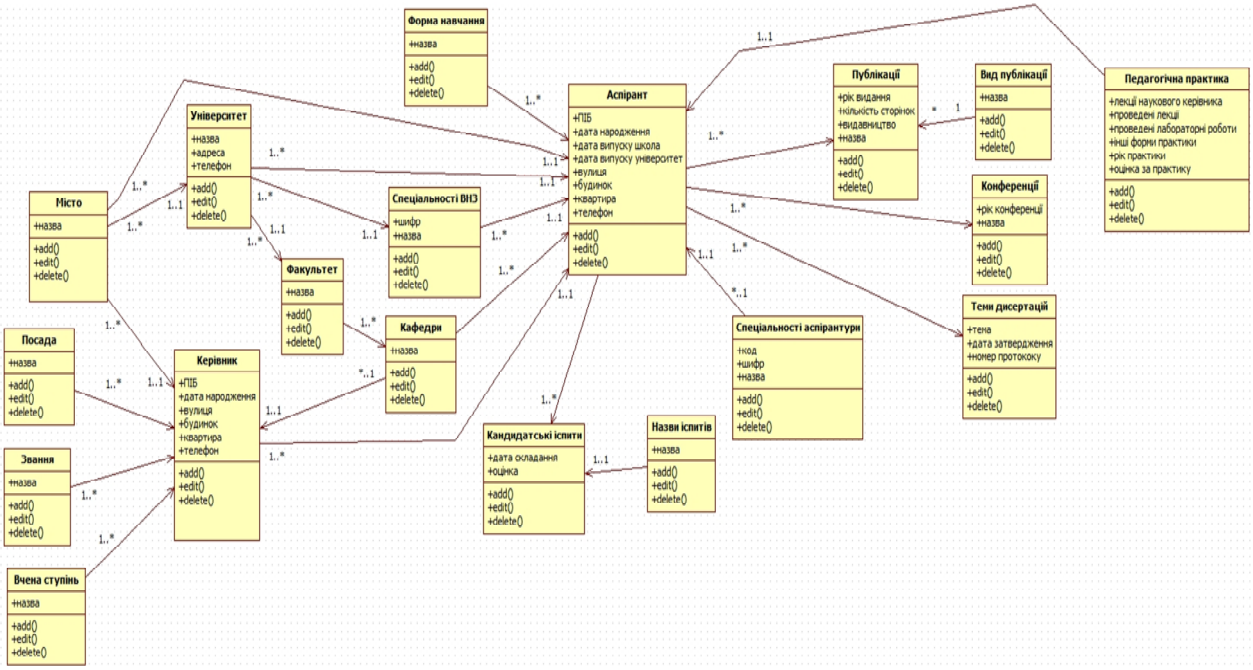


Рисунок 1 – Зв'язок об'єктів у предметній області «Аспірантура»

У результаті в АІС «Аспірантура» використовуються наступні дані:

- інформація про міста (назва міста);
- інформація про університети (назва, юридична адреса, телефон);
- інформація про кафедри (назва університету, назва кафедри);
- інформація про спеціальності вищих навчальних закладів (шифр, назва);
- інформація про наукові спеціальності (шифр, назва);
- інформація про наукових керівників (прізвище, ім'я по-батькові, дата народження, адреса, телефон, місце роботи, посада, вчене звання);

- інформація про аспірантів (прізвище, ім'я по-батькові, дата народження, адреса, телефон, дата закінчення школи, дата закінчення університету, назва закінченого ВНЗ, назва отриманої спеціальності, керівник, кафедра);
- інформація про теми дисертаційних робіт (аспірант, тема, дата затвердження на Вченій раді ВНЗ, номер протоколу Вченої ради ВНЗ про затвердження теми дисертації);
- інформація про публікації (аспірант, назва, рік видання, кількість сторінок, видавництво);
- інформація про конференції, відвідані аспірантами (аспірант, рік, назва);
- інформація про кандидатські іспити (аспірант, назва іспиту, дата складання);
- інформація про педагогічну практику (аспірант, відвідування лекцій наукового керівника, проведені лекції, проведені лабораторні роботи, інші форми практики, рік, оцінка).

