

МОДЕЛІ ТА МЕТОДИ СИНТЕЗУ ОПИСУ РАЦІОНАЛЬНОЇ АРХІТЕКТУРИ ІНФОРМАЦІЙНОЇ СИСТЕМИ

© Євланов М. В., Васильцова Н. В., Панфьорова І. Ю., 2015

Розглянуто моделі та методи, використання яких дає змогу розв'язати задачу синтезу опису архітектури інформаційної системи. Розроблено метод синтезу варіантів опису архітектури інформаційної системи. Запропоновано теоретико-ігрову модель синтезу опису раціональної архітектури інформаційної системи. Розглянуто особливості використання синтезованого опису для визначення проектних обмежень.

Ключові слова: інформаційна система, архітектура, вимога, ІТ-послуга, семантична мережа.

This paper describes the models and methods, the use of which can solve the problem of synthesis of architecture description of information system. The method of synthesis of options of architecture description of information system was developed. A game-theoretical model describing the synthesis of rational architecture describe of information system. Features of using synthesized description to determine project constraints are described.

Key words: information system, architecture, requirement, IT-accommodation, semantic network.

Вступ. Загальна постановка проблеми

Сучасний підхід до організації процесів створення, впровадження, експлуатації та модернізації інформаційних систем (ІС) ґрунтується на запропонованих стандартом ISO 15288:2002 описах технічних процесів життєвого циклу систем. За цим стандартом весь життєвий цикл ІС складається з таких процесів [1]:

- 1) процес визначення вимог правовласників;
- 2) процес аналізу вимог;
- 3) процес проектування архітектури;
- 4) процес реалізації елементів системи;
- 5) процес комплексування;
- 6) процес верифікації;
- 7) процес передачі;
- 8) процес валідації;
- 9) процес функціонування;
- 10) процес технічного обслуговування;
- 11) процес вилучення та списання.

Аналіз наведених процесів вказує, що одним з найважливіших артефактів життєвого циклу ІС є архітектура системи [1]. Під архітектурою розуміють фундаментальні поняття і властивості системи в середовищі, що її оточує, які втілено в її елементах, відношеннях, а також у принципах її проектування та розвитку [2]. Це визначення, на думку авторів стандарту ISO/IEC 42010:2010, є максимально загальним визначенням, придатним для опису архітектур практично будь-яких систем, зокрема – ІС.

Тепер архітектура ІС розглядається переважно як сукупність описів системи на різних рівнях абстракції. При цьому можуть використовуватися різні мови, що забезпечують формування таких описів – Rapide, SysML, ArchiMate, ACME, xADL тощо. Але, незалежно від підходу до формального опису архітектури ІС, виникає необхідність розв'язання задачі синтезу такого опису архітектури системи, який задовольняв би як основні проектні обмеження, так і вимоги, що

ставляться до створюваної ІС. Тому дослідження моделей і методів, які дозволяють формалізувати цей процес та розв'язувати задачу синтезу раціонального опису архітектури ІС, є актуальними як з теоретичного, так і з практичного погляду.

Аналіз основних особливостей архітектури інформаційної системи

Поки що не існує єдиного погляду на формальний опис архітектури ІС. Різні автори вживають цей термін із різним змістом або ж виділяють у складі архітектури різну кількість складових компонентів – від двох до семи чи більше [3–7]. Так, наприклад, у [4] пропонується розуміти термін “архітектура ІС” як концепцію, що визначає модель, структуру, виконувани функції та взаємозв'язок компонентів ІС. В [7] пропонується розглядати архітектуру ІС як сталу сукупність структурних, функціональних та споживацьких характеристик системи, що розробляється, наявність яких забезпечує такі властивості або якості системи: задана продуктивність системи (system performance); надійність функціонування (reliability); захищеність / безпека експлуатації системи (safety / security); можливість її ефективного супроводження та розвитку (maintenance and evolution). Іноді, визначаючи термін “архітектура ІС”, додатково говорять про конкретні естетичні вимоги до створюваної ІС [5, 6]: добре спроектована ІС повинна мати набір ознак привабливості для користувача (to be presentable for user). Важливою також є відмінність терміна “архітектура ІС” від терміна “структура системи”: якщо структура системи описує тільки статичні аспекти її побудови, то архітектура ІС визначає також і динаміку поведінки (behaviour) відповідної ІС, а також визначає інтерфейси її взаємодії з навколишнім середовищем [7].

Однак аналіз сучасного визначення терміна “архітектура системи”, який зафіксовано у стандарті ISO/IEC 42010:2010, дає змогу вказати на такі відмінні особливості того, що розуміють під цим терміном. Так, зазначений стандарт не накладає ніяких обмежень на визначення терміна “система” для користувачів цього стандарту. Розділення термінів “фундаментальні поняття або властивості” здійснено для підтримки двох різних філософій архітектури без надання переваги будь-якій з них. Йдеться про такі філософії [2]:

- архітектура як сукупність фундаментальних понять (концепція) – абстрактне визначення системи;

- архітектура як сприйняття – сприйняття властивостей системи.

З погляду будь-якої з цих філософій архітектура є абстракцією, а не артефактом. Для визначення артефактів, які можуть використовуватися для вираження та опису архітектури, стандарт ISO/IEC 42010:2010 вживає інший термін – “опис архітектури”.

Відповідно до описів процесів життєвого циклу ІС, наведених у [1], основними даними, необхідними для розв'язання задачі синтезу опису архітектури ІС, є вимоги до цієї системи. Але аналіз досліджень показує [8], що проблему перетворення неформальних описів потреб правласників ІС на формальні описи вимог до створюваної ІС дослідники вирішують переважно на концептуальному рівні. Серед напрямів дослідження проблеми формального опису вимог до ІС тепер особливо виділяють семантичне моделювання вимог. Але дослідження робіт з цього напрямку [9, 10] визначає мінімальний рівень формалізації описів моделей вимог до ІС та методів формування і аналізу цих вимог. Загалом треба визнати, що проблема формального опису вимог до ІС як вихідних даних для синтезу архітектури системи, методів формування та аналізу вимог – зокрема на рівні знань, що видобуваються з описів вимог до ІС, поки що не має прийняттого для практичного використання рішення.

Формулювання цілі статті

За результатами досліджень автор у [11] запропонував формальну постановку та узагальнену модель задачі синтезу опису раціональної архітектури ІС. Ця задача розглядається як гра Постачальника із Споживачем ІТ-послуг (далі – Постачальник та Споживач), яка має такі особливості:

- за кількістю гравців – гра двох осіб;
- за кількістю стратегій – скінченна гра, кількість чистих стратегій якої визначається кількістю типових функціональних модулів ІС, що пропонуються Постачальником Споживачеві ІТ-послуг;

- за типом взаємовідношень гравців – безкоаліційна гра, до закінчення якої Постачальник та Споживач ІТ-послуг не можуть діяти сумісно (наприклад, не можуть затверджувати документ “Технічне завдання на розробку інформаційної системи”) [12];

- за характером вирашів – гра з нульовою сумою (загальний капітал Постачальника та Споживача ІТ-послуг перерозподіляється між ними залежно від результатів гри);

- за виглядом функції вирашу – біматрична гра, оскільки функції вирашів Постачальника та Споживача ІТ-послуг різні.

В загальному випадку функції вирашу Постачальника та Споживача визначаються метою їхніх дій під час створення ІС. Глобальною метою Постачальника слід вважати надання Споживачу такого набору ІТ-послуг, який найкраще відповідає комплексу вимог, які визначив Споживач, та особливостям бізнес-процесів Споживача з обмеженнями на вартість, час виконання, якість та зміст робіт з надання цього набору ІТ-послуг Споживачу. Глобальною метою Споживача вважається пошук та організація взаємодії з таким Постачальником, який надає набір ІТ-послуг, що найповніше відповідає комплексу вимог, визначених Споживачем та особливостями бізнес-процесів Споживача за обмежень на вартість, час виконання, якість та зміст робіт з надання цього набору ІТ-послуг, вибраних Постачальником [13, 14].

Зважаючи на це, основна ціль статті – розроблення моделей і методів синтезу опису раціональної архітектури ІС на основі формальних моделей вимог до ІС.

Метод синтезу варіантів опису архітектури інформаційної системи

Як основну формальну модель функціональної вимоги до ІС під час розв’язання задачі синтезу опису раціональної архітектури ІС пропонується використовувати модель представлення функціональної вимоги до ІС на рівні знань. Ця модель дає змогу описати функціональну вимогу до ІС фрагменти семантичної мережі, який в загальному випадку має такий вигляд [15]:

$$K_i^f = \{ \langle d_n^{ij}, \{ \langle d_{el_fr}^{ij}, d_{el_fr_t}^{ij} \rangle \} \rangle, \langle d_g^{ij}, \{ \langle d_{el_if}^{ij}, d_{el_if_t}^{ij} \rangle \} \rangle, \langle d_{fr_rel_n}^{ij}, \{ \langle d_{el_fr_rel}^{ij}, d_{el_fr_rel_t}^{ij} \rangle \} \rangle \} \} \quad (1)$$

де d_n^{ij} – опис найменування фрейму; $d_{el_fr}^{ij}$ – опис елемента фрейму; $d_{el_fr_t}^{ij}$ – опис типу елемента фрейму; d_g^{ij} – опис найменування інтерфейсу; $d_{el_if}^{ij}$ – опис елемента інтерфейсу; $d_{el_if_t}^{ij}$ – опис типу елемента інтерфейсу; $d_{fr_rel_n}^{ij}$ – опис назви зв’язку між інтерфейсами та/або фреймами; $d_{el_fr_rel}^{ij}$ – опис елемента зв’язку; $d_{el_fr_rel_t}^{ij}$ – опис типу елемента зв’язку.

