УДК 004.415.5; 004.891
© О. Харчеко[2], І. Боднарчук[3], В. Яцишин[2], 2013

ЕКСПЕРТА СИСТЕМА ПРОЕКТУВАННЯ АРХІТЕКТУРИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Розглядається система автоматизації процесів проектування програмної архітектури з врахуванням вимог якості. Особливістю цих процесів є слабка їх формалізація та обов'язкова участь експертів у прийнятті рішень. До задач аналізу і розробки вимог якості до архітектури, формування альтернативних архітектур, порівняльного оцінювання архітектур з врахуванням конфліктів між атрибутами якості та пошуку компромісів розроблені моделі та математичне забезпечення. Для використання проектувальниками створено репозиторії архітектурних патернів та база знань класифікації альтернативних архітектур за функціональними вимогами та вимогами якості.

Вимоги якості до архітектури визначаються на основі експертної технології парних порівнянь та технології QFD [1]. Для порівняльного оцінювання архітектур використано метод аналізу ієрархій з оптимізаційним алгоритмом визначення ваг, який дозволяє порівнювати значно більшу кількість альтернатив (n>15), ніж стандартний [0]. Прийняття остаточного рішення по вибору архітектури на множині альтернатив, проранжованих по окремих атрибутах якості і по їх сукупності, виконується шляхом аналізу конфліктів і побудови областей компромісів [0].

The system for automation of processes of software architecture design is discussed in the paper with accounting of quality requirements. For problems of requirements analysis and development for architecture, composing of alternative architectures, comparative evaluation of architectures with accounting of conflicting quality attributes and tradeoffs the models and mathematical background is developed. To use by software designers the repositories of architectural patterns are created and knowledge base for classification of alternative architectures according to functional requirements and requirements of quality is created too.

Quality requirements for the architecture are developed on the base of expert technic of pairwise comparisons QFD [1]. For comparative evaluation of architectures the Analytical Hierarchic Process is applied with optimization algorithm for definition on weights, which allows to compare considerably more number of alternatives (n>15)thanstandard one [0]. The final decision making for choosing of architecture among the alternatives ranged for individual quality attributes and for their totality is made by means of analysis of conflicts and creation of compromisesareas [0].

[2] Національний авіаційний університет, м. Київ
[3] Тернопільський національний технічний університет ім. Івана Пулюя

1. ВСТУП

У зв'язку із зростаючою складністю програмних систем (ПС) стає все важче задовольняти вимоги якості при їх проектуванні. Для розв'язання цієї задачі з мінімальними втратами цей процес переносять на більш ранні стадії проектування, а саме при проектуванні архітектури. Архітектура при цьому визначається як набір компонентів, які інкапсулюють логіку обчислень і зв'язки, які забезпечують взаємодію компонентів та створюють їх конфігурацію. Архітектура ПС забезпечує абстрактну модель високого рівня для представлення структури і ключових властивостей ПС і створює передумови забезпечення якості ПС.

Процес проектування архітектури включає декілька етапів [0]:

– визначення вимог до ПС, як функціональних, так і вимог якості, яке виконується на основі аналізу потреб всіх зацікавлених сторін.

Також необхідно визначити відносну важливість атрибутів якості. Після цього необхідно провести комунікацію вимог якості до ПС на вимоги якості до архітектури;

– вибір альтернативних проектних рішень.

На основі аналізу вимог створюються альтернативні проектні рішення, які в подальшому будуть розглядатись для пошуку кращого з них. Для створення альтернативних архітектур повинна використовуватись технологія, базована на патернах[0].

– аналіз і оцінювання проектних рішень.

Кожен варіант проектного рішення повинен бути оцінений і порівняний з іншими. Архітектор повинен при цьому враховувати те, що альтернативи по різному впливають на реалізацію атрибутів якості, а атрибути, у свою чергу, мають різну відносну важливість. Оскільки вимоги до ПС можуть змінюватись як в процесі проектування, так і під час експлуатації, то будуть змінюватись і пріоритети атрибутів, що може вплинути на порядок ранжування альтернатив. Це також необхідно враховувати при виборі варіантів рішення;

– загальний архітектурний аналіз і прийняття рішення.

Використовуючи результати попереднього етапу, архітектор обирає найкращий варіант з точки зору задоволення всіх вимог якості. Якщо такого варіанта архітектури немає, то досліджується конфлікти між критеріями якості і будуються області компромісів, на основі аналізу яких обирається рішення.

Приведемо короткий огляд існуючих методів оцінювання і вибору архітектури програмних систем з аналізом повноти реалізації в них наведених вище етапів.

2. ОГЛЯД МЕТОДІВ ОЦІНЮВАННЯ І ВИБОРУ АРХІТЕКТУРИ ПС НА ОСНОВІ ВИМОГ ЯКОСТІ

Існує раннє і пізнє оцінювання архітектур. Раннє оцінювання використовується тоді, коли ще не створено програмних компонентів або їх моделей. Таке оцінювання базується на досвіді розробників та логічному обгрунтуванні, оскільки відсутні артефакти, які дають змогу імітувати роботу ПС. Методи, які реалізують раннє оцінювання, базуються на сценаріях. До цих методів належать наступні: SAAM і ATAM [0]. В методі SAAM для коректного порівняння архітектур, існуючих та тих, що розглядаються, запропоновано аналізувати їх у трьох аспектах, а саме – функціональність, структура та розміщення. На основі пріоритетів зацікавлених сторін визначаються критерії якості. Для перевірки задоволення кожного атрибута якості розробляється сценарій і проводиться оцінка рівня задоволення даного атрибуту варіантом архітектури.

