

УДК 005.004

В.Б. Задоров, О.О.Васильєв

Київський національний університет будівництва і архітектури, Київ

**МАТЕМАТИЧНІ МОДЕЛІ ОБ'ЄКТІВ І ПРОЦЕСІВ БУДІВНИЦТВА  
В СЕРЕДОВИЩІ МОВИ ІМІТАЦІЙНОГО МОДЕЛЮВАННЯ «АІМО»**

*Розглянуто систему математичних моделей опису та імітаційного моделювання об'єктів і процесів технології будівництва із застосуванням мови «АІМО», а саме, основні елементи мови: компілятор АІМО, віртуальна машина або засіб моделювання, візуальне середовище розробки (ІДЕ). Наведено опис процесу імітаційного моделювання, основної функції імітаційного моделювання: аналізу і комплексної оцінки проектних рішень з прийняття ОПР, КР, ОТР, УПР на всіх етапах системи підготовки і управління будівництвом, математичне представлення задачі динамічного програмування організаційно-технологічних процесів, опис системи обліку ризиків при виконанні моделювання цільових процесів будівництва.*

**Ключові слова:** система підготовки і управління будівництвом; ієрархічні об'ємно-планувальні, об'ємно-планувальні рішення (ОПР), конструктивні рішення (КР), організаційно-технологічні рішення (ОТР), управлінські рішення (УПР), конструктивні моделі будівель та споруд, цільові ієрархічні організаційно-технологічні моделі процесів будівництва (ОТМ), мова програмування і моделювання будівельних технологій АІМО; компілятор; віртуальна машина; середовище розробки; об'єктно-орієнтоване програмування; таксономічна ієрархія; мереологічна ієрархія; імітаційне моделювання; критерії оптимізації; динамічне програмування; метод розгалужень і обмежень; метод Монте-Карло; ризики при виконанні моделювання організації і технології зведення будівлі

*Рассматривается комплекс математических моделей языка моделирования строительных технологий «АИМО», основные элементы языка: компилятор АИМО, виртуальная машина или способ моделирования, визуальная среда разработки (ІДЕ). Приводится описание процесса имитационного моделирования, математическое представление задачи динамического программирования, описание системы учета рисков при выполнении моделирования организационно-технологических решений возведения зданий.*

**Ключевые слова:** система подготовки и управления строительством; иерархические объемно-планировочные, объемно-планировочные решения (ЛПР), конструктивные решения (КР), организационно-технологические решения (ОТР), управленческие решения (УПР), конструктивные модели зданий и сооружений, целевые иерархические организационно-технологические модели процессов строительства (ОТМ), язык программирования и моделирования строительных технологий АИМО; компилятор; виртуальная машина; среда разработки; объектно-ориентированное программирование; таксономическая иерархия; мереологическая иерархия; имитационное моделирование; критерии оптимизации; динамическое программирование; метод ветвлений и ограничений; метод Монте-Карло; риски при выполнении моделирования организации и технологии возведения здания

*The author examines the complex of mathematical models modeling language building technologies «АИМО», the basic elements of the language: АИМО by the compiler, виртуальная машина or way of modeling, visual development environment (ІДЕ). Describes the process of simulation modeling, mathematical representation of the task of dynamic programming, description of the accounting system risks when performing modeling of organizational and technological solutions for the erection of buildings.*

**Keywords:** system of preparation and construction management-hierarchical space planning, space-planning decision constructive solutions (KR), organizational and technological solutions (FTS),

*management solutions (SPS) constructive models of buildings and structures, target hierarchical organizational-technological models of processes of construction (GCP), a language for programming and simulation of building technologies AIMO; the compiler; virtual machine; development environment; object-oriented programming; the taxonomic hierarchy; мереологічна hierarchy; simulation modeling and optimization criteria; dynamic programming; the method of branches and limitations; Monte-Carlo method; risks in the performance modeling of organization and technology of construction of the building*

## Постановка проблеми

Специфіка предметної області будівництва полягає з одного боку у певній абсолютній індивідуальності конкретних будівель та споруд, з іншого боку – у значній повторюваності аналогічних об'ємно-планувальних, конструктивних і організаційно-технологічних рішень, а також у великому усередненні різних чинників системи нормалізації і стандартизації будівництва.

Система підготовки будівництва на різних фазах життєвого циклу створення і експлуатації будівель і споруд (ЖЦ СЕ БС) має враховувати принцип сумісного розв'язання задач, що враховують критерії та обмеження, які виникають під час підготовки та прийняття об'ємно-планувальних, конструктивних, технологічних та організаційних рішень в будівництві [2]. Архітектори перевіряють і оцінюють свої об'ємно-планувальні рішення, враховуючі обмеження з конструктивних рішень. Під час проектування унікальних будівель та споруд конструктори розробляють нові конструктивні рішення з врахуванням вимог архітекторів. Все частіше з метою оптимізації витрат виникають ситуації, в яких прийняття об'ємно-планувальних і конструктивних рішень залежить від обмежень і можливостей прийняття організаційно-технологічних рішень з виконання будівельних робіт, тобто при комплексній оцінці вартості і термінів будівництва необхідні так звані «технологічні кошториси», що враховують прийняті організаційно-технологічні рішення не тільки на рівні проектів організації будівництва (ПОБ), а навіть на рівні елементарних процесів і операцій (ЕПОП) під час розробки нормативно-технологічних карт (НТК), які мають входити до складу проектів виробництва робіт (ПВР). Але досі технологічні карти в будівництві не стали обов'язковою складовою в системі підготовки будівництва. Звернемо увагу, що сенс створення і використання таких карт з'являється лише тоді, коли в них відображуються не тільки технологічні, але і організаційні рішення використання ресурсів і часу. Трудомісткість створення таких організаційно-технологічних карт (ОТК) гальмує їх масову розробку і використання. Існуючи в невеликій кількості нормативно-технологічні карти для

окремих будівельних конструктивних елементів і процесів, що пропонуються будівельникам, не відповідають сучасним вимогам. Добрим виключенням є комплект нормативно-технологічних карт, розроблених для гіпсокартонних перегородок на основі конструктивної і організаційно-технологічної системи «КНАУФ» [5]. З їх використанням розроблена сукупність індивідуальних ресурсних елементних кошторисних норм (РЕКН). Але така зацікавленість в розробці НТК і можливість їх створення вручну з'являється тільки у крупних виробників масових конструктивних елементів будівель і споруд або у великих спеціалізованих будівельних організаціях.

