

6. Барсегян А.А. Методы и модели анализа данных OLAP и DataMining [Текст] / А. А Барсегян, М.С. Куприянов, В. В. Степаненко, И. И.Холод. – СПб: БХВ-Петербург. – 2008. 7. Енюков И.С. Факторный, дискриминантный и кластерный анализ [Текст] / Енюков И.С. – М.: Финансы и статистика, 1989. 8. Кулаишев А. П. Методы и средства комплексного анализа данных [Текст] / Кулаишев А.П. – М: ИНФРА-М. 2006. 9. Мандель И.Д. Кластерный анализ [Текст] /Мандель И.Д. – М.: Финансы и статистика. – 1988. 10. Пакет статистической обработки. Кластерный анализ при большом количестве наблюдений [Электронный ресурс]. – Режим доступа: http://www.sati.archaeology.nsc.ru/stat/methods_info.php. 11. Уиллиамс У. Т., Ланс Д. Н. Методы иерархической классификации. Статистические методы для ЭВМ [Текст] // под ред. М. Б. Малютов. – М.: Наука, 1986. 12. Чубукова И. А. Data Mining: учебное пособие [Текст] / И. А. Чубукова. — М.: Интернет-университет информационных технологий : БИНОМ: Лаборатория знаний, 2006.

УДК 004.896

М.В. Євланов

Харківський національний університет радіоелектроніки

ПАТЕРНИ ПРОЕКТУВАННЯ ВИМОГ ДО ІНФОРМАЦІЙНОЇ СИСТЕМИ

© Євланов М.В., 2014

Запропоновано визначення терміна “патерн проектування вимог до інформаційної системи”. Запропоновано теоретико-категорну модель такого патерна, визначено місце патернів у інформаційній технології формування та аналізу вимог до інформаційної системи.

Ключові слова: інформаційна система, вимога, патерн, теоретико-категорна модель.

This paper is devoted to the definition of "pattern design requirements for an information system". A theoretic-categorical model of the pattern is proposed, the place of patterns in the information technology development and analysis of requirements for information systems is defined.

Key words: information system, requirement, pattern, theoretic-categorical model.

Вступ

Створення сучасних інформаційних систем (ІС) являє собою сукупність процесів, описаних у міжнародних стандартах ISO/IEC 15288:2002 та ISO/IEC 12207:2008. Ці стандарти визначають суть та основні особливості процесів проектування архітектури ІС та систем, що її забезпечують [1, 2]. Тепер ці стандарти визнані багатьма країнами світу, серед яких й Україна.

Відповідно до положень стандартів ISO/IEC 15288:2002 та ISO/IEC 12207:2008, основним ресурсом процесів проектування архітектури ІС є вимоги, що ставляться до цих систем. Тому розв’язання задачі синтезу архітектури ІС як сукупності функцій системи значною мірою залежить від того, наскільки ефективно та якісно сформульовано вимоги до цієї системи, а витрати часу та ресурсів на створення цієї ІС великою мірою визначаються результатами виконання процесів визначення вимог правласників системи та аналізу вимог.

Одним з напрямів зменшення витрат ресурсів на виконання процесів визначення вимог правласників системи та аналізу вимог є повторне використання вимог, поставлених до розроблених раніше ІС, під час створення нової ІС або модернізації чи розвитку ІС, що експлуатуються. Цей підхід не тільки може привести до скорочення часу виконання ІТ-проекту створення ІС, але й дозволить зменшити витрати ресурсів за рахунок використання основних компонентів видів

забезпечень, що реалізують повторно використовувані вимоги. Тому розробка моделей і технологій, які дозволять проектувальникам ІС визначати можливість повторного використання вимог до ІС та автоматизовано розв'язувати задачу синтезу архітектури ІС з врахуванням повторного використання вимог, є актуальними як з теоретичної, так і з практичної позицій.