Метод ATAM подібний до SAAM, але в ньому на основі аналізу сценаріїв для відібраних архітектур проводиться оцінка ризиків задоволення атрибутів якості. Оцінку ризиків проводить група експертів, яка також ранжує альтернативні варіанти за рівнем ризику і визначає так звані точки чутливості у компонентах чи зв'язках архітектури, також аналізуються компроміси між критеріями якості.

Методи ATAM і SAAM поєднані єдиною концепцією і часто використовуються в сукупності.

Вимоги якості до архітектури в даних методах визначаються експертами, не використовуються формальні методи. Тому має місце суттєвий вплив суб'єктивних факторів і відсутні методи автоматизації цих процесів.

Аналіз проектних рішень відбувається послідовно по одному атрибуту якості, при виборі варіанта архітектури не використовуються методи оптимізації. Рівень автоматизації процесів низький через недостатнє використання формальних методів.

Для обгрунтованого вибору рішення в методі SAAM/ATAM вибрані альтернативні архітектури аналізуються на ефективність витрат методом CBAM [0]. Цей метод забезпечує економічний аналіз ПС, яка базується на вибраних в попередніх методах варіантах архітектури та сценаріях моделювання. Експерти призначають оцінки критеріям якості в балах від 1 до 100 і ранжують архітектури за значенням, яке ці архітектурні рішення забезпечують для атрибуту якості. Оцінка кожного варіанта архітектури обчислюється за формулою:

$$B(A_i) = \sum_{j=\overline{1,K}} \left(Cont_{i,j} \cdot Q_j \right) \; i = \overline{1,n}. \qquad (1)$$

Тут $Cont_{ij}$ – вага *i*-ї архітектури відносно *j*-го атрибута,

Q_j – пріоритет *j*-го атрибута.

Метод забезпечує оцінку затрат на реалізацію кожної альтернативи і дає можливість обчислити показник бажаності як відношення прибутку до затрат. На основі отриманих даних проводиться вибір кращого рішення.

Метод CBAM використовує архітектурні рішення і атрибути якості, отримані із SAAM/ATAM, а забезпечує лише оцінку рішень, тобто фактично реалізує третій і частково четвертий етапи проектування архітектури.

Часто виникають задачі створення ПС на базі існуючої шляхом перепроектування для задоволення нових вимог якості. Для вирішення таких задач було створено метод реінжинірингу архітектури ПС на основі сценаріїв SSAR [0], який є сукупністю чотирьох методів оцінки архітектур відносно атрибутів якості:

- оцінка на основі сценаріїв;
- моделювання;
- математичне моделювання;
- оцінка на базі практичного досвіду.

При використанні SSAR обирається один із методів, але основним є метод оцінювання на основі сценаріїв. Цей метод подібний до того, що реалізується в SAAM.

При використанні моделювання основні компоненти ПС реалізуються в коді, а інші моделюються комп'ютером, утворюючи виконувану систему.

При використанні математичного моделювання характеристики якості ПС оцінюються за допомогою математичних моделей операцій, на яких ці характеристики реалізуються.

Оцінювання на базі практичного досвіду дає можливість виявити дефекти проектних рішень та проблеми, які необхідно усунути.

Метод SSAR не містить процедур вибору альтернативних архітектур, а також виявлення конфліктів і пошук компромісів між атрибутами якості. Оцінювання проводиться послідовно по кожному атрибуту якості без використання процедури оптимізації. Спільним недоліком розглянутих методів є послідовне оцінювання архітектури по одному параметру, що робить процес вибору трудомістким і неформалізованим. Тому поява робіт, в яких було використано процедуру аналізу ієрархій, дозволив значно покращити процес вибору архітектури і формалізувати його [0], [0], [0].

В методі SAHR використовується порівняльне оцінювання альтернатив стосовно реалізації атрибутів якості. Він дає змогу визначити

відносні ваги альтернатив по кожному атрибуту якості і проранжувати їх. За призначеними зацікавленими сторонами пріоритетами атрибутів якості обчислюється їх усереднене значення і визначаються ваги альтернатив відносно сукупності атрибутів якості.

Отримані відносні оцінки альтернатив можуть використовуватись для аналізу конфліктів між атрибутами якості і пошуку компромісного рішення.

Перевагами методу SAHR є оцінювання альтернатив по всіх атрибутах якості, оптимізація рішень та досить високий рівень формалізації, що дає змогу автоматизувати процес.

З проведеного аналізу слідує, що методи оцінювання архітектур базуютьсяв основному на експертній інформації. При цьому широко використовуються знання та досвід проектувальників. Тому для підвищення ефективності цих методів необхідно використовувати їх у складі експертної системи, в якій знання формалізовані в базі знань, а процеси введення та обробки експертної інформації автоматизовані з допомогою апаратно-програмної платформи .

3. СТРУКТУРА І ФУНКЦІОНАЛЬНІСТЬ ЕКСПЕРТНОЇ СИСТЕМИ

Загальний вигляд системи зображено на рисунку 1.

Інтерфейсна підсистема забезпечує взаємодію користувачів з системою. База знань використовується для формування альтернативних архітектур зі стандартних патернів. Тут зберігаються правила побудови архітектури у відповідності з вимогами.