На сьогодні не існує інформаційних систем і технологій, в яких би комплексно вирішувалися всі ці питання оптимізації витрат ресурсів і часу в будівництві, і які відповідали б вимогам часу. Це пов'язано з одного боку з відсутністю єдиної системної методології декомпозиції і структурування моделей об'єктів і процесів в будівництві, з другого боку – зі складністю інформаційного моделювання на рівні формалізованих знань та даних. Специфікації витрат потрібних ресурсів різних типів, що формуються у складі проектної документації, не завжди враховують структуру їх ринкової номенклатури, або вимоги організаційно-технологічних схем їх використання. Існуюча ресурсна кошторисна нормативна база (РЕКН) досі відбиває усередненість вимог до кошторисів у неринкових умовах. Благі наміри удосконалення всього цього на основі підвищення якості державної і виробничої ресурсної нормативної бази зіштовхуються з великою трудомісткістю. Але сучасна система підготовки і управління в будівництві для підвищення комплексності і ефективності рішень, що приймаються на різних фазах ЖЦ СЕ БС, повинна спиратися на сучасні засоби інформаційних технологій і ефективно структуровані бази знань та даних.

Таким чином розробка методів зв'язування інформаційних моделей об'ємно-планувальних, конструктивних і організаційно-технологічних рішень проекту будівництва для вирішення завдань вибору ефективних комплексних проектних рішень на всіх стадіях систем підготовки і управління

будівництвом і на всіх рівнях ієрархічного опису об'єктів і процесів будівництва є важливою вимогою часу. Ефективним рішенням тут є стиківка об'ємно-планувальної, конструктивної і організаційно-технологічної ієрархій з метою створення цілісної системи моделей будівельних об'єктів і процесів. Зв'язки між вузлами цих ієрархічних структур можуть забезпечити побудову системи комплексного сприйняття зведення об'єкта будівництва в часі.

Розглянута вище специфіка предметної області будівництва полягає у взаємному впливі і взаємозв'язку розглянутих структурних ієрархій, що створюють елементи нормативної бази з конструктивних і технологічних складових, а також ресурсної бази з конструктивних і об'ємно-планувальних складових для багатократного використання і параметризації.

На сьогодні існують різні підходи щодо врахування такої специфіки будівництва в інформаційних системах. Суттєве місце займає напрямок розвитку імітаційного моделювання. Імітаційне моделювання є загальним поняттям, що використовується в дуже широких тлумаченнях:

- ситуатійне моделювання, тобто будівництва та програмування в часі моделі системи в деякій ситуації з метою отримання стійких статистичних даних;

- метод дослідження реальної системи та її моделі з проведенням експериментів для уточнення опису системи;

- деякий випадок математичного моделювання об'єктів і процесів, коли аналітична модель замінюється на імітаційну.

Іноді імітаційним моделюванням називають отримання чисельних рішень за допомогою деяких евристичних численних методів. Крім того, сьогодні розвиваються такі типи імітаційного моделювання:

- агентне моделювання: використовується для дослідження систем, поведінка яких визначається не глобальними правилами і закономірностями, а навпаки, коли вони є результатом дій групи деяких активізованих в конкретних типових (нетипових) ситуаціях індивідуальних елементів цих систем;

- дискретно-подійне моделювання, коли функціонування системи представляється як хронологічна послідовність подій;

- системна динаміка, як різновид імітаційного моделювання поведінки системи побудованого на структурі моделей елементів і взаємодії між ними на основі причинно-наслідкових, зворотних зв'язків, затримок реакцій системи, впливу зовнішнього середовища тощо [4].

На сучасному етапі вважається за необхідно здійснювати пошук створення таких технологій імітаційного моделювання, в яких в тому чи іншому обсязі містяться ознаки всіх цих напрямків

імітаційного моделювання. У нашому випадку йдеться про створення комплексної прикладної багатофункціональної інструментальної технології імітаційного моделювання аналізу і комплексної оцінки проектних рішень з прийняття ОПР, КР, ОТР, УПР на всіх етапах системи підготовки і управління будівництвом.

З метою забезпечення гнучкості формування логіки об'єктів і процесів у будівництві в межах цілісної системи авторами запропонована мова моделювання і програмування будівельних технологій (АІМО) з візуальними інтерфейсами для зручного задавання і редагування елементів конструктивної і технологічної ієрархій та побудови логічних залежностей між елементами. При цьому для моделювання вузлів конструкцій можна програмувати логічні правила, формули, цикли і фрагменти моделі.

Специфіка предметної області будівництва обумовлює вимоги до мови АІМО, що є основною складовою розробленої імітаційної системи моделювання, аналізу і комплексної оцінки проектних рішень з прийняття ОПР, КР, ОТР, УПР на всіх етапах системи підготовки і управління будівництвом. Реалізація цих вимог має забезпечувати:

- можливість формування гнучких структурних (об'ємно-планувальних, конструктивних) і технологічних ієрархій як основних сутностей, і ресурсів як допоміжних сутностей, з багатократним використанням і параметризацією для мінімізації дублювання та систематизацією як «просторів імен» у C#;

- програмування логіки ієрархій;
- варіювання показників ієрархій залежно від вхідних параметрів;

- програмування моделей використання ресурсів;

- наявність потужної системи опису правил організації і технології процесів будівництва.

### **Елементи мови програмування і моделювання будівельних технологій «АІМО»**

Мова АІМО складається з таких елементів:

- Компілятор – дозволяє задавати структуру організаційно-технологічних моделей будівельних процесів на всіх етапах будівництва, перетворює вихідні коди мови АІМО у базу даних та знань, що описує повну модель зведення будівлі. Видає повідомлення про помилки, якщо такі зустрічаються.

