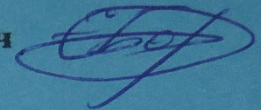


**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
БУДІВНИЦТВА І АРХІТЕКТУРИ**

БОРОДАВКА Євгеній Володимирович



УДК 004.021: 004

**МЕТОДОЛОГІЯ СТВОРЕННЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ
АВТОМАТИЗАЦІЇ ЖИТТЄВОГО ЦИКЛУ БУДІВЕЛЬНИХ ОБ'ЄКТІВ**

Спеціальність 05.13.06 – інформаційні технології

Автореферат
дисертації на здобуття наукового ступеня
доктора технічних наук

Київ-2017

На правах рукопису.

Робота виконана на кафедрі інформаційних технологій проектування та прикладної математики Київського національного університету будівництва і архітектури Міністерства освіти і науки України.

Науковий консультант доктор технічних наук, професор,
Цюцюра Світлана Володимирівна, завідувач кафедри інформаційних технологій Київського національного університету будівництва і архітектури, МОН України, м. Київ.

Офіційні опоненти: доктор технічних наук, старший науковий співробітник,
Барабаш Марія Сергіївна, професор кафедри комп'ютерних технологій будівництва Національного авіаційного університету, МОН України, м. Київ;

доктор технічних наук, професор,
Лантух-Лященко Альберт Іванович, професор кафедри мостів і тунелів Національного транспортного університету, МОН України, м. Київ;

доктор технічних наук, професор,
Зачко Олег Богданович, професор кафедри управління проектами, інформаційних технологій та телекомунікацій Львівського державного університету безпеки життєдіяльності ДСНС України, м. Львів.

Захист дисертації відбудеться «07» квітня 2017 року об 11⁰⁰ годині на засіданні спеціалізованої вченої ради Д 26.056.01 у Київському національному університеті будівництва і архітектури Міністерства освіти і науки України за адресою: 03680, м. Київ, Повітрофлотський просп., 31, ауд. 366.

З дисертацією можна ознайомитися в бібліотеці Київського національного університету будівництва і архітектури за адресою: 03680, м. Київ, Повітрофлотський просп., 31

Автореферат розісланий «06» березня 2017 р.

Вчений секретар
спеціалізованої вченої ради,
кандидат технічних наук, доцент

М.І. Цюцюра

ЗАГАЛЬНА ХАРАКТЕРИСТИКА РОБОТИ

Актуальність теми дослідження. В сучасній світовій економіці спостерігається безперервний невпинний розвиток у всіх галузях. Не стала виключенням і будівельна галузь. Навіть не зважаючи на періодичні кризові явища, вона не стоїть на місці – постійно вдосконалюються будівельні технології, замість старих споруд будуються нові, збільшуються об'єми будівництва житлового фонду, оскільки населення планети постійно зростає. Разом з постійним розвитком будівельної галузі зростають і вимоги до споруджуваних об'єктів – вони повинні споруджуватися максимально швидко і ефективно. А це, в свою чергу, неможливе без використання інформаційних технологій в галузі автоматизації життєвого циклу будівельних об'єктів. Життєвий цикл будівельного об'єкта складається з декількох етапів, але найбільш об'ємними, з точки зору використовуваних ресурсів, є етапи проектування будівельного об'єкта та управління будівництвом будівельного об'єкта. На першому з них використовуються системи автоматизованого проектування (САПР), а на другому автоматизовані системи управління (АСУ).

Поняття САПР та АСУ досить широкі і охоплюють багато галузей економіки. Більшість сучасних друкованих джерел розглядають САПР та АСУ на рівні «глобальних абстракцій» і описують їх в контексті галузевої прив'язки із загальними вимогами до них і відповідною класифікацією. Фактично за останні 50 років в теоретичних викладах, що пов'язані з САПР та АСУ особливих змін не відбувається. Розглядаються все ті ж принципи побудови САПР та АСУ, і все та ж класифікація. Звичайно, ці фундаментальні речі лежать в основі САПР та АСУ з самого їх зародження, але темпи розвитку інформаційних технологій в сучасному світі найбільші серед усіх інших галузей економіки. Тому необхідно постійно вдосконалювати теоретичну базу, що використовується при побудові САПР та АСУ в усіх галузях виробництва.

За останні роки з'явилося досить багато робіт, що пов'язані з розробкою САПР та АСУ в будівництві. Всі вони є важливим науковим надбанням і зайняли достойне місце в справі розвитку будівельних САПР та АСУ. Метою даної роботи є не просто розробка чергової будівельної САПР чи АСУ, а створення теоретичного підґрунтя та опис методології побудови універсальних розширюваних інформаційних систем на основі життєвого циклу будівельного об'єкта. В цій роботі пропонуються моделі подання інформації про будівельний об'єкт, методи перетворення і використання цієї інформації, основні алгоритми, що необхідні для реалізації інформаційної технології автоматизації життєвого циклу будівельного об'єкта, з урахуванням нових вимог динаміки розвитку економіки та будівельної галузі в тому числі.

В сучасних західних інформаційних технологіях у будівництві з'явилося поняття ВІМ – інформаційне моделювання будівель. Фактично ВІМ це частковий випадок управління життєвим циклом продукту (PLM) у будівництві. На

сьогоднішній день в Україні не достатньо розвинені методологічні основи, що описують не лише САПР та/або АСУ для вирішення певних задач в проектуванні будівельних об'єктів, а надають певний спектр моделей, методів, засобів та інструментів їх створення для розробки інформаційних технологій автоматизації життєвого циклу будівельних об'єктів. Тому сформульована і розв'язана в даній роботі науково-технічна задача розробки методології створення інформаційної технології автоматизації життєвого циклу будівельних об'єктів (ІТАЖЦБО) є **актуальною**.

Методологічною основою для обраного напрямку дослідження є праці вітчизняних та зарубіжних вчених. Великий вклад в розвиток теоретичного базису розробки САПР та АСУ в Україні зробили: Городецький О.С., Барабаш М.С., Демченко В.В., Кислоокій В.Н., Козачевський А.І., Лантух-Лященко А.І., Лященко А.А., Сазонов К.А., Слободян Я.О.,Цюцюра С.В., Яловець А.Л.та інші. В їх роботах сформульовані наукові підходи, моделі та методи, що дозволяють реалізовувати необхідні програмні інструменти для забезпечення автоматизованого проектування будівель і споруд різного призначення та управління процесом їх будівництва. В основному, всі праці відомих учених направлені на вирішення конкретних задач, що виникають під час проектування будівельних об'єктів на окремих етапах їх життєвого циклу. Такий підхід дозволяє створити ефективні системи автоматизованого проектування та управління, але для забезпечення повної підтримки всіх етапів життєвого циклу будівельного об'єкта виникає необхідність в інтеграції цих окремих систем в одну ІТАЖЦБО, що є досить складним завданням.

Дисертація спрямована на вирішення актуальної науково-технічної задачі розробки методології створення уніфікованих інформаційних технологій автоматизованого проектування будівельних об'єктів з урахуванням особливостей їх моделей подання інформації на різних етапах життєвого циклу, а також програмної реалізації запропонованої методології та оцінки ефективності її використання.

Зв'язок роботи з науковими програмами і планами. Робота виконувалася в Київському національному університеті будівництва і архітектури в рамках наукових держбюджетних тем: «Впровадження сучасних інтернет-технологій в роботу органів виконавчої влади. Розроблення та створення системи "Прозорийбюджет"» (номер державної реєстрації – 0111U007013); «Система проактивногоуправління якістю навчального процесу» (№ державної реєстрації0114U004596).

Мета і задачі дослідження. Метою дисертаційної роботи є розробка методології створення уніфікованих інформаційних технологій автоматизованого проектування будівельних об'єктів на основі дослідження та розробки моделей подання інформації про будівельний об'єкт на всіх етапах його життєвого циклу, а також методів побудови підсистем комп'ютерної графіки.

Основними задачами дослідження є:

- дослідження етапів життєвого циклу будівельного об'єкта та моделей подання інформації про будівлю на кожному з етапів;
- дослідження сучасних засобів САПР та їх класифікація для визначення етапів життєвого циклу будівлі, для яких виконується автоматизоване проектування, а також моделей якими вони оперують;
- визначення основних вимог до уніфікованих розширюваних інформаційних технологій автоматизації, що охоплюють увесь життєвий цикл будівельного об'єкта;
- аналіз та класифікація типів будівельних елементів та способів їх подання на кожному з етапів життєвого циклу будівлі;
- визначення базового набору графічних примітивів та атрибутів для опису ядра моделі будівельного об'єкта на всіх етапах його життєвого циклу;
- розробка методів розширення ядра моделі будівельного об'єкта на основі модульного принципу;
- створення концептуальної моделі структури даних для опису будівельного об'єкта на всіх етапах його життєвого циклу;
- дослідження та аналіз основних методів оптимізації структур збереження даних;
- розробка методу бінарного пакування даних для зменшення об'єму структури даних та пришвидшення обробки інформації;
- створення моделі інваріантної інформації та набору метаданих для розширювання структури даних про будівельний об'єкт;
- дослідження та аналіз основних методів просторової індексації об'єктів для визначення оптимальної структури організації графічних даних;
- розробка модифікованого R-дерева для оптимального використання ієрархії графічних об'єктів, що моделюють будівлю;
- дослідження основних методів побудови підсистем комп'ютерної графіки та виділення найбільш уніфікованих підходів;
- проектування та розробка прототипу інформаційної технології автоматизації життєвого циклу будівельного об'єкта з використанням розробленої методології;
- оцінка ефективності розробленого прототипу інформаційної технології за пропонованою методологією.

Об'єкт дослідження: процес створення інформаційних технологій автоматизації основних етапів життєвого циклу будівельних об'єктів.

Предмет дослідження: сучасні засоби автоматизації проектувальних робіт в будівництві та сучасні засоби автоматизації управління будівництвом об'єктів, а також їх моделі подання даних, елементи, об'єкти та поняття, якими оперують ці засоби, а також методи розробки цих засобів.

Методи дослідження. Під час проведення досліджень в дисертаційній роботі використані: онтологічні методи дослідження предметної області для визначення ієрархії елементів та сутностей будівельних об'єктів; еволюційний

метод пошуку бази розширюваних систем для визначення множини елементів базових моделей будівлі на різних етапах її життєвого циклу; теорія графів для реалізації ієрархічних зв'язків елементів будівель; теорія множин для визначення ступеня спорідненості окремих моделей будівельного об'єкта; методи об'єктно-орієнтованого моделювання з використанням уніфікованої мови моделювання (UML) для створення базових об'єктно-орієнтованих розширюваних моделей, що описують будівельний об'єкт.

Наукова новизна одержаних результатів полягає у розробці методології створення ефективної інформаційної технології автоматизації життєвого циклу будівельних об'єктів на основі моделей, методів та засобів. *Вперше*:

- запропоновано класифікацію засобів автоматизації, що використовуються в процесі будівництва;
- створено узагальнену модель засобів автоматизації будівельних об'єктів;
- розроблено модель уніфікованої розширеної інформаційної технології автоматизації життєвого циклу будівельного об'єкта;
- створено ядро моделі будівельного об'єкта на основі базового набору графічних примітивів та атрибутів;
- запропоновано метод бінарного пакування інформації для оптимізації формування і використання баз даних;
- створено модифікацію R-дерева для оптимізації індексації об'єктів.

Отримали подальший розвиток:

- методи відображення об'єктно-орієнтованих моделей на реляційні бази даних;
- класифікація зв'язків між сутностями будівельного об'єкта.

Теоретична цінність. Розроблена методологія створення інформаційної технології автоматизації життєвого циклу будівельних об'єктів є узагальнюючою роботою, що об'єднала багаторічні попередні дослідження в галузі розробки систем автоматизованого проектування будівельних об'єктів на основних етапах їх життєвого циклу та систем управління їх будівництвом, які доповнилися додатковими дослідженнями окремих етапів життєвого циклу будівлі. Одержані наукові результати формують методологію створення інформаційної технології автоматизації життєвого циклу будівельних об'єктів і дозволяють вирішити основні та допоміжні задачі, що постають в процесі розробки, а також вказують найбільш ефективні методи та моделі подання інформації про будівельний об'єкт як в процесі проектування, так і при виконанні будівельних робіт.

Практичне значення одержаних результатів. На основі розробленої методології, методів та моделей ІТАЖЦБО розроблюються ефективні системи та засоби для забезпечення автоматизованого проектування, управління та супроводження будівництва конкретних об'єктів. Оскільки методологія універсальна, то результати роботи можуть бути адаптовані в різних сферах промислового і цивільного будівництва: промислових споруд, заводів, машинобудування, фабрик, цехів в різних галузях промисловості.

Проведені дослідження та розроблена в дисертації методологія була використана в процесі розробки інтегрованих систем проектування та управління в будівництві: зокрема розроблена методологія створення ІТАЖЦБО отримала застосування в програмних розробках ТОВ «АДА», а також використовується в навчальному процесі КНУБА, що підтверджено довідками актами впровадження.

Апробація результатів дисертації. Основні результати роботи доповідалися, обговорювалися й одержали позитивну оцінку на: ХІХ науково-практичній конференції «Інформаційні технології в економіці, менеджменті і бізнесі. Проблеми науки, практики і освіти» («Європейський університет», травень 2013, м. Київ), міжнародній науково-практичній конференції «САПР Allplan. Інноваційне проектування в архітектурі і будівництві» (КНУБА, лютий 2014), міжнародній науково-практичній конференції WORLD Science «Science and Education – Our Future» (Абу-Дабі, лютий 2014), міжнародній науково-практичній конференції WORLD Science «Scientific and Practical Results in 2014. Prospects for Their Development» (Дубай, грудень 2014).

Особистий внесок здобувача. Наукові положення, розробки та висновки дисертаційної роботи є результатом самостійно проведених автором досліджень у галузі інформаційних технологій; в галузі систем автоматизованого проектування та управління будівництвом в період з 2009 р. по 2016 р. Особистий внесок підтверджується науковими публікаціями, в яких викладені основні результати дисертаційного дослідження.

Основні результати за темою дисертації були опубліковані у 8 одноосібних працях. Авторський внесок у роботах із співавторами, розкритий у списку опублікованих праць за темою дисертації.

Публікації. Основні результати дисертаційної роботи опубліковані в 25 друкованих працях, з них – 21 статей у фахових виданнях, серед яких, 8 – іноземних публікацій; 3 – тез доповідей на міжнародних конференціях; 1 – науково-навчальний посібник.

Структура і обсяг роботи. Дисертація складається зі вступу, п'яти розділів, основних висновків, списку використаних літературних джерел та додатків. Загальний обсяг дисертації складає 295 сторінок, у тому числі: 111 рисунків, 25 таблиць, використаних літературних джерел із 171 найменування на 16 сторінках і 5 додатків на 33 сторінках.