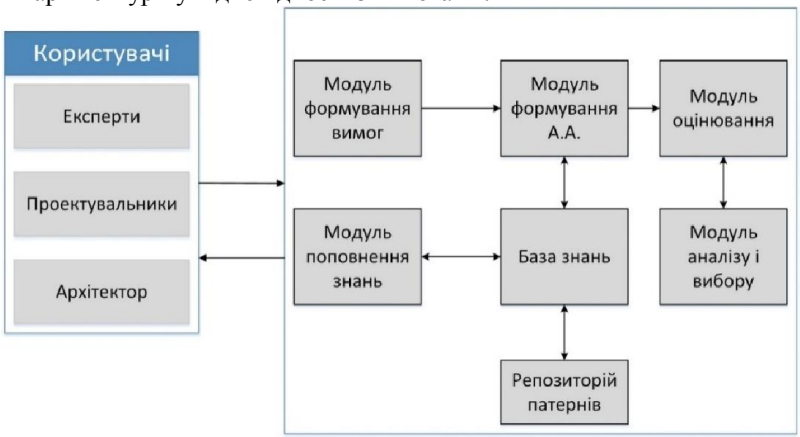

Рис. 1. Структура експертної системи (А.А. – альтернативні архітектури)

3.1 Модуль визначення вимог якості архітектури

Множина вимог якості архітектури $\{K_i^2\}, i=\overline{1,n}$ повинна визначатися таким чином, щоб побудована на основі даної архітектури ПС задовольняла вимогам якості до ПС $\{K_i^1\}, i=\overline{1,m}$. Для визначення $\{K_i^2\}$ по заданих $\{K_i^1\}$ в [0] розроблена процедура і програмний засіб, базовані на технології QFD, яка полягає в прямому визначенні експертами кореляцій між вказаними множинами. Для зменшення впливу суб'єктивних факторів на результат в даному модулі використано алгоритм методу базових протоколів і експертний метод парних порівнянь [0].

Специфікації вимог – це сукупність властивостей, яким повинна задовольняти поведінка ПС. Ці властивості можуть бути представлені за допомогою формалізму базових протоколів.

Базовий протокол представляє собою вираз виду $\forall x(\alpha\langle u\rangle\beta)$, де x – список типізованих параметрів, α і β – формули базової мови,u – процес протоколу. Формула α називається передумовою, а формула β – постумовою базового протоколу. Післяумова виражає стан системи після виконання базового протоколу. Базовий протокол розглядається, як формула темпоральної логіки, яка виражає той факт, що коли (для певних значень параметрів) стан системи має розмітку, яка задовольняє умові α, то процес uпереводить систему в стан, в котрому розмітка задовольняє вимоги умовиβ. Базові протоколи формалізують вимоги до ПС. Для запису α і β використовується мова, близька до мови специфікації. Для означення процесів використовується мова алгебри процесів, така як CCS, CSP або такі мови, як UML, MCS чи SQL. В роботі [0] цей формалізм був успішно застосований для верифікації функціональних вимог до телекомунікаційних систем.

В нашому випадку передумовою α є вимоги якості до ПС, специфіковані в термінах стандарту ISO/IEC 25010, а постумовоюβ – вимоги якості до архітектури ПС.Процес uє процесом комунікації (перетворення) α в β, який реалізується експертною технологією парних порівнянь. Суть полягає в побудові матриці $B^s\left(b_{ij}^s\right)$, де b_{ij}^s виражає, на скільки вплив i-го критерію якості архітектури переважає вплив j-го критерію на реалізацію s-го критерію якості ПС. З побудованої матриці обчислюються вагові множники $w_i^s\left(i=\overline{1,n}\right)$ критеріїв якості архітектури. Призначивши пріоритети критеріям якості ПС $P_s\left(s=\overline{1,m}\right)$, обчислюємо інтегральні вагові множники

$$\overline{w}_i = \sum_{s=1}^{m} w_i^s \cdot P_s \qquad (2)$$

і, ввівши обмеження $\overline{w}_i \le \overline{w}_t$, визначимо критерії якості архітектури, які включаються в β.

Використання формалізму базових протоколів дозволив уніфікувати представлення α і β, що дало змогу зменшити кількість помилок в специфікаціях, а також формалізувати процедуру комунікації вимог.

3.2 Модуль формування альтернативних архітектур

При створенні даного модуля для подання архітектури була прийнята концепція шарів, розвинута М.Фаулером [0], в якій функціональність розділена на шари. Корпорація Microsoft розробила технологію, яка базується на даній концепції [0]. В цій технології для кожного шару розроблено набори компонентів (патернів), які реалізують функціональність даного шару. Компоненти згруповані у категорії і призначені для вирішення певних стандартних задач.

Таким чином, кожне програмне застосування, яке буде проектуватись, можна розділити на логічні частини, які відповідають шарам. Визначивши категорії задач, які будуть розв'язуватись певним шаром, можна вибрати деякий компонент з існуючого набору і, таким чином, створити каркас архітектури. А оскільки для кожної категорії розроблено, як правило, декілька компонентів, то можна створити множину альтернативних архітектур.

Для автоматизації цього процесу пропонується використовувати експертну технологію, засновану на базі знань. Знання у системі організовані у вигляді фрейму, зображеного на рисунку 2.

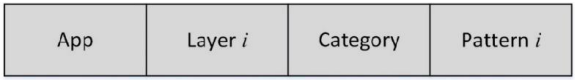

Рис. 2. Структура фрейму бази знань експертної системи.
App – ім'я програми; Layer – ім'я шару; Category – ім'я категорії задач;
Pattern – ім'я шаблону

Експерти (архітектор) ділять застосування на шари і визначають задачі, які розв'язуються на певних шарах. Потім заповнюється фрейм-шаблон, в якому незаповненим є останній слот. На основі пошуку у репозиторії шаблонів знаходиться відповідний компонент і поміщується в каркас архітектури, створюючи таким чином архітектуру.

