

*Запропонована авторська концептуальна принципова схема конфігуратора програмного комплексу та підхід до створення конфігуратора з врахуванням можливості оновлення компонентів з використанням методу аналізу ієрархій як інструменту для формування переліку конфігурацій та їх властивостей. Множина альтернатив (оптимальних конфігурацій) визначається як множина Парето. Глобальні ваги альтернатив знаходяться методами дистрибутивного, ідеального та мультиплікативного синтезів*

*Ключові слова: конфігуратор, конфігурована система, аналітична система модельної конфігурації, метод аналізу ієрархій*

*Предложена авторская концептуальная принципиальная схема конфигулятора программного комплекса и подход к созданию конфигулятора с учетом возможности обновления компонентов с использованием метода анализа иерархий как инструмента для формирования перечня конфигураций и их свойств. Множество альтернатив (оптимальных конфигураций) определяется как множество Парето. Глобальные веса альтернатив находятся методами дистрибутивного, идеального и мультипликативного синтезов*

*Ключевые слова: конфигулятор, конфигурированная система, аналитическая система модельной конфигурации, метод анализа иерархий*

# МЕТОД АНАЛІЗУ ІЄРАРХІЙ ЯК МЕТОД КОНФІГУРАЦІЇ СКЛАДНИХ СИСТЕМ

**В. Б. Задоров**

Кандидат технічних наук, професор\*

E-mail: vbz@voliacable.com

**О. О. Васильєв**

Аспірант\*

E-mail: anvasy@ukr.net

\*Кафедра інформаційних технологій

Київський національний університет

будівництва і архітектури

пр. Повітрофлотський, 31, м. Київ,

Україна, 03680

## 1. Вступ

Поняття «конфігурація» узагальнено означає стан структури для будь-якої системи. В інформаційних технологіях використовується, як правило, під час встановлення структури інформаційної системи на основі її складових з тієї чи іншої точки зору. Найбільш суттєвим та складним у створенні сучасних ІТ є конфігурування програмних комплексів, що призначені для використання в реальних умовах змін ситуаційних вимог до їх функціональності на основі інтеграції існуючих пакетів програм та тих, що розроблюються для виконання нових функцій (основних, додаткових, забезпечуючих) і для забезпечення сумісності. Таке конфігурування звичайно доводиться здійснювати в умовах вибору варіантів альтернатив окремих складових багатфункціональних програмних комплексів для вирішення різних задач в тих чи інших предметних областях, які розглядаються як складні системи. Необхідність періодичного переконафігурування в умовах ситуаційних змін особливо стосується складних програмних комплексів з підготовки та управління будівництвом, специфіка яких позначається широким спектром існуючих пакетів програм, за допомогою яких мають бути комплексно реалізовані певні функції архітектурно-будівельного, інженерного, технологічного, організаційно-технологічного проектування, вирішення задач нормативного і економічного, планового напрямків тощо. Для таких взаємопов'язаних складних систем, як сучасні ІТ в

будівництві, існуючі підходи до періодичного переконафігурування програмних комплексів потребують подальшого розвитку.

## 2. Цілі та задачі дослідження

В сучасних умовах розвитку ІТ з підготовки та управління будівництвом важливим є вирішення проблеми конфігурування та переконафігурування програмних комплексів на основі запропонованої нової концепції, яка враховує ситуаційні вимоги до змін їх функціональності.

Розглянемо запропонований підхід більш детально.

## 3. Аналіз літературних даних, вимоги до конфігураторів і обґрунтування нової концепції їх побудови

*Аналіз конфігуратора.* Конфігуратор – це комплекс програм, який дозволяє структурувати і організовувати комплексні ІТ для інженерних, комунікаційних, економічних, організаційних предметних областей з реалізацією певних функціональних задач із заданою кінцевою метою і адаптацією до конкретних типових ситуацій.

Авторський аналіз конфігуратора як системи показав, що в будь-якому конфігураторі можна виділити три частини [1] (рис. 1):

- систему, яка піддається конфігурації (конфігуровану систему);
- систему, яка визначає конфігурацію (аналітичну систему модельної конфігурації);
- систему, яка виконує конфігурування (конфігуруючу систему).