- Віртуальна машина або засіб імітаційного моделювання процесів будівництва. Віртуальна машина дозволяє, варіюючи вихідні дані (ОПР, КР,

ОТР), ресурсні та інші обмеження, пріоритети і ступінь ризиків, виконувати моделювання структури організаційно-технологічних процесів з метою отримання варіантів термінів, вартості, потреб у ресурсах, для вирішення завдань всебічного аналізу інвестицій чи витрат в будівництві. Побудова організаційно-технологічних ієрархічних моделей процесів будівництва здійснюється згідно з об'ємно-планувальною і конструктивною ієрархіями будівель та споруд, тобто здійснюється моделювання вузлів ОПП і КР і зв'язків між ними, а також моделювання дерева організаційно-технологічних робіт. Дерево організаційно-технологічних робіт остаточно формується перед початком імітаційного моделювання. Цей процес має назву інстанціювання сутностей бази даних та знань, отриманих компілятором АІМО. Також виконується моделювання параметрів ресурсів, вартостей (на різних рівнях їх деталізації), пріоритетів, потоку постачання, з метою вирішення задач предметної області: демонстрації результатів імітаційного моделювання у вигляді діаграм Ганта, сітьових графіків, графіків використання ресурсів, кошторису, оцінки на етапі інвестування показників вартості і термінів будівельних процесів залежно від характеристик конструктиву, які, в свою чергу, варіюються залежно від критеріїв оптимізації, фінансування, постачання ресурсів, а також аналізу вартості будівництва залежно від варійованих джерел ресурсів і нормативних баз.

- Візуальне середовище розробки (IDE) являє собою програму, що надає можливість зручно програмувати на мові АІМО. Після компіляції надає можливість переглядати у вигляді дерева новоутворені ієрархічні структури, переглядати їх опис (атрибути).

Далі розглянемо математичні моделі і методи, що використовуються у різних складових інформаційної системи імітаційного моделювання, що створюється [3].

### Компілятор АІМО. Математична модель перетворення вихідних даних в ієрархічну структуру даних

Основна задача компілятора АІМО – перетворити вихідний код конструктивних і технологічних моделей, а також ресурсів у ієрархічні структури даних.

Для цієї мети використовуються класичні принципи побудови аналізаторів і компіляторів коду: синтаксичний розбір, лексичний аналіз, побудова дерев виразів, генерація і оптимізація байт-коду. Для математичної моделі тут використовується алгебра предикатів.

**Вузол Node.** Мова АІМО використовує принцип об'єктно-орієнтованого програмування задля формування ієрархічної структури. Базовим елементом ієрархічної структури є вузол *Node*. Підтипи вузлів описані далі.

Кожен вузол характеризується множинами зв'язків, полів та логіки коду.

#### 1. Множина зв'язків (relations):

$$Rel = (Rel_h, Rel_c).$$

Множина зв'язків поділяється на ключові елементи: відношення в ієрархії  $Rel_h$ , що дають змогу задавати таксономічну ієрархію класів сутностей з ОПП, КР, ОТР в їх ієрархічній структурі, і відношення вміщення  $Rel_c$ , що дають змогу задавати мереологічну ієрархію ОПП, КР, ОТР.

Відношення в ієрархії і відношення вміщення характеризуються певними аксіомами, які обумовлюють властивості ієрархій.

У термінах об'єктно-орієнтованого програмування відношення в ієрархії формуються механізмом наслідування класів, а відношення вміщення реалізуються завдяки опису класу або простору імен.

а)  $Rel_h$  – відношення в таксономічній ієрархії:

$$Rel_h = (Node_{parent}, Node_{childs}),$$

де  $Node_{parent}$  – батьківський вузол,  $Node_{childs}$  – множина дочірніх вузлів.

Аксіоми:

Нехай  $Parent(N_1, N_2)$  – предикат, що  $N_1$  батьківський до  $N_2$ .  $Child(N_1, N_2)$  – предикат, що  $N_1$  дочірній до  $N_2$ .

$$Parent(N_1, N_2) \leftrightarrow Child(N_2, N_1),$$

$$Parent(N_1, N_2) \rightarrow \neg Parent(N, N_2)$$

для будь-якого  $N \neq N_1$  (лише один батьківський вузол).

$$Parent(N_1, N_2) \leftrightarrow \begin{matrix} N_1 : Rel_h = (\dots, (N_2, \dots)) \\ N_2 : Rel_h = (N_1, (\dots)) \end{matrix}.$$

б)  $Rel_c$  – відношення вміщення (containing):

$$Rel_c = (Node_{container}, N_{contained}),$$

де  $Node_{container}$  – вузол, що вміщує даний вузол,

$N_{contained}$  – множина вузлів, що вміщуються даним вузлом.

Аксіоми:

Нехай  $Container(N_1, N_2)$  – предикат, що  $N_1$  вміщує  $N_2$ .  $Contained(N_1, N_2)$  – предикат, що  $N_2$  вміщує  $N_1$ .

$$\begin{aligned} & Container(N_1, N_2) \leftrightarrow Contained(N_2, N_1), \\ & Container(N_1, N_2) \rightarrow \neg Container(N, N_2) \end{aligned}$$

для будь-якого  $N \neq N_1$  (лише один вузол-контейнер).

$$Container(N_1, N_2) \leftrightarrow \begin{matrix} N_1 : Rel_c = (\dots, (N_2, \dots)) \\ N_2 : Rel_c = (N_1, (\dots)) \end{matrix}$$

**2. Поля характеризують безпосередньо властивості вузла.** Множина полів поділяється на власні поля, поля-параметри, що використовуються для параметризації вузла та поля опису правил створення екземплярів вузлів при інстанціюванні даного вузла.