Аналіз сучасного представлення підходу до повторного використання елементів інформаційної системи

Нині найвживанишим підходом до організації повторного використання елементів ІС є ідея патернів проектування як шаблонів, що визначають розв'язання окремих задач, які часто повторюються у різних проектах програмних систем. Найповніше цю ідею реалізовано у об'єктно-орієнтованому програмуванні 1990-х років. Першим кроком у розвитку цієї ідеї стала праця [3], перше видання якої вийшло у 1995 р. Її автори підходять до розгляду патернів проектування як результатів документування досвіду розробників об'єктно-орієнтованих програм і вважають, що патерни спрощують повторне використання вдалих проектних та архітектурних рішень. Для впорядкування описів патернів у [3] введено класифікацію, яка розділяє патерни за їх призначенням (породжуючі, структурні, поведінки) та рівнем використання (клас, об'єкт). Породжуючі патерни пов'язані зі створенням екземплярів об'єктів; всі вони визначають засоби логічної ізоляції клієнта від створюваних об'єктів. Структурні патерни об'єднують класи чи об'єкти в більші структури. Патерни поведінки стосуються до взаємодії та розподілення обов'язків між класами та об'єктами. Патерни класів описують визначення відношень між класами за допомогою наслідування. Відношення між патернами класів визначаються на рівні компіляції. Патерни об'єктів описують відношення між об'єктами, передусім ті, що стосуються композиції. Відношення між патернами об'єктів зазвичай визначаються на стадії виконання, тому мають більшу динамічність та гнучкість [4]. Автори праці [3] також пропонували можливість використання патернів проектування для створення спеціалізованих CASE-засобів.

Але в кінці 1990-х – 2000-х рр. почали виявлятися основні обмеження використання патернів в процесі створення програмних систем. Ці обмеження пов'язані насамперед зі складнощами, які можуть виникнути під час реалізації патернів у конкретному проекті програмної системи. Можна стверджувати, що критерій простоти використання, оснований на порівнянні складності розроблюваного програмного коду, є основним критерієм застосування патернів проектування. Але прийняття рішення про доцільність використання патернів на базі цього критерію є нині неформалізованою процедурою. Крім того, на прийняття такого рішення можуть вплинути результати аналізу можливих наслідків використання патернів.

Результати досвіду використання патернів проектування, що розглянуті у [4], дають підставу зробити такі висновки:

1. Патерни вигідні у випадку, якщо створювана програмна система повинна бути гнучкою та передбачає постійні зміни своєї структури та змісту (додавання функцій, зміна функціональної структури тощо).
2. Патерни вигідні у разі проведення рефакторингу чи інших процедур, що спрямовані на вдосконалення організації програмної системи чи її окремих елементів [5].
3. Патерни не вигідні, якщо витрати ресурсів на створення і супроводження програмної системи, що зумовлені складністю цієї системи, перевищують ефект від гнучкості цієї системи.

З урахуванням цих висновків можна описати сучасні погляди на патерни як на природне узагальнення результатів процесів проектування програмних систем. При цьому використання патернів проектування повинно бути не самоціллю, а необхідністю, що викликана особливостями конкретних проектів (чи портфелю проектів) [4]. Така позиція дозволяє, своєю чергою, припустити, що ідея використання патернів у процесах проектування програмних систем є прагненням формалізувати та використовувати з прикладною метою знання розробників таких систем про методи, засоби та стилі програмування тих чи інших проблем практичного характеру. Це припущення підтверджується прикладом GRASP-патернів (General Responsibility Assignment Software Patterns – загальні зразки розподілення обов'язків) [6].

Формулювання цілі статті

Розглянута концепція патернів проектування орієнтована виключно на створення програмного забезпечення ІС. Використання цієї концепції в інших процесах життєвого циклу системи (за стандартом ISO/IEC 15288:2002) неможливе без серйозних змін. Особливо це стосується процесів визначення вимог правласників системи, аналізу вимог та проектування архітектури системи.

Водночас зазначимо, що спроби прямого повторного використання вимог до раніше створених ІС під час виконання робіт з розробки нової системи майже ніколи не давали ефективного результату. Приклад та основні причини цих невдач розглянуто, наприклад, у звіті [7].

Тому основною метою статті вважаємо розроблення основних визначень та моделей, які дозволять формалізовано описати підхід до використання патернів як шаблонів, що дають змогу використовувати знання, накопичені у попередніх проектах створення ІС, під час виконання ІТ-проекту створення нової системи.