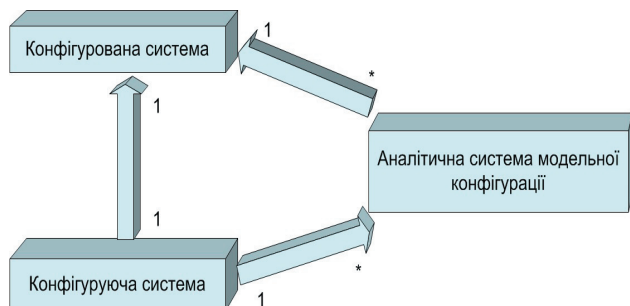


Рис. 1. Авторська принципова схема конфігуратора:  
 → – залежність/дія, 1/\* – тип зв'язку: один до багатьох,  
 1/1 – тип зв'язку: один до одного

Web-сторінки виконують роль *конфігуруючої системи*, тоді як *конфігурована система* найчастіше є лише відображенням реальних предметів. Наприклад, конфігуратори автомобілів, комп'ютерів, інтер'єра, меблів тощо. *Аналітична система модельної конфігурації* в даному випадку – це інформація про те, які складові, в якій кількості і з якими властивостями використовуються для того, щоб описати конкретний автомобіль, зібраний ПК або оснащену кімнату. Потім, як правило, ця інформація (вже як *конфігурована система*) зберігається і надсилається менеджеру для погодження з клієнтом.

Розглянемо більш складні конфігуратори – ті, які конфігурують реальні IT системи. В таких конфігураторах всі три складові (конфігурована система, аналітична система модельної конфігурації і конфігуруюча система) є повноцінними і явно вираженими. До них, наприклад, належать:

- конфігуратори Web – додатків, такі як конфігуратори Інтернет – форумів, конфігуратори Інтернет – магазинів, конфігуратори будь-яких систем, що базуються на Web-інтерфейсі або мають Web-інтерфейс (приклад: багтрекінгова система JIRA);

- конфігуратори IT систем загалом і розподілених IT систем: 1С: Бухгалтерія, 1С – Битрикс. Управление сайтом [2]; конфігуратори інженерних САПР – спеціальні програмні продукти AutoDesk, конфігуратори економічних організаційних систем – це, наприклад, SAP R3 [3] і BAAN 4 [4], конфігуратор системних блоків для спрощення процесу підбору комплектуючих у склад збірного виробу [5], Shoreline Firewall [6] – інструмент для настройки міжмережевого екрану в Linux.

*1С - Предприятие* – універсальний комплекс програм масового призначення для автоматизації функцій управління, бухгалтерського і податкового обліку в організаціях, що здійснюють виробничу і комерційну діяльність [2].

*1С - Битрикс: Управление сайтом* – професійна система управління Web – проектами. Цей універсальний програмний продукт розроблений для створення, підтримки і успішного розвитку сайтів. Система оріє-

єнтована на корпоративні сайти, інформаційні і довідкові портали, соціальні мережі, Інтернет - магазини, сайти ЗМІ, а також може бути придатна для створення інших видів Web – ресурсів [2].

Для зберігання даних сайту використовується реляційна СУБД. Підтримуються наступні СУБД: MySQL, Oracle, MS SQL. Продукт працює на Microsoft Windows і UNIX – подібних платформах, включаючи Linux. *1С – Битрикс: Управление сайтом* реалізований в одній з семи складених фірмою-розробником редакцій (*Старт, Стандарт, Эксперт, Малый бизнес, Бизнес, Портал, Большой бизнес*). Кожен з цих комплектів визначає свій набір модулів і функціональність конфігурованої системи.

*Система SAP R/3* складається з набору прикладних модулів, які підтримують різні бізнес-процеси компанії і інтегровані у єдиний програмний комплекс, що функціонує в масштабі реального часу [3].

Навіть найкоротший огляд функцій системи SAP R/3 показує її здатність вирішувати основні завдання, що стоять перед крупними організаціями. SAP R/3 – одна з небагатьох систем, що є повноцінним конфігуратором, який має в своєму складі конфігуруючу систему і забезпечує аналітичну систему модельної конфігурації, а також створює конфігуровану систему. Підприємство працює з індивідуальною сконфігурованою версією системи, налагодженою саме під його параметри. Показником технічного рівня конфігуратора може служити спосіб його настройки. Чим ширше можливості конфігуратора і настройки системи без необхідності її переписування, тим вище технічний рівень даної системи. За допомогою спеціалізованого інструменту бізнес-інжинірингу (Business Engineer) можна конфігурувати і налаштувати систему SAP R/3 так, щоб вона задовольняла потребам підприємства, підтримувати цю відповідність протягом всього життєвого циклу системи. Бізнес-конфігуратор SAP R/3 підтримує процедури створення і ведення аналітичних моделей конфігурації підприємства з автоматичною генерацією відповідних завдань і профілів настройки.

BAAN IV – система для розробки рішень для управління підприємствами з високотехнологічним виробництвом і корпоративною логістикою. Основні модулі ERP-системи BAAN IV: *Моделирование предприятия*, яка містить засоби конфігуруючої системи і засоби аналітичної модельної конфігурації [4], і *Производство*, що включає планування потреб, конфігуратор продукції. Також розроблені *BAAN – Процесс*, *BAAN – Финансы*, *BAAN - Сбыт*, *BAAN - Снабжение*, *BAAN – Проект*, *BAAN - Администратор деятельности предприятия*, *BAAN – Транспорт* і *BAAN – Сервис*.