Множина полів (fields):

$$Fields = (Fields_{own}, Fields_{params}, Fields_{chinst}),$$

де  $Fields_{own}$  – власні поля даного вузла  $Node$  ;

$Fields_{params}$  – поля-параметри вузла  $Node$  ;

$Fields_{chinst}$  – поля опису правил створення екземплярів вузлів при інстанціюванні даного вузла.

В термінах об'єктно-орієнтованого програмування поля представляються змінними (власні поля) і параметрами конструктора класу. Поля опису правил створення екземплярів вузлів при інстанціюванні даного вузла не мають прямого аналогу в об'єктно-орієнтованому програмуванні і реалізуються комбінуванням вищенаведених двох способів представлення.

### 3. Множина логіки коду (codes):

$$Codes = (Code_{own}, Code_{params}, Code_{chinst}),$$

де  $Code_{own}$  – власний код даного вузла  $Node$  ;

$Code_{params}$  – допоміжний код для ініціалізації вузла параметрами  $Fields_{params}$ ;  $Code_{chinst}$  – допоміжний код для інстанціювання вузлів при інстанціюванні даного вузла.

Логіка коду дає змогу формувати логіку предметної області в контексті даного вузла. Власний код даного вузла в термінах об'єктно-орієнтованого програмування – це вміст функцій класу. Допоміжний код для ініціалізації вузла параметрами  $Fields_{params}$  – це код конструктора класу у термінах об'єктно-орієнтованого програмування. Допоміжний код для інстанціювання вузлів при інстанціюванні даного вузла не має прямих аналогів у об'єктно-орієнтованому програмуванні і, як правило, є частиною коду конструктора класу.

### Підтипи вузлів:

$$NodeSubtypes = (ConstructionNode, TechnologyNode, ResourceNode).$$

АІМО підтримує різні підтипи вузлів, основними з яких є конструктивні, технологічні та ресурсні вузли.

Кожен з підтипів вузлів має додаткові елементи у множинах полів, логіки коду і множинах зв'язків, які зумовлені специфікою конкретного підтипу вузла.

У конструктивних вузлах можна задавати довільну кількість атрибутів. За замовчуванням вузли мають поля площі та об'єму. Ресурсні вузли мають передвстановлене поле вартості одиниці ресурсу, а кожен організаційно-технологічний вузол характеризується зв'язком з конструктивним вузлом, організацію і технологію створення якого він описує.

#### Конструктивні вузли. Construction Node:

$$Fields_{own} = (Fields_{own}Node, Square_{field}, Volume_{field}),$$

де  $Fields_{own}Node$  – власні поля, описані для базового, тобто батьківського за класифікаційною ієрархією по відношенню до конструктивного вузла

$Node$  ;  $Square_{field}$  – характеристика узагальненої (власної) площі вузла у одиницях площі;  $Volume_{field}$  – характеристика узагальненого (власного) об'єму у одиницях об'єму.

#### Організаційно-технологічні вузли.

Technology Node:

$$Fields_{params} = (Fields_{params}Node, ConstrNode),$$

де  $Fields_{params}Node$  – поля-параметри, описані для базового  $Node$  ;  $ConstrNode$  – конструктивний вузол, роботу над яким описує дана технологія.

#### Ресурсні вузли. Resource Node:

$$Fields_{own} = (Fields_{own}Node, Cost_{field}),$$

де  $Cost_{field}$  – характеристика базової вартості одиниці ресурсу.

**Процес інстанціювання.** Інстанціювання є перехідною ланкою між компіляцією і імітаційним моделюванням. Інстанціювання призначене для розгортання відношення в ієрархії  $Rel_h$  та відношення вміщення  $Rel_c$  у повну систему.

У термінах об'єктно-орієнтованого програмування процес інстанціювання схожий на процес створення екземплярів класу, при якому

ініціалізуються дані для поточного класу і ланцюжка батьківських класів.

$$Inst = Instantiat e(N),$$

*Instantiat e* – головна функція інстанціювання (мається на увазі розгортання відношень: побудування по шаблону класу екземпляра об'єкта); *N* – множина вузлів (*Node*) та підтипів,

$$Instantiat e = (Instantiat e_{Node 1}, Instantiat e_{Node 2}, \dots, Instantiat e_{Node N}).$$

Інстанціювання батьківського вузла *Instantiat e<sub>Node</sub>* можна поділити на інстанціювання полів вузла *Instantiat e<sub>fields</sub>*, інстанціювання коду вузла *Instantiat e<sub>code</sub>* та інстанціювання вкладених вузлів *Instantiat e<sub>child inst</sub>*:

$$Instantiat e_{Node} = (Instantiat e_{fields}, Instantiat e_{code}, Instantiat e_{child inst}) \text{ (для } Node \text{)}.$$

У свою чергу, інстанціювання полів *Instantiat e<sub>fields</sub>* поділяється на інстанціювання власних полів *Instantiat e<sub>fields own</sub>*, інстанціювання полів-параметрів *Instantiat e<sub>fields params</sub>* та інстанціювання полів опису правил створення екземплярів вузлів при інстанціюванні даного вузла *Instantiat e<sub>fields ch inst</sub>*:

$$Instantiat e_{fields} = F(Fields_{own}, Fields_{params}, Fields_{ch inst}, Rel_h),$$

$$Instantiat e_{fields own} = (F_{own}, N_{1F own}, N_{2F own}, \dots, N_{NF own}),$$

де *Parent(N<sub>i</sub>, N<sub>i-1</sub>)* (для  $1 < i \leq N$ ) та *Parent(N<sub>1</sub>, Node)*.

$$Instantiat e_{fields params} = (F_{params}, N_{1F params}, N_{2F params}, \dots, N_{NF params}),$$

де *Parent(N<sub>i</sub>, N<sub>i-1</sub>)* (для  $i = \overline{2, N}$ ) та *Parent(N<sub>i</sub>, Node)*.

Для кожної з цих множин інстанційована множина знаходиться як об'єднання повної множини у поточному класі з інстанційованими множинами ланцюжка батьківських класів.

