

УДК 72.02:721

DOI <https://doi.org/10.32782/2415-8151.2025.35.35>

ТЕРМІНОЛОГІЇ ОБЧИСЛЮВАЛЬНОГО ДИЗАЙНУ. ПРОБЛЕМАТИКА ВИЗНАЧЕННЯ ТЕРМІНІВ

Откидач Андрій Іванович

аспірант кафедри дизайну та основ архітектури,
Національний університет «Львівська політехніка», Львів, Україна,
e-mail: andrii.i.otkydach@lpnu.ua, orcid: 0009-0008-1847-872X

Анотація. У статті розглянуто проблему неоднозначності та відсутності уніфікації термінології в обчислювальному дизайні, зокрема понять «генеративний дизайн», «алгоритмічний дизайн» та «параметричний дизайн» в українському науковому середовищі. Проведений аналіз англomовних досліджень дав змогу виявити основні трактування цих понять, ключові особливості кожного підходу, їхні відмінності та спільні риси, а також специфіку їх застосування.

Мета статті – розглянути термінологію в англomовному науковому середовищі, визначити відмінності між методами, запропонувати уніфікацію термінології обчислювального дизайну в Україні.

Методологія. Для досягнення цілей дослідження проаналізовано найуживаніші терміни обчислювального дизайну. Із нього відібрано найбільш уживані в літературі й ті, що заслуговують на уточнення, тобто терміни з еквівалентним уживанням: «обчислювальний дизайн», «генеративний дизайн», «параметричний дизайн», «алгоритмічний дизайн» у літературі. Такий аналіз слугує орієнтиром для розробки запропонованої термінології в українському дослідницькому й практичному середовищі дизайну та архітектури, яка визначає концептуальну й операційну сферу застосування кожного терміна обчислювального дизайну, уникає еквівалентного використання різних термінів та уточнює можливі комбінації між ними.

Дослідження передбачає три етапи: 1. Визначити найбільш релевантні терміни обчислювального дизайну. 2. Зібрати та порівняти визначення термінів. 3. Запропонувати послідовну й обґрунтовану термінологію обчислювального дизайну.

Результати. Описано та уніфіковано поняття «обчислювальний дизайн», «генеративний дизайн», «параметричний дизайн», «алгоритмічний дизайн», описано ієрархічну модель, яка демонструє взаємозв'язки між термінами, зведено в матрицю унікальні особливості та обмеження цих методів.

Наукова новизна. Уперше в українській науковій діяльності відмежовані такі поняття: «обчислювальний дизайн», «генеративний дизайн», «алгоритмічний дизайн», «параметричний дизайн».

Практична значущість. Результати дослідження та пропозиція уніфікації термінів можуть сприяти уніфікації термінології, що полегшить наукову комунікацію між дослідниками та практиками, запропонована модель взаємозв'язків між термінами й матриця особливостей та обмежень цих методів може бути використана для створення навчальних матеріалів з обчислювального дизайну.

Ключові слова: обчислювальний дизайн, генеративний дизайн, алгоритмічний дизайн, параметричний дизайн, цифровий дизайн, методи дизайну, термінологія, автоматизація дизайну.

ВСТУП

Обчислювальний дизайн є інноваційним підходом у дизайні, який стрімко розвивається в архітектурі та дизайні. Цей підхід дизайну використовує алгоритмічні процеси й обчислювальні потужності комп'ютерних систем для створення великої кількості варіантів дизайнерських рішень. Наукова діяльність цього підходу в дизайні як в Україні, так і в усьому світі лише встановлюється й формується, у її розвитку бере участь значна кількість спеціалістів-практиків у проєктних бюро чи спеціалістів з комп'ютерних технологій. Тому процеси формування термінологічної бази супроводжуються проблемою неоднозначності й різноманітності визначення термінів навіть в англійській літературі, їх смислового наповнення чи застосування різних слів для визначення одного поняття чи методу. Для українського наукового середовища відсутність єдиних визначень понять, як-от «генеративний дизайн», «параметричний дизайн» та «алгоритмічний дизайн», створює труднощі для українських науковців і практиків, додається також проблема перекладу цих термінів і визначення часто інтерпретують із неповним змістом або посилаються застарілою інформацією з англійських джерел. Це ускладнює обмін знаннями, перешкоджає співпраці між дослідниками та призводить до плутанини в академічних і практичних контекстах. Актуальність цього дослідження зумовлена необхідністю уніфікувати термінологію в Україні для забезпечення ефективного комунікаційного поля між фахівцями у сфері обчислювального дизайну та для сприяння подальшому розвитку обчислювального дизайну в українській архітектурній школі.