ОСНОВНИЙ ЗМІСТ РОБОТИ

У **вступі** дається короткий виклад історичних передумов наукового і практичного розвитку інформаційних технологій в будівництві. В дисертації формулюються цілі роботи, обґрунтована актуальність, наукова новизна та практична цінність роботи, подана її загальна характеристика, проведено аналіз

поточного стану розглянутих проблем, як на національному, так і на світовому рівнях.

Перший розділ містить аналіз принципів побудови існуючих архітектурно-будівельних САПР, їх класифікацію, способи подання моделі будівельного об'єкта та їх порівняння.

Існує декілька класифікацій систем автоматизованого проектування за різними критеріями вузьконаправленого використання, цільового використання та узагальнюючого класичного напрямлення.

1. За рівнем формалізації вирішуваних задач: системи побудовані на повністю формалізованих методах вирішення проектних задач, що проводять проектні роботи, які не піддаються повній формалізації; системи, коли для вирішення неформалізованих задач розроблюються додаткові прийоми; системи, що організують пошук вирішення неформалізованих задач.

2. За функціональним призначенням: системи розрахунково-оптимізаційні; графічні та графоаналітичні системи; системи автоматизованого проектування конструкцій; системи підготовки технічної документації; системи обробки результатів експериментальних досліджень; інформаційні системи; системи технологічної підготовки програм для станків з числовим програмним управлінням.

3. За спеціалізацією: спеціалізовані системи та інваріантні системи.

4. За технічною організацією: системи з центральним процесорним керуванням; системи з автоматизованими робочими місцями конструктора (АРМ); системи з власними обчислювальними ресурсами.

У вітчизняній науці та літературі поняття САПР досить широке і охоплює великий спектр задач в різних галузях. В зарубіжній літературі застосовуються більш вузькоспеціалізовані поняття, які в останні роки все частіше проникають і в нашу літературу. Зараз поширений англomовний загальний термін, що позначає різноманітні технології автоматизації за допомогою комп'ютера САх – ComputerAided. В свою чергу технології САх поділяються за функціональними напрямками. Серед них виокремлюють три основні складові:

– ComputerAidedDesign (CAD) – автоматизоване проектування. Термін використовується для позначення широкого спектру комп'ютерних інструментів, які допомагають інженерам, архітекторам та іншим професіоналам у здійсненні проектування. Засферами застосування CAD традиційно поділяються на: архітектурно-будівельні (ArchitectureEngineeringandConstruction CAD – AEC CAD); механічні (Mechanical CAD – MCAD); електронні (Electronic CAD – ECAD або ElectronicDesignAutomation – EDA); технологічні (Computer-Aided ProcessPlanning – CAPP).

– ComputerAidedEngineering (CAE) – автоматизоване конструювання. Використання спеціального програмного забезпечення для проведення інженерного аналізу міцності та інших технічних характеристик компонент і збірок, виконаних в системах автоматизованого проектування (CAD).

– ComputerAidedManufacturing (CAM) – автоматизоване виробництво. Термін використовується для позначення програмного забезпечення, основною метою якого є створення програм для керування верстатами з ЧПУ (ComputerizedNumericalControl – CNC).

З усіх наведених класифікацій, найбільш точною і зрозумілою, на наш погляд, є класифікація, що прийшла до нас із зарубіжної літератури, яка розділяє всі САПР за напрямками, що робить її досить зручною та зрозумілою. Для більшої зручності та розуміння ієрархії систем САХ пропонується їх класифікацію подати у вигляді схеми (рис. 1).

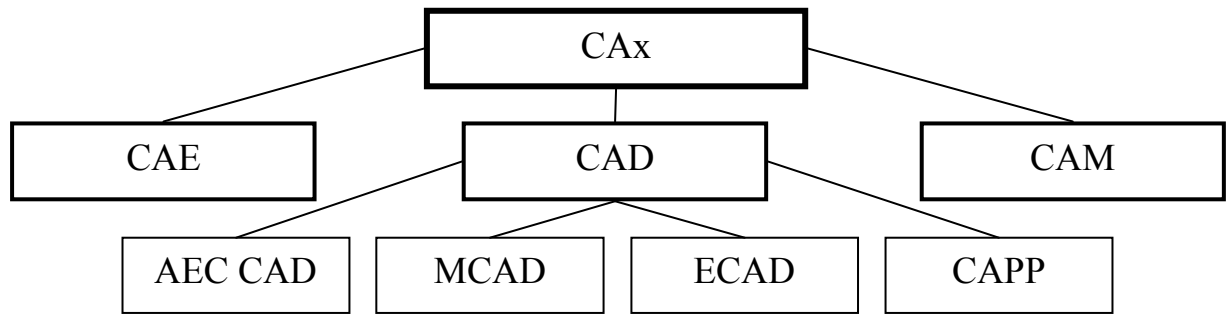


Рис. 1. Класифікація САХ-систем

Звичайно це далеко не повний перелік всіх напрямків САХ-систем, декомпозицію можна продовжувати і далі, але нас цікавить лише напрямок архітектура та будівництво. САХ-системи цього напрямку в подальшому будемо називати просто АЕС САД або архітектурно-будівельні САПР, щоб уникати неоднозначності поняття САПР, яке в цілому набагато ширше, як уже зазначалося вище.

Для початку спробуємо вдосконалити класифікацію саме АЕС САД, причому проведемо це різними методами. Проведення класифікації важлива частина створення методології побудови систем проектування будівельних об'єктів, оскільки дозволяє формалізувати задачі, які стоять перед розробниками конкретних програмних засобів для всього життєвого циклу будівельних об'єктів.

Класифікацію АЕС САД пропонується виконати у вигляді ієрархічної структури, де на кожному рівні буде здійснюватися декомпозиція за функціональними напрямками. Пропонується наступна класифікація АЕС САД за типами будівельних об'єктів:

1. АЕС САД промислових будівель.
2. АЕС САД цивільних будинків.
3. АЕС САД специфічних споруд (мости, дороги, тунелі і т.д.).

Далі пропонується всі АЕС САД класифікувати за напрямками автоматизації проектувальних робіт і прив'язати до типів будівельних об'єктів.

В наведених класифікаціях були розглянуті лише АЕС САД відокремлено від всього життєвого циклу будівельного об'єкта. Але в сучасній науці і техніці САХ-системи розглядаються в контексті системи управління життєвим циклом продукту (PLM). Тому при розробці будь-якої системи автоматизованого

проектування будівельних об'єктів необхідно враховувати можливість її включення в PLM цих об'єктів. А для цього необхідно розглянути всі складові життєвого циклу будівельного об'єкта (проектування, будівництво, експлуатація, утилізація), виокремити та деталізувати ті з них, в яких можуть застосовуватися САх-системи, програмні засоби та інформаційні технології. На сучасному етапі розвитку будівництва, найбільш автоматизовані стадії життєвого циклу будівельного об'єкта це проектування та управління будівництвом, а також частково стадія експлуатації. Якщо виокремити та деталізувати лише їх, то ми отримаємо наступну схему (рис. 2).

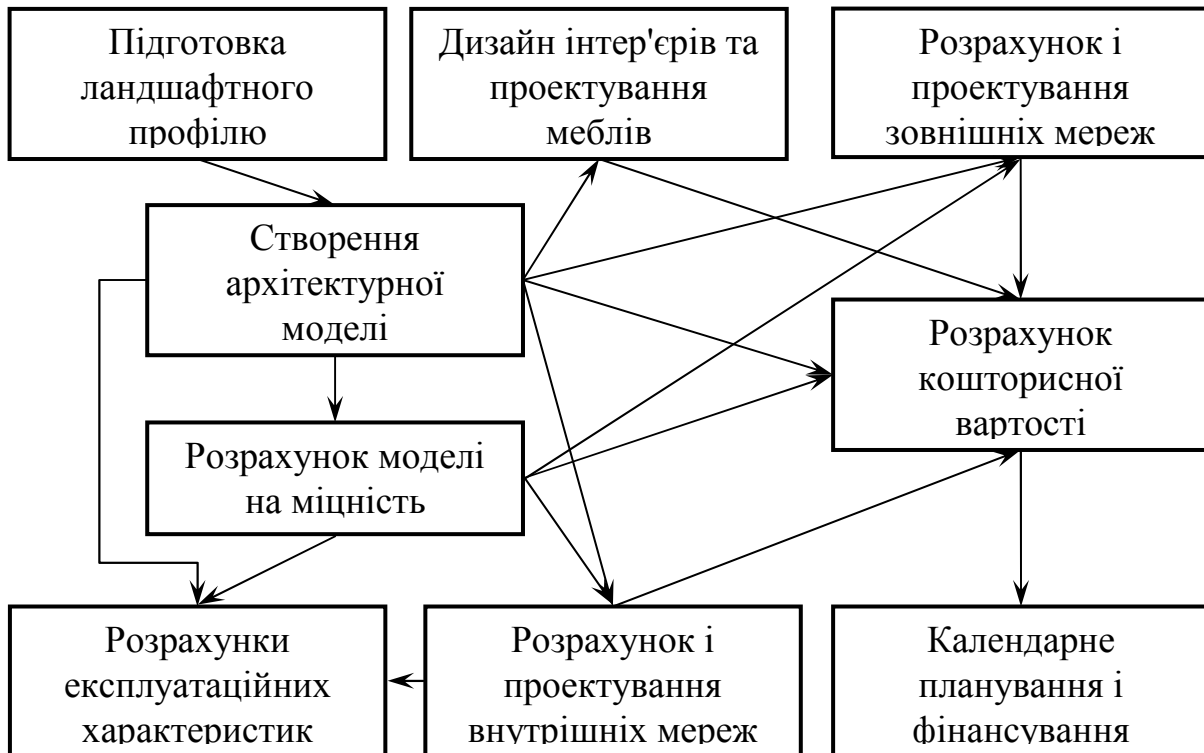


Рис. 2. Деталізований фрагмент життєвого циклу будівельного об'єкта

Як бачимо зі схеми (рис. 2), самих лише АЕС САД недостатньо для автоматизації всіх складових життєвого циклу будівельного об'єкта. Класифікацію АЕС САД потрібно розширити за рахунок її доповнення компонентами автоматизованих систем управління (АСУ) та програмних засобів (ПЗ) розрахунків експлуатаційних характеристик (освітлення, шумоізоляція, рівень забруднення навколишнього середовища і т. п.) будівельного об'єкта. Всі наведені компоненти підпадають під загальну назву САх, оскільки використовують автоматизацію з використанням комп'ютера. Розгляд цих компонент дозволить виявити спільні риси для всіх САх-систем, що використовуються при створенні будівельного об'єкта. А це досить важливо при розробці методології створення уніфікованих розширюваних систем автоматизації життєвого циклу будівельних об'єктів, оскільки дозволить виявити і врахувати напрямки розширення та інтеграції АЕС САД з іншими компонентами, що використовуються в PLM.

В результаті проведеного аналізу робимо *висновок*: дослідження і класифікація лише АЕС САД не дає повного уявлення про інформацію та види її подання для використання в PLM будівельних об'єктів. Тому поряд з АЕС САД необхідно розглянути і розробити нові АСУ та спеціалізовані ПЗ для використання їх в системі PLM.

Також в першому розділі розглянуті способи подання будівельного об'єкта на різних етапах його життєвого циклу (рис. 3).

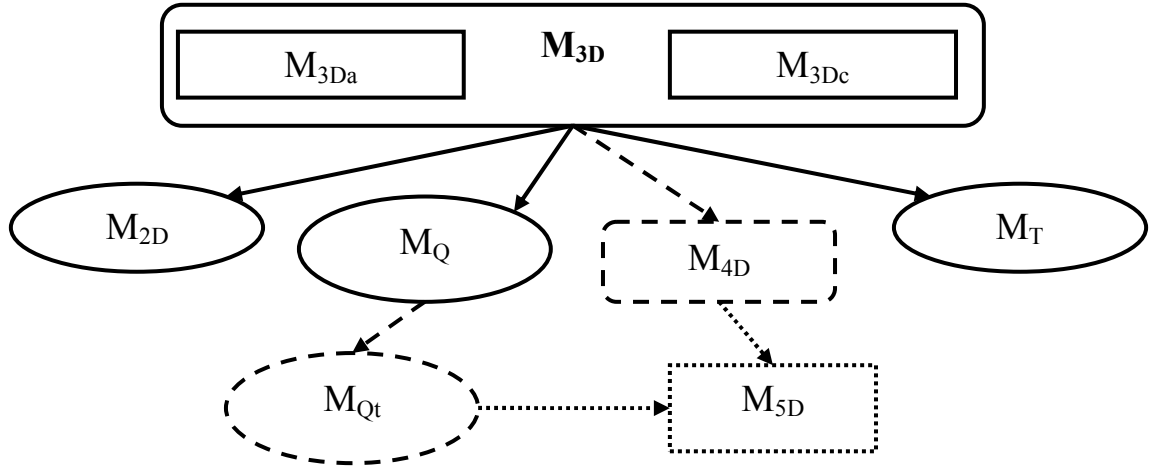


Рис. 3. Типи моделей будівельного об'єкта та їх взаємозв'язки

Далі проведемо класифікацію інформації, якою оперують розглянуті типи моделей. Для виокремлення типів інформації скористаємось методом *декомпозиції*. Тобто кожен із розглянутих типів моделей спробуємо розкласти на типи інформації, якими вони оперують.

Тривимірна модель (M_{3D}) перш за все містить *геометричну* інформацію (I_G) про елементи будівельного об'єкта. Це інформація про розміщення елементів у просторі, їх форму та розмірні характеристики:

$$I_G = \left\{ \sum_k \left[O_k(X, Y, Z, W), S_k \left(\sum_n G_{k,n} \left[\sum_i (x_i, y_i, z_i, w_i) \right] \right) \right] \right\}, \quad (1)$$

де O_k – точка прив'язки кожного k -го елемента в глобальних однорідних ($W \neq 1$) або звичайних координатах ($W = 1$);

S_k – геометрична форма кожного k -го елемента тривимірної моделі;

$G_{k,n}$ – множина графічних примітивів з яких складається графічна форма кожного k -го елемента тривимірної моделі.

Наступний тип інформації, який обов'язково повинен бути присутнім в тривимірній моделі це *топологічна* інформація (I_T). Ця інформація вказує про геометричні взаємозв'язки між елементами – які елементи зв'язані між собою, які є дочірніми для інших чи вирізають в них отвори і т.п.:

$$I_T = \left\{ \bigcup_i \bigcup_j [e_i(\cup | \infty | \cap) e_j]; e_i, e_j \in E, i \neq j \right\} \quad (2)$$

де E – множина всіх елементів тривимірної моделі будівельного об'єкта.