Між елементами моделі (1) існують такі відношення належності [15]:

- відношення належності елементів фрейму конкретному фрейму;
- відношення належності елементів інтерфейсу конкретному інтерфейсу;
- відношення належності елементів зв’язку конкретному зв’язку.

Методи формувань представлень функціональних вимог до ІС на рівні знань, які відображають погляди Постачальника, Споживача та загальносистемний погляд на створювану ІС, також розглянув автор у [15]. В результаті використання цих методів під час виконання процесів визначення та аналізу вимог до створюваної ІС формується сукупність формальних загальносистемних представлень функціональних вимог до ІС на рівні знань $K_i^{f_{is}}$. Ці представлення є вихідними даними для розв’язання задачі синтезу опису раціональної архітектури ІС.

Але перед тим, як розпочати розв’язання задачі, необхідно проаналізувати сформульовані функціональні вимоги до ІС з метою виявлення вимог, що дублюють одна іншу. Таке дублювання призводить до необхідності витратити час та ресурси на дослідження та розроблення тієї самої ІТ-послуги у кількох різних варіантах. Наслідками цього є необхідність перепроєктування ІС з метою виправлення допущених під час її проектування помилок, а також зменшення ефекту від експлуатації ІС внаслідок дублювання ІТ-послуг. Водночас допустимий ступінь дублювання вимог може бути необхідний для бізнес-процесів Споживача. Тому в процесі розв’язання задачі синтезу

опису раціональної архітектури ІС необхідно враховувати всі варіанти опису архітектури створюваної системи, які були б не гірші від початкового варіанта опису за якимось критерієм чи критеріями.

Під початковим варіантом опису архітектури створюваної ІС розумітимемо опис семантичної мережі $Arch_{base}$, який формується на основі сукупності представлень окремих функціональних вимог до ІС K_i^{fis} так:

$$Arch_{base} = \bigcup_{i=c+1}^e K_i^{fis}, \quad (2)$$

де $c+1, e$ – ідентифікатори першої та останньої функціональної вимоги до ІС з сукупності сформульованих функціональних вимог до створюваної ІС.

Мережа (2) описуватиме варіант архітектури створюваної ІС, в якому кожна функціональна вимога буде описувати окрему ІТ-послугу (вважається, що всі функціональні вимоги сформульовані правильно). Для виявлення інших можливих варіантів описів архітектури створюваної ІС як сукупності ІТ-послуг слід враховувати такі особливості мереж (2), що будуть їх описувати:

- відповідно до множини патернів, яка визначає правила та особливості формального опису представлень функціональних вимог на рівні знань [14], варіанти описів архітектури створюваної ІС планується зберігати в спеціалізованому сховищі даних як сукупність транзакцій, кожна з яких описує окреме представлення K_i^{fis} ;
- представлення K_i^{fis} є категорійними даними (формалізовані структуровані описи фреймів, інтерфейсів та зв'язків), числове подання яких ускладнено;
- оцінка ступеня дублювання представлень K_i^{fis} повинна ґрунтуватися на оцінці кількості повторень описів елементів мереж (2) у різних представленнях;
- виявлення дублювання функціональних вимог повинно приводити до синтезу нових варіантів описів архітектури, в яких описи нових ІТ-послуг формують, групуючи початкові представлення K_i^{fis} .

Враховуючи зазначені особливості, задачу синтезу варіантів архітектури створюваної ІС треба розглядати як задачу кластеризації категорійних даних, алгоритм розв'язання якої ґрунтується на оптимізації глобального критерію. Як такий алгоритм пропонується використати алгоритм CLOPE [16]. Використання цього алгоритму дає змогу формалізувати задачу синтезу варіантів описів архітектури створюваної ІС.

Маємо семантичну мережу (2), яка складається з множини представлень $\{K_i^{fis}\}$. Кожне представлення K_i^{fis} є набором об'єктів ob (фреймів, інтерфейсів та зв'язків); опис представлення K_i^{fis} аналогічний (1). Множина описів ІТ-послуг IT_{acm} – це розбиття мережі (2), таке, що $\{IT_{acm_1}, \dots, IT_{acm_k}\} = Arch_{base}$ та $(IT_{acm_i} \neq \emptyset) \wedge (IT_{acm_i} \cap IT_{acm_j} = \emptyset)$ для $1 \leq i, j \leq k$, де k – кількість ІТ-послуг створюваної ІС. Кожна ІТ-послуга IT_{acm_j} має такі характеристики:

- $D|IT_{acm}|$ – множина унікальних об'єктів ob (фреймів, інтерфейсів та зв'язків);
- $Occ(ob, IT_{acm_j})$ – кількість входжень об'єкта ob в опис ІТ-послуги IT_{acm_j} ;
- $S(IT_{acm_j}) = \sum_{ob \in D(IT_{acm_j})} Occ(ob, IT_{acm_j})$;
- $W(IT_{acm_j}) = |D(IT_{acm_j})|$;
- $H(IT_{acm_j}) = S(IT_{acm_j}) / W(IT_{acm_j})$.

Тоді розрахунок функції вартості ІТ-послуги $Pr ofit(IT_{acm}, r)$ відповідно до положень алгоритму CLOPE виконаємо за формулою [16]:

$$\text{Profit}(IT_{acm}, r) = \frac{\sum_{j=1}^k \frac{S(IT_{acm_j})}{W(IT_{acm_j})^r} \times |IT_{acm_j}|}{\sum_{j=1}^k |IT_{acm_j}|} \quad (3)$$

де $|IT_{acm_j}|$ – кількість загальносистемних представлень функціональних вимог на рівні знань K_i^{fis} , що описують ІТ-послугу IT_{acm_j} .

Задачу синтезу варіантів описів архітектури створюваної ІС можна сформулювати так: для заданих $D(Arch_{base})$ та r знайти розбиття $IT_{acm} : \text{Profit}(IT_{acm}, r) \rightarrow \max$.

Але не можна забувати, що множина сформульованих вимог до ІС практично ніколи не описує створювану ІС повністю [14]. Тому постановку задачі синтезу варіантів описів архітектури створюваної ІС слід модифікувати так: для заданих $D(Arch_{base})$ та r знайти розбиття $IT_{acm} : \text{Profit}(IT_{acm}, r) \in [\text{Profit}_{\max} - e; \text{Profit}_{\max}]$, де Profit_{\max} – максимальне значення функції (3), e – величина допустимої похибки. У випадку, якщо величину e не вдається визначити за допомогою методів експертних оцінок, рекомендується вважати $e = 0,1 \times \text{Profit}_{\max}$.

Точне визначення значення коефіцієнта відштовхування r потребує спеціальних досліджень успішних ІТ-проектів створення ІС. Відповідно до результатів дослідження законів композиції елементів функціональної структури ІС, розглянутих в [17], рекомендується вибирати значення r , на підставі наведених в табл. 1 даних.

Таблиця 1

Залежність значення коефіцієнта відштовхування r від варіантів організації архітектури інформаційної системи, що створюється як сукупність ІТ-послуг

Варіант організації архітектури інформаційної системи	Значення коефіцієнта відштовхування r
Функціональна (підсистеми)	1,5
Модульна	2
Сервісна	≥ 3

Для розв'язання поставленої задачі алгоритм CLOPE необхідно модифікувати. Зокрема, треба змінити дії з ініціалізації розподілу набору транзакцій на кластери, оскільки, як вказано вище, початковий варіант опису архітектури створюваної ІС та, відповідно, розділення представлень K_i^{fis} між ІТ-послугами IT_{acm_j} вже визначено.

Результатом цієї модифікації є метод синтезу варіантів описів архітектури створюваної ІС, який складається з таких етапів.

Етап 1. Сформулювати початковий варіант опису архітектури створюваної ІС $Arch_{base}$.

Крок 1.1. Прийняти $n = |K_i^{fis}|$.

Крок 1.2. Сформулювати множину описів ІТ-послуг IT_{acm} , виконавши операції $IT_{acm_j} = K_i^{fis}$.

Крок 1.3. Для множини представлень $\{K_i^{fis}\}$ виконати операцію (2).

Етап 2. Встановити значення коефіцієнта відштовхування r та розрахувати значення функції $\text{Profit}(IT_{acm}, r)$ за виразом (3).

Етап 3. Здійснити синтез кращих та/або прийнятних варіантів описів архітектури створюваної ІС.

Крок 3.1. Прийняти $\text{Profit}_{\max} = \text{Profit}(IT_{acm}, r)$, $i = 1$, $j = 1$, $k = |IT_{acm}|$.

Крок 3.2. Вибрати представлення K_i^{fis} .

Крок 3.3. Якщо $K_i^{fis} \in IT_{acm_j}$, то вилучити K_i^{fis} з IT_{acm_j} . В іншому разі перейти до кроку 3.13.

Крок 3.4. Прийняти $z = 0$, $m = 0$.

Крок 3.5. Якщо $z = j$, то прийняти $z = z + 1$.

Крок 3.6. Якщо $z > k$, то перейти до кроку 3.12.

Крок 3.7. Включити K_i^{fis} до IT_{acm_z} .