Даними з вибірки відомостей про аспірантів є:

- тема дисертації;
- публікації аспіранта;
- конференції, в яких прийняв участь аспірант;
- складені аспірантом кандидатські іспити;
- педагогічна практика аспіранта.

Однією з основних складових автоматизованої інформаційної системи «Аспірантура» є інформаційне забезпечення (ІЗ). Інформаційне забезпечення – це сукупність форм документів, нормативної бази і реалізованих рішень щодо обсягу, розміщення і форм організації інформації, яка циркулює в системі автоматизованого оброблення економічної інформації чи в інформаційній системі [1].

Основними принципами створення інформаційного забезпечення АІС «Аспірантура» є:

- цілісність;
- достовірність;
- контроль;
- захист від несанкціонованого доступу;
- єдність і гнучкість;
- стандартизація та уніфікація;
- адаптивність;
- мінімізація помилок введення-виведення інформації.

Завдяки ефективному ІЗ АІС «Аспірантура» забезпечує:

- єдність і зберігання інформації, необхідної для розв'язання задач;
- єдність інформаційних масивів для всіх задач інформаційної системи;
- однократність уведення інформації та її багатоцільове використання;
- різні методи доступу до даних;
- низьку вартість витрат на зберігання та використання даних, внесення змін.

Основною складовою інформаційного забезпечення є інформаційна база (ІБ) – сукупність упорядкованої інформації, яка використовується під час функціонування АІС. Інформаційна база задачі, що досліджується, складається з нормативно-довідкової інформації, первинних документів, вихідних інформаційних повідомлень. Бази даних, які використовуються для розв'язання задачі, організовані на основі реляційної моделі даних, тобто у вигляді відношення, де кожному елементу рядка відповідає тільки один

елемент стовпця. Дана модель підтримується засобами системи управління базами даних (СУБД) MySql 5.5.

Під час аналізу предметної області був обраний наступний варіант розв'язання задачі, що досліджується (рис.2). Цей варіант базується на принципі MVC («model»-«view»-«controller») [2].

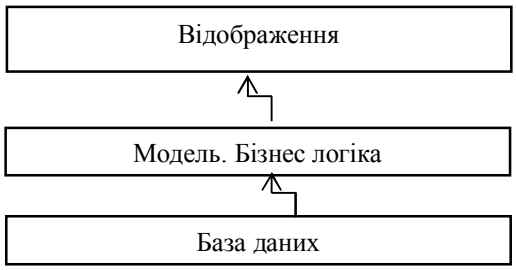


Рисунок 2 – Обраний підхід до розв'язання задачі, що досліджується

Логічна структура бази даних складається з 18 основних таблиць (табл.1):

Таблиця 1 – Логічна структура бази даних

Довідники	Вхідна інформація
«Місто» (City); «Посада» (Post); «Звання» (Title); «Екзамени» (Exam) ; «Вид публікації» (KindPublishing); «Науковий ступінь» (ScientificDegree); «Форма навчання» (FormEducation); «Наукова спеціальність» (SpecPostGraduate); «Університет» (university); «Факультет» (Faculty); «Кафедра» (Department); «Спеціальність» (Specialty); «Аспірант» (GraduateStudent); «Керівник» (Supervisor).	«Конференції» (Conference); «Публікації» (Publishing); «Кандидатські екзамени» (CandidateExaminations); «Тема дисертації» (DissertationTheme).

Фізична модель даних представляє собою опис об'єктів предметної області в рамках засобів конкретної системи управління базами даних (СУБД). У фізичній моделі описується вся інформація про конкретні фізичні об'єкти: таблиці, колонки, індекси та збережені процедури [1] (рис.3).

Для відображення даних використано програмне забезпечення Qt Creator 3.2., для програмної реалізації – мова програмування C++.