За тим самим сценарієм проводиться інстанціювання коду:

$$Instantiat e_{code} = (Code_{own}, Code_{params}, Code_{ch inst}, Rel_h),$$

$$Instantiat e_{code own} = (Code_{own}, N_{1code own}, N_{2code own}, \dots, N_{Ncode own}),$$

де *Parent(N<sub>i</sub>, N<sub>i-1</sub>)* (для  $i = \overline{2, N}$ ) та *Parent(N<sub>i</sub>, Node)*.

### Комплекс математичних моделей віртуальної машини АІМО

Імітаційне моделювання віртуальної машини АІМО реалізує процес генерування, оцінки та вибору квазіоптимальної стратегії підходу до зведення будівельного об'єкта та детального плану будівництва або окремих його складових з оцінками строків, обсягів ресурсів, бюджету, ризиків тощо.

На рис. 1 наведена схема внутрішньої частини віртуальної машини АІМО. Стрілками показані потоки даних. Стрілки згруповані за типами потоків даних: параметри, ієрархії (структури), бази ресурсів, критерії оптимізації.



Рис. 1. Внутрішня частина віртуальної машини АІМО

Система імітаційного моделювання є складною системою, що обробляє великий обсяг даних, моделює повний цикл будівництва (ПЦБ) з багатокритеріальною оптимізацією. Віртуальна машина виконує оптимізацію за множиною критеріїв, кожен з яких має свій пріоритет. Для формалізації імітаційного моделювання повного циклу будівництва (ПЦБ) або його складових введемо такі позначення:

$$F(F_{o1}, F_{o2}, \dots, F_{on}) \rightarrow \min,$$

$F$  – головна функція оптимізації (комбінований критерій);

$F_{oi}$  – функції-критерії оптимізації,  $i = \overline{1, n}$ ;

$$F = \sum_{i=1}^n F_{oi} k_i, \text{ де } k_i \text{ – вага кожного критерію.}$$

Прикладами критеріїв оптимізації ПЦБ є, мінімізація потреби у фінансових потоках, матеріальних, трудових ресурсах, час виконання робіт. Система також дозволяє задавати користувацькі критерії, що виникають під час виконання робіт ПЦБ, наприклад, пріоритет використання певних ресурсів.

Конструктивні і організаційно-технологічні ієрархічні моделі проекту будівництва забезпечують такі дані:

1.  $IRW_{ij}$  – інтегральна потреба  $i$ -ї роботи у  $j$ -му ресурсі (в одинице-днях),  $1 \leq i \leq N_W$  ( $N_W$  – кількість робіт, загальна),  $1 \leq j \leq N_R$  ( $N_R$  – кількість ресурсів, загальна).

Інтегральна потреба  $i$ -ї роботи у  $j$ -му ресурсі  $IRW_{ij}$  стосується сумарної кількості ресурсів, потрібних для виконання певної роботи. При цьому на даному етапі імітаційного моделювання ще не визначені час початку роботи і час завершення роботи, її тривалість.

Загальна кількість робіт і ресурсів охоплює всі роботи ПЦБ, отже залежить від обсягів і складності будівлі. Загальна кількість робіт може значно перевищувати кількість робіт, описаних на мові АІМО, оскільки АІМО дозволяє багаторазово використовувати фрагменти і шаблони конструктивних і організаційно-технологічних ієрархій, які на даному етапі розгортаються у плоску (не ієрархічну) множину атомарних робіт.

2.  $LW_i$  – множина обмежень  $i$ -ї роботи ( $1 \leq i \leq N_W$ ),

$$LW_i = (LW_{iTimeMin}, LW_{iTimeMax}, LW_{iTimeStart} \dots),$$

де  $LW_{iTimeMin}$  – обмеження для  $i$ -ї роботи, а саме, її мінімальна тривалість;  $LW_{iTimeMax}$  – обмеження для  $i$ -ї роботи, а саме, її максимальна тривалість;  $LW_{iTimeStart}$  – директивне обмеження для часу початку  $i$ -ї роботи.

Класичний підхід враховує можливість варіювання тривалості роботи і кількості використаних ресурсів, залишаючи при цьому незмінною інтегральну потребу  $i$ -ї роботи у  $j$ -му ресурсі  $IRW_{ij}$ .

Проте, часто через технологічні особливості виконання, робота може конкретизуватися тільки у певних межах.

3.  $LW_{ij}$  – множина обмежень відношень пари  $i$ -та робота,  $j$ -та робота (обмеження може включати довільну кількість робіт – не тільки пару) ( $1 \leq i \leq N_W, 1 \leq j \leq N_W$ ),

$$LW_{ij} = (LW_{ijOrder}, LW_{ijAlter}, \dots),$$

$LW_{ijOrder}$  – обмеження порядку слідування робіт  $i$  та  $j$  (початок однієї роботи вимагає закінчення іншої, одночасність робіт, тощо);  $LW_{ijAlter}$  – можливість альтернативності робіт  $i$  та  $j$ .

Множина обмежень пари ( $i$ -та робота,  $j$ -та робота) включає одні з головних обмежень віртуальної машини АІМО, які описують такі класичні елементи планування ПЦБ, як правила слідування та залежності між роботами.

У доповнення до класичних обмежень (послідовних, паралельних) АІМО дозволяє задавати альтернативні роботи з гнучкою системою пріоритетів.

Для опису процесу імітаційного моделювання введемо поняття основної функції імітаційного моделювання, яка приймає як вхідні дані множини робіт і комплекс обмежень. Як вихідні дані ця функція видає конкретизовану модель виконання робіт ПЦБ або його складових з визначеними значеннями часу початку і закінчення кожної роботи поряд з іншими даними.

Основна функція імітаційного моделювання:

$$M_{Res} = M(IRW_{ij}, LW_i, LW_{ij}, LSR_i, LSC, \dots),$$

де  $LSR_i$  – множина обмежень потоку  $i$ -го ресурсу ( $1 \leq i \leq N_R$ );  $LSC$  – множина обмежень потоку

фінансування  $M_{Res} = (RW_{ij}(t), TS_i, TF_i, \dots)$ ;

$RW_{ij}(t)$  – інтенсивність споживання  $j$ -го ресурсу (в одиницях) у момент часу  $t$  у рамках  $i$ -ї роботи;  $TS_i$  – розрахунковий час початку  $i$ -ї роботи;  $TF_i$  – розрахунковий час закінчення  $i$ -ї роботи.