АНАЛІЗ ПОПЕРЕДНІХ ДОСЛІДЖЕНЬ

Аналіз останніх досліджень у галузі обчислювального дизайну демонструє значний інтерес до інноваційних методів формування в архітектурі та дизайні. Одними із важливих є роботи Інес Каetano, у яких авторка досліджує методи проєктування, пов'язані з обчислювальним дизайном, як-от «генеративний дизайн», «параметричний дизайн» та «алгоритмічний дизайн».

У книзі Джона Фрейзера «Еволюційна архітектура», у якій автор досліджує основи генеративного та еволюційного дизайну для генерації форми, описується, як можна застосовувати принципи еволюції до архітектури й проєктування.

Проаналізовано дослідження Рівки Оксман та сучасні розуміння параметричного дизайну як концептуального й практичного підходу в архітектурі та дизайні. Рівка Оксман

детально аналізує різні теорії, що пояснюють процес параметричного дизайну, включно з алгоритмічними підходами, системним мисленням та інтерактивним проєктуванням.

Бранко Коларевич одним із перших почав досліджувати параметричний дизайн як процес, що ґрунтується на змінних параметрах, а не на статичних формах. Його робота стала основою для подальших досліджень у цій сфері, зокрема для розуміння взаємозв'язків між параметричним та генеративним дизайном.

Проаналізовано книгу Костаса Терзідіса «Алгоритмічна архітектура», яка є однією з ключових робіт у сфері алгоритмічного дизайну. У своєму дослідженні Терзідіс акцентує на алгоритмічному мисленні як основі процесу проєктування, пропонуючи новий підхід до архітектури, що базується на алгоритмах та логічних операціях. Автор визначає алгоритмічний дизайн як парадигму, яка дає змогу створювати складні елементи та структурні форми за допомогою математичних інструкцій і коду.

МЕТА

Мета статті – розглянути термінологію в англійському науковому середовищі, визначити відмінності між методами, запропонувати уніфікацію термінології обчислювального дизайну в Україні.

РЕЗУЛЬТАТИ ТА ОБГОВОРЕННЯ

Архітектори й дизайнери завжди намагаються використовувати інноваційні ідеї, матеріали, техніки та нові підходи для покращення процесу проєктування. Справжньою революцією в підході до проєктування стало застосування цифрових технологій у середині XX століття. Першими продуктами в цифровому дизайні були примітивні програми, здатні відтворювати геометричні фігури, пізніше – 3D-геометрія. Тому коли професіонали почали застосовувати комп'ютери в дизайні природним чином, новий підхід стали називати «цифровим дизайном» (англ. Digital design) – поняття, що застосовується до характеристики процесу проєктування за допомогою комп'ютера як інструмента [6].

В українському науковому середовищі поява «цифрового» феномену в дизайні на перетині століть породила багато назв. Одне з найчастіше вживаних – «дигітальна» – фактично транслітерація з англійської мови слова *digital*, яке в перекладі з англійської мови означає «цифровий», тобто результати якого відображаються на цифрових моніторах. Тому використання терміна зі словом «дигітальний» є англіцизмом, правильний переклад терміна *digital design* українською – «цифровий дизайн».

Цифрові технології дали поштовх у розвитку вузькоспеціалізованим програмам для дизайну, як-от «система автоматизованого проєктування» (англ. Computer-aided design), що визначають як використання комп'ютерів для допомоги у створенні, модифікації, аналізі чи оптимізації дизайну [14].

На початку XXI століття в англomовній літературі виникло поняття «обчислювальний дизайн» (англ. computational design), який суттєво відрізняється від традиційних методів цифрового дизайну, оскільки ґрунтується на представленні дизайну, на його обчислювальній логіці, а не на геометричних аспектах [5].