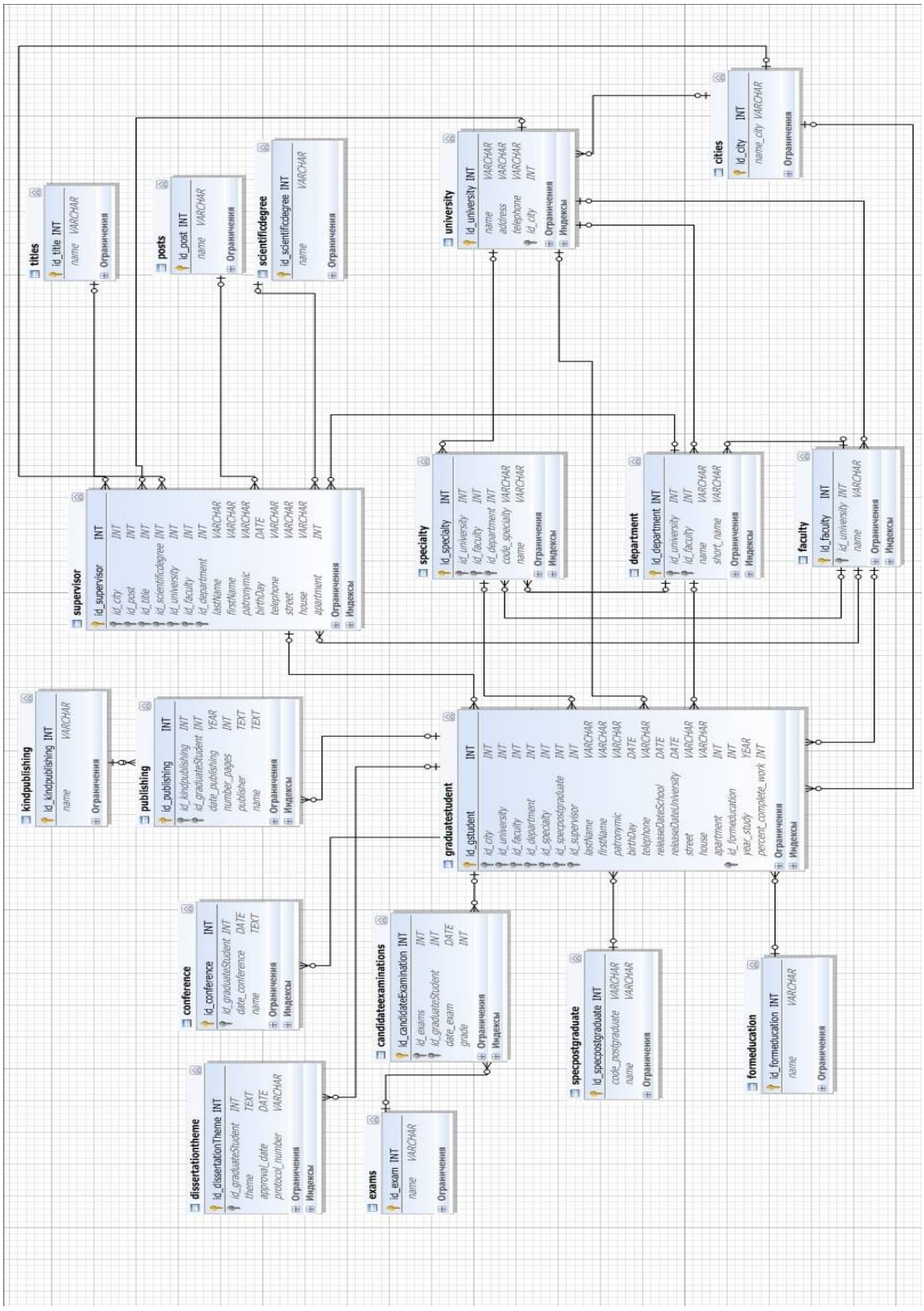


Рисунок 3 – Фізична структура бази даних

Діалог користувача з системою організовується за допомогою меню. Структуру меню зображено на рис.4.