В цілому, описаних вище типів інформації достатньо для побудови базової тривимірної моделі будівельного об'єкта. Проте, архітектурно-будівельні САПР найчастіше використовують або архітектурне або розрахункове розширення тривимірної моделі. Тому для повноти моделі необхідно використати ще один тип інформації – *атрибутивну* (I_A). Атрибутами може бути будь яка інформація: числова, текстова, графічна тощо. Для будівельного об'єкта і його елементів важливими атрибутами є тип елемента, його матеріал та інші додаткові специфічні атрибути (наприклад розрахункові формули) в залежності від задач, що стоять перед конкретним засобом АЕС САД:

$$I_A = \left\{ \sum_i \left[T_i, M_i, \sum_j P_{i,j} \right] \right\}, \quad (3)$$

де T_i – тип кожного i -го елемента тривимірної моделі будівельного об'єкта;

M_i – матеріал кожного i -го елемента тривимірної моделі об'єкта будівництва (не обов'язковий елемент);

$P_{i,j}$ – j -й специфічний параметр i -го елемента тривимірної моделі будівельного об'єкта.

Для двовимірної моделі (M_{2D}) характерні ті ж самі типи інформації, що і для тривимірної. Тобто головним тут є геометричний тип інформації (I_G), але в спрощеному вигляді. Формула 1 справедлива і для двовимірної моделі, при відсутності четвертої однорідної координати ($W=1$) і при наступних умовах:

$$(Z = 0; X, Y \in R) \vee (X = 0; Z, Y \in R) \vee (Y = 0; Z, X \in R). \quad (4)$$

Як і в тривимірній моделі, в двовимірній теж присутня топологічна інформація (I_T), що описує взаємозв'язки між елементами моделі (2). Також в моделі M_{2D} , як і в моделі M_{3D} , є бажаним присутність атрибутивної інформації (I_A), що наповнює модель більш широким змістом. Тому можемо сказати, що модель M_{2D} оперує тими типами інформації, що і модель M_{3D} навіть дещо в спрощеному вигляді стосовно геометричної інформації.

Тепер розглянемо топологічну модель (M_T). З її назви стає зрозумілим, що основним типом інформації, що тут використовується є топологічна інформація (I_T), оскільки для цієї моделі найважливішим моментом є взаємозв'язки між елементами моделі. Звичайно елементи моделі повинні мати певні характеристики, тому цілком очевидним є необхідність використання в цій моделі атрибутивної інформації (I_A).

Фактично наведених типів інформації цілком достатньо для функціонування топологічної моделі. Але для більшої виразності і зручності використання,

топологічну модель часто використовують в поєднанні з двовимірною моделлю (M_{2D}), а вона за собою «тягне» окрім описаної, ще й геометричну інформацію (I_G). Хоча формально в модель (M_T), геометрична інформація не входить.

Нормативно-кошторисна модель M_Q фактично складається з одного типу інформації – атрибутивної (I_A). Але класичного поняття атрибуту, що описується парою «ім'я – значення», недостатньо для повноцінного опису цієї моделі. Для опису моделі M_Q необхідна інформаційна структура, яка б дозволяла описувати матрицю параметрів. Такий тип інформації пропонується назвати *параметрична* (I_P). Тому фактично модель M_Q складається з одного типу інформації – параметричної, що дозволяє будь-який елемент чи групу елементів будівельного об'єкта описувати матрицею атрибутів, що може розширюватися як у «висоту», так і в «ширину». Тут можна зробити зауваження, що атрибутивний тип інформації (I_A) є підмножиною інформації параметричного типу (I_P):

$$I_A \subseteq I_P. \quad (5)$$

Розширені моделі M_{4D} та M_{Qt} наслідують інформаційний склад своїх базових моделей – M_{3D} та M_Q відповідно. Часові характеристики, що в них використовуються, легко можуть бути віднесені до атрибутивного типу інформації (I_A). Тому ніяких додаткових типів інформації вони не породжують і не використовують.

Таким чином ми визначили інформаційний склад всіх моделей будівельного об'єкта, застосувавши до них декомпозицію. Тепер для побудови класифікації типів інформації використаємо метод *агрегації*, тобто об'єднаємо типи інформації всіх моделей, будівельного об'єкта (I_B):

$$I_B = (I_G, I_T, I_A) \cup (I_G, I_T, I_A) \cup (I_T, I_A) \cup (I_P) = (I_G, I_T, I_A, I_P). \quad (6)$$

В результаті ми отримали чотири типи інформації, яка використовується в усіх поданнях моделі будівельного об'єкта.

Оскільки метою даної роботи є розробка методології створення ІТАЖЦБО, то цілком очевидно, що така інформаційна технологія повинна оперувати всіма наведеними моделями. Тобто модель будівельного об'єкта в ІТАЖЦБО (M_B) це буде об'єднання всіх моделей, що розглянуті вище або їх розширень, що є більш універсальним:

$$M_B = M_{2D} \cup M_{4D} \cup M_T \cup M_{Qt}. \quad (7)$$

Інформація, що циркулює в ІТАЖЦБО, поділяється на чотири типи, що були визначені вище. Але тут необхідно розглянути ще один аспект. Ми розглянули інформацію, як складову частину кожної моделі. Проте будь-який тип інформації можна розділити на два види: інформація, яка знаходиться власне в *конкретній* моделі (I_B) та *інваріантна* (загальна) інформація (I_I), що може бути використана для будь-якої моделі будівельного об'єкта. Наприклад, в окремо взятому будівельному об'єкті використовуються елементи типу A і типу B , а елементи типу C і D не використовуються. Однак вони повинні бути десь описані необхідними

типами інформації. Для ІТАЖЦБО склад інваріантної інформації (I_I) такий же, як і модельної інформації (I_B).

Щоб забезпечити розширюваність ІТАЖЦБО без втручання розробників, необхідно забезпечити можливість формування нових інформаційних об'єктів. Тобто необхідно мати такий тип інформації, що дозволяє породжувати інформаційні об'єкти одного з чотирьох типів, що описані вище. Таку інформацію будемо називати *метаінформація* (I_M), тобто інформацією, що належить до виду інваріантної інформації, оскільки теж є загальною для моделей всіх можливих об'єктів будівництва.

У **другому розділі** розглядається процес створення інформаційного ядра будівельного об'єкта та методів його розширювання. В сучасній будівельній галузі використовується досить широкий набір різноманітних будівельних конструкцій та елементів з різними характеристиками та ознаками. З плином часу цей набір постійно розширюється новими елементами, а існуючі елементи вдосконалюються. З точки зору розробника систем підтримки життєвого циклу будівельних об'єктів важливим є виокремлення основних будівельних елементів і конструкцій та групування їх за схожими ознаками. Це дозволить створити уніфіковані структури даних та класи для опису і роботи з цифровим поданням цих елементів та конструкцій.

Звичайно проаналізувати всі можливі елементи та конструкції до їх появи не можливо. Тому необхідно для початку класифікувати всі відомі та найбільш часто вживані будівельні елементи, і передбачити можливість появи нових елементів, що можуть бути приєднані до однієї з груп класифікації.

Найпростіші будівельні елементи будуються з використанням плоских просторових багатокутників. Для утворення закінченої тривимірної моделі таких будівельних елементів достатньо здійснити переміщення утворюючого багатокутника вздовж деякої кривої. В найпростішому випадку ця крива є відрізком прямої.

Складні будівельні елементи неможливо описати простою плоскою геометричною фігурою. Вони складаються з декількох геометричних примітивів і є комплексними об'єктами. Для побудови тривимірної моделі таких елементів найпростіше використовувати граничні моделі. В ідеальному випадку бажано, щоб це були полігональні моделі з триангульованими гранями.

Для спрощеного опису будівельних об'єктів, які є композицією плоских полігональних моделей введено додаткову абстрактну геометричну ознаку «групові». Вона необхідна, наприклад, для опису ферм. Ферма це складний комплексний елемент, але описувати його через граничну модель не досить зручно. Простіше описати її як групу стержнів, кожен з яких це плоска полігональна модель.

Звичайно лише геометричних характеристик недостатньо для опису будь-якого елемента будівельного об'єкта. Тому потрібно мати додатковий набір параметрів та атрибутів, щоб зберігати специфічну інформацію по кожному елементу.

В сучасних спеціалізованих будівельних САПР базовою ознакою для будь-якого об'єкту моделі є його *тип*. Такий підхід дозволяє досить просто класифікувати та розподіляти об'єкти, що використовуються. Але, в той же час, такий підхід обмежує гнучкість системи. Наприклад, якщо для об'єкта *стіна* визначено, що доступні типи геометричного подання лише ті, які підпадають під тип «плоскі полігональні», то створити стіну як об'єкт з геометрією у вигляді граничної моделі неможливо. Для реалізації такої можливості розробникам необхідно буде здійснювати доробку системи для розширення її функціональності, навіть у випадку коли в системі вже є об'єкти для яких доступне подання у вигляді граничної моделі.

В результаті дослідження встановлено, що для різних типів елементів використовуються як спільні так і відмінні варіанти геометричного подання. Тому, зважаючи на викладені міркування, для уніфікованої розширюваної системи доцільно зробити базовою ознакою саме *геометричний* тип.

Оскільки на різних етапах підтримки життєвого циклу будівлі використовуються різні елементи, що мають різне функціональне призначення, пропонується об'єднати однорідні елементи в класи. Наприклад можуть мати місце такі класи: будівельні об'єкти, геодезичні, електротехнічні, сантехнічні, об'єкти інтер'єру, кошторисно-фінансові і т.д. Оскільки за основу ми вирішили взяти геометричні примітиви, то проаналізуємо, які спільні параметри вони можуть мати. Елементом назвемо об'єкт вищого рівня ієрархії, що може складатися з одного або декількох геометричних примітивів.

Першим обов'язковим параметром для будь-якого елемента повинен бути його унікальний ідентифікатор (*ID*). За допомогою цього ідентифікатора здійснюються всі операції з елементом (модифікація, видалення і т.п.), а також забезпечується його зв'язок з іншими елементами.

Наступним параметром елемента є його тип, який визначає функціональне призначення елемента та правила його обробки.

Будь-який елемент незалежно від геометричної будови та типу повинен мати просторове визначення та топологію, що описує його взаємозв'язки з іншими елементами.

Для багатьох елементів визначення однієї точки прив'язки неможливе зважаючи на їх складну геометрію. Тому для першочергового однозначного визначення просторового розташування елемента пропонується використовувати просторовий обмежувальний паралелепіпед (*spatialboundingbox*, *SBB*). *SBB* описується всього дома просторовими точками, що визначають його головну діагональ. Подібна структура дозволяє описати межі об'єкта з геометрією будь-якої складності.

Топологічні зв'язки бувають різних типів в залежності від специфіки елемента. Пропонується виокремити три типи топологічних зв'язків, що описують всі можливі взаємовідношення між елементами. Перший тип це *включення* – такий тип зв'язку, коли дочірній елемент впливає на базову геометричну

структуру батьківського елемента. Наприклад стіна-вікно, плита-отвір. Другий тип зв'язку це *групування* – такий зв'язок, коли батьківський елемент не має власної геометричної структури, а його повністю складає група дочірніх елементів. Прикладом може слугувати поверх будинку, ферма, багатоскатний дах. Третій тип зв'язку це зв'язок *посилання* – такий зв'язок, коли батьківський елемент має свою геометричну структуру і він зв'язаний з дочірніми елементами, які не включені в нього геометрично, але їх окреме існування не має сенсу. Прикладом може слугувати земля та елементи ландшафту, двері та наличники, вікна та підвіконня, електрообладнання та з'єднувальні провідники. Слід зазначити, що в деяких випадках на один і той же дочірній елемент можуть посилатися різні батьківські елементи, наприклад з'єднувальний провідник має два кінця, тому з'єднує два «батьківські» елементи і кожен з них має посилання на нього.

Останнім спільним параметром для будь-якого елемента пропонується вести *матеріал*. Але багато елементів мають складну структуру з використанням декількох матеріалів, тому параметр матеріал ми використовуємо як атрибут геометричного примітиву.

За результатами наведених міркувань можемо сформуванати інформаційну модель базового елемента, що використовується в процесі створення ІТАЖЦБО(рис. 4).

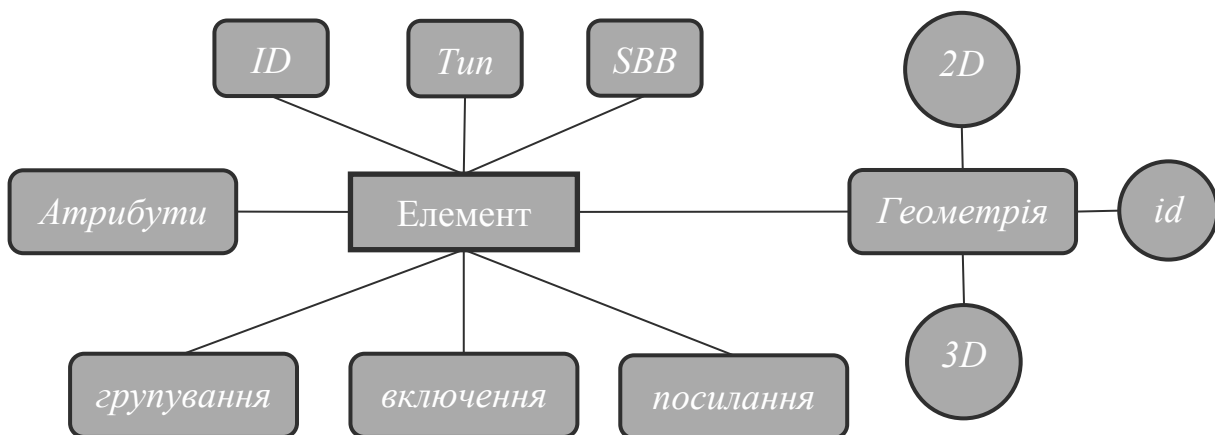


Рис. 4. Інформаційна модель базового елемента

Базовий елемент є ядром моделі будівельного об'єкта. Він може бути описаний будь-яким примітивом, а також має опис топологічних зв'язків:

$$M_E = \langle ID, T, SBB, G(id, 2D, 3D), A, R(g, i, l) \rangle, \quad (8)$$

де ID – ідентифікатор елемента;

T – тип елемента;

SBB – просторовий обмежувальний паралелепіпед;

$G(id, 2D, 3D)$ – геометрія елемента (ідентифікатор, 2D-модель, 3D-модель);

A – атрибути елемента;

$R(g, i, l)$ – відношення елемента (групування, включення, посилання).

В принципі, використання лише базового елемента для побудови моделі будівельного об'єкта цілком достатньо. Але в такому випадку ми отримаємо лише

візуальну форму майбутнього об'єкта, а важливі етапи життєвого циклу будівельного об'єкта не зможуть бути автоматизовані належним чином без додаткової ручної обробки моделі.

У **третьому розділі** пропонується модель розширеної структури збереження даних про будівельний об'єкт та методи її оптимізації.

Будівельний об'єкт має досить складну структуру, і його моделі на різних етапах життєвого циклу доволі кардинально відрізняються. Для створення максимально уніфікованої структури даних, що описуватиме будівлю на всіх етапах життєвого циклу, необхідно знати складові моделі, що подає будівельний об'єкт на цьому етапі. В попередніх розділах був проведений такий аналіз і деякі складові моделі на етапі проектування навіть були розглянуті досить детально. В багатьох складових були виявлені спільні ознаки, що дозволяють об'єднати їх в один клас інформації. Також в попередньому розділі проаналізовано і визначено основні методи опису геометрії складових елементів моделі будівельного об'єкта на різних етапах життєвого циклу та визначені основні графічні примітиви для опису цих елементів.