Обчислювальний дизайн – це сучасний та швидко прогресувальний напрям в архітектурі й дизайні. Цей метод використовує алгоритмічні процеси та потужності комп'ютерних систем, щоб створювати безліч різноманітних дизайнерських рішень.

В англomовній літературі виокремлюють відмінності в поняттях «обчислювальний дизайн» та «цифровий дизайн», які полягають у тому, що цифровий дизайн розглядається як спосіб використання комп'ютерних інструментів у процесі проєктування, тоді як обчислювальний передбачає використання обчислень комп'ютера для розробки проєктів. У цій перспективі обчислювальний дизайн є ортогональним до цифрового. Тобто можна використовувати обчислювальний дизайн без цифрових інструментів, можна застосовувати цифрові інструменти й не спиратися на обчислювальний дизайн або використовувати обидва [6].

Тож термін «цифровий дизайн» в англomовній літературі стосується саме способу виконання діяльності з використання комп'ютерних програм як-от CAD, BIM, 3D моделювання та інші.

За останні два десятиліття дизайнери й архітектори затвердили концепцію обчислювального дизайну як спосіб удосконалення типових процесів проєктування та пришвидшення роботи. Використання обчислювального дизайну часто вимагає спеціалізованого досвіду й мислення, що змушує дизайнерів здобувати додаткові знання з інших галузей, зокрема програмування. У результаті комбінації різних полів науки з'явилися нові підходи та методи дизайну цього напрямку, з яких виникли нові терміни. Деякі з них або мають неоднозначне визначення, або охоплюють два чи більше суперечливих понять, або перетинаються в значенні з іншими термінами, як-от «параметричний дизайн», «генеративний дизайн» та «алгоритмічний дизайн». Ці терміни використовують у літературі найбільше й найчастіше як ключові слова, часто вживаючи їх паралельно та

плутаючи між собою. І параметричний, і генеративний, й алгоритмічний дизайн – це терміни, що з'явилися майже одночасно в літературі в 1970-х роках і є частиною поняття обчислювального дизайну [6].

Вони пропонують архітекторам нові шляхи проєктування, порушуючи передбачувані зв'язки між формою та репрезентацією створених людиною на користь комп'ютерно генерованих, дозволяючи так розробку нових типологій, виявлення більшої кількості варіантів виконання того ж завдання. Вони зміщують акцент зі «створення форми» на «пошук форми» [3].

Розглянемо ці терміни, проаналізуємо в порівняльній характеристиці та знайдемо суперечності. Для цього звернемося до мети використання обчислювальної методики дизайну, а саме – формоутворення, що є одним із важливих питань у дизайні та архітектурній практиці.

Параметричний дизайн дослідники описують як досить теоретично, так і досить філософськи, наприклад, як підхід, який описує дизайн символічно, на основі використання параметрів або параметрика – це більше про спосіб мислення, який деякі дизайнери можуть уважати чужим, але першою вимогою є мислення, яке прагне виразити й дослідити взаємозв'язки [6; 19].

Параметричний дизайн розглядають як створення геометрії за допомогою комп'ютерних розрахунків, що автоматично оновлюються та візуалізуються на екрані після зміни параметра. Це процес використання геометричних обмежень, а також співвідношення розмірів і даних для визначення форми. Процес, за допомогою якого оголошуються параметри конкретного дизайну, а не його форма – так можна створити кілька рішень [12; 15].

Параметричний дизайн може генерувати креативні рішення, мати справу зі складними зв'язками та керувати ними параметрично, а також використовувати параметри для встановлення зв'язків між елементами дизайну з метою визначення діапазону формальних альтернатив [8].

За принципом роботи параметричний дизайн визначають як математичний, де взаємозв'язок між елементами дизайну відображається у вигляді параметрів, які можна переформулювати для створення складних геометрій. Ці геометрії базуються на параметрах елементів, змінюючи їх, водночас створюються нові форми [8].

Параметричне проєктування дає змогу архітекторам і дизайнерам визначати ключові параметри своєї моделі та вносити зміни

в інтерактивному режимі. Коли вносяться зміни, решта моделі оновлюється автоматично [18].

