

О.Є. КОВАЛЕНКО

КОМПОЗИЦІЙНА КОНВЕРГЕНЦІЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ У СИСТЕМАХ СИТУАЦІЙНОГО УПРАВЛІННЯ НА ОСНОВІ МОДЕЛІ ЗНАТЬ ПРЕДМЕТНОЇ ОБЛАСТІ

***Анотація.** Запропоновано знання-орієнтовану модель конвергенції інформаційних технологій шляхом їх композиції. Визначено відповідність між формальним описом проблемної області та засобами інформаційних технологій. Наведено класифікацію категорій інформаційних технологій та представлено формальну модель їх композиції в системах ситуаційного управління. Запропонований підхід забезпечує облік та конвергенцію засобів інформаційних технологій для вирішення цільових проблем ситуаційного управління шляхом композиції етапів онтологічного аналізу вимог, функціональної декомпозиції, предметної інтерпретації та фізичної реалізації. Використання запропонованої моделі процесу проектування дозволить забезпечити формалізований синтез систем ситуаційного управління для цільової предметної області шляхом конвергенції інформаційних технологій на основі моделей знань.*

***Ключові слова:** система ситуаційного управління, модель знань, конвергенція технологій, композиційна модель.*

DOI: 10.35350/2409-8876-2019-15-2-40-45

Вступ

Створення та використання складних технічних систем різного призначення обумовило потребу у виробленні систематичних і цілісних підходів до вирішення проблем їх підтримки впродовж всього життєвого циклу [1]. Побудова і використання систем ситуаційного управління (ССУ) у різних сферах вимагає розробки методів і засобів конвергенції інформаційних технологій для здійснення адекватного ситуаційного управління у цільовій проблемній області. Конвергенція визначається як глибока інтеграція знань, засобів та усіх раціональних видів людської діяльності для досягнення спільної мети, можливостей отримання відповідей на нові питання по зміні відповідної фізичної або соціальної екосистеми [2, 3].

1. Загальна постановка задачі, об'єкт, предмет та мета досліджень

Традиційні технологічні підходи до створення інформаційно-комунікаційних систем (ІКС) потребують додаткових витрат ресурсів (людських, часових, фінансових) для конвергенції технологій у рамках цільової ССУ. Для вирішення цієї проблеми використовуються різні методології модельно-орієнтованої розробки (model-driven development, MDD) систем [4], зокрема такі як предметно-орієнтована розробка (domain-driven development, DDD) [5, 6], розробка, керована вимогами [7, 8], розробка, керована функціональністю (feature driven development, FDD) [9] та ін. Крім того, одним з напрямків підвищення

ефективності розробки ІКС є можливість багаторазового використання готових програмно-технологічних компонентів у нових розробках. Цей напрямок одержав назву "складальне програмування" [10, 11]. Однак такий підхід висуває жорсткі вимоги до функціональної універсальності та стандартизації архітектури інформаційних технологій, а також забезпечення їх портативності в гетерогенному інформаційному середовищі. Крім того, у створенні спеціалізованих ІКС, зокрема ССУ, беруть участь не лише ІТ-спеціалісти, а й фахівці з предметних сфер застосування ІКС. У зв'язку з цим доцільно створення засобів автоматизації з проблемно-орієнтованої розробки ІКС на основі вимог і знань фахівців з предметної області [12].

2. Конвергенція технологій на основі моделей знань

Розробка конвергентних систем в першу чергу вимагає оцінки їх доцільності. Коли ж рішення про необхідність розробки такої системи прийнято, слід формалізувати опис предметної області і провести декомпозицію задачі створення ССУ на підзадачі.

Формальний опис предметної області ситуаційного управління (СУ), для якої створюється проблемно-орієнтована ССУ, є ієрархією концептів (понять) і функціональних перетворень, якими будуть оперувати користувачі. Формальний опис предметної області має також містити узагальнений опис моделі процесів СУ. Композиція технологій СУ повинна здійснюватися з урахуванням архітектури ССУ.