3.3 Модуль порівняльного оцінювання архітектур

Із скомпонованих альтернативних архітектур необхідно визначити найкращу. Для цього проведемо порівняння як по кожному показнику

якості, так і по їх сукупності. Структурна схема процесу порівняння зображена на рисунку 3.

$K_i^1, i = \overline{1, m1}$ – критерії якості ПС у відповідності зі стандартом ISO/IEC 25010;

$K_i^2, i = \overline{1, m2}$ – критерії якості архітектури;

$A_i, i = \overline{1, n}$ –альтернативні архітектурні рішення.

Перелік критеріїв якості у використанні визначається і специфікується розробником ПС разом із замовником в модулі, розглянутому раніше. Критерії якості архітектури $\{K_i^2\}$можна визначити шляхом комунікації критеріїв $\{K_i^1\}$ з допомогою модуля, описаного вище, з врахуванням рекомендацій стандарту ISO/IEC 25010.

Архітектурне рішення вибирається з умови оптимізації сукупності критеріїв $\{K_i^1\}, \{K_i^2\}$. Це задача багатокритеріальної ієрархічної оптимізації і для розв'язання таких задач найчастіше використовується метод аналізу ієрархій Сааті.

При використанні MAI для рішення таких задач ваги альтернатив (критеріїв) w_i на кожному рівні знаходяться з використанням матриць парних порівнянь $B(b_{ij})$, які заповнюють експерти (тут b_{ij} визначає перевагу *i*-тої альтернативи над *j*-ю).

Коефіцієнти матриць повинні бути узгодженими, тобто $b_{ij} = w_i / w_i \ \ \forall b_{ij} \in B$. Вагові множники в цьому випадку знаходяться як компоненти власного вектору матриці парних порівнянь, які відповідають максимальному характеристичному числу матриці. Але при значній кількості альтернатив (n>9) в силу дії на експертів різних факторів матриця $B(b_{ij})$ є неузгодженою і її ранг буде відмінним від одиниці, тобто матриця буде мати декілька власних значень. Тому в роботі [0] для рішення задачі вибору оптимальної архітектури при великій кількості альтернатив (n>9) використано алгоритм обчислення ваг w_i в MAI з умови мінімізації неузгодженості матриці $B(b_{ij})$. Ця задача зводиться до задачі лінійного програмування, математична модель якої визначається вибраною формою неузгодженості матриці $B(b_{ij})$. В роботі [0] приведено рішення цієї задачі для наступної форми міри неузгодженості:

$$\left| \frac{w_i}{w_j} - b_{ij} \right| \le \delta_t \cdot b_{ij}, \quad \delta_t \ge 0, \tag{3}$$

де δ_t – вибране порогове значення.

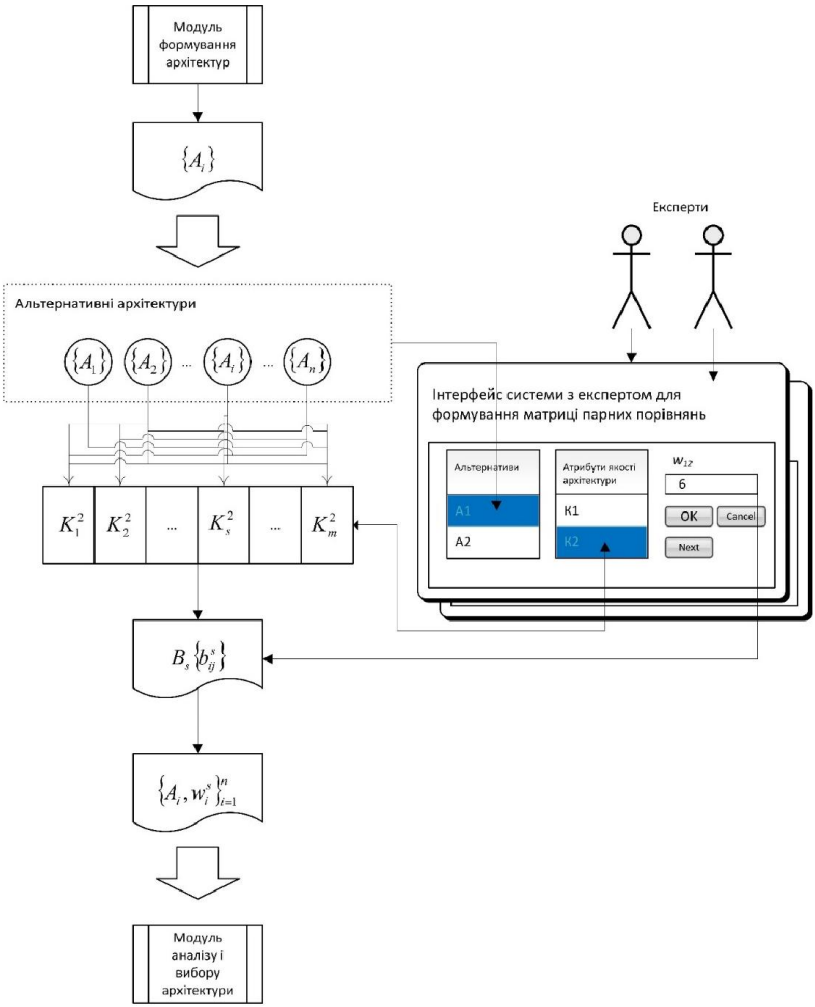

Рис. 3. Ієрархічне представлення задачі оптимізації архітектури

Задача мінімізації (3) зводиться до задачі лінійного програмування

$$\min_{\{w_i\}} \sum_{i=1}^{n} \sum_{j=1}^{n} \left(y_{ij}^{+} - y_{ij}^{-} \right)$$