Як один з прикладів – BIM-проектування, яке має ознаки параметричного дизайну. У разі зміни значення одного параметра змінюються інші пов'язані параметри того ж самого об'єкта чи інших об'єктів, групи об'єктів чи характеристики їх взаємодії.

Розглянемо термін «генеративний дизайн». В українській літературі термін «генеративний дизайн» плутають з терміном «алгоритмічний дизайн». Адже обидва способи дизайну потребують використання алгоритмів для виконання завдань.

Відповідно до Кембриджського словника, алгоритм – це набір математичних інструкцій або правил, які допоможуть обчислити відповідь на проблему. Дослідження генеративного дизайну розпочалися на початку 1970-х років, і джерелом цього дизайну є алгоритми, що імітують природу [10].

Генеративний дизайн, який належить до методів обчислювального проектування, може автоматично проводити проектні дослідження в умовах обмежень та параметрів, які визначені дизайнером. Його визначають як методологію проектування, що відрізняється від інших підходів до дизайну тим, що в процесі проектування дизайнер взаємодіє з матеріалами та виробами не безпосередньо в ручному режимі, а через генеративну систему [9].

Генеративний дизайн описують як метод проектування, де генерація форми базується на правилах або алгоритмах, часто похідних від обчислювальних інструментів, як-от Processing, Rhinoceros, Grasshopper та інші скриптові платформи [3].

Процес генеративного проектування зазвичай містить розробку алгоритму та його похідні, налаштування параметрів і форми, а також вибір найкращого варіанта. Деякі з найбільш використовуваних генеративних систем вміщують системи на основі правил (граматики форм, L-системи та фрактали), системи самоорганізації (клітинні автомати та ройові системи) та обчислювальну геометрію (діаграми Вороного та алгоритм пошуку A*) [1].

Генеративний дизайн долає недоліки традиційних ручних методів проектування шляхом використання обчислювальної потужності для розв'язання проблем швидкості, точності, складності, а також розвиток винахідливості, оскільки він збільшує кількість і діапазон варіацій, які містять «щасливі випадки», тобто несподівані результати, що позитивно впливають на процес проектування [7].

Використання цього методу дизайну стосується будь-якої практики проектування або

дизайну, де дизайнер використовує систему комп'ютерної програми, для вирішення проблеми проектування з певним рівнем автономності [2].

Генеративний дизайн уперше застосований у проєктах з проблемами невідомих результатів, які характеризуються великими обсягами даних для проектування. У цьому контексті термін «генеративний дизайн» стосується низки інструментів, що реалізують методи та алгоритми штучного інтелекту, які застосовують для виконання проектних завдань. З практичного боку, інструменти генеративного дизайну шукають рішення проблеми, вираженої математичним рівнянням. Це часто призводить до ітеративного процесу оптимізації, який намагається мінімізувати цільове значення. Відповідно, генеративний дизайн виявився корисним для виявлення незвичайних рішень, які не підпадають під типовий набір форм або конфігурацій, що використовують у процесі дизайну [4].

Тож можемо підсумувати: генеративний дизайн – це поняття, яке визначає спосіб дизайну, заснований на алгоритмах, що дають змогу створювати численні варіанти рішень на основі вихідних параметрів, який за допомогою комп'ютера та автономності в процесі самостійно генерує форми, часто на основі еволюційних чи стохастичних моделей.

Алгоритмічний дизайн – це метод дизайну, який використовує алгоритми для створення моделей. Алгоритмічний дизайн стає важко відрізнити від генеративного дизайну. Однак в алгоритмічному дизайні є кореляція між алгоритмом і генерованою моделлю, що забезпечує можливість відстеження та дає змогу користувачеві ідентифікувати частини алгоритму, які згенерували певну частину моделі. Відповідно до цього визначення, алгоритмічний дизайн є підмножиною генеративного, де розробка алгоритму зосереджена на передбаченому дизайні завдяки отриманню меншої кількості несподіваних результатів [6].