*Конфігуратор системних блоків* призначений для спрощення процесу підбору комплектуючих до складу збірного виробу. З його допомогою можна легко і швидко розрахувати підсумкову вартість системного блоку з комплектуючих, що є в наявності на складі компанії. Отриману модельну конфігурацію можна як зберегти (при цьому конфігурація отримує свій індивідуальний код, за яким надалі до неї можна буде повернутися), так і відправити замовлення на її збирання [5].

*Shorewall або точніше Shoreline Firewall* – інструмент для настройки міжмережевого екрану в Lin-

ух, програмне забезпечення під вільною ліцензією GNU GPL [6] є надбудовою над підсистемою Netfilter (iptables/ipchains) ядра Linux і забезпечує спрощені методи конфігурації даної підсистеми. Shorewall не передбачає GUI для конфігурації, правка конфігураційних файлів може виконуватися в будь-якому текстовому редакторі.

Розглянуті приклади показують, що визначенню конфігуратора можуть відповідати практично будь-які системи ІТ, оскільки скрізь можна виділити три складові конфігуратора, нехай навіть у виродженому вигляді.

Дійсно, чіткої межі не існує. Проте, при створенні інтелектуальних складних ІТ бізнес-систем, що розвиваються, доцільно вважати, що конфігуратором може бути лише ІТ система з наявністю явно виражених і функціонально виділених трьох складових: конфігуруючої системи, аналітичної системи модельної конфігурації і конфігурованої системи. Як правило, такі системи і отримують назву від своїх авторів, пов'язану з поняттям «конфігуратор», наприклад «Конфігуратор – Битрикс».

*Типи конфігураторів.* Метод аналізу ієрархій використовується для всіх типів конфігураторів за рівнем складності.

*Прості конфігуратори.* Не змінюють структури і модулі ПЗ, а змінюють лише настройки інтерфейсу, параметри виконання конкретних функцій. Зберігаються на клієнтській стороні системи. Цей тип конфігураторів найчастіше зустрічається на практиці і найбільш простий в реалізації. Як правило, параметри зберігаються у \*.ini – файлах, в системному реєстрі або у \*.config-файлах. Без подібного рівня конфігурації система не може вважатися повноцінною. Будь-який комерційний проект містить мінімальний набір можливостей для настройки. Як приклад таких параметрів конфігурації можна навести розмір/стиль вікон інтерфейсу, тип і розмір шрифту, наявність/відсутність панелей і рядків стану тощо.

*Ускладнені конфігуратори.* Вони можуть видозмінювати структуру і модулі ПЗ. Зберігаються на клієнтській стороні. Зустрічаються в великих багатифункціональних програмних продуктах, наприклад, графічних пакетах (3DMax [7], AutoCad [8]). Через наявність великого числа функцій виникає необхідність в їх структуризації і налаштуванні для конкретного користувача.

*Розподілені конфігуратори.* Вони мають ті ж властивості, що і в попередньому типі, проте дані конфігурації можуть зберігатися на віддаленому сервері і завантажуватися для конкретного користувача на локальний комп'ютер.

*Конфігуратори з можливістю завантаження компонентів.* До цього розглядалися конфігуратори, які використовували лише наявні компоненти. Цей тип конфігураторів, у свою чергу, дозволяє в міру необхідності розширювати складові блоки системи, що конфігурується. «Підвантаження» може виконуватися з будь-якого можливого джерела – локально, по локальній мережі від сервера, або навіть з Інтернету.

*Вибір найбільш ефективної моделі реалізації конфігуратора.* Враховуючи вимоги до створення, експлуатації та розвитку бізнес-систем, був обраний підхід до створення конфігуратора [1, 9], який враховує мож-

ливість оновлення компонентів, що є найважливішим критерієм для складних змінюваних систем.

Концептуальна принципова схема запропонованого конфігуратора наведена на рис. 2.