Крок 3.8. Розрахувати значення $Profit(IT_{acm}, r)$ за виразом (3).

Крок 3.9. Якщо $Profit(IT_{acm}, r) > Profit_{max}$, то зафіксувати варіант опису архітектури створюваної ІС як множину IT_{acm} , скориговану з урахуванням виконання кроку 3.3 та кроку 3.7, прийняти $m = 1$ та перейти до кроку 3.12.

Крок 3.10. Якщо $Profit(IT_{acm}, r) \in [Profit_{max} - e; Profit_{max}]$, то зафіксувати варіант опису архітектури створюваної ІС як множину IT_{acm} , скориговану з врахуванням виконання кроку 3.3 та кроку 3.7.

Крок 3.11. Прийняти $z = z + 1$. Якщо $z \leq k$, то перейти до кроку 3.5.

Крок 3.12. Якщо $IT_{acm_j} = \emptyset$ та $m = 1$, то вилучити IT_{acm_j} з множини IT_{acm} і прийняти $k = k - 1$.

Крок 3.13. Прийняти $j = j + 1$. Якщо $j \leq k$, то перейти до кроку 3.3.

Крок 3.14. Прийняти $i = i + 1$. Якщо $i \leq n$, то перейти до кроку 3.2.

Крок 3.15. Якщо $m = 1$, то прийняти $i = 1$ та $j = 1$, після чого перейти до кроку 3.2. В протилежному випадку завершити виконання етапу методу.

Етап 4. Вилучити з розгляду всі зафіксовані на етапі 3 варіанти описів архітектури створюваної ІС IT_{acm} , для яких не виконується умова $Profit(IT_{acm}, r) \in [Profit_{max} - e; Profit_{max}]$. Завершити виконання методу.

Результатом виконання методу синтезу варіантів описів архітектури створюваної ІС є множина прийнятних варіантів описів архітектури, в яких ступінь дублювання описів предметної галузі в функціональних вимогах зведено до необхідного мінімуму. Кожен з цих варіантів описів являє собою множину ІТ-послуг IT_{acm} . Своєю чергою, кожна з ІТ-послуг IT_{acm_j} є результатом об'єднання, аналогічного (2), окремих представлень K_i^{fis} , що введено до опису цієї ІТ-послуги за результатами виконання етапу 3 розробленого методу.

Теоретико-ігрова модель синтезу опису раціональної архітектури інформаційної системи

Запропонований метод допомагає уточнити теоретико-ігрову модель синтезу опису раціональної архітектури створюваної ІС. Як наведено у [11], ця модель має вигляд

$$\Gamma_{IS} = \langle \{Pr, U\}, \{X_j\}_{j \in \{Pr, U\}}, \{f_j\}_{j \in \{Pr, U\}} \rangle, \quad (4)$$

де Γ_{IS} – позначення гри Постачальника та Споживача ІТ-послуг; $\{Pr, U\}$ – множина гравців, які беруть участь у грі Γ_{IS} ; $\{X_j\}_{j \in \{Pr, U\}}$ – множина стратегій гри Γ_{IS} ; $\{f_j\}_{j \in \{Pr, U\}}$ – множина функцій вигравів гри Γ_{IS} для Постачальника та Споживача.

У [11] стверджувалося, що функції виграшу Постачальника та Споживача визначаються в загальному випадку формалізованими описами їхніх глобальних цілей під час виконання процесів створення ІС. Використання загальносистемних представлень функціональних вимог до ІС на рівні знань K_i^{fis} дає змогу розглядати оператори вдоволення потреб $r^{Pr}(K_i^{fPr})$ та $r^U(K_i^{fU})$ функцій мети Постачальника та Споживача відповідно як функції, що порівнюють загальносистемне представлення i -ї функціональної вимоги на рівні знань та представлення цієї вимоги на рівні знань Постачальника та Споживача. В загальному випадку такі функції матимуть вигляд

$$r^{Pr}(K_i^{fPr}) = f(K_i^{fis}, K_i^{fPr}) = 1 - \frac{|K_i^{fis} \setminus K_i^{fPr}|}{|K_i^{fis}|}, \quad (5)$$

$$r^U(K_i^{fU}) = f(K_i^{fis}, K_i^{fU}) = 1 - \frac{|K_i^{fis} \setminus K_i^{fU}|}{|K_i^{fis}|}, \quad (6)$$

де $K_i^{f_{Pr}}$ – представлення i -ї функціональної вимоги до створюваної ІС на рівні знань з позиції Постачальника, формалізований опис якого аналогічний (1); $K_i^{f_U}$ – представлення i -ї функціональної вимоги до створюваної ІС на рівні знань з позиції Споживача, формалізований опис якого аналогічний (1).

Тоді функція виграшу Постачальника в грі (4) має вигляд

$$F_{Pr} = \sum_{i=c+1}^e 1 - \frac{|K_i^{f_{IS}} \setminus K_i^{f_{Pr}}|}{|K_i^{f_{IS}}|} \rightarrow \max. \quad (7)$$

Функція виграшу Споживача в грі (4) має вигляд

$$F_U = \sum_{i=c+1}^e 1 - \frac{|K_i^{f_{IS}} \setminus K_i^{f_U}|}{|K_i^{f_{IS}}|} \rightarrow \max. \quad (8)$$

Досягти виграшу Постачальник може в результаті виконання таких дій [14]:

- розроблення нових ІТ-послуг та ІТ-сервісів, які задовольняють вимоги, що ставлять Споживачі;

- адаптація розроблених раніше ІТ-послуг та ІТ-сервісів до особливостей вимог конкретних Споживачів.

Досягти виграшу Споживач може в результаті виконання таких дій [14]:

- формулювання вимог до ІС, ІТ-послуг та ІТ-сервісів, які враховують загальні та індивідуальні особливості бізнес-процесів об'єкта автоматизації;

- аналіз та вибір ІТ-послуг та ІТ-сервісів, які пропонуються Постачальниками та відповідають сформульованим вимогам.

Фізичний зміст функції виграшу Постачальника (7) полягає в максимізації повторного використання у ІТ-проекті створення ІС ІТ-сервісів, що реалізують сформульовані функціональні вимоги до цієї системи. В цьому випадку основним видом ІТ-проектів створення ІС з погляду Постачальника буде адаптація типових ІС, ІТ-послуг та ІТ-сервісів, що реалізують ці послуги, до особливостей процесів Споживача. Створення ІС “з нуля” Постачальник розглядатиме як частковий, найменш бажаний варіант ІТ-проекту, який, однак, після успішного завершення може істотно розширити можливості Постачальника.

З погляду Споживача основний ефект від виконання ІТ-проекту створення ІС на його вимоги i , відповідно, основний виграш, що описано функцією (8), досягається у випадку максимізації виконання вимог до ІС, що поставив Споживач і прийняв до реалізації Постачальник. Споживача цікавить передусім реалізація описів онтологій саме тієї сукупності термінів предметної галузі, які характерні для його процесів.

З урахуванням наведеного вище матриця виграшів Постачальника має такий вигляд:

$$Pr = \begin{pmatrix} 1 - \frac{|K_{1(c+1)}^{f_{IS}} \setminus K_{1(c+1)}^{f_{Pr}}|}{|K_{1(c+1)}^{f_{IS}}|} & \dots & 1 - \frac{|K_{1i}^{f_{IS}} \setminus K_{1i}^{f_{Pr}}|}{|K_{1i}^{f_{IS}}|} & \dots & 1 - \frac{|K_{1e}^{f_{IS}} \setminus K_{1e}^{f_{Pr}}|}{|K_{1e}^{f_{IS}}|} \\ \dots & \dots & \dots & \dots & \dots \\ 1 - \frac{|K_{j(c+1)}^{f_{IS}} \setminus K_{j(c+1)}^{f_{Pr}}|}{|K_{j(c+1)}^{f_{IS}}|} & \dots & 1 - \frac{|K_{ji}^{f_{IS}} \setminus K_{ji}^{f_{Pr}}|}{|K_{ji}^{f_{IS}}|} & \dots & 1 - \frac{|K_{je}^{f_{IS}} \setminus K_{je}^{f_{Pr}}|}{|K_{je}^{f_{IS}}|} \\ \dots & \dots & \dots & \dots & \dots \\ 1 - \frac{|K_{k(c+1)}^{f_{IS}} \setminus K_{k(c+1)}^{f_{Pr}}|}{|K_{k(c+1)}^{f_{IS}}|} & \dots & 1 - \frac{|K_{ki}^{f_{IS}} \setminus K_{ki}^{f_{Pr}}|}{|K_{ki}^{f_{IS}}|} & \dots & 1 - \frac{|K_{ke}^{f_{IS}} \setminus K_{ke}^{f_{Pr}}|}{|K_{ke}^{f_{IS}}|} \end{pmatrix}. \quad (9)$$

Матриця виграшів Споживача має такий вигляд:

$$U = \begin{pmatrix} 1 - \frac{|K_{1(c+1)}^{f_{IS}} \setminus K_{1(c+1)}^{f_U}|}{|K_{1(c+1)}^{f_{IS}}|} & \dots & 1 - \frac{|K_{1i}^{f_{IS}} \setminus K_{1i}^{f_U}|}{|K_{1i}^{f_{IS}}|} & \dots & 1 - \frac{|K_{1e}^{f_{IS}} \setminus K_{1e}^{f_U}|}{|K_{1e}^{f_{IS}}|} \\ \dots & \dots & \dots & \dots & \dots \\ 1 - \frac{|K_{j(c+1)}^{f_{IS}} \setminus K_{j(c+1)}^{f_U}|}{|K_{j(c+1)}^{f_{IS}}|} & \dots & 1 - \frac{|K_{ji}^{f_{IS}} \setminus K_{ji}^{f_U}|}{|K_{ji}^{f_{IS}}|} & \dots & 1 - \frac{|K_{je}^{f_{IS}} \setminus K_{je}^{f_U}|}{|K_{je}^{f_{IS}}|} \\ \dots & \dots & \dots & \dots & \dots \\ 1 - \frac{|K_{k(c+1)}^{f_{IS}} \setminus K_{k(c+1)}^{f_U}|}{|K_{k(c+1)}^{f_{IS}}|} & \dots & 1 - \frac{|K_{ki}^{f_{IS}} \setminus K_{ki}^{f_U}|}{|K_{ki}^{f_{IS}}|} & \dots & 1 - \frac{|K_{ke}^{f_{IS}} \setminus K_{ke}^{f_U}|}{|K_{ke}^{f_{IS}}|} \end{pmatrix}, \quad (10)$$

де k – кількість варіантів описів архітектури створюваної ІС, які сформовано в результаті виконання методу синтезу варіантів описів архітектури створюваної ІС.