Алгоритмічний дизайн визначають як метод проектування, що базується на створенні комп'ютерних програм, які генерують простір і форму, використовуючи логіку, засновану на правилах, що відповідають архітектурним завданням, типологіям і будівельним нормам. Так, алгоритмічний дизайн дає змогу дизайнерам використовувати обчислювальну можливість і креативне використання комп'ютерів у робочому процесі проектування, створювати складну геометрію з використанням відносно простих правил і зв'язків [17; 20].

У цьому процесі генерація форм контролюється визначеними дизайнером сценаріями,

складеними в текстовому або графічному синтаксисі. Сценарії дають змогу дизайнерам генерувати та контролювати результати, які інакше були б надто складними для ручного моделювання [11]. В алгоритмічному дизайні є можливе використання сценаріїв, які допомагають користувачеві проєктувати безпосередньо за допомогою взаємодії з кодом [13]. Програму, яка створює модель будівлі, окремо створюючи її плити, колони, балки, стіни, вікна тощо, варто вважати прикладом алгоритмічного дизайну, оскільки відстеження частин коду, які створюють певну частину модель, легка [6].

Тож наведені визначення в літературі описують алгоритмічний дизайн як процес, що безпосередньо використовує програмування або математичні алгоритми для створення та контролю форми. Основна різниця з генеративним дизайном полягає в чіткій залежності результату від алгоритму, що дає змогу відстежити зв'язок між алгоритмічним кодом і створеною формою, на відміну від генеративного дизайну, де зв'язок може бути менш передбачуваним.

Розглянувши принцип роботи наведених способів, можемо окреслити спільне та відмінне в цих методах. Порівнюючи «параметричний» і «генеративний» дизайни, обидва способи працюють із параметрами для створення варіацій форм. Однак генеративний дизайн використовує більш автономні алгоритмічні описи, що може призвести до непередбачуваних результатів, тоді як параметричний дизайн передбачає контрольоване й передбачуване варіювання параметрів.

У підходах генеративного дизайну після запуску генеративного процесу, система виконує закодовані інструкції, поки не буде виконано «критерій зупинки». Натомість параметричний – залежний та контрольований користувачем.

Генеративний та алгоритмічний дизайн: обидва залучають алгоритми, але в генеративному дизайні часто використовують еволюційні алгоритми або методи машинного навчання, що дають змогу отримувати нові варіанти, тоді як в алгоритмічному дизайні відбувається чітке дотримання алгоритму без еволюційних змін. Алгоритмічний дизайн також забезпечує відстеження зв'язку між кодом і формою, що часто є нечітким у генеративному підході

Параметричний і алгоритмічний дизайн: суперечності виникають з огляду на залежність параметрів та алгоритмів. Параметричний дизайн використовує параметри як основу для варіацій, але не вимагає використання алгоритмічного програмування, тоді як алгоритмічний дизайн передбачає конкретні математичні алгоритми для контролю форм і структур.

ВИСНОВКИ

Усі три методи – параметричний дизайн, генеративний дизайн і алгоритмічний дизайн – працюють із параметрами й алгоритмами, використовують їх для створення варіацій форм і структур. Їхньою спільною метою є автоматизація процесу дизайну та надання можливості досліджувати велику кількість варіантів, що може бути обмеженим у традиційному підході. Окрім того, усі методи залучають певний рівень програмування або обчислень, що дає змогу дизайнерам керувати процесом формоутворення більш ефективно та експериментувати з новими рішеннями.

Унікальні особливості й обмеження кожного із цих підходів можемо узагальнити в матриці.

Так, хоча всі три методи прагнуть автоматизувати процес дизайну, кожен із них має свій акцент: параметричний дизайн забезпечує суворий контроль, генеративний надає творчі варіанти з певною автономністю, а алгоритмічний метод пропонує точне дотримання алгоритму з можливістю відстежувати процес. Ці відмінності є особливо важливими для розуміння їх застосування в різних проєктах, що вимагають різного рівня контролю, передбачуваності та креативності.

Тож урахувавши ключові особливості та принципи застосування цих методів, можемо навести пропозиції для уніфікації термінів.

Параметричний дизайн – підхід, який використовує параметри для варіювання форми, забезпечуючи передбачуваність і контроль над процесом.