$$w_i \geq a,\ i = \overline{1,n}, \tag{4}$$

$$w_i - b_{ij} w_j = y_{ij}^{+} - y_{ij}^{-}.$$

$$-\delta_t \cdot b_{ij} \cdot w_j \leq w_i - b_{ij} \cdot w_j \leq \delta_t \cdot b_{ij} \cdot w_j,$$

$$y_{ij}^{+}, y_{ij}^{-} \geq 0;\ i, j = \overline{1,n}. \tag{5}$$

Для отримання порівняльних оцінок якості архітектур по кожному з критеріїв формується матриця $B\left(b_{ij}^{s}\right)$, де b_{ij}^{s} показує, наскільки *i*-та альтернатива переважає *j*-ту по реалізації *s*-го критерію. Для отриманих матриць визначаються набори вагових множників $\left\{w_i^s\right\}$, $i = \overline{1,s}$, $s = \overline{1,7}$ як рішення задачі лінійного програмування.Для ранжування альтернатив по множині критеріїв необхідно знайти їх пріоритети. Для визначення пріоритетів критеріїв якості архітектури $\left\{K_i^2\right\}$ по їх впливу на реалізацію критеріїв якості ПС $\left\{K_i^1\right\}$ експертами заповнюється матриця парних порівнянь $B^s\left(b_{ij}^{s}\right)$, де величина b_{ij}^{s} визначає, на скільки вплив критерію K_i^2 переважає вплив критерію K_j^2 на реалізацію критерію якості ПС K_s^1. Розв'язавши задачу (4), (5), отримаємо набори пріоритетів критеріїв якості архітектури $\left\{P_i^{1s}\right\}$, $i = \overline{1,m2}$, $s = \overline{1,m1}$.Тоді вага альтернативної архітектури A_i відносно реалізації критерію якості ПС K_s^1 визначатиметься за формулою:

$$J_i^{1s} = \sum_{j=1}^{m2} p_j^{1s} \cdot w_i^j, \quad i = \overline{1,n}, \quad s = \overline{1,m1}, \tag{6}$$

де w_i^j –вагові множники, визначені на попередньому етапі.

Тепер можна ранжувати альтернативи $\left\{A_i\right\}$ за величиною $\left\{J_i^s\right\}$ для кожного $s = \overline{1,m1}$.Тобто отримаємо множини альтернатив і відповідні їм значення ваг $\left\{A_i^s, P_i^{1s}\right\}$, $s = \overline{1,m1}$.

В даній системі, після оцінювання альтернатив, може рішатись задача їх оперативного корегування. Така задача виникає тоді, коли експерти і архітектор при виборі надають перевагу певній альтернативі A_j, хоча вона за деякими критеріями має найкращі оцінки. Ставиться

задача збільшити оцінки за цим критеріями за рахунок зменшення за іншими, але так, щоби оцінки альтернати A_j ви за всіма критеріями були не гірші за інші.

Така ситуація є типовою при проектуванні розподілених програмних застосувань, коли бажана архітектура відома, а корегування критеріїв може здійснюватись шляхом підбору стандартних функціональних компонентів [12].

До розв'язування цієї задачі можна застосувати аксіоматичний підхід В. Подіновського [13], який полягає в попарному заміщенні критеріїв. Критерії K_r і K_s є порівняними за заміщенням, якщо для деякої альтернативи A_i можлива компенсація за перевагою будь-якої зміни критерію K_r зміною критерію K_s.

Тобто, якщо A_i^p – це альтернатива, яка заміщує A_i шляхом корекції K_r і компенсації K_s, то їх скореговані значення будуть

$$\overline{K}_r^{ip} = \overline{K}_r^i - \delta_r,\ \overline{K}_s^{ip} = \overline{K}_s^i + \delta_{si},\ \delta_{si} = f\left(r, s, \overline{K}, \delta_r\right), \qquad (7)$$

тут $\overline{K}$ – вектор значень критеріїв.

Запишемо співвідношення для компенсації при заміщенні для множини компонент вектору $\overline{K}^i$ альтернативи A_i, яку ми хочемо зробити кращою за A_j:

$$\delta\overline{K}_r^{ir_z} = C_r^{ir_z} \cdot \delta K_r^i,\ r_z \in R_i^2(r),\ r \in R_i^1, \qquad (8)$$

де $\delta\overline{K}_r^{ir_z}$ – можливе зменшення компоненти $\overline{K}_r^i$ з метою збільшення $\overline{K}_{r_z}^i$;

R_i^1 – множина індексів r, для яких $\overline{K}_r^{iz} > \overline{K}_r^j$, $j = \overline{1,n};\ i \neq j$;

$R_i^2(r)$ – задана для R_i^1 множина індексів, така, що компоненти $\overline{K}_r^i, r \in R_i^1$ можуть брати участь у заміщенні компонентів $\overline{K}_s^i, s \in R_i^2(r)$;

$C_r^{ir_z}$ – задані коефіцієнти пропорційності.