Складність при розробці уніфікованої структури даних про будівельний об'єкт також полягає в тому, що один і той же фізичний елемент будівлі на різних етапах життєвого циклу може мати різну інтерпретацію в контексті дій, що виконуються над ним. Наприклад найпоширеніший елемент «стіна» на етапі архітектурного проектування інтерпретується як деяка просторова призма з двома паралельними сторонами (хоча і не завжди так) і можливими отворами в ній. На етапі конструктивних розрахунків на міцність будівлі ця ж «стіна» трансформується в плоский полігон з отворами і для неї, окрім геометричного подання та топологічного розташування, важливі такі характеристики як матеріал, з якого вона зроблена та її здатність сприймати навантаження (це несуча стіна чи просто перегородка). На етапі управління будівництвом в більшості випадків не важлива ні геометрія стіни ні її здатність нести навантаження. Тут важливими є кількісні показники – об'єм і площі поверхонь, а також матеріал і розташування (зовнішня чи внутрішня). Також на цьому етапі важливим є час виконання будівельних робіт для зведення цього елемента та кількість робітників і техніки, що необхідно при цьому задіяти. Подібні міркування можна застосувати до будь-якого елемента будівлі. Крім того є ще досить велика кількість елементів не належних безпосередньо до будівельного об'єкта але які мають пряме відношення до нього на різних етапах життєвого циклу: геодезичне планування місця будівництва, профілювання земельної ділянки, риття котловану, прокладання зовнішніх інженерних мереж і т. п. Щоб створити максимально ефективну та уніфіковану структуру даних про будівельний об'єкт необхідно передбачити можливість для опису будь-яких можливих типів інформації та можливість розширення структури даних при мінімальних затратах на переробку програмних засобів.

Ще одним важливим аспектом, про який вже згадувалося в першому розділі, є орієнтація на засоби моделювання будівельного об'єкта. Якщо планується

створення засобів проектування моделей будівлі на різних етапах життєвого циклу, а не просто графічне відображення готових моделей (двовимірних та тривимірних), то в структурі даних повинно бути це враховано, оскільки складність такої структури даних буде значно вищою.

Для початку створимо модель структури даних про будівельний об'єкт на найвищому рівні декомпозиції – тобто розділимо всю інформацію за можливими типами, а потім будемо деталізувати і знаходити можливі місця перетину різних типів інформації. Можливий варіант поділу інформації за її належністю до певного етапу життєвого циклу будівлі не є прийнятною, оскільки на кожному з етапів може використовуватися одна й та ж сама інформація або деяка її частина. Тому поділ даних за типом інформації, що вони несуть, є більш доцільним.

Деякі типи інформації вже розглядалися в першому розділі. Також там описувалися моделі будівельного об'єкта на різних етапах життєвого циклу та типи інформації, якими вони оперують. Враховуючи додаткову інформацію, яка була отримана в процесі дослідження, що було проведене в другому розділі, створена узагальнена концептуальна модель структури даних про будівельний об'єкт на метарівні декомпозиції (рис. 5).

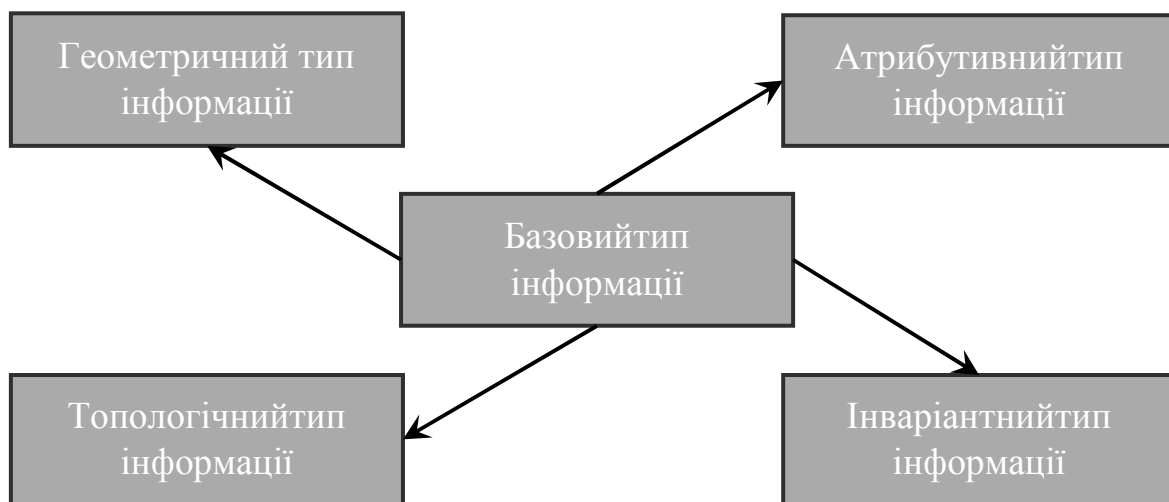


Рис. 5. Концептуальна модель структури даних на метарівні декомпозиції

Першим розглянемо *базовий* тип інформації, оскільки він є сполучною ланкою між іншими типами інформації. По суті це специфічний набір атрибутів, що властиві будь-якому елементу. Ми точно можемо стверджувати, що кожен елемент повинен мати унікальний ідентифікатор, а також базовий тип, про який писалося в попередньому розділі. До базової інформації також віднесемо посилання на різні варіанти геометрії елемента (двовимірна та тривимірна), а також посилання на набір атрибутів, що описують даний елемент.

Найбільш вживаний та об'ємний за кількістю даних *геометричний* тип інформації. Для візуалізації як елементів будівлі, так і супутніх сутностей необхідно мати їх геометричну інтерпретацію.

Наступний тип інформації це *атрибутивний*. Він дозволяє описати якісні та кількісні характеристики будь-якого елемента.

Топологічний тип інформації призначений для опису зв'язків між різними елементами та їх атрибутами.

Інваріантний тип інформації призначений для опису стандартизованої інформації, що може бути використана в якості атрибутивної для різних елементів різних будівельних об'єктів.

Геометричний тип інформації досить складний і має досить велику кількість структур для збереження різноманітних форм подання геометричних об'єктів. Але перш за все варто розділити геометричну інформацію на два види: двовимірну та тривимірну. Один і той же об'єкт може мати як двовимірну так і тривимірну інтерпретацію і, відповідно, кожна із них повинна описуватися окремим набором даних, що взаємопов'язані між собою.

Цілком зрозуміло, що для опису об'єктів в двовимірному просторі достатньо оперувати простішими «плоскими» графічними примітивами. У випадку використання окремих плоских моделей є можливість відразу мати двовимірні структури для опису ізотетичних проекцій, тобто таких, коли елемент розташований паралельно до координатних площин. Такі проекції будуть максимально інформативними і зрозумілими для кінцевого користувача, а також цілком придатними для редагування в інтерактивному режимі.

Для універсальності обробки об'єктів доцільно використати лише одну уніфіковану просторову структуру. Такою структурою може бути триангуляційна модель або будь-яка інша полігональна модель. Кожен тривимірний об'єкт може бути описаний набором полігонів і таким чином зберігатися в структурі даних. Якщо всі тривимірні об'єкти будуть мати набір однакових просторових даних, то їх обробка при побудові в тривимірному просторі буде ідентична. Єдиний суттєвий недолік цього підходу це реалізація можливості здійснювати побудову та редагування об'єктів у тривимірному просторі. Хоча цей недолік можна обійти – для редагування та побудови використовувати спрощені моделі подання тривимірних об'єктів, а для їх фінального рендерінгу та зберігання використовувати уніфіковану структуру. Хоча це призведе до ускладнення структури даних але в деяких випадках такий підхід може мати сенс. Проте очевидно, що єдиного універсального рішення, яке було б оптимальне з усіх боків не існує.

Основні концептуальні моменти моделі структури даних будівельного об'єкта сформульовані. Таким чином, програмна реалізація структури даних буде заснована на об'єктно-орієнтованому підході, що дозволяє провести дослідження методів та засобів адекватного та оптимального відображення об'єктно-орієнтованої концептуальної моделі структури даних на реляційну базу даних, що дає можливість зберігати моделі об'єкта для повторного їх використання.

Слідуючи сучасним тенденціям в програмуванні, система підтримки життєвого циклу будівельного об'єкта повинна бути реалізована об'єктно-

орієнтованою. Це дозволить створити систему стійкою до внесення помилок і простою з точки зору розширювання тарефакторингу.

Оскільки система матиме розгалужену ієрархічну структуру класів, то є сенс автоматизувати відображення класів на структуру даних та максимально її оптимізувати. Процес відображення системи класів на реляційну базу даних називається ObjectRelationalMapping (O/RM), а посередник між об'єктно-орієнтованим середовищем та реляційною базою даних відповідно називається ObjectRelationalMapper (ORM).

Пряме повторення полів класу в таблиці бази даних зазвичай недостатнє для створення адекватного реляційного подання об'єктної моделі. Окрім звичайної інформації, що присутня в класах, необхідно зберігати також інформацію про зв'язки між класами, деяку інформацію про поведінку класу, додаткову інформацію для управління об'єктом (так звана «shadowinformation»), а також метаінформацію у випадку опису складних класів-списків чи інших складних класів.

В простій проекції кожен атрибут класу проектується в одне поле таблиці бази даних, що має такий же тип даних. Звичайно так просто ніколи не буває. Зазвичай типи атрибутів класу і полів таблиці можуть відрізнитися, або один атрибут проектуватися в більше, ніж одне поле. Крім того між діаграмою класів та її проекцією на базу даних є деякі відмінності пов'язані з введенням додаткових полів в таблиці БД, що використовуються в якості внутрішніх та зовнішніх ключів.

В результаті проектування об'єктної моделі в реляційну виникає необхідність її рефакторингу у відповідності до отриманої реляційної моделі. Оскільки в таблицях БД з'явилася додаткова інформація, що потрібна для підтримки роботи ORM, то UML діаграма класів буде приведена у відповідність до реляційної структури даних. Таким чином в об'єктній моделі з'являються нові атрибути і методи в класах, що забезпечить їх максимально просте та ефективно збереження та відновлення з бази даних.

Тепер розглянемо більш складні ієрархічні об'єктні моделі. Реляційні бази даних не підтримують ієрархії автоматично, тому необхідно мати спеціальні підходи для реалізації запису/зчитування ієрархії класів об'єктної моделі з її реляційного аналогу. Для реалізації проектування ієрархічних об'єктних моделей на реляційні БД існує чотири основні підходи:

- проекція всієї ієрархії класів в одну таблицю;
- проекція кожного *конкретного* класу в окрему таблицю;
- проекція кожного класу в окрему таблицю;
- проекція класів в узагальнену структуру таблиць.

Кожен з наведених варіантів має свої переваги та недоліки і може бути застосований в окремих випадках в залежності від обставин.

Жодна з описаних стратегій не є ідеальною для будь-якого випадку в проектуванні. Іноді варто почати проектування використовуючи одну стратегію, а в подальшій роботі виконати рефакторинг до іншої моделі.

Оскільки жодна з моделей не є універсальною, то при виборі стратегії необхідно зважати на предметну область її застосування. В загальному випадку в одному проекті можна використовувати комбінацію цих стратегій для різних частин ієрархії класів.

Структура БД, що наближена до ідеальної, повинна:

- займати якомога найменший об'єм пам'яті;
- надавати швидкий доступ до даних, що в ній зберігаються;
- не допускати марнування простору;
- легко відновлювати всі ієрархічні та інші зв'язки між об'єктами.

Жодна з запропонованих стратегій одночасно не задовольняє всім цим умовам. Деякі стратегії задовольняють одним, деякі іншим умовам. Тому ми можемо застосувати комбінаційний метод, що буде включати різні стратегії для окремих елементів БД, однак це лише фрагментує всю БД на окремі частини, кожна з яких буде близька до ідеальної.

Якщо брати в промислових масштабах проектування, то реальними стратегіями для проектування об'єктної моделі на реляційну є дві: проекція всієї ієрархії класів в одну таблицю та проекція кожного класу в окрему таблицю. Недоліком першої є марнування простору БД особливо при досить розгалуженій ієрархії класів, недоліком другої – занадто велика кількість таблиць в БД, що неминуче веде до збільшення часу доступу до даних. Проте перший підхід нейтралізує недолік другого – час доступу до даних практично миттєвий, а другий в свою чергу нейтралізує недолік першого – ніякого марнування простору БД, оскільки в кожному запису всі поля практично завжди зайняті інформацією. Логічним висновком є бажання схрестити ці два підходи з застосуванням деякої технології, що дозволить нейтралізувати недоліки обох підходів, а залишить лише їх переваги.

Сучасним прогресивним підходом є пакування таблиці в одне бінарне поле. В сучасних СУБД такі поля дозволяють зберігати до 2 Гбайт даних, що цілком достатньо для реалізації запропонованого підходу. За такої стратегії побудови БД, частини таблиці, що знаходиться на стороні відношення «багато», пакується в одне поле і записується в таблицю, що знаходиться на стороні відношення «один». Наприклад, якщо розглянути класичну організацію зберігання ламаної лінії, то ми повинні отримати дві таблиці: точки (Points) та власне ламані (Polylines). В запропонованому методі, частини таблиці «Points», що відповідають окремим ламаним, пакуються в бінарне поле та записуються у відповідні рядки (табл. 1).

За такого підходу, кожен об'єкт повинен «вміти» запакувати себе в бінарний потік, який потім буде об'єднаний з потоками інших об'єктів і записаний в поле БД. Оскільки в будь-якому випадку при об'єктно-реляційному проектуванні класи потребують модифікації для забезпечення їх запису та зчитування з реляційної

БД, то для реалізації бінарного пакування достатньо розширити ці стандартні методи класів.

Використання методу бінарного пакування даних слід застосовувати обережно і особливо слідкувати за суворим дотриманням послідовності зчитування та запису. Оскільки якщо ця послідовність порушиться хоча б на 1 байт, то вся інформація, що буде прочитана після зсуву, буде недостовірною або намагання її прочитати навіть може привести до аварійного завершення програми.

Таблиця 1. Приклад бінарного пакування

POLYLINES			POINTS				
ID	Points		ID_PL	ID	X	Y	Z
1	BLOB	←	2	1	-34	2.4	7.2
2	BLOB		2	2	-1.2	35	4.6
3	BLOB		2	3	0.3	-2.5	2.7
			2	4	4.4	3.5	-4.2
			2	5	0.3	-2.3	-3.7

Розглянутий приклад використання бінарного пакування інформації застосовний до об'єктів, що знаходяться в зв'язку «агрегація». В такому випадку всі агреговані об'єкти зберігаються в полі таблиці, що описує об'єкт агрегатор. Дані в бінарному полі у всіх рядках таблиці мають однакову структуру і зчитуються одними методами. Такий метод бінарного пакування назовемо *агрегованим*, або *простим*.