Рядки матриці виграшів Постачальника (9) визначають значення функції виграшу Постачальника від реалізації сукупності ІТ-послуг j -го варіанта опису архітектури ІС. Стівці матриці виграшів Постачальника (9) визначають значення функції виграшу Постачальника від реалізації i -ї функціональної вимоги до ІС у різних варіантах описів архітектури цієї ІС. Рядки матриці виграшів Споживача (10) визначають значення функції виграшу Споживача від реалізації сукупності ІТ-послуг j -го варіанта опису архітектури ІС. Стівці матриці виграшів Споживача (10) визначають значення функції виграшу Споживача від реалізації i -ї функціональної вимоги до ІС у різних варіантах описів архітектури цієї ІС.

У випадку, якщо всі описи фреймів, інтерфейсів та зв'язків, що характеризують предметну галузь в i -й функціональній вимозі до ІС та використовуються для формування представлення $K_{ji}^{f_{IS}}$, будуть вибрані з бібліотеки описів раніше реалізованих вимог до ІТ-послуг ІС, виграш Постачальника від реалізації i -ї функціональної вимоги у j -му варіанті опису архітектури ІС дорівнюватиме 1. Якщо ж всі описи фреймів, інтерфейсів та зв'язків, що характеризують предметну галузь в i -й функціональній вимозі до ІС та використовуються для формування представлення $K_{ji}^{f_{IS}}$, є для Постачальника новими, то виграш Постачальника від реалізації i -ї функціональної вимоги у j -му варіанті опису архітектури ІС дорівнюватиме 0. Якщо ж новими для Постачальника є тільки частина описів цих фреймів, інтерфейсів чи зв'язків або нові описи фреймів, інтерфейсів чи зв'язків ґрунтуються на наявних у бібліотеці описах фреймів, інтерфейсів чи зв'язків, що характеризують абстрактні терміни предметної галузі, виграш Постачальника від реалізації i -ї функціональної вимоги у j -му варіанті опису архітектури ІС міститиметься в проміжку від 0 до 1.

У випадку, якщо всі описи фреймів, інтерфейсів та зв'язків, що характеризують предметну галузь в i -й функціональній вимозі до ІС та використовуються для формування представлення $K_{ji}^{f_{IS}}$, відобразатимуть тільки предметну галузь Споживача, виграш Споживача від реалізації i -ї функціональної вимоги у j -му варіанті опису архітектури ІС буде дорівнювати 1. Якщо всі описи фреймів, інтерфейсів та зв'язків, що характеризують предметну галузь в i -й функціональній вимозі до ІС та використовуються для формування представлення $K_{ji}^{f_{IS}}$, є для Споживача новими (тобто їх не вживає Споживач для опису своєї предметної галузі), то виграш Споживача від реалізації i -ї функціональної вимоги у j -му варіанті опису архітектури ІС дорівнюватиме 0. Якщо ж новою для Споживача є тільки частина описів цих фреймів, інтерфейсів чи зв'язків або нові описи фреймів чи інтерфейсів абстрагують сформульовані Споживачем фрейми чи інтерфейси, виграш Споживача від реалізації i -ї функціональної вимоги у j -му варіанті опису архітектури ІС буде в проміжку від 0 до 1.

Зважаючи на наведене вище, опис раціональної архітектури ІС як набору ІТ-послуг буде сформовано під час знаходження результату гри (4).

Для пошуку результату гри (4) пропонується використовувати метод пошуку рівноваг Неша для біматричної гри в чистих стратегіях. Цей метод призначений для пошуку ситуації рівноваги за Нешем – таких стратегій, ухиляючись від яких, ані Постачальник, ані Споживач не зможуть збільшити значення своїх функцій виграшу.

Для визначення ситуації рівноваги за Нешем під час синтезу опису раціональної архітектури ІС уточнимо описи стратегій гри (4). Як впливає з опису функцій виграшів (7) та (8), в загальному випадку ці стратегії визначаються множиною загальносистемних представлень функціональних вимог до ІС на рівні знань, які формують варіанти описів архітектури створюваної ІС. З урахуванням цього будь-яку стратегію $X_j^{(Pr,U)} \in \{X\}^{(Pr,U)}$ можна представити як семантичну мережу, яка являє собою результат виконання операції (2) над множиною загальносистемних представлень $\{K_{ji}^{f_{is}}\}_{i=(c+1),\dots,e}^{(Pr,U)}$.

Тоді ситуація рівноваги за Нешем у гри (4) являє собою один чи кілька варіантів описів архітектури створюваної ІС $\{K_{ji}^{f_{is}}\}_{i=(c+1),\dots,e}^{(Pr,U)}$, для яких виконується умова

$$\{K_{ji}^{f_{is}}\}_i^{(Pr,U)} \in \arg \max_{\{K_{ji}^{f_{is}}\}_i \in \{\{K_{i^s}^{f_{is}}\}_{i^s \in \{1,\dots,k\}}\}^{(Pr,U)}} f^{(Pr,U)}(\{K_{-ji}^{f_{is}}\}_i^{(Pr,U)} \parallel \{K_{ji}^{f_{is}}\}_i^{(Pr,U)}), \quad (11)$$

де $j = 1, \dots, k$, $i = (c+1), \dots, e$.

Згідно з цією умовою, метод пошуку рівноваг за Нешем у чистих стратегіях біматричної гри (4) буде складатися з таких етапів.

Етап 1. Сформувати матрицю виграшів Споживача (10) та матрицю виграшів Постачальника (9).

Етап 2. Позначити максимальні елементи в кожному стовпці матриці (10). Якщо в якомусь стовпці цієї матриці кілька максимальних елементів, то необхідно позначити всі такі елементи.

Етап 3. Позначити максимальні елементи в кожному рядку матриці (9). Якщо в якомусь рядку матриці кілька максимальних елементів, то необхідно відмітити всі такі елементи.

Етап 4. Знайти перетинання результатів етапу 2 та етапу 3 (тобто максимальних елементів, які розміщені на тих самих позиціях у кожній з матриць).

Етап 5. Якщо результат виконання етапу 4 не є пустою множиною, зафіксувати відповідні варіанти описів архітектури створюваної ІС як описи раціональної архітектури цієї системи $\{K_{ji}^{f_{is}}\}_{i=(c+1),\dots,e}^{(Pr,U)}$. В іншому разі визнати, що опису раціональної архітектури ІС на чистих стратегіях не існує. Завершити виконання методу.

Ситуація, коли множина елементів, які відзначено в матриці (10) (максимальних за стовпцями) та множина елементів, які відзначено в матриці (9) (максимальних за рядками) не перетинаються, означає, що рівноваг у чистих стратегіях для цього варіанта гри не існує. В цьому випадку пропонується розглянути множину змішаних стратегій, оскільки рівновага за Нешем існує для змішаного розширення будь-якої біматричної гри (4). Під час гри Постачальник та Споживач повинні використовувати з додатними вірогідностями тільки чисті стратегії, що дає змогу звести пошук рівноваги в змішаних стратегіях до розв'язання систем лінійних рівнянь та нерівностей. Для цього доцільно використовувати стандартні методи та алгоритми, зокрема алгоритм Лемке – Хаусона [18].

Недоліком використання методу пошуку рівноваг за Нешем у чистих стратегіях біматричної гри (4) слід вважати збільшення кількості ризиків прийняття рішень, пов'язаних зі створенням ІС. Однак для розглянутої задачі синтезу опису раціональної архітектури ІС як сукупності ІТ-послуг цей недолік не є критичним, оскільки процес синтезу архітектури передбачає координацію дій Постачальника та Споживача після завершення гри з метою узгодження та затвердження документів ІТ-проекту на створення ІС.

