



УДК 658.512.2

АЛГОРИТМІЧНЕ ПРОЄКТУВАННЯ ЯК НОВИЙ ЕТАП РОЗВИТКУ ТЕХНОЛОГІЙ ДИЗАЙНУ

БЕРДИНСЬКИХ Святослав, ГОРОДЕЦЬКА Вікторія
Університет економіки та права «КРОК», Київ, Україна
sviatoslavbo@krok.edu.ua, horodetskavo@krok.edu.ua

Здійснено аналіз технології алгоритмічного проєктування як прогресивного інструмента створення складних архітектурно-дизайнерських форм та оптимізації проєктних рішень. Проведено порівняння алгоритмічного підходу з традиційними CAD- та BIM-репрезентаціями, а також іншими обчислювальними технологіями, що використовуються у сфері проєктування та будівництва.

Ключові слова: алгоритмічне проєктування, CAD, BIM, обчислювальне проєктування, генеративне проєктування, дизайн середовища.

ВСТУП

Архітектурно-проєктна практика впродовж історії впроваджувала новітні технологічні досягнення, спричиняючи зміни у способах представлення та концептуалізації проєктних рішень. З поширенням персональних комп'ютерів і появою інструментів CAD на початку 1980-х років методи презентації дизайну впевнено еволюціонували від паперових до цифрових. Незважаючи на неприйняття фахівцями-проєктувальниками в період раннього розвитку, впровадження цифрових методів проєктування поступово зростало протягом наступних десятиліть, поки не набуло широкого розповсюдження. На сучасному етапі актуальним є втілення в дизайн новітніх процесів та методик.

ПОСТАНОВКА ЗАВДАННЯ

Метою роботи є системний огляд алгоритмічного проєктування та аналіз переваг у порівнянні з традиційними проєктними підходами.

РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ ТА ЇХ ОБГОВОРЕННЯ

Спосіб представлення різних елементів споруди та їхніх просторових взаємозв'язків на кресленнях, створених за допомогою CAD, залишався схожим на традиційний спосіб рукотворного виконання. Завдяки стандартизації графічного представлення архітектурні проєкти стали зрозумілими для широкого кола фахівців. Проте для некваліфікованих осіб розуміння архітектурних креслень може бути складним [1].

Виникнення BIM-інструментів суттєво вплинуло на архітектурні презентації. Із зростанням складності проєктів збільшилась і кількість спеціалістів, залучених до реалізації, що вимагало наявності єдиної центральної бази зберігання інформації. Така база має бути доступною усім учасникам проєкту та містити всю необхідну інформацію про процеси



проектування, будівництва, експлуатації й обслуговування будівлі. У цій системі об'єкт можна представити як у 2D, так і у 3D форматі, що зменшує проблему синхронізації результатів роботи різних фахівців та мінімізує відмінності між задумом проєкту і його реальним виконанням [1]. Однак існує кореляція між інтуїтивністю BIM і її гнучкістю: система обмежена фізичними взаємозв'язками та правилами конструктивної логіки. Вона ускладнює реалізацію об'єктів, яких неможливо кількісно оцінити, та перешкоджає розробці комплексних або нетрадиційних рішень [1].

Вирішити проблему гнучкості допомогло *обчислювальне проектування* (Computational Design, або CD), яке спричинило «обчислювальний поворот» у проектуванні будівель і здійснило революцію в традиційних процесах, які донедавна значною мірою ґрунтувалися на ручному створенні креслень [2].

Обчислювальне проектування (CD) сформувалося як дослідницька тема лише наприкінці 1990-х років, однак його початком можна вважати 60-ті роки XX ст. Воно зародилося під впливом модерністських ідей та технологічних досліджень [2]. Проте протягом останніх двох десятиліть CD досягло значного прогресу, перевершуючи рівень автоматизації креслярських завдань, інтегруючи різноманітні обчислювальні технології та нові методи конструювання. Це відкрило шлях до інновацій у проектуванні. Архітектори прийняли парадигму CD не лише як засіб оптимізації робочих процесів, а й як можливість досліджувати нові напрями в архітектурній творчості [2]. Одним з прикладів обчислювального дизайну є *алгоритмічне проектування* (AD), яке значно розширило можливості творчого процесу архітекторів [1].