Якщо застосовувати стратегію бінарного пакування даних для ієрархічної об'єктної моделі, що спроектована на реляційну БД, то потрібно розширити базову таблицю двома полями: перше поле для зберігання кінцевого типу об'єкта, друге – бінарне, для зберігання даних дочірнього об'єкта. В такому випадку, кожен рядок таблиці в бінарному полі може містити різні набори даних, що описують різні кінцеві об'єкти дочірніх класів (рис. 6).

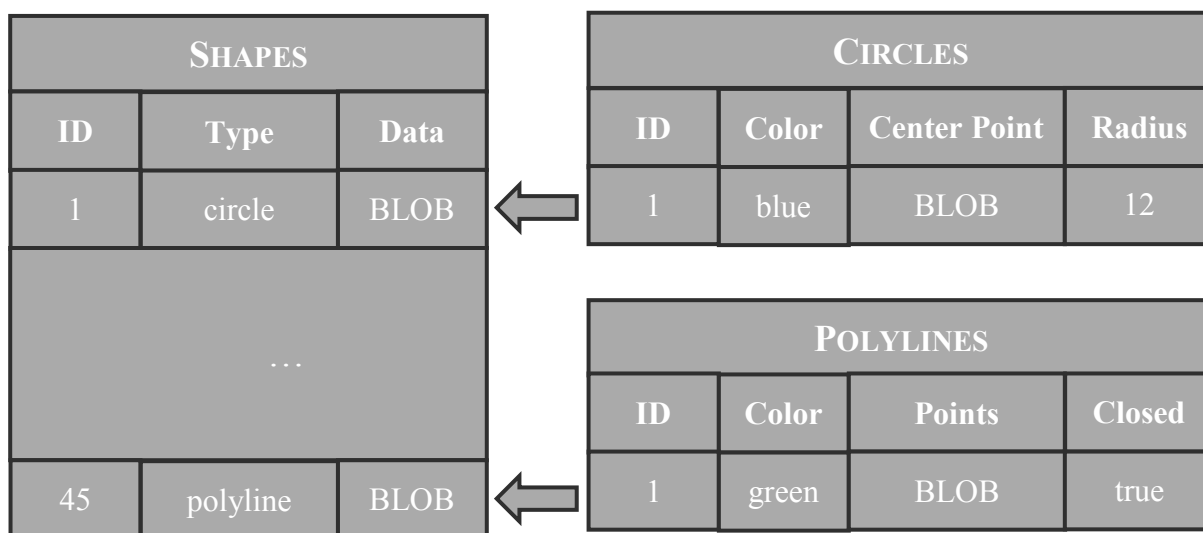


Рис. 6. Принцип бінарного пакування ієрархічної моделі

Як і в попередньому простому методі, кожен клас повинен мати методи, що вміють його пакувати та розпаковувати з бінарного поля. Загальну диспетчеризацію цього процесу покладено на базовий клас, який в залежності від типу кінцевого об'єкта, що записаний у відповідному полі, знає методи якого дочірнього класу необхідно викликати для обробки конкретного кінцевого об'єкта, що зберігається в конкретному рядку таблиці. Такий метод бінарного пакування назвемо *ієрархічним* або *складним*.

В загальному випадку може мати місце доволі глибока вкладеність ієрархічних об'єктів, а також спільне використання агрегації. За такої складної моделі можливі випадки, коли одні бінарні поля включають інші, що в свою чергу включають інші і т. д. Але з використанням спеціальних методів для управління процесом запису та зчитування даних, це буде виконуватися автоматично в процесі створення та знищення об'єктів ієрархії.

Єдиним значним недоліком використання методу бінарного пакування даних є те, що бінарні поля не можуть бути використані в SQL-запитах. Тому якщо інформація планується до використання в SQL-запитах її варто винести в окреме поле з визначеним типом і не пакувати в бінарне.

Перевагами використання методу бінарного пакування даних є:

- зменшення об'єму БД;
- збільшення швидкості доступу до даних;
- спрощення зчитування та запису складних ієрархічних моделей;
- автоматичне шифрування даних, оскільки якщо стороння особа відкриє БД, то без знання структури кожного бінарного поля витягнути з них інформацію неможливо.

Окрім структури даних для збереження моделі будівельного об'єкта необхідно також розробити структуру даних для інваріантної моделі.

В розрізі моделі будівельного об'єкта, набір метаданих в першу чергу призначений для опису нових будівельних елементів, а саме їх геометричного подання, набору необхідних параметрів та атрибутів. Тобто метадані є необхідною і невід'ємною частиною інваріантної інформації. Тому моделі інваріантних даних та метаданих будуть розглядати як одну модель даних.

Інваріантна модель будівельного об'єкта обов'язково повинна містити деяку базову інформацію, що є початковою і незмінною. Для будівельного об'єкта такою інформацією є базові геометричні та графічні примітиви, що дозволяють описувати двовимірне та тривимірне подання будівельних елементів, як основних складових будівлі. Також обов'язковою є параметрична інформація та інформація про базові атрибути, що притаманні складовим елементам будівлі. Фактично перелічена інформація є метаданими на основі яких здійснюється опис будь-яких сутностей інваріантної моделі (рис. 7).



Рис. 7. Модель інваріантної інформації для будівельного об'єкта

На рис. 7 показаний лише один із можливих варіантів інваріантної моделі для будівельного об'єкта. З огляду на завдання, що повинна вирішувати розроблювана система ця модель може мати інші форми. Хоча базові частини моделі будуть присутні в будь-якій реалізації.

Для прикладу розглянемо, як на основі базових атрибутів створити модель будівельного елемента «стіна» на різних етапах життєвого циклу будівельного об'єкта. Для початку необхідно в інваріантну модель інформації додати тип сутності «стіна». Далі, з точки зору проектування, модель стіни необхідно описати геометричними примітивами для двовимірного і тривимірного подання, а також вибрати якими графічними примітивами вона буде відображатися. Якщо йти простим шляхом і вважати, що всі стіни мають бути прямокутними паралелепіпедами, то для двовимірного подання нам необхідно мати прямокутник з можливістю розташування під будь-яким кутом до осей координат, а для тривимірного подання нам потрібен прямокутній паралелепіпед, що буде складатися із шести просторових прямокутників.

Тепер необхідно задати набір атрибутів, що притаманні стіні в якості будівельного елемента. Звичайно це будуть такі атрибути як: матеріал, шар, товщина, об'єм, площа поверхонь і т. п. Якщо ж розглядати стіну з точки зору конструктора, то для неї будуть важливими такі параметри як: несуча стіна чи перегородка, навантаження на стіну, матеріал, товщина і т. п. На етапі кошторисного розрахунку вартості будівлі для стіни важливі такі характеристики: матеріал, товщина, зовнішня чи внутрішня, поверх, на якому зведена стіна, об'єм, площа поверхонь. Ці характеристики необхідні для одного – призначити кошторисні нормативи для побудови цієї стіни. Тому всі вони можуть бути замінені на параметр, що описує стіну з точки зору кошторисного проектування. З точки зору експлуатації будівлі, модель стіни повинна мати характеристики: теплопровідність, вогнестійкість, стійкість до корозії і т. п. Всі зазначені атрибути і параметри можна описати в структурі інваріантних даних і зв'язати з типом будівельного елемента «стіна» і використовувати у всіх подальших моделях будівель.

З написаного вище, стає зрозумілим, що одна і та ж сутність будівлі, на різних етапах її життєвого циклу, має різний набір атрибутів. Цей набір можна змінювати динамічно, якщо він реалізований за допомогою інваріантної структури даних та метаінформації. Зокрема для стіни можна записати наступну математичну модель подання:

$$M_W = I_G \cup \sum_i a_i \cup \sum_j p_j, \quad (9)$$

де M_W – модель стіни;

I_G – графічна інформація про стіну;

a_i – i -й атрибут стіни на всіх етапах життєвого циклу будівлі;

p_j – j -й параметр стіни на всіх етапах життєвого циклу будівлі.

У четвертому розділі розглядаються моделі та методи побудови підсистем двовимірної та тривимірної графіки в розрізі їх застосування під час створення інформаційної технології автоматизації життєвого циклу будівельного об'єкта.

Будь-яка інформаційна технологія автоматизації життєвого циклу будівельного об'єкта обов'язково повинна включати графічні підсистеми – двовимірну та тривимірну. Ці підсистеми створюють найбільш зручне для сприйняття подання будівельного об'єкта: плани, розрізи, схеми конструкцій, просторова модель будівлі та окремих її елементів і т.п.

Оскільки графічні системи доволі вимогливі до апаратних ресурсів, швидкість роботи таких підсистем залежить від кількості графічних примітивів, що використовуються для побудови моделі, а також від ефективності використаних структур даних та алгоритмів рендиру. Тому постає необхідність в розробці ефективного методу просторової індексації геометричних об'єктів.

Дерево є досить поширеним і загально вживаним поняттям в інформатиці та математиці. Дерево може розглядатися як зв'язаний граф (такий граф, між будь-якою парою вершин якого існує принаймні один шлях), що не містить циклів.

Основні типи дерев наступні: бінарні, червоно-чорні, AVL, B, B+-дерева, 2-3, kd, R, R+, R*, Quad. Проаналізувавши різні варіанти дерев, приходимо до висновку, що найбільш універсальним рішенням для графічних підсистем в інформаційній технології автоматизації життєвого циклу будівельного об'єкта є застосування R-дерева та його модифікацій.

R-дерево – деревовидна структура даних, що використовується для організації доступу до просторових даних, тобто для індексації багатовимірної інформації. Оскільки просторові об'єкти можуть бути складними за формою і мати нерівні контури, то використовується так звана мінімальна обмежуюча область (minimal bounding box – mbb). Сукупність обмежуючих областей розбиває простір і утворює множину ієрархічно вкладених прямокутників (для двовимірного простору). У разі тривимірного або багатовимірного простору це будуть прямокутні паралелепіпеди (кубоїди) або паралелотипи.

Для підвищення ефективності доступу до просторових даних та їх обробки використовуються оптимізації алгоритмів побудови R-дерев в наступних напрямках:

- скорочення просторових перетинів між вузлами;
- мінімізація мертвого простору;
- зменшення мінімальної обмежуючої області;
- максимізація використання ресурсів зберігання.

Для розщеплення переповнених вершин можуть застосовуватися різні алгоритми, що породжує ділення R-дерев на підтипи: квадратичні і лінійні.

Спочатку R-дерева не гарантували хороших характеристик для найгіршого випадку, хоча добре працювали на реальних даних. Проте, в 2004-му році був опублікований новий алгоритм, що визначає пріоритетні R-дерева (LarsArge та

інші). Стверджується, що цей алгоритм ефективний, як і найбільш ефективні сучасні методи, і в той же час є оптимальним для найгіршого випадку.

Кожна вершина R-дерева має змінну кількість елементів (не більш за деякий заздалегідь заданий максимум). Під час проектування R-дерева треба визначити деякі константи: максимальне число нащадків у вершині M , мінімальне число нащадків у вершині m , за винятком кореня.

Особливості та характеристики R-дерев дозволили їх використовувати в багатьох областях, таких як бази даних, комп'ютерні системи та робототехніка, географічні інформаційні системи і системи автоматизованого проектування.

Отже, використання R-дерев дозволяє підвищити ефективність обробки і використання просторових даних під час проектування:

- ефективно зберігати дані локалізованих в просторі груп об'єктів;
- збалансований і швидкий пошук даних внайгіршому випадку;
- додавання/видалення однієї точки не вимагає істотної перебудови дерева;
- підвищена чутливість до порядку нових даних;
- обмежуючі прямокутники вершин можуть перекриватися.

Останні два пункти є досить вагомими недоліками. Тому були розроблені деякі модифікації.

Існує декілька основних алгоритмів побудови R-дерев та їх модифікацій: базовий, клітинний, R^* , «Розділяй і володарюй». Кожен з цих методів також має модифікацію з уточненнями. Найкращий з точки зору мінімізації перекриття вузлів є метод «розділяй і володарюй», але в той же час він є найбільш витратним за часом.

Алгоритм розділити на групи (алгоритм «Розділяй і володарюй»). Задана множина з N об'єктів E_i ділить на K груп з дотриманням обмеження на кількість об'єктів в групах знизу m^{L-1} і зверху M^{L-1} , де L – поточний рівень дерева.

1. Розподіл на дві групи. Викликати алгоритм *розділення на дві групи*, передавши як аргументи всю множину об'єктів E_i і співвідношення розподілу ($[K/2]$): $(K - [K/2])$ для отримання груп G_1 і G_2 .

2. Рекурсивний розподіл. Якщо $[K/2] > 1$, то викликати алгоритм *розділення на групи*, передавши як аргументи отриману групу G_1 з параметром ділення $[K/2]$, інакше видати в якості результату G_1 . Якщо $(K - [K/2]) > 1$, то викликати алгоритм *розділення на групи*, передавши як аргументи отриману групу G_2 з параметром ділення $(K - [K/2])$, інакше видати в якості результату G_2 .

3. Алгоритм розділити на дві групи. Задана множина з N об'єктів E_i ділить на 2 групи в заданому співвідношенні площ $k:l$ з дотриманням обмеження на кількість об'єктів знизу $k \cdot m^{L-1}$ і зверху $k \cdot M^{L-1}$ для першої групи, а також знизу $l \cdot m^{L-1}$ і зверху $l \cdot M^{L-1}$ для другої, де L – поточний рівень дерева, що будується. Цей алгоритм аналогічний алгоритму *розщеплення вузла* звичайного R-дерева, за винятком перевірок, при яких порівнювані значення зважуються по величинах k та l .

4. Вибір осі координат для сортування. Вибрати таку вісь координат i , на якій досягається максимальне значення виразу:

$$\max_{\text{вісі}} \left(\frac{\max_{\text{фігури } j} R_a^{j,i} - \min_{\text{фігури } j} R_b^{j,i}}{\max_{\text{фігури } j} R_b^{j,i} - \min_{\text{фігури } j} R_a^{j,i}} \right), \quad (10)$$

де $R_a^{j,i}$ і $R_b^{j,i}$ – відповідно менше і більше значення i -ої координати фігури (мінімально обмежуючого прямокутника).

5. Розподіл по групах. Розділити всі об'єкти на три частини за значенням $P^i(j) = (R_a^{j,i} + R_b^{j,i})/2$ для обраного i по квантилях рівня a і $1-a$ ($a \in [0;0,5]$ – параметр розподілу по групах): з найменшої за значенням частини утворити першу групу об'єктів, а з найбільшої – другу групу. Об'єкти, що залишилися, розподілити по групах в кроках 3 і 4.