Інтенсивність споживання  $j$ -го ресурсу (в одиницях) у момент часу  $t$  у рамках  $i$ -ї роботи.

$RW_{ij}(t)$  – одна з основних компонент вихідних даних функції імітаційного моделювання, яка забезпечує повну загальну інформацію щодо використання ресурсів кожної роботи у рамках модельованої структури повного циклу будівництва або його складових.

Отримані дані можуть бути надалі представлені у табличній, графічній, ієрархічній формі. Відображення у ієрархічній формі забезпечується за рахунок здатності АІМО пов'язувати результуючі дані моделювання з вихідними даними.

Обмеження потоку ресурсів:

$$LSR_i = (LSR_{i1}, LSR_{i2}, \dots, LSR_{iN_O}),$$

$N_O$  - кількість обмежень.

$$LSR_{ij} = (t_S, t_F, S_{Max}, \dots),$$

$1 \leq i \leq N_R$ ;  $1 \leq j \leq N_O$ ;  $t_S$  – час початку обмеження;  $t_F$  – час кінця обмеження;  $S_{Max}$  – ліміт обмеження (інтегральний, за період).

Обмеження потоку ресурсів забезпечує гнучку систему задавання правил використання ресурсів (матеріальних ресурсів, транспортних засобів, будівельних машин та робочої сили) для виконання робіт ПЦБ.

Система може використовуватися для формування обмежень, пов'язаних з постачальниками та внутрішньою логістикою підприємства.

Перевірка критерію (для обмеження наявного обсягу  $j$ -го ресурсу):

$$\sum_{i=1}^{N_W} \int_{t_S}^{t_F} RW_{ij}(t) dt \leq LSR_{jk},$$

(це заданий інтегральний графік поставок матеріальних ресурсів, у користувацькому представленні можна задати складову залишків ресурсів на початок періоду)

$$1 \leq i \leq N_W, 1 \leq j \leq N_R, 1 \leq k \leq N_O.$$

Обмеження потоку фінансування:

$P_i(t)$  – вартість одиниці  $i$ -го ресурсу у момент часу  $t$ ,  $1 \leq i \leq N_R$ , або одинице-дня, якщо це людський чи машинний ресурс.

Критерій:

$$\sum_{i=1}^{N_W} \sum_{j=1}^{N_R} RW_{ij}(t) \cdot P(t) \leq LSC_k(t),$$

$$1 \leq i \leq N_W, \quad 1 \leq j \leq N_R, \quad 1 \leq k \leq N_{SC},$$

де  $LSC_k(t)$  – функція обмеження потоку фінансування для часу  $t$ ;  $N_{SC}$  – кількість обмежень потоку фінансування.

Система розраховує інтенсивність фінансування як суму інтенсивності усіх робіт, які виконуються у даний момент часу. Інтенсивності робіт обчислюються як добуток інтенсивності споживання  $j$ -го ресурсу (в одиницях) у даний момент часу  $t$   $i$ -ї роботи  $RW_{ij}(t)$  на вартість одиниці певного ресурсу  $P_i(t)$  у момент часу  $t$ .

АІМО дозволяє задавати довільне правило формування вартості ресурсів залежно від часу. Так, користувач може сформувати дану залежність, беручи до уваги зовнішні чинники компанії, а також враховуючі глобальні чинники, наприклад, фактор інфляції тощо.

**Критерій оптимізації.** АІМО підтримує широкий спектр можливостей задавання критеріїв оптимізації, які можуть формуватися на основі будь-яких даних, доступних в рамках імітаційного моделювання. Нижче наведені основні наперед визначені критерії.

**Потоки ресурсів.** Мінімізація пікового значення:

$$\text{Max}_{t=0}^{\text{Max}t} \left( \sum_{i=1}^{N_W} RW_{ij}(t) \right) \rightarrow \min$$

(для  $j$ -го ресурсу);

$$\text{Max}t \quad \text{– повний час моделювання,} \\ 1 \leq j \leq N_R.$$

Критерій можна широко використовувати у випадках задавання користувачем обмежень постачальників, складів, логістики у забезпеченні ресурсами робіт ПЦБ.

Стабільність потоку:

$$\text{Max}_{t=0}^{\text{Max}t} \left( \sum_{i=1}^{N_W} RW_{ij}(t) \right) \text{–} \\ \text{– Min}_{t=0}^{\text{Max}t} \left( \sum_{i=1}^{N_W} RW_{ij}(t) \right) \rightarrow \min$$



Критерій можна використовувати для забезпечення рівномірності споживання ресурсів, що є важливим для логістики підприємства.

**Оптимізація за часом.** Основною оптимізацією за часом є оптимізація загального часу виконання комплексу робіт (зведення об'єкта, його окремих частин тощо):  $Maxt \rightarrow \min$ ,  $Maxt$  - повний час моделювання.

Цей критерій є одним з найбільш поширених, однак, має, як правило, нижчий пріоритет, ніж інші критерії.

Мінімізація тривалості для певної роботи:

$$TF_i - TS_i \rightarrow \min, 1 \leq i \leq N_W,$$

$TS_i$  - час початку  $i$ -ї роботи;

$TF_i$  - час закінчення  $i$ -ї роботи.

Даний критерій є гнучким механізмом формування особливих обмежень користувача, які визначаються внутрішніми або зовнішніми чинниками підприємства.

**Оптимізація за інтегральною кількістю ресурсів:**

$$\sum_{i=1}^{N_W} \int_{t_S}^{t_F} RW_{ij}(t) dt \rightarrow \min$$

для  $j$ -го ресурсу  $1 \leq j \leq N_R$ ;  $t_S$ ,  $t_F$  - час початку і кінця періоду оптимізації.