AD – це процес проектування, заснований на алгоритмах. Основна увага його зосереджена на демонстрації послідовності дій, кроків і методів, які приводять до формування ідеї дизайну, а не на остаточному результаті. Алгоритм передає інформацію про дизайн у вигляді загальної абстрактної моделі, яка не має фіксованих характеристик до моменту, коли задаються конкретні параметри для його виконання. Цей підхід забезпечує гнучкість у проектуванні, оскільки дозволяє адаптувати проєкт до різних умов і потреб, просто змінюючи параметри алгоритму. Однак це визначає відмову від безпосередньої візуалізації простору проектування, що частково перешкоджає розумінню візуальної складової. Для порівняння CAD- та BIM-репрезентації мають діаметрально протилежну філософію, надаючи перевагу результату над процесом, що зменшує їхню гнучкість, але полегшує розуміння проєктного рішення [1]. Концепції, які складно передати, часто залишаються незрозумілими через недостатню ефективність наявного методу репрезентації. У AD правила проектування викладаються явно, тоді як у CAD та BIM вони лише передбачаються загальним макетом результату. Це може призводити до нечіткої ідентифікації правил, що викликає хибне тлумачення задуму проєкту та зміни, які порушують початкову ідею. Варто зазначити, що парадигма AD не веде до кращого розуміння дизайну. Проте, у поєднанні з візуалізацією можливих варіантів, AD дає змогу побачити можливості та ефект від змін параметрів і правил [1].

В алгоритмічному проектуванні геометричні відношення між елементами виражаються безпосередньо через алгоритмічний опис,



використовуючи такі операції як віднімання, об'єднання чи перетин. Однією з важливих особливостей AD є збереження інформації про примітивні фігури та встановлені між ними взаємовідношень. У CAD цього можна досягти шляхом комбінування фігур за допомогою попередньо визначених подібних операцій. Натомість у BIM ці операції, як правило, не існують у явному вигляді, оскільки їх складно перекласти на імітоване конструктивне середовище [1]. З точки зору зрозумілості, CAD і BIM приховують послідовність операцій, які призводять до появи кінцевої форми. Тому визначення того, як був отриманий результат, може бути неочевидним. У парадигмі AD розуміння результату значно залежить від назв, які використовуються для опису концепцій дизайну в програмі. Розуміння того, «як» було отримано результат, в AD часто є явним, тоді як розуміння того, «що» є результатом, вимагає виконання алгоритмічного опису для його візуалізації. На противагу цьому, в парадигмах CAD та BIM розуміння «що» є простішим ціною втрати «як» [1]. CAD і BIM обмежені у виразності щодо рекурсії та випадковості, адже їхній процес створення є надзвичайно трудомістким і вимагає багато часу. Представлення самих рекурсивних правил часто неможливе, що призводить до непараметричних описів, в яких зміна одного елемента не впливає на інші. Тому для тестування нових варіантів доводиться починати весь процес спочатку. На відміну від цього, AD забезпечує параметричний опис рекурсивних правил, дозволяючи автоматично і швидко генерувати варіанти дизайну. Хоча абстрактне представлення рекурсії в AD може бути менш інтуїтивним, воно значно ефективніше для складних правил, які важко сприймати візуально [1].

Незважаючи на сучасний вигляд, AD є природним наслідком прагнення автоматизувати завдання моделювання – архітектор розробляє алгоритми, реалізація яких створює цифрову модель проєкту замість того, щоб вручну моделювати її за допомогою цифрового інструменту проєктування. Порівняно з традиційними процесами цифрового моделювання, AD має переваги точності, гнучкості, автоматизації та легкості змін, що дозволяє легко та швидко досліджувати ширші простори проєктування [3].