Таким чином, основну задачу композиції ІКТ в рамках конвергентної архітектури ССУ можна визначити як встановлення взаємно однозначної, коректної, оптимальної відповідності між формальним описом прикладної області та засобами інформаційних технологій:

$$F: \langle A, D \rangle \rightarrow \langle B, C, R \rangle, \quad (1)$$

де F – функція відображення моделі A предметної області і моделі вимог D до проблемно-орієнтованої ССУ на множину технологічних засобів (алфавіт) СУ B і множину функцій управління C цими технологічними засобами на множині R допустимих ситуацій у предметній галузі.

Множину B технологічних засобів СУ, в залежності від їх семантичного змісту та функціональних спроможностей, можна розділити на підмножини, які визначають класи технологічних засобів СУ:

$$B = \bigcup_{i=1}^n B^{(i)},$$

де n – визначає кількість технологічних засобів. В основу розбивки множини B на підмножини може бути покладена ознака функціонального призначення, тобто технологічна орієнтація функції, що спирається на певний клас $B^{(i)}$ технологічних засобів.

За функціональним призначенням можна виділити такі технологічні засоби СУ:

- засоби візуалізації;
- засоби отримання та збереження інформації;
- засоби фільтрації та очищення даних;
- засоби аналітичної обробки;
- засоби персоналізації;
- засоби інформаційної безпеки;
- системні засоби;
- конфігураційні засоби тощо.

У свою чергу всі технологічні засоби можна згрупувати за категоріями застосування:

- організаційна;
- системна;
- предметна;
- допоміжна.

Множина C функцій управління технологічними засобами включає в себе допустимі операції маніпулювання цими засобами на множині R допустимих ситуацій.

Таким чином, кінцеву конфігурацію технологій в рамках окремої реалізації ССУ можна представити як процес P конвергенції технологій в проблемно-орієнтованій ССУ:

$$P = \langle \tilde{B}, \tilde{C}, \tilde{R}, \alpha \rangle, \quad (2)$$

де $\tilde{B} \subset B, \tilde{C} \subset C, \tilde{R} \subset R, \alpha$ – відношення лінійного порядку на множині \tilde{C} .

Вихідними даними для процесу P (2) конвергенції технологій в проблемно-орієнтованій ССУ є узгоджена модель знань:

$$\Pi = \langle \tilde{A}, \tilde{D}, \beta \rangle, \quad (3)$$

де $\tilde{A} \subset A, \tilde{D} \subset D, \beta$ – функція інтерпретації вимог \tilde{D} до ССУ у контексті моделі знань предметної області \tilde{A} .

Процес P конвергенції може бути розділений на декілька етапів, що відповідають ступеню наближення конфігурації технологій в ССУ до задоволення вимог \tilde{D} . Важливим при реалізації процесу P конвергенції є забезпечення поступовості, тобто збереження результатів проміжних конфігурацій конвергентної архітектури незалежно від ступеня задоволення вимог. Отже, кожен етап процесу P повинен передбачати збереження його результатів.

Першим етапом функціонального відображення F (1) є логіко-семантичний (онтологічний) аналіз вимог до архітектури технологій ССУ. У процесі відображення отримуємо онтологічне подання моделі вимог з конкретизованими атрибутами, що відповідають моделі предметної області. Для ССУ таке відображення буде мати вид:

$$F_1 : \langle A, D \rangle \rightarrow \langle A_s, D_o, D_s, D_c, D_r, S \rangle, \quad (4)$$

де $D = D_O \oplus D_S \oplus D_C \oplus D_r$; F_1 – функція онтологічного відображення вимог на предметну область; D_O – множина вимог до організаційних технологій; D_S – множина системних вимог; D_C – множина вимог до технологій предметної області; D_r – множина вимог до допоміжних технологій; S – множина відношень між концептами предметної області A ; A_S – онтологічне представлення моделі знань предметної області.