Виклад основного матеріалу статті

Для досягнення поставленої цілі пропонується уточнити термін “вимога до ІС”. Оскільки самостійного визначення цього терміна поки що не існує, як основу для уточнення використаємо визначення терміна “вимога”, задеклароване стандартом IEEE 610.12-1990. Це визначення є найузагальненішим і використовується практично у всіх галузях комп’ютерних наук. На основі цього визначення термін “вимога до ІС” можливо визначити так [8]:

- умова чи можливість, що необхідні Споживачу ІТ-послуг для вирішення проблеми або досягнення мети;
- умова чи можливість, яку повинна мати ІС чи компонент ІС (ІТ-послуга, ІТ-сервіс) з погляду Споживача ІТ-послуг та які відповідають домовленості, стандарту, специфікації чи іншому офіційному документу;
- документоване представлення умови чи можливості, подібних до описаних у попередніх двох визначеннях.

На базі цього визначення запропоновано концепцію представлення вимог до ІС, яка являє собою сукупність таких положень [9]:

- відмова від розгляду тільки множини сформульованих вимог до ІС та первісне подання вимог до ІС як елементів універсуму, що поєднує як відомі, так і невідомі Постачальнику ІТ-послуг, Споживачу ІТ-послуг чи їм обом вимоги до ІС, а також методи формування цих вимог;
- первісна різноманітність представлень вимог до ІС у вигляді даних, інформації та знань;
- процесний підхід до опису вимог, який визначає мінімальну процесну атрибутивну модель вимоги до ІС;
- підхід до управління вимогами до ІС, оснований на відомих принципах управління вимогами.

Ця концепція визначає основні особливості формалізованого опису груп вимог до ІС та окремих вимог, їх моделей різного виду, методів та засобів формування, аналізу та управління цими вимогами, а також можливих моделей і методів синтезу опису архітектури ІС, що проектується, на різних рівнях представлення.

Запропоновані визначення вимог до ІС та концепція подання цих вимог дозволяють сформулювати основне визначення патерна проектування вимог до ІС. За цим визначенням патерн проектування вимог до ІС – це результат виділення та повторного використання з прикладними цілями Постачальником ІТ-послуг таких видів знань [10]:

- про умову чи можливість, які необхідні Постачальнику ІТ-послуг для вирішення проблеми, що виникла перед ним, або для досягнення поставленої мети;
- про умову чи можливість, яку повинна мати ІС або компонент ІС (ІТ-послуга, ІТ-сервіс) з погляду Постачальника чи Споживача ІТ-послуг та яка відповідає домовленості, стандарту, специфікації чи іншому офіційному документу;

- про документоване (з використанням природної чи формальної мови) представлення умови чи можливості, подібних до тих, що описані у перших двох визначеннях.

Використовуючи запропоноване визначення, патерни проектування вимог до ІС можна класифікувати так:

- за призначенням – структурні патерни та патерни поведінки;
- за рівнем використання – патерни подання вимог до ІС у вигляді знань, патерни представлення вимог до ІС у вигляді інформації, патерни представлення вимог до ІС у вигляді даних.

Визначення кожного виду патернів проектування вимог до ІС наведено у таблиці.

**Визначення видів патернів проектування вимог до інформаційної системи
згідно із запропонованою класифікацією**

Вид патерна	Структурний патерн	Патерн поведінки
Патерн представлення вимог до ІС у вигляді знань	Патерн знань про структури даних, що вимагаються, які формують єдине цілісне інформаційне представлення об'єкта чи процесу	Патерн знань про процес обробки структур даних, які формують єдине цілісне інформаційне представлення об'єкта чи процесу
Патерн представлення вимог до ІС у вигляді інформації	Патерн виконаного природною чи формальною мовами опису вимоги до структур даних, які формують єдине цілісне інформаційне представлення об'єкта чи процесу	Патерн виконаного природною чи формальною мовами опису вимоги до процесу обробки структур даних, які формують єдине цілісне інформаційне представлення об'єкта чи процесу
Патерн представлення вимог до ІС у вигляді даних	Патерн ІТ-сервісів та їх елементів, що забезпечують реалізацію в межах ІТ-послуги структур даних, які формують єдине цілісне інформаційне представлення об'єкта чи процесу	Патерн ІТ-сервісів та їх елементів, що забезпечують реалізацію в межах ІТ-послуги процесу обробки структур даних, які формують єдине цілісне інформаційне представлення об'єкта чи процесу