Використання опису раціональної архітектури інформаційної системи для визначення проектних обмежень

Для перевірки, чи дає змогу Постачальнику та Споживачеві отриманий у результаті гри (4) опис раціональної архітектури створюваної ІС досягти їхньої глобальної мети, слід перевірити

проектні обмеження IT-проекту створення ІС на основі синтезованого опису. Але треба враховувати, що основою розрахунків вартості, часу та якості виконання IT-проекту в переважній більшості методів аналізу проектних витрат є оцінка обсягу робіт зі створення ІС. Тому системи проектних обмежень, відповідно до розглянутої у [11] постановки задачі синтезу опису раціональної архітектури створюваної ІС, матимуть такий вигляд:

$$\left\{ \begin{array}{l} \sum_{i=c+1}^e a_i^{\text{Pr}} \text{pay}(PM(K_i^{f_{IS}}, K_i^{f_{Pr}})) \geq \text{pay}^*(PM(K_i^{f_{IS}}, K_i^{f_{Pr}})); \\ \sum_{i=c+1}^e b_i^{\text{Pr}} t(P(K_i^{f_{IS}}, K_i^{f_{Pr}})) \leq t^*(PM(K_i^{f_{IS}}, K_i^{f_{Pr}})); \\ \sum_{i=c+1}^e g_i^{\text{Pr}} q(PM(K_i^{f_{IS}}, K_i^{f_{Pr}})) \geq q^*(PM(K_i^{f_{IS}}, K_i^{f_{Pr}})); \end{array} \right. \quad (12)$$

$$\left\{ \begin{array}{l} \sum_{i=c+1}^e a_i^U \text{pay}(PM(K_i^{f_{IS}}, K_i^{f_U})) \leq \text{pay}^*(PM(K_i^{f_{IS}}, K_i^{f_U})); \\ \sum_{i=c+1}^e b_i^U t(PM(K_i^{f_{IS}}, K_i^{f_U})) \leq t^*(PM(K_i^{f_{IS}}, K_i^{f_U})); \\ \sum_{i=c+1}^e g_i^U q(PM(K_i^{f_{IS}}, K_i^{f_U})) \geq q^*(PM(K_i^{f_{IS}}, K_i^{f_U})); \end{array} \right. \quad (13)$$

де a_i^{Pr} – нормативний коефіцієнт вартості виконання вимоги tr_i , який враховує індивідуальні особливості Постачальника [11]; $\text{pay}(PM(K_i^{f_{IS}}, K_i^{f_{Pr}}))$ – вартість створення Постачальником ІС на основі опису її раціональної архітектури; $PM(K_i^{f_{IS}}, K_i^{f_{Pr}})$ – функція розрахунку обсягу робіт зі створення ІС з погляду Постачальника; $\text{pay}^*(PM(K_i^{f_{IS}}, K_i^{f_{Pr}}))$ – мінімально допустима для Постачальника величина вартості створення ІС на основі опису її раціональної архітектури; b_i^{Pr} – нормативний коефіцієнт тривалості виконання вимоги tr_i , який враховує індивідуальні особливості Постачальника [11]; $t(P(K_i^{f_{IS}}, K_i^{f_{Pr}}))$ – величина часу створення Постачальником ІС на основі опису її раціональної архітектури; $t^*(PM(K_i^{f_{IS}}, K_i^{f_{Pr}}))$ – максимально допустима для Постачальника величина часу створення ІС на основі опису її раціональної архітектури; g_i^{Pr} – нормативний коефіцієнт якості виконання вимоги tr_i , який враховує індивідуальні особливості Постачальника [11]; $q(PM(K_i^{f_{IS}}, K_i^{f_{Pr}}))$ – величина показника якості створення Постачальником ІС на основі опису її раціональної архітектури; $q^*(PM(K_i^{f_{IS}}, K_i^{f_{Pr}}))$ – максимально допустиме для Постачальника значення показника якості створення ІС на основі опису її раціональної архітектури; a_i^U – нормативний коефіцієнт вартості виконання вимоги tr_i , який враховує індивідуальні особливості Споживача [11]; $\text{pay}(PM(K_i^{f_{IS}}, K_i^{f_U}))$ – вартість створення ІС на основі опису її раціональної архітектури за вимогами Споживача; $PM(K_i^{f_{IS}}, K_i^{f_U})$ – функція розрахунку обсягу робіт зі створення ІС з погляду Споживача. $\text{pay}^*(PM(K_i^{f_{IS}}, K_i^{f_U}))$ – мінімально допустима для Споживача вартість створення ІС на основі опису її раціональної архітектури; b_i^U – нормативний коефіцієнт тривалості виконання вимоги tr_i , який враховує індивідуальні особливості Споживача [11]; $t(PM(K_i^{f_{IS}}, K_i^{f_U}))$ – величина часу створення ІС на основі опису її раціональної архітектури за вимогами Споживача; $t^*(PM(K_i^{f_{IS}}, K_i^{f_U}))$ – максимально допустима для Споживача величина часу створення ІС на основі опису її раціональної архітектури; g_i^U – нормативний коефіцієнт якості виконання вимоги tr_i , який враховує індивідуальні особливості Споживача [11]; $q(PM(K_i^{f_{IS}}, K_i^{f_U}))$ – значення показника якості створення ІС на основі опису її раціональної архітектури за вимогами

Споживача; $q^*(PM(K_i^{fis}, K_i^{fu}))$ – максимально допустиме для Споживача значення показника якості створення ІС на основі опису її раціональної архітектури.

Фізичний зміст систем обмежень (12) та (13) полягає в перевірці таких умов ефективного виконання ІТ-проекту створення ІС на основі опису її раціональної архітектури:

- вартість ІТ-проекту створення цієї ІС повинна бути не нижчою від величини, мінімально прийнятної для Постачальника, і не вищою від величини, максимально прийнятної для Споживача;
- час виконання ІТ-проекту створення цієї ІС повинен бути не більшим від максимально допустимих строків з погляду як Постачальника, так і Споживача;
- оцінка якості виконання ІТ-проекту створення цієї ІС повинна бути не меншою за мінімально прийнятні для Постачальника та Споживача значення.

Тоді праві частини систем (12) та (13) являтимуть собою умови, що пропонують Постачальник та Споживач під час формування та узгодження договору на створення ІС. Ліві частини систем (12) та (13) в цьому випадку являтимуть собою результати розрахунків умов ефективного виконання ІТ-проекту створення ІС на основі опису її раціональної архітектури.

Для визначення конкретного вигляду функцій $P(K_i^{fis}, K_i^{fv})$ та $P(K_i^{fis}, K_i^{fu})$ треба врахувати, що основний обсяг робіт зі створення ІТ-сервісів, що реалізують ІТ-послуги ІС на основі опису її раціональної архітектури, формують роботи зі створення інформаційного (ІЗ) та програмного (ПЗ) забезпечень цих сервісів. Тому використання для оцінювання обсягу таких робіт множин представлень функціональних вимог до ІС на рівні знань $\{K_i^{fis}\}$, $\{K_i^{fv}\}$ та $\{K_i^{fu}\}$ дає змогу досягнути таких переваг:

- будь-який з потоків даних ІТ-послуги можливо одночасно описати і як набір сутностей, що складаються з окремих атрибутів (за класичним методом функціональних точок), і як набір класів, що формують такі об'єкти, як екрани чи звіти (за методом об'єктних точок, який трактує кожен окремих потік даних як екран чи звіт);
- будь-який процес перетворення даних ІТ-послуги можливо у такий спосіб одночасно описати і як набір внутрішніх логічних файлів (ILF за класичним методом функціональних точок), і як набір 3GL чи аналогічних їм програмних модулів (за методом об'єктних точок).

Отже, використання множин представлень $\{K_i^{fis}\}$, $\{K_i^{fv}\}$ та $\{K_i^{fu}\}$ й опису раціональної архітектури ІС $\{K_{ji}^{fis}\}_{i=(c+1), \dots, e}^{Pr, U}$, який формується на їх основі, дає змогу використовувати їх як єдину основу для формування специфікацій на розроблення ІЗ та ПЗ створюваної ІС. Однак ця перевага є також і додатковою умовою, яка обмежує використання цих представлень і опису раціональної архітектури як основи для оцінювання обсягу робіт зі створення ІС. Розглянемо цю умову детальніше.

З погляду Споживача ІТ-послуга створюваної ІС повинна містити інформаційну та програмну реалізацію тільки тих концептів предметної галузі, фрейми та інтерфейси яких є у представленнях функціональних вимог до цієї ІТ-послуги на рівні знань. Однак такий погляд незручний для Постачальника, оскільки призводить до необхідності створення за вимогами різних Споживачів великої кількості унікальних ІС, що реалізують однакові ІТ-послуги в різних варіантах.

З погляду Постачальника заявлені Споживачем концепти та фрейми, що описують їх, є частковими випадками семантичної мережі фреймів, яку задано на ієрархіях абстрактних концептів (таксономіях), що деталізують максимально абстрактні концепти, використовуваних в різних ІТ-послугах типової ІС, до максимально конкретних концептів, використовуваних у конкретній ІТ-послуді цієї ІС. Така позиція великою мірою визначена концепцією патернів проектування, яка використовується для розроблення ПЗ типових ІС [19]. Що стосується розроблення ІЗ типової ІС, то така позиція ґрунтуватиметься на припущенні про можливість представлення схеми даних створюваної ІС як реалізації багатовимірної моделі даних [20], для якої реляційна модель даних буде частковим випадком.