Наступним етапом відображення (1) є функціональна декомпозиція F_2 , в процесі якої здійснюється розбиття логіко-семантичного представлення, отриманого в результаті відображення F_1 (4), на функціонально однотипні групи, відповідні класам сервісів проблемної області, системної підтримки, функцій організаційної підтримки, допоміжних функцій (наприклад, функцій розширення та інтеграції ССУ):

$$G^{(i)} = \langle A_S^{(i)}, D_O^{(i)}, D_S^{(i)}, D_C^{(i)}, D_r^{(i)}, S^{(i)} \rangle.$$

Тут $\bigcap_{i=1}^n G^{(i)} = \emptyset$, де n – перевірена (верифікована) та підтверджена (валідована) кількість категорій конвергентних технологічних сервісів.

Таким чином, відображення F_2 можна визначити як:

$$F_2 : \langle A_S, D_O, D_S, D_C, D_r, S \rangle \rightarrow \left\langle \bigcup_{i=1}^n G^{(i)}, \Gamma \right\rangle, \quad (5)$$

де Γ – функція фільтрації (верифікації та валідації) логіко-семантичної моделі за категоріями сервісів. Наступним етапом відображення (1) є предметна інтерпретація результатів відображення (5):

$$F_3 : \left\langle \bigcup_{i=1}^m G^{(i)}, \Gamma \right\rangle \rightarrow \langle B, C, R \rangle, \quad (6)$$

де $m \leq n$ – число функціональних груп, що визначають кінцеву конфігурацію технологій ССУ. Якщо кортеж $\langle B, C, R \rangle$ відповідає вимогам до сервісів технологій конвергентної архітектури ССУ, то кінцева конфігурація технологій ССУ може бути представлена у вигляді (2). Якщо ж кортеж $\langle B, C, R \rangle$ являє собою узагальнену конвергентну модель технологій ССУ (наприклад, опис у вигляді моделей формальної нотації), то необхідно провести перетворення узагальненої моделі в екземпляр (варіант) її конкретної фізичної реалізації:

$$F_4 : P \rightarrow P'.$$

Таким чином, відображення (1) можна представити у вигляді суперпозиції відображень:

$$F = F_1 \circ F_2 \circ F_3 \circ F_4,$$

де o – знак суперпозиції, причому, якщо $P = P'$, то F_3 – тотожне відображення.

Висновки

Встановлення відповідності між формальним описом предметної області і засобами інформаційних технологій дозволило застосувати орієнтований на знання підхід до розробки систем ситуаційного управління. Запропонований підхід забезпечує формування репозиторію та конвергенцію засобів інформаційних технологій для вирішення цільових проблем ситуаційного управління шляхом композиції етапів онтологічного аналізу вимог, функціональної декомпозиції, предметної інтерпретації та фізичної реалізації. Використання запропонованої моделі процесу проектування дозволить забезпечити формалізований синтез ССУ для цільової предметної області шляхом конвергенції потрібних ІКТ на основі моделей знань.