Компоненти вектору $\overline{K}^i$ після заміщення визначаються наступними співвідношеннями:

$$\overline{K}_r^{ip} = \overline{K}_r^i - \sum_{r_z \in R_i^2(r)} C_r^{ir_z} \cdot \delta\overline{K}_{r_z}^i, \quad r \in R_i^1;$$
$$\overline{K}_r^{ip} = \overline{K}_{r_z}^i + \sum_{r \in R_i^1} \sum_{r_z \in R_i^2(r)} \delta\overline{K}_{r_z}, \quad r_z \in s, s \overline{\in} R_i^1, r_z \in R_i^2(r). \tag{9}$$

А.А. Павловим в роботі [14] сформульовані задачі оптимізації заміщення (9), які зводяться до задач лінійного програмування, математичні моделі яких залежать від стратегії прийняття рішень.

Для Парето-оптимальної стратегії модель оптимізації буде такою:

$$\max\left\{\sum_{r \in R_r^1} d_r + \sum_{s \overline{\in} R_j^1} d_s\right\} = \max\{y\} \tag{10}$$

при обмеженнях:

$$d_r, d_s \geq 0, r \in L_j^1, s \overline{\in} R_j^1;$$
$$\overline{K}_r^j - \sum_{r_z \in R_j^2(r)} \delta\overline{K}_r^{jr_z} \geq \max_i\left(\delta\overline{K}_r^i\right) + d_r, i \in \overline{1,n}, i \neq j, r \in R_j^1;$$
$$\overline{K}_s^j + \sum_{r \in R_j^1} \sum_{r_z \in R_j^2(r)} \frac{1}{d_r^{jr_z}} \delta\overline{K}_r^{jr_z} \geq \max_i\left(\overline{K}_s^i\right) + d_s, \quad i = \overline{1,n}, i \neq j, r \overline{\in} R_j^1, r_z = s, \tag{11}$$
$$\exists r, s \in R_j^2(r);$$
$$\sum_{r_z \in L_j^2(r)} \delta\overline{K}_r^{ir_z} \leq b_{K_r^j}, \quad j \neq i, r \in R_j^1, r_z = s.$$

Змінними тут є $\delta K_r^{jr_z}, d_r, d_s$.

Після деяких перетворень для розв'язування задачі застосовується стандартний симплекс-метод.

3.4 Модуль аналізу і прийняття рішень

Для прийняття остаточного рішення по вибору архітектури проводиться аналіз результатів, отриманих в модулі оптимізації.

Оскільки в розробці ПС беруть участь декілька груп фахівців,які мають різні пріоритети для кожного з атрибутів якості, то визначаються пріоритети кожної з груп шляхом формування ними матриць парних порівнянь, до яких застосовується МАІ і знаходяться пріоритети критеріїв $\left\{P_i^S\right\}, i = \overline{1,k_2}, s$ – номер групи експертів. Компромісне рішення приймається як середнє геометричне $P_{ij}^* = \sqrt[n]{P_{ij}^1 \cdot P_{ij}^2 \cdot \ldots \cdot P_{ij}^n}$, або як усереднене, з врахуванням показника компетентності груп експер-

тів $P_{ij}^{*} = P_{ij}^{\alpha_1} \cdot P_{ij}^{\alpha_2} \cdot \ldots \cdot P_{ij}^{\alpha_n}$. ($\alpha_1, \alpha_2, \ldots, \alpha_n$ – показники компетентності).

Для досягнення компромісу при прийнятті остаточного рішення, а також при зміні вимог до ПС в процесі проектування, можуть змінитися пріоритети відносно критеріїв якості. В цьому випадку, для корегування пріоритетів критеріїв використовується співвідношення

$$D'_{s,i,j} = \frac{\left|J_i - J_j\right|}{\left|w_i^s - w_j^s\right|} \cdot \frac{100}{P_s}. \qquad (12)$$

Тут $D'_{s,i,j}\left(s = \overline{1,m2};\ i, j = \overline{1,n};\ i \neq j\right)$ – мінімальна зміна величини пріоритету P_K критерію якості K_s, яка змінює порядок слідування сусідніх альтернатив A_i та A_j на зворотній. Найменше значення $D'_{s,i,j}$ показує, що пріоритет P_s атрибуту K_s є критичним до змін оцінок в парних порівняннях.

Використовуючи співвідношення (12), для кожного критерію K_s можна знайти інтервал ΔP_s, в якому експерти можуть проводити корекцію пріоритетів P_s, безпосередньо або через корекцію значень парних порівнянь, без зміни ранжування альтернатив $D_s^{*} = \min\limits_{i} D_{s,i,j} = \Delta P_s,\ i = \overline{1,n},\ s = \overline{1,k_2}$.

Якщо потрібно визначити ранжування альтернатив відносно глобальної якості ПС, то необхідно визначити пріоритети критеріїв якості ПС $\left\{P_i^2\right\},\ i = \overline{1,m1}$, застосувавши модифіковану процедуру МАІ.

Визначити ваги альтернатив відносно реалізації глобального критерію якості ПС можна за значенням показника:

$$J_i^0 = \sum_{s=1}^{m1} J_i^{1s} \cdot P_s^2, \quad i = \overline{1,n}. \qquad (13)$$

Тоді показником важливості альтернативи A_i по множині критеріїв буде

$$J_i = \sum_{j=1}^{m2} P_j \cdot w_j^i, \quad i = \overline{1,n}, \qquad (14)$$

і ранжування $\left\{A_i\right\}$ проводиться за значеннями $\left\{J_i\right\}$.

Отримані проранжовані,як по окремих показниках якості,так і по їх сукупності,альтернативні архітектури використовуються для прийняття остаточного рішення по вибору архітектури.