6. Перевірка завершення розподілу. Якщо всі об'єкти розподілені, то закінчити процес розподілу. Якщо одна з груп в результаті подальших операцій буде не доповнена (кількість об'єктів в групах $p_1 < k \cdot m^{L-1}$ або $p_2 < l \cdot m^{L-1}$), то вставити в неї всі об'єкти, що залишилися і на цьому закінчити. Якщо одна з груп в результаті подальших операцій буде переповнена (кількість об'єктів в групах $p_1 < k \cdot M^{L-1}$ або $p_2 < l \cdot M^{L-1}$), то вставити в іншу групу решту об'єктів, що залишилися.

7. Якщо ж залишився нерозподілений об'єкт, то необхідно його помістити в групу, розмір якої збільшиться в мінімальній мірі при зважуванні по k та l перейти на крок 3.

У наступній теоремі встановлюється трудомісткість роботи глобального алгоритму побудови R-дерева, що використовує запропонований алгоритм розбиття «Розділяй і володарюй».

Теорема (Трудомісткість глобального алгоритму побудови R-дерева «Розділяй і володарюй»). Нехай задано N об'єктів, на яких необхідно побудувати R-дерево. Тоді трудомісткість глобального алгоритму побудови R-дерева, що використовує алгоритм «Розділяй і володарюй», складає: $O(N \log^2 N)$.

Доведення. Спочатку встановимо трудомісткість роботи алгоритму розбиття «Розділяй і володарюй». Його трудомісткість можна записати наступним чином:

$$T(N) = R(N) + 2 \cdot T(N/2), \quad (11)$$

де $R(N)$ – трудомісткість алгоритму розбиття множини об'єктів на дві частини.

В алгоритмі розподілу на дві частини найбільш трудомістким етапом є другий крок, в якому виконується розподіл множини об'єктів на три частини по заданих квантилях. Якщо використовувати лінійні за складністю алгоритм розподілу по квантилях Хоара або алгоритм цифрового сортування, то виходить $O(R(N)) = O(N)$, оскільки інші кроки алгоритму, що розглядається не більше ніж лінійні, що виходить з опису алгоритму.

Таким чином, загальна трудомісткість алгоритму розбиття «Розділай і володарюй» складає:

$$\begin{aligned}
 O(T(N)) &= O(N) + 2 \cdot O(T(N/2)) \\
 &= O(N) + 2 \cdot (O(N/2) + 2 \cdot O(T(N/4))) \\
 &= O(N) + 2 * (O(N/2) + 2 \cdot (O(N/4) + 2 \cdot O(T(N/8)))) \\
 &= \dots = O(N) \cdot O(\lceil \log_2 N \rceil) = O(N \log N)
 \end{aligned} \tag{12}$$

Для встановлення загальної трудомісткості глобального алгоритму побудови R-дерева «Розділай і володарюй» скористаємося формулою:

$$\begin{aligned}
 O(S(N)) &= O\left(\sum_{i=0}^{\lceil \log_k N \rceil - 1} K^i \cdot T(N/K^i)\right) \\
 &= O\left(\sum_{i=0}^{\lceil \log_k N \rceil} K^i \cdot O(N/K^i \cdot \log(N/K^i))\right) \\
 &= \sum_{i=0}^{\lceil \log_k N \rceil - 1} O(N \cdot \log N) = O(N \cdot \log N) \cdot O(\lceil \log_k N \rceil) \\
 &= O(N \cdot \log^2 N),
 \end{aligned} \tag{13}$$

що і вимагалось довести.

На другому кроці роботи алгоритму розподілу «Розділай і володарюй» присутній параметр a розподілу по групах. Для встановлення його впливу на якість отриманого розбиття було проведено експериментальне моделювання роботи глобального алгоритму побудови R-дерева, що використовує алгоритм «Розділай і володарюй». При цьому було встановлено, що зменшення значення a призводить до деякого поліпшення структури R-дерева на нерівномірних розподілах і розподілах зі значним перекриттям об'єктів. З іншого боку, це призводить до зниження швидкості роботи алгоритму і зниження якості розбиття на точкових і регулярних наборах даних.

Підсумовуючи переваги і недоліки поведінки алгоритму при різних значеннях a на різних розподілах об'єктів, було вибрано емпіричне значення параметра розподілу для універсального застосування, рівне 0,45.

R-дерево автоматично підтримує ієрархію, а тому можна використати цю властивість для зберігання топологічних зв'язків об'єктів і створити систему класів на основі R-дерева, яка підтримувала б не тільки ієрархію об'єктів, а й всі інші аспекти обробки даних об'єкта: побудову, модифікацію та знищення.

Для створення такої системи класів необхідно дещо модифікувати класичні підходи до побудови R-дерев. По-перше необхідно зняти обмеження на мінімальну та максимальну кількість нащадків для кожного вузла дерева. В модифікованому варіанті R-дерева, що пропонується для практичного застосування в підсистемах двовимірної графіки, кількість нащадків будь-якого

вузла дерева може бути довільною. Модифікований варіант R-дерева дозволяє одержати високу гнучкість та простоту роботи.

Модифіковане R-дерево може напряму описувати ієрархію графічних об'єктів, які застосовуються для моделювання об'єктів будівництва. Для прикладу можна розглянути план поверху будь-якого будинку. Поверх є коренем дерева, а всі будівельні об'єкти, що розташовані на ньому є листям цього дерева. Поверх може ділитися на секції, а ті в свою чергу на приміщення. Приміщення складаються із колон, перекриттів, стін. Стіни можуть містити двері та вікна, а ті в свою чергу мають у складі дрібніші елементи: відкоси, підвіконня, коробки і т. п.

Ще однією важливою модифікацією R-дерева є те, що на відміну від класичного R-дерева, в модифікованому дереві можливий випадок, коли один і той же об'єкт присутній в декількох різних гілках, що дозволяє повністю покрити можливі типи топологічних зв'язків «*посилання*». В наведеному вище прикладі такий зв'язок можливий для об'єктів «приміщення» та «стіна», оскільки одна й та ж стіна може посилатися на декілька приміщень, оскільки є кордоном між ними. Те ж саме стосується й інших будівельних елементів. З урахуванням зазначеного про зв'язок «посилання», отримаємо наступне дерево (рис. 8).

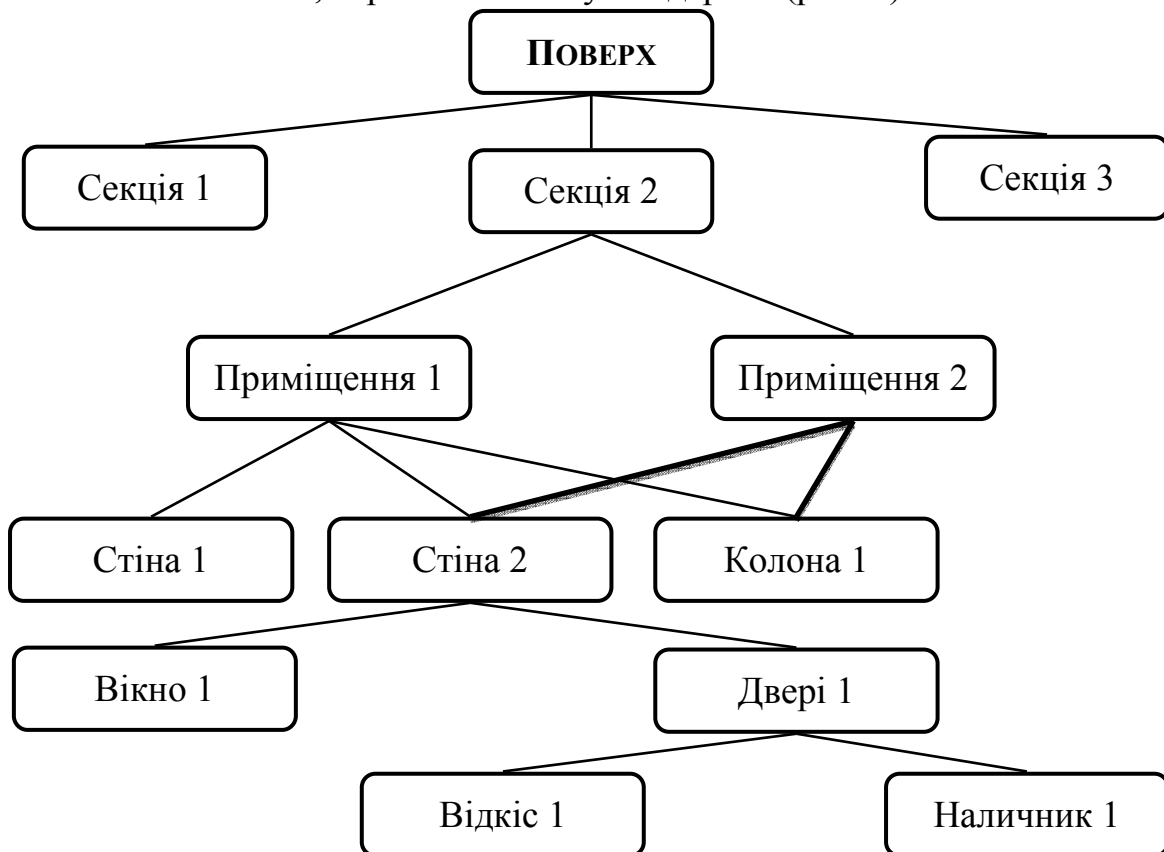


Рис. 8. Приклад модифікованого R-дерева з додаванням зв'язку «посилання»

Після додавання зв'язку «посилання» отримана нова структура, що не є деревом в класичному розумінні. Це фактично деякий граф, з елементами ієрархії, а така структура досить складана і заплутана. Тому необхідно знайти вирішення проблеми підтримки зв'язку «посилання» іншими методами.

Для усунення негативних наслідків, що притаманні варіанту дерева на рис. 8 пропонується для побудови модифікованих R-дерев використовувати за основу типи зв'язків «включення» та «групування». Тоді дерева будуть мати чітку ієрархію і не будуть мати листів з декількома «батьками». Зв'язок типу «посилання» варто використовувати лише для об'єктів на одному рівні ієрархії для отримання горизонтальних переходів (рис. 9).

Програмна реалізація таких типів зв'язків досить проста – достатньо в кожному об'єкті мати три списки для кожного типу зв'язків. Хоча зв'язки «включення» та «групування» повинні бути взаємовиключними, тому для них можна використати лише один список з додатковим прапорцем, але це не принципово і залежить від конкретної реалізації.

Розширення класичного R-дерева на тривимірний простір називається BVH-деревом (BoundingVolumeHierarchy). Всі ознаки класичного R-дерева тут зберігаються, а в якості обмежувальних прямокутників використовуються обмежувальний об'єм, який найчастіше є прямокутним паралелепіпедом, але може бути також і будь-яким іншим тілом (призма, сфера, піраміда і т. п.).

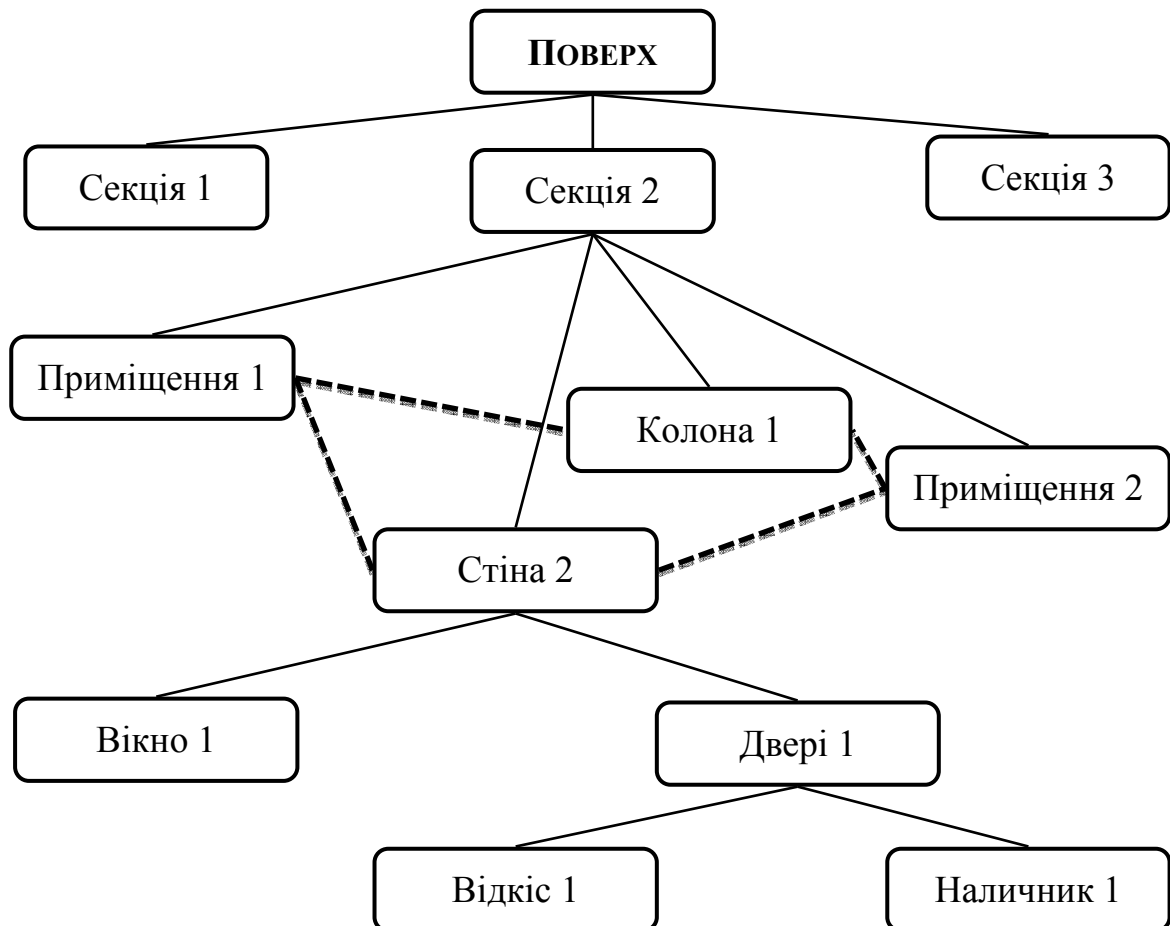


Рис. 9. Приклад модифікованого R-дерева із зв'язком «посилання» на одному рівні ієрархії

Розширення модифікованого R-дерева, що описане в попередньому пункті, на тривимірний простір зберігає всі описані раніше властивості, а в якості

обмежувального об'єму використовує прямокутний паралелепіпед. З точки зору реалізації це найпростіше об'ємне тіло, що дозволяє здійснювати ефективний пошук в тривимірному просторі.

Як і для двовимірного випадку, в тривимірному модифікованому R-дереві присутні всі три типи зв'язків між об'єктами – «включення», «групування» та «посилання». Фактично сама структура дерева для тривимірного простору нічим не відрізняється від дерева для двовимірного простору. Всі відмінності зводяться до алгоритмів подання об'єктів та їх пошуку в дереві.