Порівняльний аналіз позицій Постачальника та Споживача дає змогу виділити як основну компромісну умовну одиницю для оцінювання обсягу робіт зі створення ІС окрему гілку таксономії, що утворюють семантичну мережу фреймів опису раціональної архітектури створюваної ІС. Кожна така гілка буде складатися з окремих фреймів та інтерфейсів, які описують концепти різних рівнів абстракції предметної галузі в онтології функціональної вимоги до ІС та об'єднані відношеннями вкладення (або узагальнення (наслідування)). Під час створення ІТ-сервісів ІС такої гілці будуть відповідати [14, 21]:

- взаємопов'язана сукупність таблиць нормалізованих схем даних, схем даних типу “зірка” або “сніжинка” ІЗ відповідної ІТ-послуги ІС, яка забезпечує зберігання значень даних концептів предметної галузі, що формуються під час експлуатації ІТ-послуги;
- сукупність класів ПЗ відповідної ІТ-послуги ІС, яка реалізує бізнес-логіку, екранні форми та звіти, що необхідні для введення обробки та виведення значень даних концептів предметної галузі, що формуються під час експлуатації ІТ-послуги.

Тоді, за аналогією з функціональними та об'єктними точками як назвами умовних одиниць обсягу робіт зі створення ІС, пропонується іменувати кожен гілку таксономії фреймів, яка міститься в описі раціональної архітектури створюваної ІС, терміном “онтологічна точка”. Показник, який характеризує обсяг робіт зі створення ІС як кількість онтологічних точок, дасть змогу оцінити як обсяг робіт зі створення ІЗ ІС (як різновид програмування мовою SQL або іншою мовою аналогічного призначення), так і обсяг робіт зі створення ПЗ цієї ж ІС.

У загальному випадку будь-який зв'язок між фреймами або між фреймом та інтерфейсом можна описати виразом [22]

$$C_{f_a, f_b}^{f_a} \in C = \langle n_C, A_{f_a}^C, A_{f_b}^C, Pow_{f_a}^C, Pow_{f_b}^C, S_{f_a}^C, S_{f_b}^C \rangle; \quad (14)$$

де $C_{f_a, f_b}^{f_a}$ – елемент множини зв'язків C ; n_C – назва зв'язку $C_{f_a, f_b}^{f_a}$; $A_{f_a}^C$ – підмножина атрибутів фрейму f_a , який бере участь в утворенні зв'язку $C_{f_a, f_b}^{f_a}$, $A_{f_a}^C \subseteq (ns_{a1}, \dots, ns_{ak})$; $A_{f_b}^C$ – підмножина атрибутів фрейму f_b , який бере участь в утворенні зв'язку $C_{f_a, f_b}^{f_a}$, $A_{f_b}^C \subseteq (ns_{b1}, \dots, ns_{bk})$; $Pow_{f_a}^C$ – потужність зв'язку $C_{f_a, f_b}^{f_a}$ для фрейму f_a ; $Pow_{f_b}^C$ – потужність зв'язку $C_{f_a, f_b}^{f_a}$ для фрейму f_b ; $S_{f_a}^C$ – ступінь участі значень фрейму f_a в утворенні зв'язку $C_{f_a, f_b}^{f_a}$; $S_{f_b}^C$ – ступінь участі значень фрейму f_b в утворенні зв'язку $C_{f_a, f_b}^{f_a}$.

Потужність зв'язку для фрейму f_k визначається як кількість елементів множини значень цього фрейму O_{f_k} , які можуть брати участь в утворенні зв'язку.

Для виділення таксономії фреймів введемо формалізований опис зв'язку виду “генералізація”. Відповідно до виразу (14) цей зв'язок можна описати так [22]

$$C^{gen} = \langle n_C, A_{f_a}^C, A_{f_b}^C, 1, 1, S_{f_a}^C, S_{f_b}^C \rangle \quad (15)$$

у разі виконання для кожного дочірнього фрейму f_b умови [22]

$$\begin{cases} \exists f_a = \{n, [(ns_1, vs_1, ps_1), \dots, (ns_k, vs_k, ps_k)], \{if_1, \dots, if_n\}, \{mt_1, \dots, mt_z\}\}; \\ o_{f_b}^i = \langle v_{f_a}^{ij} \rangle \mathbf{U} \langle v_{f_b}^{ij} \rangle; \\ F(o_{f_a}^i) = F(o_{f_b}^i); \\ F(o_{f_b}^i) \neq F(o_{f_a}^i); \end{cases} \quad (16)$$

де $o_{f_b}^i$ – множина значень фрейму f_b ; $v_{f_a}^{ij}$ – значення j -го атрибута в i -му значенні фрейму f_a ; $v_{f_b}^{ij}$ – значення j -го атрибута в i -му значенні фрейму f_b ; F – сукупність операцій над значеннями фреймів f_a і f_b , причому операції сукупності F не обов'язково належать цим фреймам.

Тоді введений термін “онтологічна точка”, визначений як окрема гілка таксономії фреймів, що є в описі раціональної архітектури створюваної ІС як мережі фреймів, можна формалізовано описати виразом

$$\begin{aligned} OntPD = & \langle FR_{OntPD} = (fr_i, \dots, fr_k, \dots, fr_j), C^{gen}, G_{OntPD} = \\ & = \langle \langle fr_i, fr_{i+1}, C^{gen} \rangle, \dots, \langle fr_{k-1}, fr_k, C^{gen} \rangle, \langle fr_k, fr_{k+1}, C^{gen} \rangle, \dots, \langle fr_{j-1}, fr_j, C^{gen} \rangle \rangle \end{aligned} \quad (17)$$

у разі виконання умови

$$\begin{aligned} \forall fr_k \in FR_{OntPD} \exists \langle fr_{k-1}, fr_k, C^{gen} \rangle \mathbf{I} \langle fr_k, fr_{k+1}, C^{gen} \rangle \in G_{OntPD}, \\ i < k < j \end{aligned} \quad (18)$$

де $OntPD$ – формалізований опис онтологічної точки; FR_{OntPD} – підмножина фреймів, які утворюють онтологічну точку, $FR_{OntPD} \subseteq FR$; G_{OntPD} – множина відображень, які задають зв’язки генералізації між фреймами, що входять до підмножини FR_{OntPD} ; i – ідентифікатор кореневого фрейму онтологічної точки; j – ідентифікатор фрейму-листа онтологічної точки.

Для формування описів онтологічних точок на основі множини загальносистемних представлень $\{K_i^{fis}\}$, що утворюють опис раціональної архітектури створюваної ІС, пропонується метод, який складається з таких етапів.

Етап 1. Вибрати не розглянуте раніше представлення $K_i^{fis} \in \{K_i^{fis}\}$, яке утворює опис раціональної архітектури створюваної ІС.

Етап 2. Зафіксувати не розглянутий раніше фрейм fr^{ij} з множини фреймів представлення K_i^{fis} .

Етап 3. Виділити підмножину зв’язків $G_{OntPD}^p \in C^{gen}$, для яких зафіксований фрейм є дочірнім.

Етап 4. Якщо $G_{OntPD}^p = \emptyset$, то перейти до етапу 6.

Етап 5. Зафіксувати один зі зв’язків підмножини G_{OntPD}^p та фрейм fr^{ik} , який для зафіксованого зв’язку є батьківським. Потім перейти до етапу 3.

Етап 6. Виділити підмножину зв’язків $G_{OntPD}^{ch} \in C^{gen}$, для яких зафіксований фрейм є батьківським.

Етап 7. Якщо $G_{OntPD}^{ch} = \emptyset$, то перейти до етапу 9.

Етап 8. Зафіксувати один зі зв’язків підмножини G_{OntPD}^{ch} та фрейм fr^{ik} , який для зафіксованого зв’язку є дочірнім. Потім перейти до етапу 6.

Етап 9. Сформувати такий, що раніше не існував, опис онтологічної точки $OntPD_m, m=1,2,3,\dots$ із зафіксованих під час виконання етапів 2–8 фреймів і зв’язків та вилучити зафіксовані зв’язки із розглядання. Якщо для фрейму fr^{ij} виконується умова $G_{OntPD}^p = \emptyset \mathbf{I} G_{OntPD}^{ch} = \emptyset$, то перейти до етапу 10. Якщо для фрейму fr^{ij} виконується умова $G_{OntPD}^p = \emptyset$, то перейти до етапу 8. В іншому разі перейти до етапу 5.

Етап 10. Вилучити фрейм fr^{ij} із розглядання. Якщо множина фреймів представлення K_i^{fis} є пустою, то виключити представлення K_i^{fis} із розглядання та перейти до етапу 11. В іншому разі перейти до етапу 2.

Етап 11. Якщо виконується умова $\{K_i^{fis}\} = \emptyset$, то завершити виконання методу. В іншому разі перейти до етапу 1.

Розробляти метод оцінювання обсягу робіт з реалізації опису раціональної архітектури ІС з використанням онтологічних точок пропонується, модифікуючи відомий та апробований на практиці метод об’єктних точок [23]. Цей вибір зумовлено порівняльною простотою методу об’єктних точок і можливістю отримання з його допомогою достатньо точних оцінок обсягу робіт зі створення ПЗ ІС на ранніх стадіях відповідних ІТ-проектів. Для усунення неможливості оцінки методом об’єктних точок обсягу робіт зі створення ІЗ ІС пропонується розглядати цей вид робіт як

програмування мовою PL/SQL (або аналогічною мовою) бази даних зі структурою, яка визначена описом раціональної архітектури створюваної ІС.

Тоді для оцінювання обсягу робіт зі створення ПЗ ІС треба встановити вигляд функції оцінювання рівнів складності робіт зі створення екранів, звітів та модулів, необхідних для реалізації кожної ІТ-послуги створюваної ІС.