Запропоновані визначення патернів проектування вимог до ІС дають змогу формалізовано описати сукупність цих патернів теоретико-категорною моделлю M_{IS}^{Pt} , яка має такий вигляд:

$$M_{IS}^{Pt} = [D_{IS}^{Pt}, I_{IS}^{Pt}, K_{IS}^{Pt}, H(D_{IS}^{Pt}), H(I_{IS}^{Pt}), H(K_{IS}^{Pt}), H(D_{IS}^{Pt}, I_{IS}^{Pt}), H(I_{IS}^{Pt}, D_{IS}^{Pt}), H(D_{IS}^{Pt}, K_{IS}^{Pt}), H(K_{IS}^{Pt}, D_{IS}^{Pt}), H(I_{IS}^{Pt}, K_{IS}^{Pt}), H(K_{IS}^{Pt}, I_{IS}^{Pt})], \quad (1)$$

де D_{IS}^{Pt} – підклас структурних патернів та патернів поведінки представлень вимог до ІС у вигляді даних; I_{IS}^{Pt} – підклас структурних патернів та патернів поведінки представлень вимог до ІС у вигляді інформації; K_{IS}^{Pt} – підклас структурних патернів та патернів поведінки представлень вимог до ІС у вигляді знань; $H(D_{IS}^{Pt})$ – підмножина морфізмів, що описують перетворення структурних патернів та патернів поведінки представлень вимог до ІС у вигляді даних самих у себе та один в інший, яка має такий вигляд:

$$\forall X, Y \in D_{IS}^{Pt} \quad H(D_{IS}^{Pt}) = H(I_X) \cup H(X, Y) \cup H(I_Y), \quad (2)$$

I_X, I_Y – одиничні морфізми, умови існування яких описані у [11]; $H(I_{IS}^{Pt})$ – підмножина морфізмів, що описують перетворення структурних патернів та патернів поведінки представлень вимог до ІС у вигляді інформації самих у себе та один в інший, яка має такий вигляд:

$$\forall X, Y \in I_{IS}^{Pt} \quad H(I_{IS}^{Pt}) = H(I_X) \cup H(X, Y) \cup H(I_Y), \quad (3)$$

$H(K_{IS}^{Pt})$ – підмножина морфізмів, що описують перетворення структурних патернів та патернів поведінки представлень вимог до ІС у вигляді знань самих у себе та один в інший, яка має такий вигляд:

$$\forall X, Y \in K_{IS}^{Pt} \quad H(K_{IS}^{Pt}) = H(I_X) \cup H(X, Y) \cup H(I_Y), \quad (4)$$

$H(D_{IS}^{Pt}, I_{IS}^{Pt})$ – підмножина морфізмів, що описують перетворення структурних патернів та патернів поведінки представлень вимог до ІС у вигляді даних у структурні патерни та патерни поведінки представлень вимог до ІС у вигляді інформації; $H(I_{IS}^{Pt}, D_{IS}^{Pt})$ – підмножина морфізмів, що описують перетворення структурних патернів та патернів поведінки представлень вимог до ІС у вигляді інформації у структурні патерни та патерни поведінки представлень вимог до ІС у вигляді даних; $H(D_{IS}^{Pt}, K_{IS}^{Pt})$ – підмножина морфізмів, що описують перетворення структурних патернів та патернів поведінки представлень вимог до ІС у вигляді даних у структурні патерни та патерни поведінки представлень вимог до ІС у вигляді знань; $H(K_{IS}^{Pt}, D_{IS}^{Pt})$ – підмножина морфізмів, що описують перетворення структурних патернів та патернів поведінки представлень вимог до ІС у вигляді знань у структурні патерни та патерни поведінки представлень вимог до ІС у вигляді даних; $H(I_{IS}^{Pt}, K_{IS}^{Pt})$ – підмножина морфізмів, що описують перетворення структурних патернів та патернів поведінки представлень вимог до ІС у вигляді інформації у структурні патерни та патерни поведінки представлень вимог до ІС у вигляді знань; $H(K_{IS}^{Pt}, I_{IS}^{Pt})$ – підмножина морфізмів, що описують перетворення структурних патернів та патернів поведінки представлень вимог до ІС у вигляді знань у структурні патерни та патерни поведінки представлень вимог до ІС у вигляді інформації.