Ще одна важлива деталь, це те, що тривимірне подання об'єктів зазвичай складається з декількох графічних примітивів, але листовим вузлом дерева є саме об'єкт, а не його графічні частини. Наприклад об'єкт «колона» в тривимірному поданні це група полігонів (найчастіше чотирикутників або трикутників), що відтворюють просторову форму. Але листовим вузлом дерева є саме об'єкт колона, а не її геометричні складові.

У **п'ятому розділі** розроблена уніфікована розширювана інформаційна технологія автоматизації життєвого циклу будівельного об'єкта.

Як вже зазначалося раніше, інваріантна частина інформації (I_1) є важливою складовою універсальних розширюваних систем автоматизації. Структура інваріантної БД повинна бути розширювана – це дозволить включати в неї нові сутності, що на теперішній час ще не існують. З точки зору життєвого циклу будівельного об'єкта, в таку БД потрібно включити всю необхідну нормативну інформацію, а також інформацію про типові будівельні елементи та кошторисно-фінансові норми.

Зазначимо, що інваріантна БД може мати не одну реалізацію, а декілька, і кожна з них фізично може бути подана у вигляді декількох БД. До того ж кожна з окремих фізичних частин інваріантної БД може бути реалізована з використання будь-якої СУБД, і не завжди однієї і тієї ж для всіх частин.

Базова модель інваріантної БД включає в себе довідник типових будівельних елементів; типових геометричних елементів, що описують будівельні; перелік основних матеріалів та їх характеристики; перелік одиниць вимірювання, що використовуються та коефіцієнтів перерахунку між ними; список кошторисних нормативів; цінові показники на основні матеріали, обладнання, машини та роботу; часові норми на зведення окремих будівельних конструкцій; скрипти автоматичної зв'язки будівельних елементів з кошторисно-фінансовими та часовими нормами їх виконання та інші необхідні довідники.

З точки зору фізичного подання, логічним є відокремлення в окремі БД кошторисних нормативів, цінових показників, часових норм та скриптів автоматичної прив'язки. Це дозволить виконувати кожен проект будівельного об'єкта максимально гнучким і з можливістю декількох фінальних реалізацій, що мають спільне архітектурно-планувальне рішення.

Запропонована типова структура інваріантної БД (рис. 10) не є повною, а лише базовою частиною. Для кожної конкретної САх-системи, що автоматизує певний етап життєвого циклу будівельного об'єкта, потрібно додавати нові

сутності та атрибути, що характерні для цього етапу. Наприклад для створення САх-системи, що призначена для конструкторського проектування необхідно детально описати всі можливі стандартні конструктивні елементи та їх характеристики.

Структура даних для зберігання інформації про будівельний об'єкт є важливою складовою системи. Від її вдалого проектування залежить гнучкість і розширюваність всієї системи. Базова структура даних призначена для опису окремого будівельного об'єкта і повинна бути сумісна з базовою структурою інваріантної інформації. В деякій мірі, базова структура даних ядра системи повинна дублювати структуру інваріантної бази даних, оскільки типова інформація буде братися з інваріантної БД і переписуватися в базову структуру.

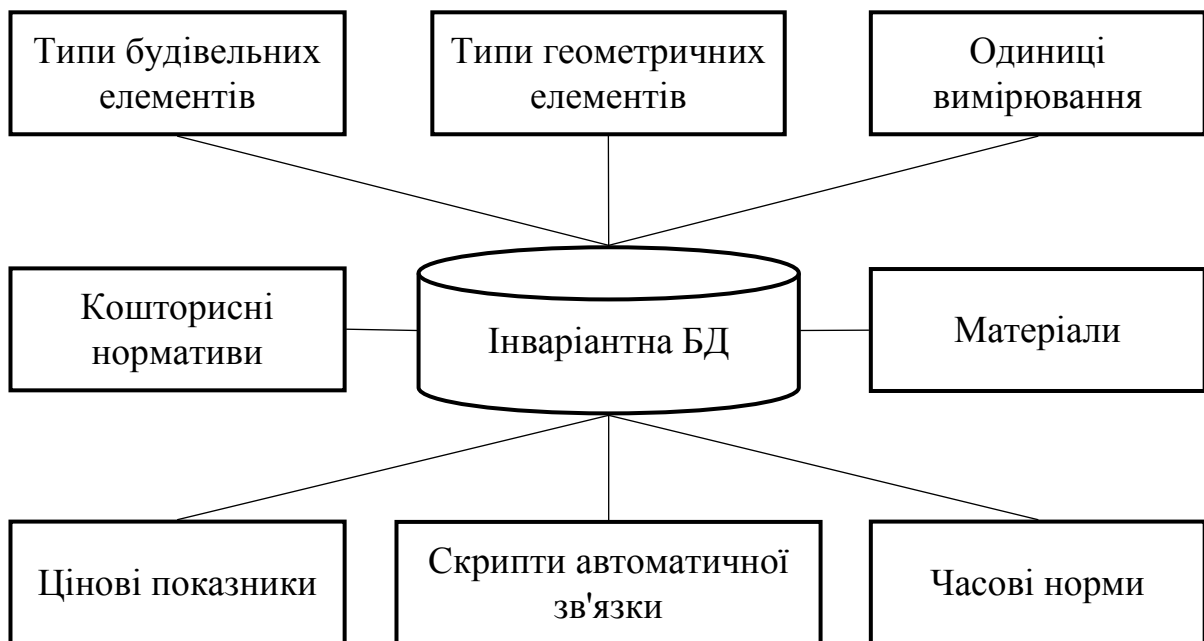


Рис. 10. Типова структура інваріантної БД

Логічно базова структура даних може бути розділена на складові у відповідності до методології, що викладена в попередніх розділах. Тобто вона повинна містити основну інформацію, геометричну, топологічну, часову і фінансову.

Базова структура даних ядра системи описує конкретний будівельний об'єкт в цифровому вигляді. Ієрархічно, на найвищому рівні, знаходиться таблиця, що містить перелік всіх поверхів будівлі (Stories). Кожен поверх включає в себе будівельні елементи (Elements) як на пряму так і в складі приміщень (Zones). Фактично між елементами і поверхами встановлений прямий зв'язок включення, а між елементами і приміщеннями – посилання.

Кожен будівельний елемент повинен мати структурний склад (Structures) – це може бути простий матеріал (Materials), композитний матеріал або група матеріалів.

Особливо важливими є таблиці, що описують геометрію елементів. Всі вони зв'язані з таблицею Geometries, яка є агрегатором різних геометричних конструкцій і напрямку прив'язана до елементів.

Інформаційна технологія автоматизації життєвого циклу будівельного об'єкта складається з набору інструментів та програмних засобів, що дозволяють виконувати двовимірну та тривимірну візуалізацію елементів будівельного об'єкта, а також додаткових інструментів редагування та створення окремих елементів. Під елементами будівельного об'єкта розуміються будь-які архітектурні та конструктивні елементи, а також елементи інженерних мереж.

Найпершим потрібно розробити засіб для перегляду двовимірних зображень планів кожного поверху будівельного об'єкта, а також його фронтальних перерізів. Потім необхідно створити засіб для візуалізації тривимірної моделі. І, насамкінець, розробити засоби, що дозволятимуть автоматизувати проектування будівельного об'єкта на всіх етапах його життєвого циклу.

Розробка всіх засобів та інструментів ведеться в парадигмі об'єктно-орієнтованого проектування та програмування. Середовище та мова програмування не важливі, але повинна бути підтримка ООП.

Розроблений прототип ІТАЖЦБО був створений на мові програмування C++. Схема найвижчого рівня декомпозиції розробленої системи подана на рис. 11.

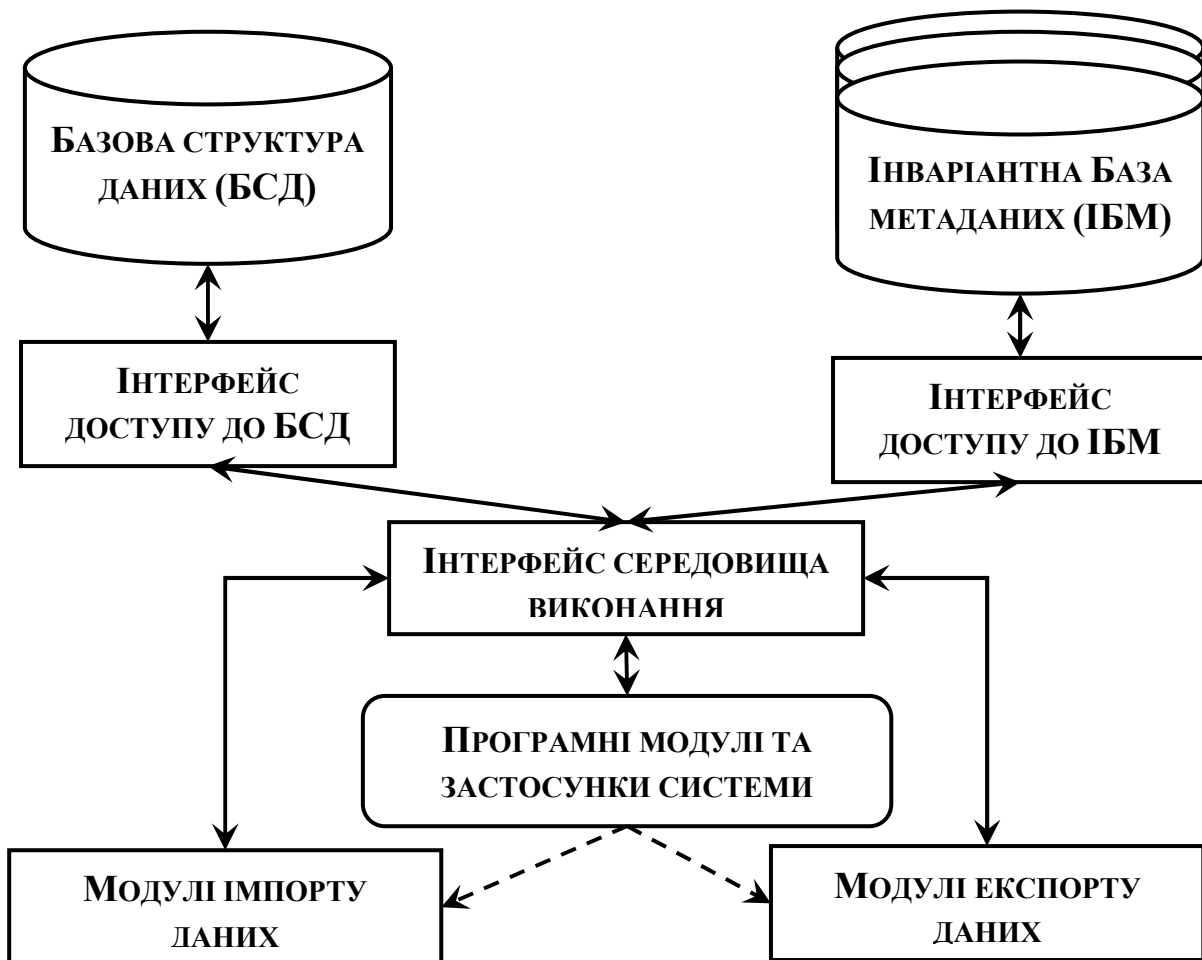


Рис. 11. Схема топ рівня розробленого прототипу системи

Прототип системи складається з базової структури даних (БСД), що містить інформацію про об'єкт будівництва на всіх етапах його життєвого циклу. БСД реалізована у вигляді реляційної БД з використанням СУБД MS Access. Принципово не важливо на якій СУБД реалізована БСД, головне, щоб для кожної реалізації на СУБД був створений інтерфейс доступу. Саме інтерфейс доступу знає тип використаної СУБД і відповідно інкапсулює в собі класи для роботи з БСД. Ці класи є точкою входу для застосунків і модулів, що використовуються в системі. Класи інтерфейсу доступу до БСД реалізують всі можливі операції з даними – додавання, видалення, редагування, пошук за різноманітними запитами. У випадку зміни чи розширення БСД зміни потрібно вносити лише в інтерфейсі доступу. Такий підхід дозволяє збільшити гнучкість системи та автономність кожної окремої частини, що в свою чергу, забезпечує її модульність і можливість паралельної розробки командою інженерів-програмістів.

Основним системним компонентом нижнього рівня є інтерфейс середовища виконання, який є диспетчером для всіх програмних застосунків і модулів системи. Він пропускає через себе звернення від сервісів вищого рівня до БСД та ІБМ, а також інкапсулює всі глобальні налаштування системи. Інтерфейс середовища виконання реалізований у вигляді системи класів і може бути розширений шляхом додавання нових класів у залежності від парадигми свого функціонування. Окрім інкапсуляції базових налаштувань та диспетчеризації запитів до рівня даних, інтерфейс середовища виконання містить ще й свій верхній рівень, що об'єднує базові сервісні функції системи. Цей верхній рівень є набором спеціалізованих бібліотек, що використовуються модулями та застосунками системи як для розрахункових операцій, так і для візуалізації моделі будівельного об'єкта.

Програмні модулі та застосунки системи це сукупність програм різноманітного призначення. Їх склад може суттєво відрізнитися в залежності від поставленої мети розробки системи. В створеному прототипі системи реалізовані наступні модулі та застосунки: модуль двовимірної графіки, модуль тривимірної графіки, модуль тріангуляції, модуль прив'язки кошторисних нормативів, модуль документації, застосунок для тривимірної візуалізації моделі, застосунок для двовимірних і тривимірних моделей поверхів, застосунок для побудови простих архітектурних елементів будівлі, застосунок для підготовки даних для календарного планування.

Допоміжними модулями, що розширюють функціонал системи, є модулі імпорту та експорту. Серед модулів імпорту можна виокремити наступні: модуль імпорту растрових зображень, модуль імпорту векторних зображень, модуль імпорту файлів DWG та DXF, модуль імпорту даних з програмного засобу ArchiCAD, модуль імпорту файлів формату IFC. Модулі експорту, які розроблені для прототипу системи, це: модуль експорту в кошторисні програми, модуль експорту в програми управління будівництвом, модуль експорту в формат IFC, модуль експорту планів поверхів в програмний комплекс AutoCAD.

ВИСНОВКИ

В дисертаційній роботі наведені теоретичні та практичні результати у відповідності з метою і задачами дослідження – розв'язання науково-практичних проблем зі створення теоретичних основ та методології формування універсальних інформаційних технологій автоматизованого проектування будівельних об'єктів та розробка моделей і методів подання інформації в процесі будівництва на всіх етапах його життєвого циклу, а також методів побудови підсистем комп'ютерної графіки.

В процесі вирішення поставлених задач були отримані наступні результати.

1. Запропонована методологія створення уніфікованих інформаційних технологій автоматизованого проектування будівельних об'єктів на всіх етапах їх життєвого циклу з урахуванням особливостей об'єктів, програмна реалізація ІТАЖЦБО та оцінка ефективності її реалізації.

2. Досліджені основні етапи життєвого циклу будівельного об'єкта та запропоновані моделі подання інформації на кожному етапі.