3.5 Аналіз компромісів між критеріями якості

У випадку коли жодне з рішень не задовольняє вимоги по всім критеріям , то проводиться аналіз конфліктів між атрибутами і пошук компромісних рішень.

Аналіз компромісів між критеріями можна проводити, побудувавши діаграми компромісів. Наприклад, на рисунку 4 зображена діаграма компромісів між портативністю та масштабованістю для варіантів архітектур, які розглядались в [0]. По осі *x* показано відносне значення критерію "Переносимість", по осі *y* – "Масштабованість".

Рисунок 4 корисний з огляду на можливість візуалізації компромісів, їхніх відносних розмірів та відношень між альтернативами розробки з точки зору компромісів.

Для кожної альтернативної архітектури точка на діаграмі є можливим компромісом, якщо вона знаходиться зліва від діагоналі координатної площини.

На рисунку 4 для вибраного рішення THTD (лівий верхній кут) архітектура вибрана з огляду на надання переваги масштабованості перед портативністю, у той час коли для рішення COAB портативність переважає над масштабованістю. Розмір компромісу відображається розміщенням точки в напрямку вверх вліво чи вниз вправо. Чим ближче точка розміщена до кута, тим більший розмір компромісу було виконано. Попадання альтернативи у нижній лівий квадрат відображає, що обидва атрибути впливають негативно.

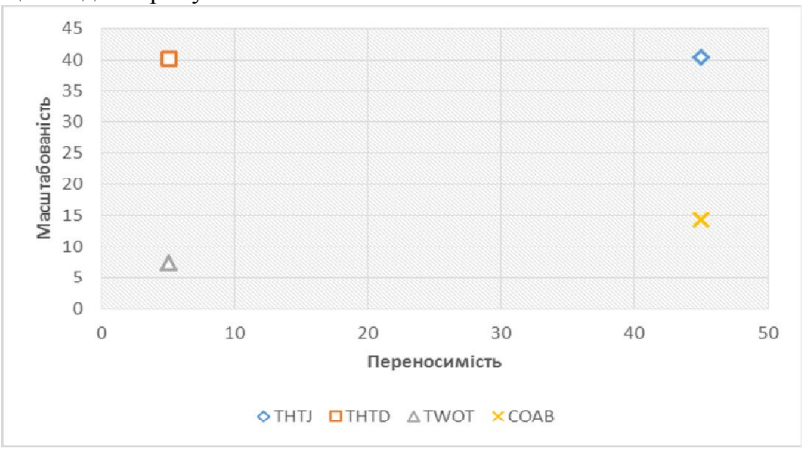

Рис. 4. Компроміси між критеріями якості

При врахуванні в аналізі компромісів ваг атрибутів якості отримані вище результати можуть відповідно змінитися. В таблиці 1 наведені

архітектурні альтернативи, відсортовані відповідно до ваг атрибутів якості.

Тоді діаграма компромісів з рис. 4 набуде вигляду рис. 5.

Таблиця 1

Архітектурні альтернативи, відсортовані відповідно до ваг атрибутів якості.

Атрибути якості	Альтернативи розробки			
	THTJ	THTD	TWOT	COAB
Модифікованість	0,1459	0,0510	0,0129	0,0700
Масштабованість	0,0330	0,0330	0,0044	0,0117
Експлуатаційні якості	0,0198	0,0198	0,0337	0,0239
Вартість	0,0224	0,0162	0,0657	0,0306
Затрати на розробку	0,0205	0,0149	0,0695	0,0301
Переносимість	0,0423	0,0047	0,0047	0,0423
Легкість встановлення	0,0297	0,0651	0,0453	0,0368

Альтернатива THTJ має значну перевагу у модифікованості порівняно з іншими альтернативами розробки. Отже, для THTJ компроміси між модифікованістю та будь-якими іншими атрибутами якості підсилюються у діаграмі компромісів. Таким чином, ці діаграми візуалізують компроміси та їх розміри, які не відображаються при використанні MAI.

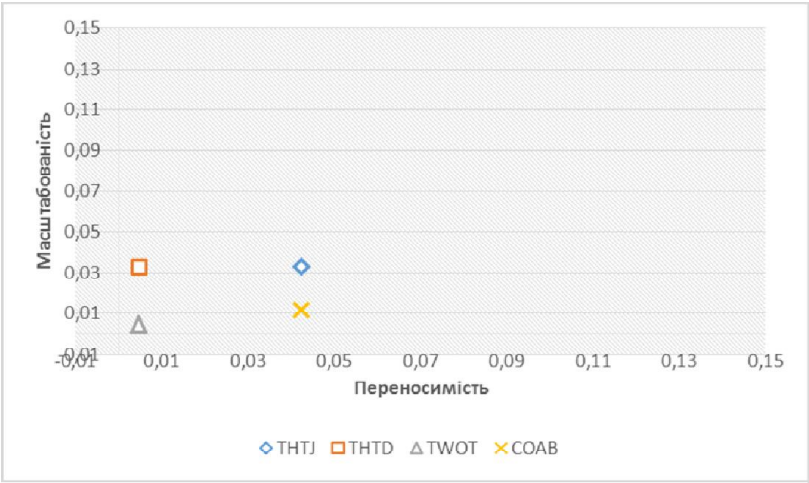

Рис. 5. Діаграма компромісів між переносимістю та масштабованістю (з врахуванням ваг пріоритетів атрибутів якості)

4. ВИСНОВКИ

Застосування запропонованої системи при проектуванні програмного забезпечення дозволить за рахунок використання формалізованих моделей, репозиторію готових архітектурних патернів та методів оптимізації підвищити якість програмних продуктів, скоротити терміни розробки і зменшити витрати.

