І Міжнародна науково-практична конференція

СУЧАСНІ ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ: ВІД ТЕОРІЇ ДО ПРАКТИКИ

ТЕЗИ ДОПОВІДЕЙ

22–23 травня, 2026 р.

Луцький національний технічний університет

ЛУЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

MINTECH 2026

1st International Scientific and Practical Conference

MODERN INFORMATION TECHNOLOGIES: FROM THEORY TO PRACTICE

BOOK OF ABSTRACTS

22–23 May, 2026

Lutsk National Technical University

ЛУЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

УДК 004
С 91

Міністерство освіти і науки України	Ministry of Education and Science of Ukraine
Волинська обласна рада	Volyn Regional Council
Луцька міська рада	Lutsk City Council
Луцький національний технічний університет	Lutsk National Technical University (Ukraine)
Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського»	National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute” (Ukraine)
Національний університет «Львівська політехніка»	Lviv Polytechnic National University (Ukraine)
Військовий інститут телекомунікацій та інформатизації імені Героїв Крут (м. Київ)	Kruty Heroes Military Institute of Telecommunications and Information Technology (Ukraine)
Рівненський державний гуманітарний університет	Rivne State University of the Humanities (Ukraine)
Державний торговельно-економічний університет (м. Київ)	State University of Trade and Economics (Ukraine)
Чернівецький національний університет імені Юрія Федьковича	Yuriy Fedkovych Chernivtsi National University (Ukraine)
Люблінська політехніка (Польща)	Lublin University of Technology (Poland)
Гронінгенський університет (Нідерланди)	University of Groningen (the Netherlands)
Кавказький університет (Грузія)	Caucasus University (Georgia)
Політехнічний університет Браганси (Португалія)	Polytechnic Institute of Bragança (Portugal)
Університет Сямень (Малайзія)	Xiamen University (Malaysia)
Мінюський університет (Португалія)	University of Minho (Portugal)

Рекомендовано до опублікування науково-технічною радою
Луцького національного технічного університету
(протокол № 10 від 27.05.2026 р.)

Редакційна колегія:
Кондіус Інна Степанівна, к.е.н., доцент (декан факультету комп'ютерних та інформаційних технологій)
Ліщина Наталія Миколаївна, к.т.н., доцент
Ліщина Валерій Олександрович, к.т.н., доцент
Андрущак Ігор Євгенович, д.т.н., професор
Тулашвілі Юрій Йосипович, д.пед.н., професор
Козубцов Ігор Миколайович, д.пед.н., с.н.с.
Ящук Андрій Анатолійович, к.т.н., доцент (головний редактор)
Повстяна Юлія Славомирівна, к.т.н., доцент (відповідальний секретар)
Суринович Олена Миколаївна, к.т.н., доцент
Газдюк Катерина Петрівна. PhD, Чернівецький національний університет імені Юрія Федьковича
Дунець Роман Богданович, д.т.н., професор, Львівська політехніка
Komada Pawel, PhD, Politechnika Lubelska (Poland)
Shulga Artem, PhD, University of Groningen (the Netherlands)

С 91 Тези доповідей I Міжнародної науково-практичної конференції «Сучасні інформаційні технології: від теорії до практики (СІнТех-2026)» (22-23 травня 2026 року). Луцьк: ЛНТУ, 2026. 1086 с.

Book of abstracts of the 1st International Scientific and Practical Conference “Modern Information Technologies: From Theory to Practice (MInTech-2026)” (22–23 May 2026). Lutsk: Lutsk National Technical University, 2026. 1086 p.

Матеріали I Міжнародної науково-практичної конференції «Сучасні інформаційні технології: від теорії до практики». Видання містить аналіз та результати досліджень, що стосуються актуальних питань інформаційних технологій в галузях освіти, науки та промисловості. Тези доповідей надано в авторській редакції. За фактичний матеріал і його інтерпретацію відповідають автори.

Proceedings of the 1st International Scientific and Practical Conference “Modern Information Technologies: From Theory to Practice”. The publication contains analyses and research results addressing current issues in information technologies in the fields of education, science, and industry. The abstracts are published as submitted by the authors. The authors are responsible for the factual content and interpretation of the submitted materials.