Використання процесного представлення функціональних вимог до ІС [14] дає змогу визначити термін “екран” методу об’єктних точок як сукупність структур даних, необхідних для виконання таких операцій:

- введення даних, які ініціюють виконання процесу;
- виведення результатів виконання процесу на екран дисплея;
- вибір із бази даних тих даних, що використовуються в процесі;
- зберігання даних, що надходять з процесу.

Термін “звіт” методу об’єктних точок у цьому випадку можна визначити як сукупність структур даних, що необхідні для виконання операції виведення результатів виконання процесу у вигляді електронного чи паперового документа.

Термін “3GL-модуль” у цьому випадку треба замінити терміном “модуль”, який позначає сукупність структур даних, необхідних для забезпечення нормального функціонування усіх екранів та звітів, запланованих до реалізації в межах ІТ-послуги.

Оскільки переважна більшість архітектур сучасних ІС не передбачає зберігання даних на комп’ютерах операторів та користувачів ІС, модифікація методу об’єктних точок, що пропонується, передбачає тільки оперування сумарними показниками кількості використовуваних таблиць (без їх класифікації на клієнтські та серверні). Водночас зберігання значень та фізичного змісту базових критеріїв дає змогу використовувати перевірені на практиці таблиці показників методу об’єктних точок [14].

Формальний опис екранів, звітів та модулів кожної конкретної ІТ-послуги визначатиметься засобом публікації функціональної вимоги до цієї ІТ-послуги. У випадку, якщо вимога подається у вигляді DFD або таких варіантів публікації, що зводяться до неї, звітами будуть тільки вихідні потоки даних, які достатньо легко виділити під час формування публікації вимоги. Усі інші потоки даних будуть являти собою екрани. Тоді рівень складності робіт зі створення екрана $Screen_{ij}$ або звіту $Report_{ik}$ можна буде оцінити на основі кількості онтологічних точок, які описують цей екран або звіт, та кількості фреймів у цих онтологічних точках:

$$Screen_{ij} = f(|\{OntPD_{ijm}\}|, |\{FR_{OntPD_{ijm}}\}|); \quad (19)$$

$$Report_{ik} = f(|\{OntPD_{ikm}\}|, |\{FR_{OntPD_{ikm}}\}|); \quad (20)$$

де $FR_{OntPD_{ijm}}$ – множина фреймів m -ї онтологічної точки, яка характеризує j -й екран i -ї ІТ-послуги;

$FR_{OntPD_{ikm}}$ – множина фреймів m -ї онтологічної точки, яка характеризує k -й звіт i -ї ІТ-послуги;

$j = 1, \dots, n$ – ідентифікатор екрана $Screen_{ij}$; $k = 1, \dots, p$ – ідентифікатор звіту $Report_{ik}$.

Значення виразу (19) наведено в табл. 2, а значення виразу (20) в табл. 3.

Таблиця 2

Значення функції оцінювання рівня складності робіт зі створення екрана програмного модуля ІТ-послуги

	$ \{FR_{OntPD_{ijm}}\} < 4$	$4 \leq \{FR_{OntPD_{ijm}}\} < 8$	$8 \leq \{FR_{OntPD_{ijm}}\} $
$ \{OntPD_{ijm}\} < 3$	$f = 1$	$f = 1$	$f = 2$
$3 \leq \{OntPD_{ijm}\} \leq 7$	$f = 1$	$f = 2$	$f = 3$
$8 \leq \{OntPD_{ijm}\} $	$f = 2$	$f = 3$	$f = 3$

**Значення функції оцінювання рівня складності робіт
зі створення звіту програмного модуля ІТ-послуги**

	$ \{FR_{OntPD_{ikm}}\} < 4$	$4 \leq \{FR_{OntPD_{ikm}}\} < 8$	$8 \leq \{FR_{OntPD_{ikm}}\} $
$ \{OntPD_{ikm}\} < 3$	$f = 2$	$f = 2$	$f = 5$
$3 \leq \{OntPD_{ikm}\} \leq 7$	$f = 2$	$f = 5$	$f = 8$
$8 \leq \{OntPD_{ikm}\} $	$f = 5$	$f = 8$	$f = 8$

Для підвищення точності оцінювання обсягу робіт використаємо підхід, прийнятий в класичному методі функціональних точок. Згідно з цим підходом обсяг робіт з реалізації програмного модуля обробки даних оцінюється на основі сукупності структур даних, що обробляються в цьому модулі. Тому для оцінювання рівня складності робіт зі створення модуля $Module_i$ і фрагмента бази даних ІС, з яким цей модуль взаємодіє під час експлуатації ІС, можна використати кількість фреймів онтологічних точок, які характеризують i -ту ІТ-послугу, та кількість атрибутів цих фреймів:

$$Module_i = f(|\{FR_{OntPD_m}\}|, |\{ATTR_FR_{OntPD_m}\}|). \quad (21)$$

Значення функції (21) наведено в табл. 4.

Таблиця 4

**Значення функції оцінювання рівня складності робіт
зі створення програмного модуля ІТ-послуги**

	$ \{FR_{OntPD_m}\} = 1$	$2 \leq \{FR_{OntPD_m}\} \leq 5$	$6 \leq \{FR_{OntPD_m}\} $
$ \{ATTR_FR_{OntPD_m}\} \leq 19$	$f = 7$	$f = 7$	$f = 10$
$20 \leq \{ATTR_FR_{OntPD_m}\} \leq 50$	$f = 7$	$f = 10$	$f = 15$
$50 < \{ATTR_FR_{OntPD_m}\} $	$f = 10$	$f = 15$	$f = 15$

У випадку, якщо вимога публікується у вигляді діаграми варіантів використання мови UML або таких варіантів, що зводяться до неї, кожна окрема онтологічна точка відповідної ІТ-послуги потребуватиме окремих екранів для введення і виведення даних. Функція (19) для таких екранів буде мати вигляд

$$Screen_{ij} = f(1, |\{FR_{OntPD_{jm}}\}|), \quad (22)$$

а її значення визначатимуться верхнім рядком табл. 2. Крім того, для обміну даними між модулем та базою даних необхідний екран, який характеризується кількістю усіх онтологічних точок цієї ІТ-послуги і оцінюється функцією (19). Наявність звітів у цьому випадку повинна бути особливо відзначена в публікації вимоги.

Тоді за аналогією з методом об'єктних точок обсяг робіт зі створення ІС на основі опису раціональної архітектури ІС можна оцінити як обсяг робіт з реалізації нових онтологічних точок так:

$$PM = \sum_{i=(c+1)}^e PM_i; PM_i = \frac{NOntP_i}{PROD}, \quad (23)$$

де $PROD$ – коефіцієнт продуктивності виконавців ІТ-проекту створення ІС, значення якого наведено в [23]; $NOntP_i$ – оцінка нових онтологічних точок, яка визначається за формулою

$$N_{OntP_i} = \frac{N_{OntP_i}(100 - \% reuse)}{100}; \quad (24)$$

$OntP_i$ – оцінка онтологічних точок ІТ-послуги, яка визначається за формулою

$$OntP_i = \sum_{j=1}^n Screen_{ij} + \sum_{k=1}^p Report_{ik} + Module_i; \quad (25)$$

$\% reuse$ – процент повторного використання фреймів онтологічних точок, який визначається за формулою

$$\% reuse = 100 * (1 - (|FR_{OntPD_i}| - |FR_{K_i^{f_{is}}}|)); \quad (26)$$

$OntPD_i$ – множина описів онтологічних точок, яка формується на основі загальносистемного представлення $K_i^{f_{is}}$ розглянутим вище методом; FR_{OntPD_i} – множина описів фреймів з множини описів онтологічних точок $OntPD_i$; $FR_{K_i^{f_{is}}}$ – множина описів фреймів представлення $K_i^{f_{is}}$.

Для розрахунку часу виконання ІТ-проекту створення ІС на основі опису її раціональної архітектури можна в цьому випадку використовувати формулу розрахунку витрат часу моделі СОСОМО II, яка має такий вигляд [23]:

$$TDEV = \left[3 \times \sqrt{PM^{(0,33+0,2 \times 0,01 \times B)}} \right] \times \frac{SCED\%}{100}; \quad (27)$$

де $TDEV$ – час виконання ІТ-проекту створення ІС, місяців; B – оцінка сукупності драйверів масштабу та економічності ІТ-проекту створення ІС, яка визначається за формулою

$$B = PREC + FLEX + RESL + TEAM + PMAT; \quad (28)$$

де $PREC$ – показник безпрецедентності створюваної ІС; $FLEX$ – показник гнучкості створюваної ІС; $RESL$ – показник переваг архітектурних чи ризикованих рішень; $TEAM$ – показник згуртованості команди виконавців ІТ-проекту створення ІС; $PMAT$ – показник зрілості процесів розробки ІС; $SCED\%$ – показник стиснення розкладу за часом.

Значення драйверів масштабу та економічності ІТ-проекту створення ІС та показника $SCED\%$ наведено у [23]. Тоді потребу в персоналі для ІТ-проекту створення ІС на основі опису її раціональної архітектури можна оцінити за формулою

$$SS = \frac{PM}{TDEV}. \quad (29)$$

Результат розрахунку оцінки потреби в персоналі за формулою (29) є основою для подальшого визначення вартості ІТ-проекту створення ІС на підставі опису її раціональної архітектури з урахуванням нормативних величин оплати праці виконавців робіт ІТ-проекту створення ІС.