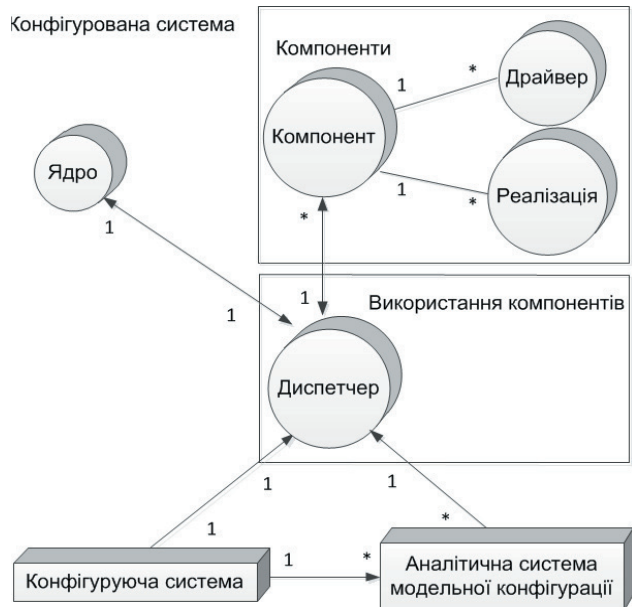


Рис. 2. Концептуальна принципова схема запропонованої структури конфігуратора: → – залежність/дія, 1/\* – тип зв'язку: один до багатьох, 1/1 – тип зв'язку: один до одного

Побудовані на базі такої структури конфігуратора складні системи повинні містити три частини:

- система, що піддається конфігурації (конфігурована система), яка забезпечує ефективне оперативне функціонування бізнес-системи в стабільних умовах і в стандартних ситуаціях;
- система, що визначає конфігурацію, тобто та, яка визначає модельні вимоги до конфігурації на ранніх етапах проектування і на етапах розвитку системи (її можна назвати засобом системної аналітичної модельної конфігурації бізнес-системи);
- система, що виконує конфігурацію (конфігуруюча система), яка забезпечує інтеграцію різнофункціональних інформаційних технологій і підтримує інструментальні засоби реструктуризації конфігурованих систем і засобів системної модельної конфігурації.

Саме для аналітичної системи модельної конфігурації та конфігуруючої системи виникає необхідність постійно вирішувати, по-перше, задачу вибору ефективних компонентів програмного забезпечення, по-друге, задачу підбору ефективного портфелю (пакету) альтернативних програмних компонентів, що максимально враховують можливі ситуаційні вимоги до змін функціональності конфігурованого програмного комплексу.

Вирішення цих задач має бути здійснено деяким спеціалізованим програмним модулем, що міститься в конфігуруючій системі. Для реалізації такого спеціалізованого програмного модулю найбільш ефективним є застосування та розвиток методу аналізу ієрархій [10–13].



### 3. Використання методу аналізу ієрархій у конфігуруючій системі

Ідея методу аналізу ієрархій [10–13] полягає у структуризації задач прийняття рішень шляхом побудови багаторівневої ієрархії, що об'єднує всі значущі компоненти задачі (головна мета, підлеглі цілі, дійові сили, критерії, результати, альтернативи), які порівнюються між собою за допомогою спеціально розроблених процедур. В результаті стає можливим отримання чисельних оцінок інтенсивності взаємного впливу елементів ієрархії, на основі яких оцінюються ступені переваги альтернатив відносно головної мети. Ідея методу відповідає принципам системного підходу.

Метод аналізу ієрархій використовується у конфігуруючій системі для створення довільної конфігурації за вимогами користувача, замовника-керівника, або адміністратора. У кожного з них можуть бути свої вимоги, які може задовольнити метод аналізу ієрархій.

При реалізації концепції конфігуратора змінюються вимоги до архітектури сучасних ІТ бізнес-систем на всіх стадіях їх життєвого циклу.

Однією з важливих задач конфігуратора є підбір множини конфігурацій, що найбільше задовольняють користувача, якнайкраще відповідають бізнес-вимогам та конкретному формату системи, яка потрібна користувачеві.

У спрощеному вигляді задача може вирішуватися в ручному режимі, однак при збільшенні кількості вимог, обмежень та особливостей стає актуальною задача пошуку оптимальної або квазіоптимальної конфігурації - множини конфігурованих програмних компонент, що у сукупності забезпечують користувачеві необхідні умови в конкретних ситуаціях з вирішення комплексів функціональних задач.

#### 3. 1. Схеми методу аналізу ієрархій для вибору ефективних конфігурацій складних комплексних багатофункціональних програмних продуктів

Авторами було запропоновано використання математичного методу аналізу ієрархій [10–13] як інструменту для формування переліку конфігурацій та їх властивостей, що найбільше задовольняють користувача в умовах конкретних ситуацій, що складаються. Метод було адаптовано до особливостей розглянутої вище предметної області – підготовки та управління в будівництві на всіх фазах життєвого циклу створення та експлуатації готової будівельної продукції.

Узагальнена схема рівнів методу аналізу ієрархій наведена на рис. 3.