Ведуться дискусії щодо того, чи є алгоритмічне проєктування (AD) тотожним іншим напрямом обчислювального дизайну, зокрема *параметричному* (PD) та *генеративному* (GD) проєктуванню. *Параметричне проєктування* (PD) символічно описує дизайн за допомогою параметрів, дозволяючи змінювати їх значення для створення різноманітних варіантів. Натомість алгоритмічне проєктування (AD) базується на використанні алгоритмів - наборі логічних інструкцій для автоматичної генерації форм або процесів. Отже, PD акцентує увагу на змінності та налаштуванні, тоді як AD фокусується на побудові цілісної логіки [2]. *Генеративне проєктування* (GD) – парадигма проєктування, що використовує більш автономні алгоритмічні описи порівняно з параметричним проєктуванням (PD). У межах GD після запуску генеративного процесу система виконує задовані інструкції доки не буде виконано критерій зупинки, що дозволяє навіть з простих алгоритмічних описів отримувати складні результати. Однак через складність таких алгоритмів їх часто важко співвіднести із кінцевим результатом – передбачити результат лише на основі алгоритмічного опису стає непросто.



AD також можна віднести до генеративного, оскільки воно використовує алгоритми для створення моделей. Втім, в AD існує взаємозв'язок між алгоритмом і згенерованою моделлю, що забезпечує можливість відстеження зв'язку і дозволяє користувачеві ідентифікувати частини алгоритму, яка згенерувала певну частину моделі. У певному сенсі, AD є ізоморфним до моделі. Таким чином, AD можна розглядати як підмножину GD, яка забезпечує передбачуваність результатів завдяки меншій кількості несподіваних ефектів. Цей підхід також забезпечує точніший ступінь контролю, спрощуючи налагодження й обслуговування [2].

Більшість інновацій переживали період неприйняття. Випадок з AD не є винятком, оскільки він передбачає витрату часу на вивчення нового методу репрезентації, що є проблемою для освоєння архітекторами. Однак, завдяки навчанню, існуючі бар'єри розуміння можуть бути усунені, і AD може бути інтегрована в загальну архітектурну практику репрезентації [1].

ВИСНОВКИ

AD - це підхід до проєктування на основі алгоритмів, який набуває все більшого поширення як в архітектурній практиці, так і в теорії завдяки значній свободі проєктування та здатності автоматизувати повторювані проєктні завдання, полегшуючи внесення змін до проєкту та пошук досконаліших рішень. Його численні переваги спонукають нове покоління архітекторів та дизайнерів частіше використовувати програмні середовища, які для них не є типовими інструментами моделювання, виходячи за межі встановлених обмежень поточного програмного забезпечення [3].

ЛІТЕРАТУРА

1. Castelo-Branco R., Caetano I., Leitão A. Digital representation methods: The case of algorithmic design, *Frontiers of Architectural Research*. 2022. Volume 11. P. 527-541. DOI: <https://doi.org/10.1016/j.foar.2021.12.008>
2. Castelo-Branco R., Caetano I., Leitão A. Algorithmic Representation Space. *Published at B-pro Prospectives Journal, Bartlett School of Architecture UCL*. 2022. <https://journal.b-pro.org/article/algorithmic-representation-space/>
3. Caetano I., Santos L., Leitão A. Computational design in architecture: Defining parametric, generative, and algorithmic design. *Frontiers of Architectural Research*. 2020. Volume 9. P. 287-300. DOI: <https://doi.org/10.1016/j.foar.2019.12.008>

BERDYNISKI H. S., HORODETSKA V.

VISUALIZATION USING ARTIFICIAL INTELLIGENCE TOOLS IN ENVIRONMENTAL DESIGN

The article considers algorithmic design technology as a modern approach to creating complex architectural forms and optimizing design solutions. The algorithmic design technology is compared with traditional CAD and BIM representations, as well as other computational technologies used in the field of design and construction.

Key words: *algorithmic design, CAD, BIM, computational design, parametric design, generative design.*