Генеративний дизайн – процес автоматизованого створення нових варіантів дизайну за допомогою алгоритмів, які генерують форми відповідно до заданих параметрів і критеріїв. Характеризується автономністю процесу та можливістю створення нових, непередбачуваних варіантів.

Алгоритмічний дизайн – метод, у якому форма створюється за допомогою математичних або програмних алгоритмів, що забезпечує чіткий контроль і відстеження взаємозв'язку між кодом та кінцевою формою на кожному етапі проєктування.

Обчислювальний дизайн – загальний термін, що охоплює всі процеси дизайну, які використовують обчислювальні інструменти та алгоритми, включно з параметричним, генеративним й алгоритмічним підходами.

Результат аналізу забезпечує основу для подальших досліджень в обчислювальному дизайні та його методах з адаптацією до потреб в українській архітектурній школі.

Таблиця 1

Матриця характеристик методів обчислювального дизайну

	Параметричний	Генеративний	Алгоритмічний
Параметри	присутні, прогнозовані; визначені параметри рівнозначні або зав'язані у стійку залежність	присутні, визначені параметри лише ті, що мають критичне значення; невідомі параметри в більшості не прогнозовані	присутні, визначені параметри лише ті, що мають критичне значення, більшість параметрів прогнозовані
Невідомі змінні	відсутні, деякими невідомими параметрами можна знехтувати, що не впливатиме на результат	присутні, впливають на процес і результат дизайну; невідомі значення параметрів визначаються в результаті генерування	присутні, детерміновані, впливають на процес і результат дизайну; невідомі значення параметрів визначаються в результаті генерування
Обчислювальний метод	формула	недетермінований алгоритм	детермінований алгоритм
Процес	одноразова або повторювана дія моделювання за параметрами, пошук результату контрольований людиною	тривала генерація варіантів контрольована алгоритмом з масиву даних, процес працює поки не буде виконано «критерій зупинки»	тривала генерація варіантів контрольована алгоритмом з масиву даних, процес працює поки не буде виконано «критерій зупинки»
Результат	очікуваний, одноваріантний, що залежить від визначеного набору значень параметрів	неочікуваний; продуктом є певна множина варіантів рішення, що є результатом обчислення з невідомими змінними; можливий результат – «щасливий випадок»	простежується, процес функціонування алгоритму зрозуміле; продуктом є певна відносно очікувана множина варіантів рішення, що є результатом обчислення з невідомими змінними, які є значенням із діапазону масиву даних
Масив даних	визначений, чітко окреслений, із відносно невеликою кількістю значень	масив даних – невизначений, генерація відбувається на основі випадковостей	визначений, межі масиву окреслені, генерація відбувається на основі відомої точки старту

Уніфікація термінології в методах обчислювального дизайну покращить комунікацію між науковцями для подальших досліджень та сприятиме впровадженню інноваційних методів у практичну діяльність.

ЛІТЕРАТУРА

- [1] Abdelmohsen S. M. Reconfiguring architectural space using generative design and digital fabrication: a project based course. *17th Conference of the iberoamerican society of digital graphics: Knowledge based design*. Blucher design proceedings. São Paulo. 2014. Vol. 1. № 7, p. 391–395. DOI: <https://doi.org/10.5151/despro-sigradi2013-0074>
- [2] Abrishami S., Goulding J., Rahimian F. P., Ganah A. Integration of BIM and generative design to exploit AEC conceptual design innovation. *Information technology in construction*. 2014. Vol. 19. P. 350–359. URL: https://www.researchgate.net/publication/265421507_Integration_of_BIM_and_generative_design_to_exploit_AEC_conceptual_design_innovation (Accessed: 22.01.2025)
- [3] Agkathidis A. *Generative design*. London : Quercus, 2016. 160 p. URL: https://www.researchgate.net/publication/299457545_Generative_Design (Accessed: 22.01.2025)
- [4] Buonamici F., Carfagni M., Furferi R., Volpe Y., Governi L. Generative design: an explorative study. *Computer-aided design and applications*. 2020. Vol. 18, № 1, P. 144–155. DOI: <https://doi.org/10.14733/cadaps.2021.144-155>

