DOI: 10.34185/1991-7848.itmm.2026.01.034

ТЕХНОЛОГІЧНИЙ ПОКАЗНИК ЯКОСТІ РОЗРОБКИ ТА МОДИФІКАЦІЇ СКЛАДНИХ СИСТЕМ - ЗВ'ЯЗНІСТЬ

Карповський Д.О.[1] [ORCID], Шинкаренко В.І.[2] [ORCID]

[1]*Український державний університет науки і технологій, аспірант, Україна*

[2]*Український державний університет науки і технологій,*

д.т.н., професор, Україна

Анотація. *У роботі розглядається зв'язність як один із ключових технологічних показників якості розробки та модифікації складних систем. Проаналізовано її роль у структурній організації програмного забезпечення та вплив на зрозумілість, підтримуваність і гнучкість системи. Показано взаємозв'язок зв'язності зі зчепленням та об'ємом, а також обґрунтовано доцільність досягнення високої зв'язності при низькому рівні залежностей між компонентами. Окрему увагу приділено прояву зв'язності в онтологічному моделюванні та організаційних системах, зокрема на виробництві та в університетському середовищі. Розглянуто, що високий рівень зв'язності забезпечує цілісність системи, спрощує її модифікацію та підвищує ефективність функціонування.*

Ключові слова: *програмне забезпечення, інформаційна система, програмна архітектура, зв'язність, онтології, складні системи, виробничі системи, організаційні структури.*

Сучасні складні системи – незалежно від того, чи йдеться про програмне забезпечення, виробничі процеси або організаційні структури – мають спільну рису: вони складаються з великої кількості взаємопов'язаних елементів. Ефективність їх функціонування значною мірою залежить не лише від окремих компонентів, а й від того, як саме ці компоненти організовані [1].

У програмуванні зв'язність є одним із ключових принципів якісної побудови програмних систем. Вона визначає, наскільки логічно пов'язані між собою елементи всередині модуля, класу або функції. Висока зв'язність означає, що всі складові компонента спрямовані на виконання однієї чітко визначеної задачі. Наприклад, клас, який відповідає виключно за обробку платежів, є більш зв'язним, ніж клас, що одночасно займається обробкою платежів, логуванням та управлінням користувачами [2].

Такий підхід безпосередньо впливає на якість коду. Високозв'язні модулі легше читати, тестувати та підтримувати, оскільки їх поведінка є передбачуваною і зрозумілою. Крім того, зміни в одному компоненті рідше впливають на інші частини системи, що зменшує ризик виникнення помилок. Це особливо важливо в умовах постійної еволюції програмного забезпечення.

Низька зв'язність, навпаки, призводить до появи так званих «божественних об'єктів» (God objects), які виконують надто багато функцій. Такі компоненти важко модифікувати, оскільки будь-яка зміна може мати непередбачувані наслідки для всієї системи. У результаті зростає складність супроводу та знижується гнучкість програмного продукту.

Поняття зв'язності виступає універсальним показником якості системи. Зв'язність разом зі зчепленням та об'ємом визначає ефективність архітектури та рівень організації складних структур.

Зв'язність є важливою не лише в програмуванні. Її прояв можна спостерігати і в реальних організаційних системах, таких як виробництво або навчальні заклади.

Розглянемо виробничу систему. Сучасне підприємство складається з багатьох підрозділів: виробничих ліній, складів, логістики, контролю якості та управління. Кожен із цих підрозділів виконує певну функцію. Якщо всередині підрозділу всі процеси чітко спрямовані на досягнення єдиної мети (наприклад, виготовлення конкретного продукту), можна говорити про високу зв'язність [3].

Наприклад, виробнича лінія, яка відповідає лише за складання одного типу виробу, є більш зв'язною, ніж підрозділ, який виконує одночасно різнорідні задачі. У першому випадку процеси узгоджені, зрозумілі та легко оптимізуються. У другому – виникає плутанина, зростає кількість помилок і ускладнюється управління [4].

Зв'язність безпосередньо впливає на ефективність функціонування виробництва. Високозв'язні підсистеми легше автоматизувати, контролювати та масштабувати. У разі необхідності змін (наприклад, модернізації

обладнання) вони можуть бути реалізовані локально, без суттєвого впливу на інші частини системи.

Аналогічна ситуація спостерігається і в університетській системі. Університет складається з факультетів, кафедр, груп студентів, адміністративних підрозділів. Кожен із цих елементів має свою функціональну роль [5].

Якщо кафедра спеціалізується на одній галузі знань і всі її дисципліни логічно пов'язані між собою, це приклад високої зв'язності. Така структура забезпечує цілісність навчального процесу, полегшує координацію роботи викладачів і покращує якість освіти.

У випадку низької зв'язності, коли в межах однієї кафедри поєднуються несумісні або слабко пов'язані дисципліни, виникають труднощі в організації навчального процесу. Студентам складніше формувати системне розуміння предметної області, а викладачам – узгоджувати навчальні програми [6].

Функціонування системи в обох випадках (виробництво та університет) значною мірою залежить від рівня зв'язності її підсистем. Висока зв'язність забезпечує:

- чіткий розподіл функцій,
- зрозумілу структуру,
- спрощення управління,
- підвищення ефективності взаємодії [7].