СПИСОК ЛІТЕРАТУРИ

1. Коваленко О. Є. Системна інженерія та життєвий цикл систем // Електронне моделювання, т. 40, № 6, С. 61–82, 2018.
2. Roco M. C., Bainbridge W. S. The new world of discovery, invention, and innovation: convergence of knowledge, technology, and society // Journal of Nanoparticle Research, т. 15, № 9, pp. 1-17, 2013.
3. Roco M. C. Convergence-Divergence Process // Handbook of Science and Technology Convergence / W. Bainbridge, M. Roco, eds. – Cham, Springer, 2016, pp. 79-93.
4. da Silva A. R. Model-Driven Engineering: A Survey Supported by a Unified Conceptual Model // Computer Languages, Systems & Structures, № 43, p. 139–155, 2015.
5. Haywood D. An Introduction to Domain Driven Design // Methods and Tools, vol. 17, № 4, pp. 18-37, Winter 2009.
6. Evans E. Domain Driven Design, Definitions and Pattern Summaries. March 2015. [Онлайнвий]. Available: https://domainlanguage.com/wp-content/uploads/2016/05/DDD_Reference_2015-03.pdf. [Дата звернення: 04 April 2019].
7. Castro J., Kolp M., Mylopoulos J. A requirements-driven development methodology // International Conference on Advanced Information Systems Engineering, pp. 108-123, June 2001.
8. Castro J., Kolp M., Mylopoulos J. A Requirements-Driven Development Methodology // Seminal Contributions to Information Systems Engineering. – Berlin, Heidelberg, Springer, 2013, pp. 265-280.
9. Ambler S. W. Feature Driven Development (FDD) and Agile Modeling. 2018. [Онлайнвий]. Available: <http://www.agilemodeling.com/essays/fdd.htm>. [Дата звернення: 04 April 2019].
10. Лаврищева Е. М. Сборочное программирование. Теория и практика // Кибернетика и системный анализ. – 2009, т. 45, № 6, С. 3-12.
11. Лаврищева Е. М. Теория объектно-компонентного моделирования программных систем. 2016. [Онлайнвий]. Available: http://www.ispras.ru/preprints/docs/prep_29_2015.pdf.
12. Kovalenko A., Velev D. Compositional Model of Design Process in the Problem-Oriented CAD of Applied Software // Engineering Simulation. – т. 16, p. 129–133, 1998.

REFERENCES

1. Kovalenko O. E. (2018). Systems Engineering and Systems Life Cycle. Electronic modeling, Vol. 40, No 6, 61–82. (In Ukrainian)

2. Roco M. C., Bainbridge W. S. (2013). The new world of discovery, invention, and innovation: convergence of knowledge, technology, and society. *Journal of Nanoparticle Research*, т. 15, № 9, 1-17.
3. Roco M. C. (2016). *Convergence-Divergence Process*. Handbook of Science and Technology Convergence. Cham: Springer, 79-93.
4. da Silva A. R. (2015). Model-Driven Engineering: A Survey Supported by a Unified Conceptual Model. *Computer Languages, Systems & Structures*, № 43, 139-155.
5. Haywood D. (2009, Winter). An Introduction to Domain Driven Design. *Methods and Tools*, vol. 17, № 4, 18-37.
6. Evans E. (2015, March). Domain Driven Design, Definitions and Pattern Summaries. Retrieved from https://domainlanguage.com/wp-content/uploads/2016/05/DDD_Reference_2015-03.pdf.
7. Castro J., Kolp M., Mylopoulos J. (2001, June). A requirements-driven development methodology. *International Conference on Advanced Information Systems Engineering*, 108-123.
8. Castro J., Kolp M., Mylopoulos J. (2013). A Requirements-Driven Development Methodology. *Seminal Contributions to Information Systems Engineering*. Berlin, Heidelberg: Springer, 265-280.
9. Ambler S. W. (2018). Feature Driven Development (FDD) and Agile Modeling. Retrieved from <http://www.agilemodeling.com/essays/fdd.htm>.
10. Lavrisheva E. M. (2009). Sborochnoe programmirovaniye. *Teoria i Practica*. [Assembly programming. Theory and Practice]. *Cybernetics and Systems Analysis*, Vol. 45, No 6, 3-12. (In Russian)
11. Lavrisheva E. M. (2016). *Teoria Ob'ektno-Componentnogo Modelirovaniya Programnih System* [Theory of object-component modeling of software systems]. Retrieved from http://www.ispras.ru/preprints/docs/prep_29_2015.pdf. (In Russian)
12. Kovalenko A., Velev D. (1998). Compositional Model of Design Process in the Problem-Oriented CAD of Applied Software. *Engineering Simulation*, Vol. 16, 129-133.

Стаття надійшла до редакції 28.05.2019.