**Оптимізація за інтегральними фінансовими затратами за певний період:**

$$\sum_{i=1}^{N_W} \sum_{j=1}^{N_R} \int_{t_S}^{t_F} RW_{ij}(t) \cdot P_j(t) dt \rightarrow \min,$$

$P_j(t)$  - вартість одиниці  $j$ -го ресурсу у момент часу  $t$ ;  $1 \leq i \leq N_W$ ,  $1 \leq j \leq N_R$ ,  $t_S$ ,  $t_F$  - час початку і кінця періоду оптимізації.

Оптимізація за інтегральною кількістю ресурсів та оптимізація за інтегральними фінансовими затратами за певний період є альтернативами оптимізації за інтенсивністю потоку споживання ресурсів, що дозволяє користувачеві формувати особливі обмеження на тривалий період часу.

Для оптимізації головної функції імітаційного моделювання віртуальної машини АІМО використовується динамічне програмування, яке полягає в тому, що переміщення функції мети до оптимуму відбувається крок за кроком, причому кожний крок визначається лише поточним станом і не залежить від попередніх подій.

Повний інтервал моделювання ПЦБ розбивається на невеликі етапи, розміри яких

варіюються, виходячи з користувацьких обмежень або інших даних.

Математичне представлення задачі динамічного програмування:

$F$  - головна функція оптимізації (комбінований критерій),  $\bar{x}$  - множина змінних,

$$F = (F_{o1}, F_{o2}, \dots, F_{on}),$$

$F_{oi}$  - функції-критерії оптимізації,  $i = \overline{1, n}$ ;

$n$  - кількість функцій-критеріїв оптимізації,

$$\bar{x} = (\overline{x_{o1}}, \overline{x_{o2}}, \dots, \overline{x_{on}}),$$

$$\sum_{i=1}^n F_{oi}(\overline{x_{oi}}) k_i \rightarrow \min,$$

де  $k_i$  - вага кожного критерію оптимізації,

$$\overline{x_{oi}} = (x_{oi1}(t), x_{oi2}(t), \dots, x_{oim_i}(t)), i = \overline{1, n},$$

$m_i$  - кількість змінних в  $i$ -й функції оптимізації.

Обмеження:

$$\sum_{j=1}^m a_{ikj} x_{ojk} \leq L_{ik}, i = \overline{1, n}, k = \overline{1, o},$$

$o$  - кількість обмежень для  $i$ -ї функції оптимізації.

**Загальний алгоритм динамічного програмування:**

1. Визначення керованих параметрів, етапів (кроків)  $t = 1, 2, \dots, T$  та стану системи після завершення управління  $S = (S_1(t), S_2(t), \dots, S_n(t))$ .

2. Попереднє визначення формул розрахунку виграшів  $\omega_t = f_t(S_t, X_t)$ , формул зміни стану системи  $S_{t+1} = \phi_t(S_t, X_t)$  та формул розрахунку ефективності

$$\omega_t = \{ \max(f_t(S_t, X_t) + \omega_{t+1}(\phi_{t+1}(S_{t+1}, X_{t+1}))) \}.$$

3. Виконання умовної оптимізації останнього кроку для визначення множини кінцевих станів  $\omega_T = f_T(S_T, X_T)$  та кроків  $t = T-1, T-2, \dots, 2, 1$ .

4. Виконання безумовної оптимізації, тобто переміщення за  $t = 1, 2, \dots, T$  із фіксацією оптимального розв'язку задачі.

Основним методом оптимізації є метод розгалужень та обмежень як частковий випадок динамічного програмування.

Кожний етап  $t = 1, 2, \dots, T$  - це окрема задача, яка розв'язується методом розгалужень і обмежень, а забезпечення цільової функції етапу - це вектор:  $Z = (z_1, z_2, \dots, z_n)$ .

Оптимум цільової функції на етапі  $t$  дійсний як відносно цього етапу, так і відносно наступних етапів  $t+1, t+2, \dots, T$ .

Використовується принцип лавиноподібного розгалуження згідно з принципом Беллмана [1].

На кожному етапі генерується велика кількість варіантів. Спочатку генеруються всі можливі стани  $X_{01}$ , потім відносно  $X_{01}$  генеруються всі можливі стани  $X_{02}$ , потім відносно  $X_{02}$  генеруються всі можливі стани  $X_{03}$  тощо.

Для кожного вузла розгалуження прогнозується максимально можлива оцінка – рекорд  $Z$ . Рекордом є досягнуте значення оцінки цільової функції або суб'єктивне значення, або обмеження.

Поточний стан розгалуження  $Z$  порівнюється з рекордом  $Z \leq \bar{Z}$ . Якщо порівняння неможливе, то всі  $Z$  запам'ятовуються з метою визначення в подальшому кращого  $Z$ .

На поточному етапі використання методу розгалужень і обмежень виконується відкидання – вибір одного чи декількох кращих розгалужень, остаточне відкидання найгірших та відкладання вбік поганих або сумнівних для подальшого аналізу.

В цілому, в даній задачі розглядається система високого рівня складності, існує багато варіантів рішень, отже на практиці не є можливим на 100% визначити оптимальне рішення.

Імітаційне моделювання дає змогу будувати моделі процесів, що описують, як ці процеси проходили б насправді.

Імітаційна модель — логіко-математичний опис поведінки об'єкта в різних ситуаціях, який може бути використаний для експериментування на комп'ютері в цілях проектування, аналізу і оцінки функціонування об'єкта.

Для моделювання повного циклу будівництва використовується метод Монте-Карло. Він об'єднує аналіз чутливості і аналіз розподілу ймовірностей вхідних змінних. Побудова моделі починається з визначення функціональних залежностей у реальній системі. Після чого можна одержати кількісний розв'язок, використовуючи теорію ймовірності й таблиці випадкових чисел.