- [5] Caetano I., Leitão A. Architecture meets computation: an overview of the evolution of computational design approaches in architecture. *Architectural science review*. 2020. Vol. 63, № 2, P. 165–174. DOI: <https://doi.org/10.1080/00038628.2019.1680524>

- [6] Caetano I., Santos L., Leitão A. Computational design in architecture: defining parametric, generative, and algorithmic design. *Frontiers of architectural research*. 2020. Vol. 9, № 2, P. 287–300. DOI: <https://doi.org/10.1016/j.foar.2019.12.008>

- [7] Chaszar A., Joyce S. C. Generating freedom: questions of flexibility in digital design and architectural computation. *International journal of architectural computing*. 2016. Vol. 14, № 2, P. 167–181. DOI: <https://doi.org/10.1177/1478077116638945>

- [8] Eltaweel A., SU Y. Parametric design and daylighting: a literature review. *Renewable and sustainable energy reviews*. 2017. Vol. 73. P. 1086–1103. DOI: <https://doi.org/10.1016/j.rser.2017.02.011>

- [9] Fischer T., Herr C. M. Teaching generative design. *Proceedings of the 4th conference on generative art*. Milan, 2001. P. 147–160. URL: https://www.researchgate.net/publication/30869860_Teaching_Generative_Design (Accessed: 22.01.2025)

- [10] Frazer J. Chapter 9 – Creative design and the generative evolutionary paradigm. *Creative evolutionary systems*. 2002. P. 253–274. DOI: <https://doi.org/10.1016/B978-155860673-9/50047-1>

- [11] Humppli H., Österlund T. Algorithm-aided BIM. *eCAADe 2016: Complexity & simplicity*, Oulu, Finland, 24–26 August 2016.

P. 601–609. DOI: <https://doi.org/10.52842/conf.ecaade.2016.2.601>

[12] Kolarevic B. *Architecture in the digital age: Design and manufacturing*, Taylor and Francis, New York, 2004, 320 p. DOI: <https://doi.org/10.4324/9780203634561>

[13] Queiroz N., Dantas N., Nome C., Vaz, C. Designing a Building envelope using parametric and algorithmic processes. *19th Conference of the Iberoamerican society of digital graphics*. Blucher design proceedings. 2015. Vol. 2, № 3. P. 797–801. DOI: <https://doi.org/10.5151/despro-sigradi2015-sp90284>

[14] Sarcar M. M. M., Rao K. M., Narayan K. L. *Computer aided design and manufacturing*. New Delhi : PHI Learning Pvt. Ltd, 2008. 186 p. URL: https://books.google.co.uk/books?id=zXdivq93WIUC&printsec=frontcover&source=gbv_vpt_reviews#v=onepage&q&f=false (Accessed: 22.01.2025)

[15] Szalapaj P. *CAD principles for architectural design*. London : Routledge, 2013. 256 P. DOI: <https://doi.org/10.4324/9781315042794>

[16] Terzidis K. 2006. *Algorithmic architecture*. London : Routledge, 2006. 176 p. DOI: <https://doi.org/10.4324/9780080461298>

[17] Terzidis K., Isorna J.M., Srinivasan V. Algorithmic design: a paradigm shift in architecture? *22nd International conference on education and research in computer aided architectural design in Europe*. 15-18 September 2004, 2004. P. 201–207. DOI: <https://doi.org/10.52842/conf.ecaade.2004.201>

[18] Jabi W. *Parametric design for architecture*. London : Laurence King Publishing, 2013. 208 p. URL: <https://archive.org/details/cvrt-ave-jabi-wassim-parametric-design-for-architecture-laurence-king-publishing-2013/page/207/mode/2up> (Accessed: 22.01.2025)

[19] Woodbury R. *Elements of Parametric Design*. 1st ed. New York : Routledge, 2010. 300 p. URL: https://edisciplinas.usp.br/pluginfile.php/7366617/mod_resource/content/1/woodbury%20elements%20of%20parametric%20design.pdf (Accessed: 22.01.2025)

[20] Zboinska M. A. Hybrid CAD/E platform supporting exploratory architectural design. *Computer-Aided Design*, 2015. Vol. 59, P. 64–84. DOI: <https://doi.org/10.1016/j.cad.2014.08.029>

REFERENCES

[1] Abdelmohsen, S. M. (2014). Reconfiguring architectural space using generative design and digital fabrication: a Project based course. In *Proceedings of the 17th Conference of the Iberoamerican Society of Digital Graphics*. DOI: <https://doi.org/10.5151/despro-sigradi2013-0074> [in English].

