

**УДК 004.415.5**

**В.І. Носа, Л.І. Шрам, М.О. Тимошик**

Тернопільський національний технічний університет імені Івана Пулюя, Україна

## **ДО ПРОБЛЕМИ БАГАТОКРИТЕРІЙНОГО ОЦІНЮВАННЯ ПРОГРАМНОЇ АРХІТЕКТУРИ НА ЕТАПІ ПРОЕКТУВАННЯ**

**V.I. Nosa, L.I. Shram, M.O. Tymoshyk**

## **TO THE PROBLEM OF MULTICRITERIA ASSESSMENT OF SOFTWARE ARCHITECTURE DURING DESIGN**

Архітектура програмної системи втілює колекцію архітектурних рішень, які відповідають вимогам багатьох атрибутів якості [1]. Для того, щоб дізнатись, який рівень вимог якості має бути досягнутий, оцінка якості архітектури має бути проведена на різних етапах життєвого циклу [2]. В один і той самий час архітектурне рішення може бути приведене у відповідність деяким атрибутам якості і перевищувати вимоги інших, всупереч природи атрибутів якості [3]. При появі конфлікту має бути встановлений пріоритет конкуруючих вимог, а потім використаний в процесі узгодження вимог для досягнення консенсусу [4]. Однак для комплексних архітектур метод багатокритеріального прийняття рішення може забезпечити більш формалізоване та числове обґрунтування. Одна з таких методик, як МАІ [5], була застосована для вибору бажаної архітектури з поміж альтернатив [4].

Застосування МАІ стандартним чином може забезпечити загальні важливі пріоритети для розроблених альтернатив. Він бере до розгляду усі атрибути якості, пріоритетні ваги розроблених альтернатив для окремих атрибутів якості та важливість кожного з атрибутів. Однак, якщо альтернатива вибрана з використанням МАІ, ключові компроміси у розробці виконуються для кожного атрибуту якості, і відносне значення компромісів не відображається явно.

МАІ ґрунтується на парних порівняннях для отримання різних пріоритетних ваг для альтернативних розробок та атрибутів якості. Це дає в результаті загальне ранжування розроблених альтернатив. У деяких випадках після кінцевого ранжування може бути найменша похибка у цих пріоритетних вагах. Ці чутливі пріоритетні ваги можуть бути інтерпретовані наступним чином:

**2.** Рішення, що призводять до малого відхилення ваг можуть бути найкритичнішими.

**3.** При постійній зміні вимог змінюються і пріоритети цих вимог. Невелике відхилення у вагових коефіцієнтах, що можуть привести до зміни ранжування, відображає чутливість результуючого ранжування до змін. Також потрібно це враховувати при прийнятті конструкторського рішення.

Стандартна практика застосування МАІ не має достатніх засобів для визначення точного рівня зроблених компромісів. Більше того, отримане ранжування може бути чутливе до найменших змін важливих пріоритетів, що може спотворити кінцеве ранжування.

Пропонується застосовувати методіку поглибленого аналізу отриманих з використанням стандартного МАІ результатів для того, щоб:

**1.** ідентифікувати можливі компроміси для архітектурної альтернативи разом з величиною цих компромісів;

**2.** виявити найкритичніші рішення у всьому процесі прийняття рішень;

**3.** оцінити чутливість кінцевого рішення та його можливості для обробки зміни пріоритетів атрибутів.

Ці результати роблять рішення більш чітким та ілюструють раціональність при прийнятті рішення на основі МАІ. Виявлення критичних рішень та виконання аналізу чутливості може виявити потенційні проблеми та спричинити кращу адаптацію

архітектури для майбутніх змін. На додачу, ця методика не обмежується лише певною предметною областю.

Оцінка архітектури програмного забезпечення стала важливою технікою в оцінюванні якості програмного забезпечення. Об'єктивний принцип оцінки архітектури полягає в тому, щоб виявити потенціал вибраної архітектури для досягнення можливостей системи задовольнити вимоги якості. Набір методів, таких як Метод аналізу архітектурних компромісів (АТАМ) [3], аналіз здатності підтримки на архітектурному рівні (ALMA) [6] були розроблені для оцінки проблем якості на архітектурному рівні. Оцінка архітектури може розглядатись як фаза процесу прийняття рішення. Процес прийняття рішення складається з наступних дій: ідентифікація задачі; аналіз задачі та розробка рішення; вибір та оцінка. Незважаючи на те, що оцінка архітектури сфокусована на вибір та дії щодо оцінки, вона (оцінка) також охоплює часто отримання рішення ітеративним шляхом.

Результати поточної оцінки архітектури протягом життєвого циклу проекту можуть бути позиціоновані на відрізку від неструктурованих (неостаточних) рішень та структурованих рішень. Оцінка архітектури дозволяє нам прогнозувати оптимальні атрибути якості, маючи справу з невизначеностями як у подальшій технології реалізації, так і у змінних вимогах. Таким чином, розглядається оцінка архітектури як процес прийняття рішення з неостаточними компонентами. Більшість методів оцінки архітектури проводять спочатку оцінку для індивідуальних атрибутів якості та об'єднують результати потім. Об'єднання вимагає процесу прийняття рішення для балансування компромісів та вибору найкращих кандидатів при конфлікті атрибутів якості. Відкритими для дослідження залишаються питання, як проводити аналіз компромісів і наскільки чутливим є кінцевий результат [7]. Пропонується розробити декілька нових методів аналізу для відповіді на ці запитання шляхом доповнення існуючих методів методом аналізу ієрархій. Більш точно, буде ідентифіковано критичні компроміси і точки чутливості у процесі МАІ.

#### **Література**

1. Fowler, Martin. Patterns of Enterprise Application Architecture / Martin Fowler, David Rice, Matthew Foemmel, Edward Hieatt, Robert Mee, Randy Stafford. – Boston: Addison-Wesley, 2002. – 533 p.
2. Bass, L. Software architecture in practice : 2nd edition / Len Bass, Paul Clements, Rick Kazman. – Boston, MA: Addison-Wesley Professional, 2003. – 528 p.
3. Kazman, R. ATAM<sup>SM</sup>: 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.
4. M. Svahnberg, C. Wholin, and L. Lundberg. A Quality-Driven Decision Support Method for Identifying Software Architecture Candidates. // Int. Journal of Software Engineering and Knowledge Engineering, 2003. 13(5): pp. 547-573.
5. Саати Т. Принятие решений. Метод анализа иерархий / Tomas Saaty; пер. с англ. Р.Г. Вачнадзе. – М.: Радио и связь, 1993. – 278 с.
6. 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.
7. Dobrica, L. A survey on software architecture analysis methods / L. Dobrica, E. Niemela // IEEE Transactions on Software Engineering. – Volume 28, Issue 7, NJ, USA: IEEE Press Piscataway – July, 2002. – pp. 638-653.