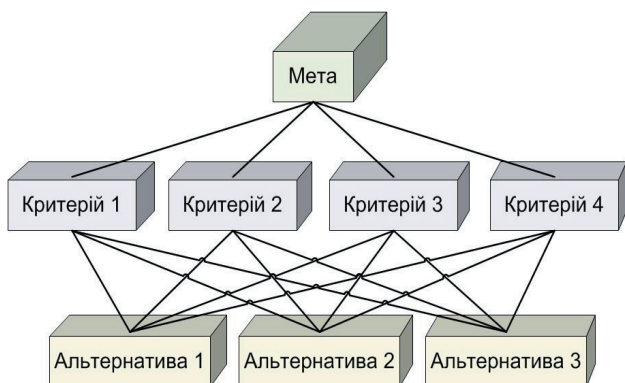


Рис. 3. Схема рівнів методу аналізу ієрархій

Ієрархії складаються з низки рівнів (мета, критерії, альтернативи), всі елементи структури пов'язані між собою певними зв'язками.

У залежності від вимог користувача процес підбору конфігурацій може бути поділений на *простий або одинарний процес підбору конфігурацій*, тобто процес підбору оптимального одинарного програмного компоненту, та *складений процес підбору конфігурацій*, тобто процес підбору множини програмних компонентів.

Одинарний процес підбору конфігурацій має на меті пошук одного компоненту програмного забезпечення, який найбільш точно відповідає всім критеріям і обмеженням (рис. 4).

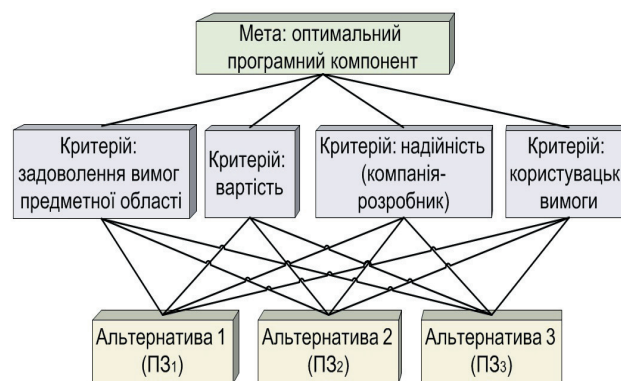


Рис. 4. Одинарний підбір конфігурації: задача підбору оптимального компоненту програмного забезпечення (одинарного)

У випадку одинарного підбору конфігурації альтернативами є безпосередньо програмні компоненти, в той час як критеріями є певні фактори, за якими оцінюється той чи інший програмний засіб на відповідність вимогам користувача.

Слід зазначити, що, як правило, критерії можуть бути піддані декомпозиції. Таким чином, розрахунок ваги критерію та значення критерію для певної альтернативи, що є основою методу аналізу ієрархій, виконується ієрархічно. Оцінка критерію для певної альтернативи визначається експертною оцінкою відповідності програмного компоненту певним вимогам, а саме бізнес-вимогам, гнучкості та надійності програмного компоненту, вимогам до моделі вартості, до компанії-розробника тощо.

Використовуються домінуючі ієрархії. Для оцінки елементів кожного рівня ієрархії використовується шкала парних порівнянь Т. Сааті [14] (табл. 1). Побудовані відповідні матриці парних порівнянь умовних показників, які дозволяють розраховувати глобальну вагомість ієрархічної піраміди.

Результати всіх парних порівнянь виражаються у чисельній формі, що дозволяє виконувати подальші розрахунки. Таким чином, може бути виконаний пошук оптимального програмного компоненту для включення в конфігуровану модель програмного комплексу.

Другим варіантом задачі є пошук складеної конфігурації, тобто пакету (портфелю) програмних компонентів (рис. 5), які повинні бути складовими оптимальної бібліотеки системи, що виконує і підтримує

ефективну конфігуровану модель програмного комплексу, тобто конфігуруючої системи.

Таблиця 1

Шкала парних порівнянь Т. Сааті

Відносна важливість (бали)	Визначення	Пояснення
1	однакова важливість	обидва елементи вносять однаковий вклад
3	перший елемент трохи важливіший за другий	досвід дозволяє поставити перший елемент трохи вище за другий
5	суттєва перевага	досвід дозволяє встановити безумовну перевагу першого над другим
7	значна перевага	перший елемент настільки важливіший за другий, що є практично значимим
9	абсолютна перевага першого над другим	очевидність переваги підтверджується більшістю
2, 4, 6, 8	проміжні оцінки між сусідніми твердженнями	компромісне рішення
обернені величини чисел, наведених вище	якщо при порівнянні першого елемента з другим отримане одне з вищевказаних чисел (1–9), то при порівнянні другого з першим, матимемо обернену величину	