Розроблена модель (1) є фактор-категорією для усіх моделей вимог до ІС, тобто є узагальненим описом різних груп вимог до ІС, встановлюючи для цих груп єдиний базис знань про вимоги до ІС, їх представлення та засоби перетворення цих представлень самих у себе та один в інший.

Такий погляд на модель патернів проектування вимог до ІС дає змогу описати підмножину інформаційних технологій формування та аналізу вимог до ІС на основі запропонованих рішень теоретико-категорною моделлю $M_{Tr_{IS}}^{Pt}$, яка матиме такий вигляд:

$$\begin{aligned} M_{Tr_{IS}}^{Pt} = [& M_{IS}^{Pt}, L_{IS}^B, L_{IS}^{IB}, L_{IS}^S, L_{IS}^f, L_{IS}^{nf}, L_{IS}^{fw}, L_{IS}^{nfw}, F_{M_{IS}^{Pt}}^{L_{IS}^B}, F_{M_{IS}^{Pt}}^{L_{IS}^{IB}}, F_{M_{IS}^{Pt}}^{L_{IS}^S}, \\ & F_{M_{IS}^{Pt}}^{L_{IS}^f}, F_{M_{IS}^{Pt}}^{L_{IS}^{nf}}, F_{M_{IS}^{Pt}}^{L_{IS}^{fw}}, F_{M_{IS}^{Pt}}^{L_{IS}^{nfw}}, F_{L_{IS}^B}^{M_{IS}^{Pt}}, F_{L_{IS}^{IB}}^{M_{IS}^{Pt}}, F_{L_{IS}^S}^{M_{IS}^{Pt}}, F_{L_{IS}^f}^{M_{IS}^{Pt}}, F_{L_{IS}^{nf}}^{M_{IS}^{Pt}}, \\ & F_{L_{IS}^{fw}}^{M_{IS}^{Pt}}, F_{L_{IS}^{nfw}}^{M_{IS}^{Pt}}, F_{L_{IS}^B}^{L_{IS}^S}, F_{L_{IS}^{IB}}^{L_{IS}^S}, F_{L_{IS}^S}^{L_{IS}^f}, F_{L_{IS}^f}^{L_{IS}^{nf}}, F_{L_{IS}^{nf}}^{L_{IS}^{fw}}, F_{L_{IS}^{fw}}^{L_{IS}^{nfw}}, F_{L_{IS}^{nfw}}^{L_{IS}^{fw}}]. \end{aligned} \quad (5)$$

де $L_{IS}^B, L_{IS}^{IB}, L_{IS}^S, L_{IS}^f, L_{IS}^{nf}, L_{IS}^{fw}, L_{IS}^{nfw}$ – теоретико-категорні моделі груп вимог до ІС, а саме: бізнес-вимог, вимог до ІС як аспекту бізнесу, вимог до ІС загалом, функціональних вимог до ІТ-послуг, нефункціональних вимог до ІТ-послуг, функціональних вимог до ІТ-сервісів, нефункціональних вимог до ІТ-сервісів [8]; $F_{M_{IS}^{Pt}}^{L_{IS}^B}, F_{M_{IS}^{Pt}}^{L_{IS}^{IB}}, F_{M_{IS}^{Pt}}^{L_{IS}^S}, F_{M_{IS}^{Pt}}^{L_{IS}^f}, F_{M_{IS}^{Pt}}^{L_{IS}^{nf}}, F_{M_{IS}^{Pt}}^{L_{IS}^{fw}}, F_{M_{IS}^{Pt}}^{L_{IS}^{nfw}}$ – функторні моделі, які описують методи, прийоми та засоби формування патернів проектування вимог до ІС як виділення знань на основі представлень раніше сформульованих груп вимог до ІС; $F_{L_{IS}^B}^{M_{IS}^{Pt}}, F_{L_{IS}^{IB}}^{M_{IS}^{Pt}}, F_{L_{IS}^S}^{M_{IS}^{Pt}}, F_{L_{IS}^f}^{M_{IS}^{Pt}}, F_{L_{IS}^{nf}}^{M_{IS}^{Pt}}, F_{L_{IS}^{fw}}^{M_{IS}^{Pt}}, F_{L_{IS}^{nfw}}^{M_{IS}^{Pt}}$ – функторні моделі, які описують методи, прийоми та засоби повторного використання патернів проектування вимог до ІС під час формування