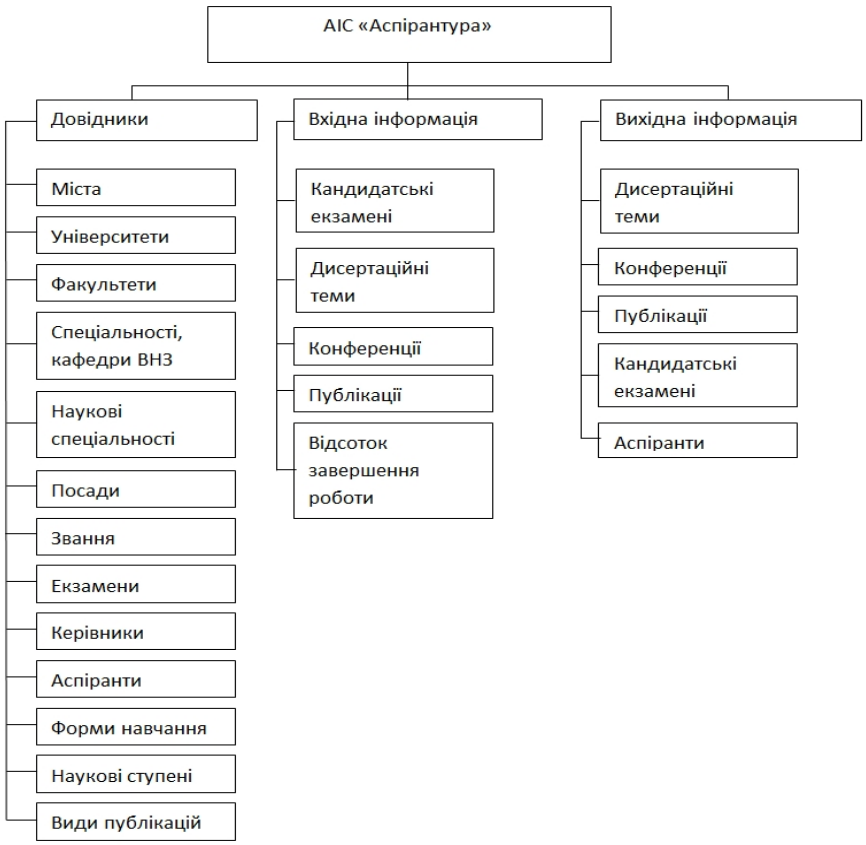


Рисунок 4 – Структура меню АІС

Вигляд головного вікна системи зображено на рис.5.