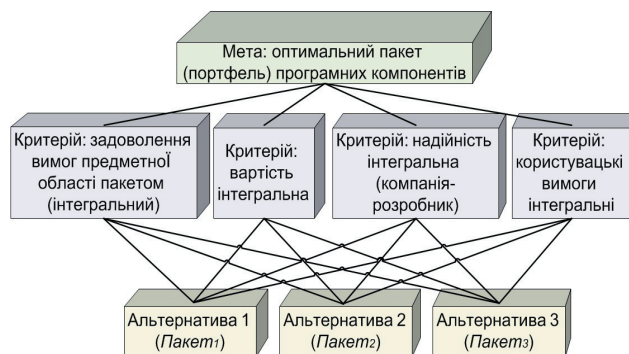


Рис. 5. Складений підбір конфігурації: задача підбору оптимального пакету (портфелю) програмних компонентів

Слід зазначити, що такий тип задачі є значно складнішим, особливо при ручному підході. В даному випадку альтернативами є пакети (портфелі) програмних компонентів, в той час як критерії також оцінюють групу програмних компонентів в комплексі. Однією з важливих складових є генерація переліку альтернатив, тобто конфігурацій, що найбільше задовольняють користувача. З математичної точки зору множина альтернатив визначається як множина Парето.

На цьому етапі система створює всі можливі комбінації доступних програмних компонентів та відбра-

ковує варіанти конфігурацій, що повністю не задовольняють користувача, обираються лише елементи, що входять до множини Парето. Кожний критерій є складним, адже оцінює в комплексі множину програмних компонентів, так само, як і для простого підбору кожен з критеріїв може бути підданий декомпозиції. Попарне порівняння альтернатив з критеріями базується на експертних оцінках як кожного окремого компоненту програмного забезпечення, що входить до пакету, так і комбінації пакетів програмного забезпечення в комплексі за даним критерієм.

### 3. 2. Математичне представлення методу аналізу ієрархії

Математично пакет програмного забезпечення представляється як множина компонентів програмного забезпечення (1):

$$\text{Пакет} = \Pi(\text{ПЗ1}, \text{ПЗ2}, \text{ПЗ3а}, \dots, \text{ПЗ3N}). \quad (1)$$

Головна функція генерації групи пакетів  $F_{\text{ПАК.ГЕНЕР}}$  базується на наборі вхідного програмного забезпечення (2–3):

$$\text{Пакет } i = \Pi(\text{Пакет1}, \text{Пакет2}, \text{Пакета}, \dots, \text{ПакетN}), \quad (2)$$

$$\text{Пакет } i = F_{\text{ПАК.ГЕНЕР}}(\text{Пакет1}, \text{Пакет2}, \text{Пакета}, \dots, \text{ПакетN}). \quad (3)$$

Розрахунок глобальних ваг альтернатив для ієрархії з двома рівнями;

*Постановка задачі розрахунку глобальних ваг альтернатив*

Задано:

$$K = \{K_j | j = \overline{1, m}\} - \text{множина критеріїв,}$$

$$A = \{A_i | i = \overline{1, n}\} - \text{множина альтернатив рішень,}$$

$V_{LA} = \{(v_{LAij}) | i = \overline{1, n}, j = \overline{1, m}\}$  – локальні ваги альтернатив відносно критеріїв, ненормовані,

$$V_K = \{(v_{Kj}) | j = \overline{1, m}\} - \text{ваги критеріїв: } \sum_{j=1}^m v_{Kj} = 1.$$

Знайти:

$V_{GA} = \{(v_{GAi}) | i = \overline{1, n}\}$  – глобальні ваги альтернатив, нормовані.

Глобальні ваги альтернатив знаходяться методами дистрибутивного, ідеального та мультиплікативного синтезів [15], які підтримуються конфігуратором.

*Дистрибутивний синтез*

Глобальна вага альтернативи  $A_i$  розраховується за формулою (4):

$$v_{GAi} = \sum_{j=1}^m v_{Kj} \cdot r_{ij}, \quad i = \overline{1, n}, \quad (4)$$

де  $r_{ij}$  - нормовані значення ваг  $v_{LAij}$  (5):

$$r_{ij} = \frac{v_{LAij}}{\sum_{p=1}^n v_{LApj}}. \quad (5)$$

Таким чином, виконується умова  $\sum_{i=1}^n r_{ij} = 1, j = \overline{1, m}$ , отже  $\sum_{i=1}^n v_{GAi} = 1$ .

**Ідеальний синтез**

Глобальна вага альтернативи  $A_i$  розгортається так само, як і в методі дистрибутивного синтезу, за допомогою адитивної згортки (6):

$$v_{GAi} = \sum_{j=1}^m v_{kj} \cdot r_{ij}, i = \overline{1, n}, \quad (6)$$

але  $r_{ij}$  – нормовані значення ваг  $v_{LAij}$ , отримуються шляхом ділення  $v_{LAij}$  на найбільше із значень ваг альтернатив за даним критерієм (7):

$$r_{ij} = \frac{v_{LAij}}{\max_{p=\overline{1, n}} v_{LApj}}, j = \overline{1, m}. \quad (7)$$

У методі ідеального синтезу на суму нормованих ваг альтернатив за кожним критерієм не накладається умова рівності одиниці.