представлень груп вимог до ІС, що створюється; $F_{L_{IS}}^{L_{IS}^B}, F_{L_{IS}}^{L_{IS}^{IB}}, F_{L_{IS}}^{L_{IS}^S}, F_{L_{IS}}^{L_{IS}^{IS}}, F_{L_{IS}}^{L_{IS}^f}, F_{L_{IS}}^{L_{IS}^{If}}, F_{L_{IS}}^{L_{IS}^{nf}}, F_{L_{IS}}^{L_{IS}^{fw}}, F_{L_{IS}}^{L_{IS}^{nfw}}$ – функторні моделі, які описують методи, прийоми та засоби перетворення представлень груп вимог до ІС, що створюється, одна в іншу під час виконання процесів створення системи.

Висновки

Використання інтелектуальної інформаційної технології формування та аналізу вимог до ІС, в основу якої покладено запропоновані моделі патернів проектування вимог до ІС, дає змогу розв'язати задачу повторного використання вимог до ІС за рахунок формування та постійного оновлення бази знань про об'єкти та процеси, що автоматизуються, а також знань про нові підходи до формування вимог та засоби їх подання. Отже, замість повторного використання окремих вимог до ІС, що не дає належного ефекту, пропонується повторно використовувати знання, виявлені під час формування та аналізу вимог, поставлених до ІС, що створювалися Постачальником ІТ-послуг раніше.

Основними напрямками подальшого дослідження у цій галузі слід вважати створення моделей конкретних патернів проектування вимог до ІС, реалізацію та апробацію запропонованої інтелектуальної інформаційної технології формування та аналізу вимог до ІС.

1. ГОСТ ИСО/МЭК 15288–2005. Системная инженерия. Процессы жизненного цикла систем. Введ. 01–01–2007. – М.: Стандартинформ, 2006. 2. ГОСТ ИСО/МЭК 12207–2010. Системная и программная инженерия. Процессы жизненного цикла программных средств. Введ. 01–03–2012. – М.: Стандартинформ, 2011. 3. Гамма Э. Приемы объектно-ориентированного проектирования. Паттерны проектирования / Э. Гамма, Р. Хелм, Р. Джонсон, Д. Влссидес. – СПб.: Питер, 2010. 4. Фримен Э. Паттерны проектирования / Э. Фримен, Э. Фримен, К. Сьерра, Б. Бейтс. – СПб.: Питер, 2011. 5. Фаулер М. Рефакторинг. Улучшение существующего кода. – М.: Символ-Плюс, 2008. 6. Larman С. Applying UML and Patterns. An Introduction to Object-Oriented Analysis and Design and Iterative Development. Third Edition. Prentice Hall PTR, 2004. 7. Волков Ю.О. Управление требованиями и автоматизация этого процесса // Сайт Юрия Волкова. – http://yurivolkov.com/articles/Requirements_management_automation_ru.html. 8. Евланов М.В. Определение понятия “требование к информационной системе” // Вісник Академії митної служби України. Серія “Технічні науки”. – 2012. – № 2. 9. Евланов М.В. Концепция представления требований к информационной системе // Информационные системы и технологии: материалы Междунар. науч.-техн. конф. – Харьков–Морское, 22–29 сентября 2012 г.: тезисы докладов / редкол.: А.Д. Тевяшев (отв. ред.) и др. – Харьков: НТМТ, 2012. 10. Левыкин В.М. Подход к использованию паттернов проектирования при работе с требованиями к информационной системе / В.М. Левыкин, М.В. Евланов, М.А. Керносов // Системний аналіз. Інформатика. Управління (САІУ-2013): матер. IV Міжнар. наук.-практ. конф. (м. Запоріжжя, 13–16 березня 2013 р.). – Запоріжжя: КПУ, 2013. 11. Цаленко М.Ш. Основы теории категорий / М.Ш. Цаленко, Е.Г. Шульгейфер. – М.: Наука, 1974.