$P_i$  – величина-параметр, що передається компілятором АІМО до віртуальної машини;  $F_{P_i}$  – функція розподілу вірогідності величини  $P_i$ .  $G(F_{P_i})$  – функція генерації випадкових значень величини  $P_i$  (базується на функції розподілу вірогідності).  $X_j = (x_{j1}, x_{j2}, \dots, x_{jN})$  – множина

значень величин-параметрів для ітерації моделювання  $j$ ,  $1 \leq j \leq k$ ,  $k$  – кількість ітерацій моделювання,  $N$  – кількість величин-параметрів.

$$X_{ji} = G_j(F_{P_i}). \quad Y_j = (y_{j1}, y_{j2}, \dots, y_{jN}),$$

$Y_j$  – множина значень розрахованих величин для ітерації моделювання  $j$ .  $Y_j = M(X_j)$ ;

$M$  – функція моделювання.  $G'$  – функція розрахунку функції розподілу вірогідності на основі множини значень розподілу вірогідності  $F_{Y_i} = G'(y_{1i}, y_{2i}, \dots, y_{ki})$ ,  $1 \leq i \leq N$ .  $P_i$ ,

$F_{P_i}$ ,  $G(F_{P_i})$  – визначаються специфічними особливостями факторів ризику у ПЦБ.  $F_{Y_i}$  – дозволяють аналізувати різні показники та величини (строки) залежно від ризиків.

#### Ризики і мова АІМО

Мова АІМО забезпечує гнучку систему обліку та врахування ризиків при виконанні моделювання організаційно-технологічних рішень виконання цільових комплексів будівельних робіт (проекування ПОС, ПВР, ОТК) на стадіях підготовки і управління будівництвом. Так чи інакше, значна частина ризиків може бути проаналізована експертом і представлена в канонічному вигляді, тобто:

- ймовірність виникнення ризику;
- наскільки при цьому змінюються змінні ризику.

Також АІМО дозволяє задавати ризики у варіанті розподілу ймовірності (залежність значення ймовірності виникнення ризику від значення величини змінної ризику).

Мова АІМО дозволяє задавати необмежену кількість ризиків, у тому числі взаємно залежних.

Змінні ризику мають бути представлені в контексті моделювання, тобто задані інженером і експертом АІМО в описі моделі. Ці змінні можуть мати різне логічне представлення.

• Вартість матеріальних ресурсів, загальні модифікатори вартості (інфляція тощо), або ж по компонентах (виникаючі дефіцити тих чи інших типів будматеріалів), ризики графіків постачань ресурсів.

• Ризики людських ресурсів, зарплатні ризики тощо.

• Технологічні ризики (збільшення термінів виконання робіт, виникнення додаткових робіт тощо) та інші.

Задавання змінних ризиків може відбуватися:

- у абсолютній формі (приріст/спад);
- у відносній;
- у процентній.

Основною особливістю АІМО є гнучкий механізм врахування і моделювання ризиків. Ризики можуть аналізуватися окремо, по групах або повністю всі. При цьому на виході (після моделювання) можуть бути набуті значення таких критеріїв оцінки ризику, як вартість, терміни та інші.

За бажанням користувача як вихідна інформація може бути представлений найбільш вірогідний хід розвитку подій (математичне сподівання), найгірший і найкращий сценарії, а також графіки розподілу ймовірності значень критеріїв. Останнє здійснюється в автоматичному режимі шляхом багатократного (ітеративного) моделювання технології з подальшим математичним аналізом результатів. Точність аналізу може змінюватися за бажанням користувача і впливати на тривалість процесу моделювання.

Віртуальна машина АІМО виконує такі операції:

- визначає область можливих вхідних даних, виходячи із заданих ризиків і вірогідностей моделі АІМО;
- випадковим чином генерує вхідні дані із визначеної вище області за допомогою деякого заданого розподілу ймовірностей;
- виконує детерміновані обчислення над вхідними даними;
- проміжні результати окремих розрахунків зводить у кінцевий результат.

### Висновки

1. Запропоновано концепцію комплексу математичних моделей і методів, як вимог, що мають бути реалізовані у середовищі мови програмування та імітаційного моделювання організаційно-технологічних рішень, призначенням якої є генерування варіантів ОПР, КР і ОТР, їх оцінки та вибору в системі підготовки і управління будівництва.

2. Запропоновано взаємозв'язані математичні моделі вирішення комплексу завдань підготовки будівництва і розроблена мова програмування і імітаційного моделювання будівельних технологій «АІМО», що складається з компілятора, віртуальної машини (засобу моделювання) та візуального середовища розробки (IDE) для програмування на мові «АІМО».

3. Розроблена мова є ядром прикладної інструментальної інформаційної технології для рішення задач організаційно-технологічної підготовки в будівництві РМВ (Pattern-Maker Building), яка створюється на основі розвитку сучасної концепції «конфігураторів».

### Список літератури

1. Беллман Р. Динамическое программирование / Р. Беллман — М.: Изд-во иностранной литературы, 1960.
2. Задоров В.Б. Интеграция информационных моделей в проектировании и управлении строительства на основе узагальненого формату знань та даних / В.Б. Задоров, О.О. Васильев // Управління розвитком складних систем: зб. наук. праць. — 2011. — № 05. — С. 52 — 60.
3. Задоров В.Б. К развитию концепции «конфигураторов» для построения архитектуры информационных технологий организационных антропогенных систем / В.Б. Задоров, А.А. Васильев // Управление развитием сложных систем: сб. науч. работ. — 2011. — Вып. 06. — С. 108 — 116.
4. Строгалев В.П. Имитационное моделирование: Учеб. пособие. / В.П. Строгалев, И.О. Толкачева. — М.: Изд-во МГТУ им. Н.Э. Баумана, 2008. — 280 с.
5. Типовые технологические карты на отделочные работы с применением комплектных систем КНАУФ. МДС 81-39.2005. Том 3. / ОАО «Тулаоргтехстрой», ООО «Кнауф Сервис». — М.: ООО РИФ «Стройматериалы», 2006. — 190 с.

Стаття надійшла до редколегії 19.06.2013

**Рецензент:** д-р техн. наук, проф. Ю.М. Тесля, Київський національний університет будівництва і архітектури, Київ.