Висновки та перспективи подальших наукових розвідок

Застосування теоретико-ігрової моделі для розв'язання задачі синтезу опису архітектури ІС може надати значні переваги над іншими підходами до розв'язання цієї задачі завдяки використанню змішаних стратегій, за якими Постачальник пропонує Споживачеві ІТ-послуг нетипові функціональні модулі створюваної ІС, а окремі ІТ-послуги, з яких складаються функціональні модулі системи, що будуть повністю адаптовані до вимог Споживача. Визначений у такий спосіб набір ІТ-послуг створюваної ІС характеризуватиметься такими особливостями:

- максимально відповідає вимогам, що поставив Споживач ІТ-послуг;
- приносить Постачальнику ІТ-послуг бажаний прибуток від повторного використання ІТ-послуг та відповідних ІТ-сервісів;
- забезпечує реалізацію безперервних технологічних ланцюжків збору, зберігання та обробки даних у межах єдиної цілісної ІС;
- гарантує відсутність у створюваній ІС ІТ-послуг, які не є обов'язковими і не відповідають поставленим вимогам.

Розроблений метод синтезу варіантів описів архітектури створюваної ІС, на відміну від стандартного алгоритму CLOPE, дає змогу виділити для подальшого аналізу варіанти опису архітектури створюваної ІС, які лише незначно гірші за оптимальний, що дає змогу розширити область пошуку компромісного варіанта архітектури створюваної ІС, який задовольняє як Постачальника, так і Споживача. Підхід до оцінювання обсягу робіт на ранніх стадіях ІТ-проекту створення ІС на основі опису її раціональної архітектури дає змогу підвищити точність оцінювання обсягу робіт, витрат часу та потреби в персоналі під час формування та аналізу вимог до цієї ІС, а також під час синтезу архітектури створюваної ІС. Запропоновані в статті методи та моделі синтезу опису раціональної архітектури створюваної ІС слід розглядати як основу сучасної інтелектуальної інформаційної технології прискороженого розроблення ІС, яка істотно зменшує тривалість створення розробки та значно підвищити ефективність та прибутковість діяльності ІТ-компаній за рахунок збільшення частки повторно використовуваних ІТ-сервісів.

1. ГОСТ ИСО/МЭК 15288–2005. Системная инженерия. Процессы жизненного цикла систем. Введ. 01–01–2007. – М.: Стандартинформ, 2006. 57 с. 2. Сайт “ISO/IEC/IEEE 42010 Website”. – <http://www.iso-architecture.org/ieee-1471/index.html>. 3. Захман Дж. Бизнес и информационные технологии: лекция / Дж. Захман. – <http://www.intuit.ru/department/itmngt/entarc/1/>. 4. Информационная система // Сайт “Глоссарий.ru”. – [http://www.glossary.ru/cgi-bin/gl_sch2.cgi?RIt\(uwsg.outt:!!xoxyls:](http://www.glossary.ru/cgi-bin/gl_sch2.cgi?RIt(uwsg.outt:!!xoxyls:). 5. Соммервил И. Инженерия программного обеспечения. – М. : Вильямс, 2002. – 624 с. 6. Foegen, M. Die Rolle der Architektur in der Anwendungsentwicklung / M. Foegen, J. Batterfeld // *Informatik-Spektrum*. Springer. – 2001. – № 5(24). – P. 290–301. 7. Архитектуры, модели и технологии программного обеспечения информационно-управляющих систем: монография / Ткачук Н. В., Шеховцов В. А., Куценко Д. В., Сокол В. Е.; под ред. М. Д. Годлевского. НТУ “ХПИ”, Харьков, 2005. – 546 с. 8. Yue T. A systematic review of transformation approaches between user requirements and analysis models / T. Yue, L.C. Briand, Y. Labiche // *Requirements Engineering*. 2010. Vol. 16. № 2. – P. 75–99. 9. Тюрганов А. Г. Особенности формализации предметной области корпоративных информационных систем / А. Г. Тюрганов // *Вестник Уфимского государственного авиационного технического университета*. – 2007. – Т. 9, № 5. – С. 72–76. 10. Cleland-Huang J. Mining Domain Knowledge [Requirements] // *IEEE Softw.* – 2015. – Vol. 32. – № 3. – P. 16–19. 11. Євланов М. В. Задача синтезу опису архітектури інформаційної системи / М. В. Євланов // *Вісник Нац. ун-ту “Львівська політехніка”*. – 2014. – № 805. – С. 114–123. 12. Chase R. B. Production and operations management: manufacturing and services / R. B. Chase, N. J. Aquilano, R. F. Jacobs. Irwin. – Boston, 1998. – 889 p. 13. Евланов М. В. Глобальные цели поставщика и потребителя ИТ-услуг / М. В. Евланов, О. Е. Неумывакина, А. Ю. Карамышева // *Восточно-европейский журнал передовых технологий*. – 2012. – № 5/2 (59). – С. 12–17. 14. Левыкин В. М. Паттерны проектирования требований к информационным системам: моделирование и применение / В. М. Левыкин, М. В. Евланов, М. А. Керносов. ООО “Компанія СМІТ”. – Харьков, 2014. – 320 с. 15. Евланов М. В. Разработка методов формирования представления сформулированного требования к информационной системе на уровне знаний // *Восточно-европейский журнал передовых технологий*. – 2015. – № 4/3 (76). – С. 4–11. 16. Yang Y. CLOPE: A fast and Effective Clustering Algorithm for Transaction Data / Y. Yang, H. Guan, J. You // *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM. – New York, 2002. – P. 682–687. 17. Левыкин В. М. Метамодел функциональной структуры информационной системы / В. М. Левыкин, М. В. Евланов // *Нові технології: наук. вісник Кременчуцького університету економіки, інформаційних технологій і управління*. – 2006. – № 1(11). – С. 67–72. 18. Матвеев В. А. Конечные бескоалиционные игры и равновесия: учебное пособие / В. А. Матвеев // ПГПИ им. С. М. Кирова, Псков, 2005. – 176 с. 19. Гамма Э. Приемы объектно-ориентированного проектирования. Паттерны проектирования / Э. Гамма, Р. Хелм, Р. Джонсон, Д. Влассидес. СПб.: Питер, 2010. – 366 с. 20. Барсегян А. А. Технологии анализа данных: Data Mining, Visual Mining, Text Mining, OLAP / А. А. Барсегян, М. С. Куприянов, В.В. Степаненко, И. И. Холод. – СПб.: БХВ-Санкт-Петербург. – 2008. – 384 с. 21. Евланов М. В. Унификация методов оценивания затрат на создание современных

информационных систем [Текст] / М. В. Евланов, Е. И. Соловьева // Вісник Кременчуцького національного університету ім. Михайла Остроградського. – 2014. – Вип. 5/2014 (88). – С. 62–67.
22. Левыкин В. М. Параллельное проектирование информационного и программного комплексов информационной системы / В. М. Левыкин, М. В. Евланов, В. С. Сугробов // Радиотехника. – 2006. – Вып. 146. – С. 89–98. 23. COCOMO II Model Definition Manual // Сайт “Center for Systems and Software Engineering”. – ftp://ftp.usc.edu/pub/soft_engineering/COCOMOII/cocomo99.0/modelman.pdf.
24. Functional Point Counting Practices Manual. Release 4.1.1. IFPLUG, Troy, 2001.

УДК 519.7

Р. М. Камінський, Г. В. Бігун

Національний університет “Львівська політехніка”,
кафедра інформаційних систем та мереж

ПОБУДОВА РЕКУРЕНТНИХ ДІАГРАМ КОРОТКИХ ЧАСОВИХ РЯДІВ ЗАСОБАМИ MS EXCEL

© Камінський Р. М., Бігун Г. В., 2015

Розкрито поняття рекурентної діаграми як матриці близькості між зображальними точками фазового портрета на фазовій площині для коротких часових рядів. Наведено формули побудови рекурентних діаграм та результати застосування запропонованого методу побудови рекурентних діаграм у середовищі табличного процесора MS Excel. Як ілюстративний матеріал використано три типи сигналів: гармонічний, меандр і трикутний та ці самі сигнали з накладеним на них рівномірним шумом. Результати мають істотне практичне значення, оскільки метод рекурентних діаграм ще не набув поширення в Україні, проте легко реалізується в табличному середовищі.

Ключові слова: нелінійна динаміка, часові послідовності, рекурентні діаграми, матриця близькостей, MS Excel.

The recurrence plot concept was disclosed as a proximity matrix between phase space depicting points on phase area for time series. Recurrence plots construction equations and application results of suggested method are given in the MS Excel. Three types of signals: harmonic, meanders, triangular are used as a graphical material and even noise distribution is imposed on this signals. Results have significant practical value, because recurrence plots method is not widely spread in Ukraine, however it is easy to implement in spreadsheet.

Key words: nonlinear dynamics, time series, recurrence plots, proximity matrix, MS Excel.

Вступ

У різноманітних сферах людської діяльності є системи, які характеризуються своєю поведінкою з погляду її складності, динамічності, особливостей процесів, що відбуваються в цій системі. Такі системи опосередковано чи безпосередньо впливають на діяльність людей, навколишнє середовище, а тому необхідно і важливо вивчати, досліджувати і моделювати їх динаміку та передбачати їх поведінку з плином часу. Складні динамічні системи переважно характеризуються нерегулярною динамікою поведінки, проявами якої є як випадкові, так і детерміновані хаотичні процеси. Спостереження за такими системами та їх експериментальні дослідження подаються часовими рядами – дискретною послідовністю випадкових величин, що є