Використання методу формалізації вимог якості дозволяє зменшити кількість помилок в специфікаціях приблизно на 30% [0]. Метод комунікації вимог якості дає змогу обґрунтовано вибирати критерії якості архітектури, а алгоритм,розроблений на його основі,дозволяє автоматизувати дану процедуру.

Застосування методу MAI дозволяє вибирати найкращу архітектуру з множини альтернативнихпо сукупностікритеріїв якості, на відміну від відомих методів ATAM, CBAM та інших, в яких архітектура оцінюється по одному критерію.

Застосування запропонованого алгоритму визначення ваг альтернатив в MAI з умови мінімуму неузгодженості матриці парних порівнянь розширює межі застосування методу на більшу кількість альтернатив (n>9) , тобто вибір рішення здійснюється на всій множині альтернатив. А при використанні стандартного MAIдля цього випадку множина альтернатив ділиться на кластери з n<9, і задача розв'язується для кожного кластера окремо, а потім порівнюються кращі варіанти з кожного кластера. А це значно збільшує час на пошук рішення. Так, для прикладу, розглянутого в [0], при n=15 множину альтернатив треба поділити на два кластери, і тоді час на пошук найкращої альтернативи при використанні стандартного алгоритму буде більш, ніж в два рази перевищувати час для модифікованого алгоритму.

Використання співвідношення (12) при зміні вимог дозволить уникнути повторного ранжування альтернатив,якщо величини змін вкладаються у визначені інтервали.

Застосування методу компоновки альтернативних архітектурта репозиторіюпатернів дає змогу ефективно використовувати готові архітектурні патерни при виборі оптимального рішення для широкого спектру категорій програмних застосувань (додатків).

1. Харченко О.Г. Інструментальний засіб розробки та комунікації вимог якості до програмних систем / О.Г. Харченко, В.В. Яцишин, І.Е. Райчев // Інженерія програмного забезпечення. – 2010. – № 2. – С. 29–34. 2. Харченко О.Г. Метод багатокритеріальної оптимізації програмної архітектури на основі аналізу компромісів / Харченко О.Г., Боднарчук І.О., Галай І.О. // Інженерія програмного забезпечення. – 2012. – № 3–4 (11–12). – С. 5–11. 3. Гамма, Э. Приемы объектно-ориентированного проектирования. Паттерны проектирования / Э. Гамма, Р. Хелм, Р. Джонсон, Д. Влиссидес. – СПб. : Питер, 2010.

– 366 с. – ISBN 978-5-469-01136-1. 4. Barbacci, M. SEI Architecture Analysis Techniques and When to Use Them / Mario R. Barbacci.– Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2002. –CMU/SEI-2002-TN-005, ADA413696. – 34 p. 5. Kazman, R. ATAM[SM]: Method for Architecture Evaluation / Rick Kazman, Mark Klein, Paul Clements. – Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, August 2000. – CMU/SEI-2000-TR-004, ADA377385. – 83 p. 6. Kazman, R. Quantifying the costs and benefits of architectural decision / Kazman, R., Asundi, J., and Klein // Proceedings of the 23rd International Conference on Software Engineering (ICSE), 2001. – pp. 297 – 306. 7. Bengtsson, Perolof Architecture-level modifiability analysis (ALMA) / Perolof Bengtsson, Nico H. Lassing, Jan Bosch, Hans van Vliet // Journal of Systems and Software. – 2004. – Vol. 69, No. 1-2. – pp. 129-147. 8. Fowler, Martin. Patterns of Enterprise Application Architecture / Martin Fowler, David Rice, Matthew Foemmel, Edward Hieatt, Robert Mee, Randy Stafford.: Addison-Wesley, 2002. – 533 p. 9. Руководство Microsoft по проектированию архитектуры приложений. 2-е издание. Microsoft, 2009. – 529 с. 10. Letichevsky A. Basic Protocols, Message Sequence Charts, and the Verification of Requerements Specifications / A. Letichevsky, J. Kapitonova, A. Letichevsky Jr., V. Volkov, S. Baranov, V. Kotlyarov//ISSRE 2004, WITUL, Rennes, 4 November 2005. – pp. 112 – 142. 11. Harchenko A. Stability of the Solutions of the Optimization Problem of Software Systems Architecture /Harchenko A., Bodnarchuk I.,Halay I. // Proceedings of the VIIth International Scientific and Technical Conference CSIT 2012. – Lviv 2012. – pp. 47-49. 12. Al-Naeem, Taric. A Quality-Driven Systematic Approach for Architecting Distributed Software Applications / Tariq Al-Naeem, Ian Gorton, Muhammed Ali Babar, Fethi Rabhi, Boualem Benatallah // Proceedings of the 27th international conference on Software engineering ICSE-05. – ACM New York, NY, USA ©2005. – pp. 244 – 253. 13. Подиновский В. В. Введение в теорию важности критериев в многокритериальных задачах принятия решений. / Подиновский В. В. – М.: Физматлит, 2007. – 64 с. 14. Павлов О.А.Оперативные алгоритмы принятия решений в иерархической системе Саати, основанные на замещении критерие / Павлов О.А., Ліщук К.І. // Вісник НТУУ "КПІ". Інформатика, управління та обчислювальна техніка. К.: "ВЕК+", 2008.– №48. – С. 78 – 81.