Відповідальний за випуск: к.т.н., доцент Ліщина Н.М.

© Колектив авторів

СЕМАНТИЧНИЙ РОЗРИВ У ЗАДАЧАХ ВІДНОВЛЕННЯ БІЗНЕС-ЛОГІКИ ПОЛІГЛОТНИХ КОРПОРАТИВНИХ ПРОГРАМНИХ СИСТЕМ

Понепаляк Андрій Андрійович
ДВНЗ «Донецький національний технічний університет», аспірант кафедри штучного інтелекту та кібербезпеки, andrii.ponepaliak.asp@donntu.edu.ua
Дорогий Ярослав Юрійович
ДВНЗ «Донецький національний технічний університет», завідувач кафедри штучного інтелекту та кібербезпеки, доктор технічний наук, професор, yaroslav.dorohyi@donntu.edu.ua

SEMANTIC GAP IN BUSINESS LOGIC RECOVERY OF POLYGLOT ENTERPRISE SOFTWARE SYSTEMS

Ponepaliak Andrii Andriiovych
SHEE "Donetsk National Technical University", postgraduate student of the Department of Artificial Intelligence and Cybersecurity, andrii.ponepaliak.asp@donntu.edu.ua
Dorohy Yaroslav Yuriyovych
SHEE "Donetsk National Technical University", Head of the Department of Artificial Intelligence and Cybersecurity, Doctor of Technical Sciences, Professor, yaroslav.dorohyi@donntu.edu.ua

The paper examines the semantic gap in business logic recovery of polyglot enterprise software systems. It is shown that classical architecture recovery methods reconstruct structural dependencies, but do not sufficiently explain domain semantics. The need to distinguish technical structure, configuration dependencies, runtime binding and business meaning is substantiated.

Keywords: Business Logic Recovery, Architecture Recovery, Semantic Gap, Enterprise Systems, Program Comprehension, Static Analysis, Polyglot Systems.

Сучасні корпоративні програмні системи рідко характеризуються простою й однорідною структурую. Їхня

бізнес-логіка не зосереджується в одному класі, модулі чи сервісі, а розподіляється між імперативним кодом, конфігураційними файлами, SQL-запитами, API-контрактами, ORM-мапінгами, зовнішніми інтеграціями та механізмами виконання фреймворків. Через це відновлення архітектури такої системи не може зводитися лише до побудови графів викликів або залежностей. Потрібно встановити не тільки те, які елементи пов’язані між собою, а й те, яке бізнес-правило стоїть за цією взаємодією.

Класичні методи Architecture Recovery (AST, графи викликів, потоків та залежностей) створюють формальну основу для аналізу, дозволяючи, за визначенням S. Ducasse та D. Pollet, реконструювати високорівневу модель системи на основі наявних артефактів [1]. Проте технічна відтворюваність таких методів не забезпечує автоматичного розуміння бізнес-змісту, оскільки структурні графи фіксують лише факти технічної взаємодії, не розрізняючи доменні та суто інженерні зв'язки. Формальний опис коду через AST чи CFG не дає відповіді на питання щодо реалізації конкретних понять предметної області, що зумовлює семантичний розрив між структурою програми та її прикладним наповненням у задачах Business Logic Recovery.

Для поліглотних Enterprise-систем ця проблема є особливо помітною. Один бізнес-сценарій може бути розбитий між сервісним методом, SQL-запитом, YAML- або XML-конфігурацією, API-контрактом і зовнішнім сервісом. Окремо кожен із цих елементів може виглядати як звичайний технічний артефакт, але разом вони утворюють бізнес-операцію. D. Perry та A. Wolf підкреслювали, що архітектура програмного забезпечення не зводиться лише до набору елементів і зв’язків, а також охоплює форму та обґрунтування проєктних рішень [2].