3. Визначені основні вимоги до універсальних розширюваних інформаційних технологій автоматизованого проектування всіх етапів життєвого циклу будівельних об'єктів.

4. Проведено вивчення та дослідження сучасних засобів систем автоматизованого проектування та проведена їх класифікація за задачами на етапах життєвого циклу будівельного об'єкта.

5. Проаналізовані та класифіковані типи будівельних елементів та способи їх подання на кожному з етапів життєвого циклу будівлі.

6. Створена концептуальна модель структури даних, що описує будівельний об'єкт на всіх етапах його життєвого циклу та розроблені методи розширення ядра моделі будівельного об'єкта на основі модульного принципу.

7. Проведено аналіз основних методів оптимізації структур збереження інформаційних даних для вибору найбільш ефективної структуризації даних при паралельному проектуванні будівельних об'єктів.

8. Розроблено метод бінарного пакування даних для оптимізації розміру інформації, що зберігається, а також для підвищення швидкості пошуку необхідних даних при проектуванні та їх шифрування.

9. Створена модель інваріантної інформації та набору метаданих для розширювання структури даних про будівельний об'єкт.

10. Досліджені та проаналізовані основні методи просторової індексації об'єктів на основі чого була визначена оптимальна структура оптимізації графічних даних та розроблене модифіковане R-дерево, що забезпечує оптимальне використання ієрархії графічних об'єктів, що моделюють будівлю.

11. Досліджені основні методи побудови підсистем комп'ютерної графіки та виокремлені найбільш уніфіковані підходи.

12. Спроектований та розроблений прототип ІТАЖЦБО з використанням розробленої методології та виконана оцінка ефективності розробленого прототипу.

13. Практична цінність даної роботи полягає в тому, що запропонована методологія створення інформаційних технологій автоматизації життєвого циклу будівельних об'єктів використовується розробниками спеціалізованих програмних комплексів для автоматизації проектування і управління будівництвом, зокрема ТОВ «АДА», а також в навчальному процесі КНУБА, про що є довідки та акти впровадження.

СПИСОК ОПУБЛІКОВАНИХ ПРАЦЬ ЗА ТЕМОЮ ДИСЕРТАЦІЇ

Публікації у закордонних виданнях

1. SergiyPaliy. MethodforSearchingof «TheIdealInterlocutor» forUsersof «InformationandOrganizationalEnvironmentfor Pre-University TrainingandSocialAdaptationofForeigners» [Текст] / SergiyPaliy, YevgeniyBorodavka, MykolaTsiutsiura // InternationalJournalofInnovativeResearchinComputerandCommunicationEngineering. – 2013. – Vol.1, Issue 8. – pp. 1611–1617.

Автором запропоновано використати сингулярне розкладення матриці для створення матриці передбачення уподобань конкретних студентів для подальшого їх використання в пошуку ідеального співрозмовника.

2. YevgeniyBorodavka. BinaryDataPackingMethodforDatabaseOptimization [Текст] / YevgeniyBorodavka, MykolaTsiutsiura // InternationalJournalofScienceandResearch. – 2014. – Vol.3, Issue 11. – pp. 903–905.

Автором запропонований метод оптимізації структури бази даних шляхом бінарного пакування не індексованих даних. Такий підхід дозволяє суттєво зменшити розмір бази даних, а також збільшити її швидкодію та надійність.

3. YevgeniyBorodavka. BuildingModelConception [Текст] / YevgeniyBorodavka, MykolaTsiutsiura // InProceedingsofthe WORLD ScienceConference “ScienceandEducation – OurFuture”. – 2014. – Issue №3, November 24-26, 2014, AbuDhabi, UAE. – pp. 53 – 55.

Автору належить концептуальна модель будівельного об'єкта на всіх етапах його життєвого циклу. Автор виокремив основні моделі, що притаманні етапам життєвого циклу будівельного об'єкта та показав їх зв'язки.

4. OlenaKryvoruchko. ModelsofMakingManagementDecisions[Текст] / OlenaKryvoruchko, YevgeniyBorodavka // InternationalJournalofScienceandResearch. – 2014. – Vol.3, Issue 12. – pp. 613–616.

Автором запропонована класифікація управлінських рішень та їх декомпозиція.

5. YevgeniyBorodavka, ConceptualModelofDataStructureforBuildingObject [Текст] / YevgeniyBorodavka, OlenaKryvoruchko //InProceedingsofthe WORLD ScienceConference “ScientificandPracticalResultsin 2014.

Prospects for Their Development”, pp. 37–40.

Автором запропоновані моделі на різних стадіях життєвого циклу де на верхній рівень концептуальної моделі структури даних об'єкту необхідне розкладання кожної частини концептуальної моделі.

6. Yevgeniy Borodavka. Modified R-Tree [Текст] / Yevgeniy Borodavka // International Journal of Science and Research. – 2015. – Vol.4, Issue 1. – pp. 1725–1727.

7. Olena Kryvoruchko. Information Technology Models for Project Management of Education Development [Текст] / Olena Kryvoruchko, Yevgeniy Borodavka, Mikola Tsiutsiura // International Journal of Computer Science and Telecommunications. – 2015. Vol. 6, Issue 2. – pp. 7–9.

Автором запропоновані моделі інформаційної технології управління інноваційними проектами модернізації та розвитку в освітній сфері та висвітлення проблем, що виникають на всіх етапах життєвого циклу цих проектів, узагальнюючи отриманий досвід.

8. Borodavka Y. The Hardware Adapted Ray Tracing Algorithm / Y. Borodavka, O. Lisovyi, D. Deineka // International Journal of Computer Science and Telecommunications. – 2016. – Vol.7, Issue 4. – pp. 1-7.

Автором запропоновано метод трасування променів, який адаптований для реалізації в апаратному забезпеченні. Також автор запропонував вдосконалені методи попереднього сортування об'єктів для пришвидшення їх обробки.

Публікації у фахових виданнях

9. Бородавка Є.В. Моделі та засоби інформаційної інтеграції систем проектування будівель і споруд [Текст] / Є.В. Бородавка // Вісник Східноукраїнського національного університету імені Володимира Даля. – 2009. – №6(136). – С. 255–259.

10. Бородавка Є.В. Методи оцінки коректності реалізації інформаційної технології інтеграції САПР будівель і споруд на основі комплексної цифрової моделі об'єкта [Текст] / Є.В. Бородавка, С.Л. Печенов // Управління розвитком складних систем. – 2010. – №2. – С. 42–44.

Автору належить розробка методу оцінки коректності реалізації інформаційної технології інтеграції САПР будівельних споруд на основі комплексної цифрової моделі об'єкта.

11. Бородавка Є.В. Productlifecyclemanagementinconstruction [Текст] / Є.В. Бородавка, С.Л. Печенов // Східноєвропейський журнал передових технологій. – 2010. – № 6/3 (48). – С. 31–34.

Автору належить розробка декомпозиції етапів життєвого циклу будівельного об'єкта та їх опис.

12. Бородавка Є.В. Аналіз та порівняння основних складових архітектурно-будівельних САПР [Текст] / Є.В. Бородавка // Управління розвитком складних систем. – 2010. – №3. – С. 71–74.

13. Бородавка Є.В. Модель розширюваної системи автоматизації життєвого циклу будівельного об'єкта [Текст] / Є.В. Бородавка // Управління розвитком складних систем. – 2010. – №4. – С. 69–71.

14. Бородавка Є.В. Класифікація архітектурно-будівельних САПР [Текст] / Є.В. Бородавка // Управління розвитком складних систем. – 2011. – №7. – С. 112–117.

15. Бородавка Є.В. Способи подання моделі будівельного об'єкта [Текст] / Є.В. Бородавка // Управління розвитком складних систем. – 2011. – №8. – С. 101–107.

16. Бородавка Є.В. «Ядро» моделі будівельного об'єкта на основі базового набору графічних примітивів та атрибутів [Текст] / Є.В. Бородавка // Управління розвитком складних систем. – 2013. – №15. – С. 111–114.

17. Бородавка Є.В. Методи розширення ядра моделі будівельного об'єкта на основі модульного принципу [Текст] / Є.В. Бородавка // Управління розвитком складних систем. – 2013. – №16. – С. 124–132.

18. Бородавка Є.В. Методи сортування геометричних об'єктів та їх реалізація на прикладі плагінаавтонумерації елементів для САПР Allplan [Текст] / Є.В. Бородавка, В.М. Квасневський // Управління розвитком складних систем. – 2014. – №18. – С. 128–132.

Автором запропоновані методики сортування геометричних об'єктів різними методами та приклади їх реалізації.

19. Цюцюра С.В. Методи проєкції об'єктних моделей на структури даних [Текст] / С.В. Цюцюра, Є.В. Бородавка // Управління розвитком складних систем. – 2015. – №21. – С. 92–98.

Автору належить аналіз методів проєкції об'єктних моделей на реляційні бази даних та порівняння їх характеристик.

20. Квасневський В.М. Геометричні методи побудови отворів та гільз для інженерних мереж в САПР Allplan [Текст] / В.М. Квасневський, Є.В. Бородавка // Управління розвитком складних систем. – 2015. – №22. – С.128–133.

Автору належить створення та оцінка складності алгоритмів, що використовуються в методах побудови отворів у суцільних об'єктах.

21. Бородавка Є.В. Методи побудови об'єктів в комп'ютерній графіці [Текст] / Є.В. Бородавка, В.М. Квасневський // Управління розвитком складних систем. – 2015. – №24. – С.106–110.

Автору належить дослідження способів подання полігональних моделей та порівняння їх ефективності.

Матеріали міжнародних наукових конференцій

22. Бородавка Є.В. Життєвий цикл будівельного об'єкта та автоматизовані засоби його підтримки [Текст] / Є.В. Бородавка // Тези доповіді ХІХ науково-практичної конференції "Інформаційні технології в економіці, менеджменті і бізнесі. Проблеми науки, практики і освіти". – Київ, "Європейський університет", 2013.

23. Бородавка Є.В. Автоматизовані засоби підтримки життєвого циклу будівельного об'єкта [Текст] / Є.В. Бородавка // Матеріали семінару міжнародної науково-практичної конференції «САПР Allplan. Інноваційне проектування в архітектурі і будівництві» – Київ, КНУБА, 2014. – С. 10–13.

24. Бородавка Є.В. Концепція моделі будівельного об'єкта та її структури даних [Текст] / Є.В. Бородавка // Щорічна конференція КНУБА, 2015.

Науково-навчальні посібники

25. Лященко А.А. Геометричне моделювання і комп'ютерна графіка: використання бібліотеки OpenGL [Текст] / А.А. Лященко, В.В. Демченко, Є.В. Бородавка, В.М. Смирнов // К.: КНУБА, 2009. – 90 с.

Автору належать розділи про двовимірну графіку в OpenGL, а також переклад тестових прикладів на мову C++.

АНОТАЦІЯ

Бородавка Є.В. Методологія створення інформаційної технології автоматизації життєвого циклу будівельних об'єктів. – Рукопис.

Дисертація на здобуття наукового ступеня доктора технічних наук за спеціальністю 05.13.06 – інформаційні технології. – Київський національний університет будівництва та архітектури. – Київ, 2017.

Дисертація присвячена вирішенню важливої науково-практичної проблеми автоматизації життєвого циклу будівельних об'єктів. Запропоновані методологічні засади розробки уніфікованої розширеної інформаційної технології автоматизації життєвого циклу будівельних об'єктів з використанням узагальненої моделі будівельного об'єкта, створено ядро моделі на основі базового набору графічних примітивів та атрибутів. Також в роботі пропонується класифікація програмних засобів, що використовуються на різних етапах життєвого циклу будівельного об'єкта та їх узагальнена модель. Для оптимізації бази даних, що містить інформацію про модель будівельного об'єкта, пропонується використати створений метод бінарного пакування даних.

Ключові слова: будівельний об'єкт, життєвий цикл, модель будівельного об'єкта, інформаційна технологія автоматизації, базовий набір графічних примітивів, методи оптимізації баз даних, ObjectRelationalMapping, модифіковане R-дерево, інваріантна база метаданих, архітектурно-будівельні САПР, САх-системи.

АННОТАЦИЯ

Бородавка Е.В. Методология создания информационной технологии для автоматизации жизненного цикла строительного объекта. – Рукопись.

Диссертация на соискание ученой степени доктора технических наук по специальности 05.13.06 – информационные технологии. – Киевский национальный университет строительства и архитектуры. – Киев, 2017.

Диссертация посвящена решению важной научно-практической проблемы автоматизации жизненного цикла строительных объектов. Предложены методологические основы разработки унифицированной расширяемой информационной технологии для

автоматизации жизненного цикла строительных объектов с использованием обобщенной модели строительного объекта, создано ядро модели на основе базового набора графических примитивов и атрибутов. Также в работе предлагается классификация программных средств, используемых на различных этапах жизненного цикла строительного объекта и их обобщенная модель.

На основе анализа существующих решений в области автоматизации жизненного цикла строительных объектов были установлены основные составляющие модели строительного объекта на разных этапах жизненного цикла. Эти составляющие были объединены в

обобщенную модель строительного объекта, что позволило выявить общие части модели на разных этапах жизненного цикла.

Проанализированы существующие архитектурно-строительные конструкции и классифицированы их геометрические характеристики, что позволило создать ядро модели строительного объекта на их основе.

Также были исследованы методы проектирования объектных моделей на реляционные базы данных и установлены их преимущества и недостатки. Для оптимизации баз данных, содержащей информацию о модели строительного объекта, предлагается использовать созданный метод бинарной упаковки данных.

Для эффективного использования графической подсистемы было разработано модифицированное R-дерево, что позволило автоматизировать построение иерархии графических объектов, а также обеспечить быстрый интерактивный доступ к ним.

На основе разработанной методологии был создан прототип информационной технологии, на котором протестированы предлагаемые модели и методы.

Ключевые слова: строительный объект, жизненный цикл, модель строительного объекта, информационная технология автоматизации, базовый набор графических примитивов, методы оптимизации баз данных, ObjectRelationalMapping, модифицированное R-дерево, инвариантная база метаданных, архитектурно-строительные САПР, САХ-системы.

ABSTRACT

Borodavka

Y.V.

Methodology of the information technology creation for the construction objects' lifecycle automation. – Manuscript.

The thesis is for the doctor of technical sciences degree, specialty 05.13.06 – information technology. – Kyiv National University of Construction and Architecture. – Kyiv, 2017.

The thesis is devoted to solve the important scientific and practical problem of the construction objects' lifecycle automation.

The methodological principles for unified extensible information technology of the construction objects' lifecycle automation with generalized model of the construction object are suggested.

The core of the construction object model based on a basic set of graphics primitives and attributes is developed.

The classification of the software tools used at different stages of the construction object lifecycle and their generalized models are proposed. The developed method of the binary packed data for the database optimization is suggested for using.

Keywords: construction object, lifecycle, construction object model, information technology of automation, basic set of graphics primitives,

database optimization techniques, Object Relational Mapping, modified R-tree, invariant base of metadata, architectural and building CAD, CAx-systems.