**Мультипликативний синтез**

У цьому методі використовується метод зваженого добутку, згідно з яким при порівнянні альтернатив  $A_i$  та  $A_p$  розраховується добуток (8):

$$P\left(\frac{A_i}{A_p}\right) = \prod_{j=1}^m \left(\frac{v_{LAij}}{v_{LApj}}\right)^{v_{kj}}; i, p = \overline{1, n}. \quad (8)$$

Якщо величина  $P\left(\frac{A_i}{A_p}\right)$  (8) більша або дорівнює

одиниці, тоді альтернатива  $A_i$  є важливішою за альтернативу  $A_p$ .

Глобальна вага  $A_i$  розраховується таким чином:

$$v_{GAi} = \prod_{j=1}^m (v_{LAij})^{v_{kj}}, i = \overline{1, n}.$$

**3. 3. Приклад**

Розглянемо декомпозицію критеріїв підбору портфелю програмних компонентів для прикладу комплексу функціональних задач з підготовки будівництва на заданій фазі життєвого циклу створення і експлуатації готової будівельної продукції, а саме на передпроектному етапі інвестиційної оцінки будівельного проекту. Для кожного критерію (групи критеріїв) виділяємо субкритерії (рис. 6).

Декомпозиція групи критеріїв – *задоволення вимог предметної області*:

- ефективність методики (ЕМ) інвестиційної оцінки даного типу будівельного проекту (підхід, врахування складових інвестиційної оцінки, ризиків, тощо);
- можливість забезпечення вихідними даними (ЗВД).

Декомпозиція критерію – *вартість*:

- вартість інсталяції пакету (ВІП) з інвестиційної оцінки даного типу будівельного проекту;
- вартість використання пакету (ВВП) з інвестиційної оцінки даного типу будівельного проекту (точна плата за підтримку тощо).

Декомпозиція групи критеріїв – *надійність*:

- універсальність пакету (УП) з інвестиційної оцінки даного типу будівельного проекту;
- рівень поширення пакету (РПП) з інвестиційної оцінки даного типу будівельного проекту;
- сучасність застосування пакету (СЗП) з інвестиційної оцінки даного типу будівельного проекту (інтегрованість, наявність можливостей експорту-імпорту тощо);
- імідж компанії-розробника (ІР).

Декомпозиція групи критеріїв – *користувацькі вимоги*:

- зручність інтерфейсу користувача (ЗІК);
- наявність ефективного керівництва користувача (ЕКК);
- наявність та забезпеченість ефективних засобів формування комплексу результуючої вихідної документації (ЕД) тощо.

Зводимо декомпозицію задачі в ієрархію (рис. 6).

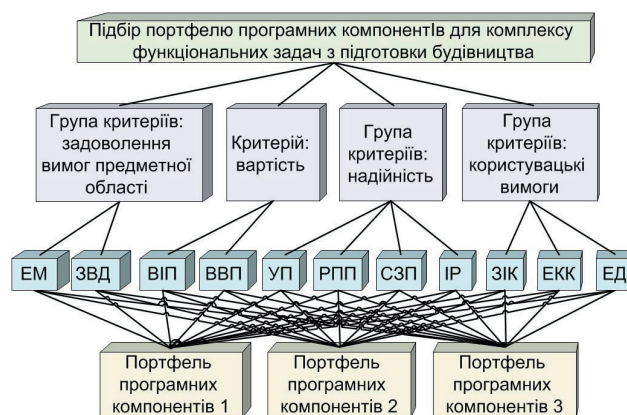


Рис. 6. Приклад декомпозиції критеріїв підбору портфелю програмних компонентів для комплексу функціональних задач з підготовки будівництва на передпроектному етапі інвестиційної оцінки будівельного проекту:

ЕМ – ефективність методики; ЗВД – забезпечення вихідними даними; ВІП – вартість інсталяції пакету; ВВП – вартість використання пакету; УП – універсальність пакету; РПП – рівень поширення пакету; СЗП – сучасність застосування пакету; ІР – імідж компанії-розробника; ЗІК – зручність інтерфейсу; ЕКК – наявність ефективного керівництва користувача; ЕД – наявність та забезпеченість ефективних засобів формування комплексу результуючої вихідної документації

Врахування суттєвих складових встановлених груп критеріїв дозволяє більш обґрунтовано здійснювати вибір з альтернативних портфелів програмних компонентів. Зауважимо, що в конкретних ситуаціях за необхідністю можуть бути побудовані наступні рівні більш детальних критеріїв в разі можливості їх кількісної чи якісної оцінки.