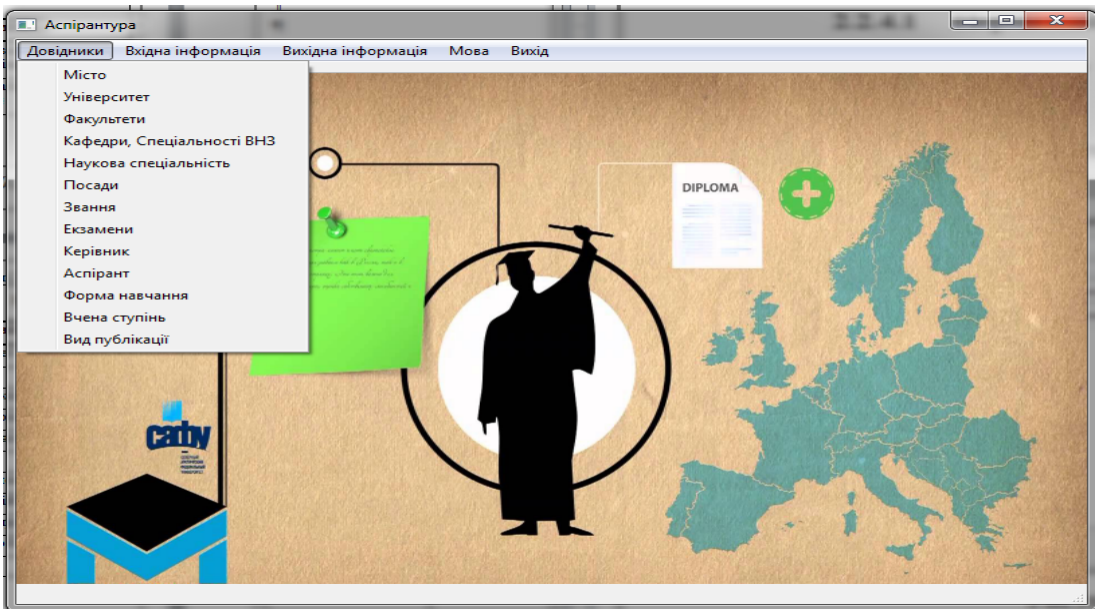


Рисунок 5 – Головне вікно системи

АІС «Аспірантура» складається з нормативно-довідникової, вхідної та вихідної інформації.

Вихідна інформація АІС – це результат виконання динамічних запитів до БД. До складу вихідної інформації АІС відносять задачі, для рішення яких створюється АІС. Вихідна інформація складається з даних трьох видів:

- вхідні параметри запиту до БД – умови, що обмежують пошук і обсяг вибірки даних;
- результати вибірки даних із БД (з таблиць документів БД);
- результати автоматичних розрахунків.

Збір інформації при побудові АІС «Аспірантура» відбувається традиційно на основі первинних документів. Носіями інформації при цьому є документи стандартної форми. Первинні документи надходять на обробку під час вступу нових аспірантів, переведення або відрахування аспірантів.

Вхідні документи перевіряються на повноту і правильність заповнення. Для перевірки правильності введення інформації використовуються програмні методи контролю. Програмний контроль включає в себе перевірку на наявність введених кодів у відповідних масивах нормативно-довідкової інформації. За своєчасне формування всіх вхідних документів відповідає завідувач аспірантури.

Організаційне забезпечення – сукупність методів і засобів, що використовуються спеціалістами для підвищення ефективності управління як на стадії створення, так і на наступних стадіях життєдіяльності інформаційної системи. Воно базується на методології, що закладена в основу функціонування системи, відображає її особливості, включає правові акти, що регулюють діяльність людино-машинної системи [1].

Суттєвих змін в організаційній структурі при автоматизації функцій працівників відділу аспірантури не відбувається. Вони виконують ті ж функції, але створюються нові умови роботи – автоматизовані робочі місця.

Висновки. Таким чином, за результатами проведених досліджень зроблено аналіз структури та методів побудови інформаційного забезпечення автоматизованої інформаційної системи, визначено склад задач по створенню та функціонуванню інформаційної бази, розглянуто організацію збору та передачі інформації.

Автоматизована інформаційна система «Аспірантура» дозволяє:

- формування та оновлення нормативно-довідникової інформації;
- формування та оновлення вхідної інформації;
- формування запитів;
- формування звітів.

Потужність та гнучкість бази даних дозволяє визнати її ефективною для користувачів. На сьогоднішній день розроблена АІС «Аспірантура» використовується у Дніпродзержинському державному технічному університеті, але зручний інтерфейс, гнучкість, масштабованість та багатомовність дозволяють використовувати створену систему у будь-якому сучасному ВНЗ.

ЛІТЕРАТУРА

1. Информационная технология. Автоматизированные системы. Термины и определения: ГОСТ 34.003-90. – [Изд. июль 2009 г. с поправкой (ИУС 1-2003)]. – М.: Стандартинформ, 2009. – 16 с. – (Национальный стандарт Украины).
2. http://www.tutorialspoint.com/design_pattern/mvc_pattern.htm. Design Patterns - MVC Pattern.

Надійшла до редколегії 27.04.2016.