У процесі проєктування складних систем важливу роль відіграє використання UML, що дозволяє формалізувати структуру та поведінку системи через взаємопов'язані діаграми. UML-проєкт охоплює різні типи діаграм, які відображають як статичні, так і динамічні аспекти системи. Через ці діаграми проявляються зв'язність і зчеплення, що дає змогу оцінювати якість архітектурних рішень ще на етапі проєктування [8].

Окремо варто розглянути зв'язність в онтологіях. У цьому випадку вона визначає, наскільки логічно пов'язані між собою поняття та відношення. Високозв'язна онтологія має чітку структуру, де кожне поняття займає своє місце і пов'язане з іншими через осмислені зв'язки [9].

Це дозволяє ефективно використовувати такі моделі в системах обробки знань, пошуку інформації та штучного інтелекту. Натомість низька зв'язність призводить до фрагментації знань і ускладнює їх автоматичне використання.

Таким чином, незалежно від природи системи – програмної, виробничої чи освітньої – зв'язність виступає ключовим фактором її ефективного функціонування. Вона визначає, наскільки добре організовані внутрішні процеси та наскільки система здатна адаптуватися до змін.

Висновок. Зв'язність є універсальним технологічним показником якості складних систем. Вона характеризує внутрішню узгодженість компонентів і визначає ефективність їх взаємодії. У програмних системах зв'язність впливає на підтримуваність і гнучкість коду. У виробничих системах – на ефективність процесів і можливість їх оптимізації. В освітніх структурах – на цілісність навчального процесу. В онтологіях – на логічну узгодженість знань. Забезпечення високої зв'язності дозволяє створювати системи, які є зрозумілими, керованими та здатними до подальшого розвитку, що робить її одним із ключових критеріїв якості в сучасній інженерії складних систем.

ЛІТЕРАТУРА / REFERENCE

1. Ameneh Naghdipour, Seyed Mohammad Hossein Hasheminejad, A. Software design pattern selection approaches: a systematic literature review. Journal of Software Engineering Studies – 2022., P. 1–20. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/spe.3176

2. Eramo, R., Tucci, M., Di Pompeo, D., Cortellessa, V., Di Marco, A., Taibi, D. Architectural support for software performance in continuous software engineering: a systematic mapping study. arXiv – 2023., P. 1–28. URL: https://arxiv.org/abs/2304.02489

3. Heiland, L., Hauser, M., Bogner, J. Design patterns for AI-based systems: a multivocal literature review and pattern repository. arXiv – 2023., P. 1–35., URL: https://arxiv.org/abs/2303.13173

4. Belle, A. B., El Boussaidi, G., Lethbridge, T. C., Kpodjedo, S., Mili, H., Paz, A. Systematically reviewing the layered architectural pattern principles and their use to reconstruct software architectures. arXiv – 2021., P. 1–25, URL: https://arxiv.org/abs/2112.01644

5. M. Ravindu An overview of structural design patterns in object-oriented software engineering2024.,P.115.,URL:https://www.researchgate.net/publication/377969013_An_Overview_of_Structural_Design_Patterns_in_Object-Oriented_Software_Engineering

6. M.Paixao. An Empirical Study of Cohesion and Coupling: Balancing Optimisation and Disruption, IEEE Transactions on Evolutionary Computation PP(99):1-1, URL:

https://www.researchgate.net/publication/316021372_An_Empirical_Study_of_Cohesion_and_Coupling_Balancing_Optimisation_and_Disruption
7. Palomba, F., Bavota, G., Di Penta, M., Oliveto, R., De Lucia, A. Detecting bad smells in source code using change history information. IEEE International Conference on Software Maintenance – 2013., P. 268–277, URL: https://fpalomba.github.io/pdf/Conferencs/C2.pdf
8. Shamaev E.V, Shinkarenko V.I. Quality charaterisitcs of uml-project, The 4th International Workshop on Computer Science and Information Technologies. Book of Abstracts/ 18-20 September 2002, University of Patras, Grees, P.26
9. Guizzardi, G., Wagner, G., Almeida, J., Guizzardi, R. Towards ontological foundations for conceptual modeling: the Unified Foundational Ontology (UFO) story. Applied Ontology – 2015.,P.259–271.,URL:https://nemo.inf.ufes.br/wp-content/papercite-data/pdf/towards_ontological_foundations_for_conceptual_modeling__the_unified_foundational_ontology__ufo__story_2015.pdf

COHESION AS A TECHNOLOGICAL QUALITY INDICATOR IN THE DEVELOPMENT AND MODIFICATION OF COMPLEX SYSTEMS

Dmytro Karpovskyi, Viktor Shynkarenko

Abstract. *This paper examines cohesion as one of the key technological quality indicators in the development and modification of complex systems. Its role in the structural organization of software and its impact on system understandability, maintainability, and flexibility are analyzed. The relationship between cohesion, coupling, and system size is demonstrated, and the importance of achieving high cohesion with low inter-component dependencies is substantiated. Particular attention is given to the manifestation of cohesion in ontological modeling and organizational systems, including manufacturing and university environments. It is shown that a high level of cohesion ensures system integrity, simplifies its modification, and enhances operational efficiency.*

Keywords: *software, information system, software architecture, connectivity, ontologies, complex systems, production systems, organisational structures.*