**4. Висновки**

Конфігуратор є ефективним і гнучким засобом структурування і організації складних бізнес-систем, а з використанням методу аналізу ієрархій, який дозволяє підбирати оптимальні конфігурації для потреб користувача, побудовані на базі конфігуратора складні

системи є значно більш ефективними в умовах ситуаційних змін за монолітні системи з постійним складом програмних продуктів. З використанням конфігуратора користувач отримує найвигідніший з точки зору бізнесової ефективності пакет програмних продуктів в умовах конкретних ситуацій з вирішення комплексів тих чи інших функціональних задач.

Було досліджено проблему конфігурування програмних комплексів на основі концепції конфігураторів та запропоновано підхід до підбору ефективного

портфелю (пакету) альтернативних програмних компонентів, який базується на використанні та розвитку методу аналізу ієрархій, що дозволяє знайти оптимальний розв'язок поставленої задачі.

Отримані результати використані при розробці експериментального конфігуратора інформаційних технологій у будівництві (KIT-Б) – Configurator of IT – the Patterns-maker of Building (C-IT PMB) [16], задачею якого є конфігурування ІТ на певних етапах життєвого циклу створення об'єкту будівництва.

## Література

1. Задоров, В. Б. К развитию концепции «конфигураторов» для построения архитектуры информационных технологий организационных антропогенных систем [Текст] / В. Б. Задоров, А. А. Васильев // Управління розвитком складних систем. – 2011. – Вип. 6. – С. 107–116.
2. 1С-Битрикс: Управление сайтом [Электронный ресурс] / Режим доступа : <http://www.1c-bitrix.ru/products/cms>.
3. ERP online. SAP R/3 [Electronic resource] / Available at : <http://www.erp-online.ru/sap>. – Title screen.
4. ERP online. BAAN [Electronic resource] / Available at : <http://www.erp-online.ru/baan>. – Title screen.
5. Business Applications Guide [Text] / High Performance Systems, Inc., 1994. – 170 p.
6. Shorewall. IPtables made easy. Important Notices [Electronic resource] / Available at : <http://Shorewall.net/Notices.html>. – Title screen.
7. Smith, Brian L. 3ds Max 2010 Architectual Visualization [Electronic resource] / Brian L. Smith. – Support Files for 3DATS. CGSchool Books, 2009. – Available at : <http://www.3dats.com/books/>. – Title screen.
8. Omura, G. Mastering AutoCAD 2013 and AutoCAD LT [Electronic resource] / G. Omura, B. C. Benton. – 2013. – Available at: <http://www.amazon.com/>. – Title screen.
9. Задоров, В. Б. К развитию концепции конфигураторов при создании информационных технологий в строительстве [Текст] : матер. IX міжн. наук.-тех. конф. / В. Б. Задоров, А. А. Васильев // Новітні комп'ютерні технології. – Київ-Севастополь. Мін-регіон України, 2011. – С. 29–31.
10. Hollowell, D. L. Analytical Hierarchy Process (AHP) – Getting Oriented [Electronic resource] / D. L. Hollowell. – Available at: <http://www.isixsigma.com/tools-templates/analytic-hierarchy-process-ahp/analytical-hierarchy-process-ahp-getting-oriented/> - 26.02.2010 p. – Title screen.
11. Analytic Hierarchy Process (AHP) [Electronic resource] / QFD Institute. – Available at: <http://www.qfdi.org/mobile/process.html/>. – Title screen.
12. Analytical Hierarchy Process: Overview. TheQualityPortal.com [Electronic resource] / Available at : [http://thequalityportal.com/q\\_ahp.htm/](http://thequalityportal.com/q_ahp.htm/). – Title screen.
13. Саати, Т. Аналитическое планирование. Организация систем [Текст] / Т. Саати, К. Керне. – М.: Радио и связь, 1991. – 224 с.
14. Саати, Т. Принятие решений. Метод анализа иерархий [Текст] / Т. Саати. – М.: Радио и связь, 1993. – 278 с.
15. Недашківська, Н. І. Оцінювання реверсу рангів у методі аналізу ієрархій [Текст] / Н. І. Недашківська // Системні дослідження та інформаційні технології. – 2005. – № 4. – С. 120–130.
16. Задоров, В. Б. Архітектура конфігуратора PMB інформаційних систем в будівництві [Текст] : матер. Другої міжн. наук.-прак. конф. / В. Б. Задоров, О. О. Васильев // Сучасні інформаційні системи і технології. – Суми: Сумський державний університет, 2013. – С. 57–58.