Додаткову складність створюють DI-контейнери, IoC-механізми, AOP-проксі, reflection, dynamic dispatch і runtime-зв’язування. Частина залежностей у таких системах формується не через прямі виклики в коді, а через конфігурації, анотації або правила фреймворку. Тому call graph може бути формально коректним, але неповним щодо фактичної поведінки системи.

Окрему роль відіграють розширені графові подання коду.

Code Property Graph поєднує абстрактне синтаксичне дерево, граф потоку керування та граф залежностей даних в одній моделі, що підвищує повноту технічного аналізу [3]. Проте навіть таке подання залишається переважно структурним, показуючи зв'язки між синтаксичними конструкціями, потоками керування й даних, але не визначаючи автоматично бізнес-сенс певного підграфа.

Ще однією проблемою є інфраструктурний шум. У великих корпоративних системах значну частину кодової бази становлять логування, кешування, транзакційність, авторизація, серіалізація, ORM-мапінги, адаптери й framework-код. Для статичного аналізатора все це є елементами системи, але для відновлення бізнес-логіки вони мають різну вагу. Якщо не відокремлювати доменні правила від технічного супроводу, отримана модель буде перевантажена деталями й менш придатна для розуміння змісту.

Проблема семантичного розриву пов'язана з напрямом Program Comprehension. Під час супроводу системи розробник формує ментальну модель того, як система функціонує. A. von Mayrhauser та A. Vans розглядають розуміння програми як процес формування такої ментальної моделі під час супроводу й еволюції програмного забезпечення [4].

Одним зі способів зменшення семантичного розриву є доповнення структурного аналізу засобами семантичного аналізу програмних артефактів. M. Allamanis та співавтори показують, що програмний код має властивості "природності", які роблять можливим застосування моделей машинного навчання до великих кодових баз [5]. Великі мовні моделі також можуть бути корисними, оскільки здатні інтерпретувати код, документацію, API-описи й конфігураційні фрагменти у природномовній формі. X. Hou та співавтори систематизують такі застосування LLM у програмній інженерії [6].

Таким чином, семантичний розрив у задачах відновлення бізнес-логіки полягає у складності переходу від технічної структури системи до її доменного змісту. Класичні методи Architecture Recovery є необхідними, але недостатніми. Їх потрібно доповнювати семантичним аналізом, типізацією зв'язків, трасуванням джерел і відокремленням бізнес-логіки від

інфраструктурного шуму. Подальші дослідження доцільно спрямувати на створення моделей, які поєднують структурні, конфігураційні та семантичні зв'язки в єдиному поданні й фіксують не лише факт залежності між артефактами, а й її тип, джерело походження, рівень достовірності та зв'язок із конкретним бізнес-сценарієм.

Список використаних джерел

1. Ducasse S., Pollet D. Software Architecture Reconstruction: A Process-Oriented Taxonomy // IEEE Transactions on Software Engineering. 2009. Vol. 35, № 4. P. 573–591. DOI: 10.1109/TSE.2009.19
2. Perry D. E., Wolf A. L. Foundations for the Study of Software Architecture // ACM SIGSOFT Software Engineering Notes. 1992. Vol. 17, № 4. P. 40–52. DOI: 10.1145/141874.141884
3. Yamaguchi F., Golde N., Arp D., Rieck K. Modeling and Discovering Vulnerabilities with Code Property Graphs // IEEE Symposium on Security and Privacy. 2014. P. 590–604. DOI: 10.1109/SP.2014.44
4. von Mayrhauser A., Vans A. M. Program Comprehension During Software Maintenance and Evolution // Computer. 1995. Vol. 28, № 8. P. 44–55. DOI: 10.1109/2.402076
5. Allamanis M., Barr E. T., Devanbu P., Sutton C. A. A Survey of Machine Learning for Big Code and Naturalness // ACM Computing Surveys. 2018. Vol. 51, № 4. Article 81. DOI: 10.1145/3212695
6. Hou X., Zhao Y., Liu Y., Yang Z., Wang K. et al. Large Language Models for Software Engineering: A Systematic Literature Review // ACM Transactions on Software Engineering and Methodology. 2024. Vol. 33, № 8. Article 220. DOI: 10.1145/3695988