[2] Abrishami, S., Goulding, J., Rahimian, F. P., & Ganah, A. (2014). Integration of BIM and generative design to exploit AEC conceptual design innovation. *Information Technology in Construction*, 19, 350–359. Retrieved from https://www.researchgate.net/publication/265421507_Integration_of_BIM_and_generative_design_to_exploit_AEC_conceptual_design_innovation [in English].

[3] Agkathidis, A. (2016). *Generative design*. Quercus, UK. Retrieved from https://www.researchgate.net/publication/299457545_Generative_Design [in English].

[4] Buonamici, F., Carfagni, M., Furferi, R., Volpe, Y., & Governi, L. (2020). Generative design: an explorative study.

Computer-Aided Design and Applications, 18(1), 144–155. DOI: <https://doi.org/10.14733/cadaps.2021.144-155> [in English].

[5] Caetano, I., & Leitão, A. (2020). Architecture meets computation: an overview of the evolution of computational design approaches in architecture. *Architectural Science Review*, 63(2), 165–174. DOI: <https://doi.org/10.1080/00038628.2019.1680524> [in English].

[6] Caetano, I., Santos, L. & Leitão, A. (2020), Computational design in architecture: Defining parametric, generative, and algorithmic design. *Frontiers of Architectural Research*, 9(2): p. 287–300. DOI: <https://doi.org/10.1016/j.foar.2019.12.008> [in English].

[7] Chaszar, A., & Joyce, S. C. (2016). Generating freedom: Questions of flexibility in digital design and architectural computation. *International Journal of Architectural Computing*, 14(2), 167–181. DOI: <https://doi.org/10.1177/1478077116638945> [in English].

[8] Eltaweel, A., & SU, Y. (2017). Parametric design and daylighting: A literature review. *Renewable and Sustainable Energy Reviews*, 73, 1086–1103. DOI: <https://doi.org/10.1016/j.rser.2017.02.011> [in English].

[9] Fischer, T., & Herr, C. M. (2001). Teaching generative design. In *Proceedings of the 4th Conference on Generative Art* (pp. 147–160). Politecnico di Milano University Milan. Retrieved from https://www.researchgate.net/publication/30869860_Teaching_Generative_Design [in English].

[10] Frazer, J. (2002). Creative design and the generative evolutionary paradigm. In *Creative evolutionary systems* (pp. 253–274). Morgan Kaufmann. DOI: <https://doi.org/10.1016/B978-155860673-9/50047-1> [in English].

[11] Humpi, H., & Österlund, T. (2016). Algorithm-aided BIM. In *Complexity & simplicity—Proceedings of the 34th eCAADe conference* (pp. 601–609). DOI: <https://doi.org/10.52842/conf.ecaade.2016.2.601> [in English].

[12] Kolarevic, B. (2005) *Architecture in the digital age: Design and manufacturing*, Taylor and Francis, New York. DOI: <https://doi.org/10.4324/9780203634561> [in English].

[13] Queiroz, N., Dantas, N., Nome, C., & Vaz, C. (2015). Designing a Building envelope using parametric and algorithmic processes. In *Proceedings of the 19th Conference of the Iberoamerican Society of Digital Graphics* (pp. 797–801). DOI: <https://doi.org/10.5151/despro-sigradi2015-sp90284> [in English].

[14] Sarcar, M. M. M., Rao, K. M., & Narayan, K. L. (2008). *Computer aided design and manufacturing*. PHI Learning Pvt. Ltd. Retrieved from https://books.google.co.uk/books?id=zXdivq93WIUC&printsec=frontcover&source=gbv_vpt_reviews#v=onepage&q&f=false [in English].

[15] Szalapaj, P. (2001). *CAD Principles for Architectural Design* (1st ed.). Routledge. DOI: <https://doi.org/10.4324/9781315042794> [in English].

[16] Terzidis, K. 2006. *Algorithmic Architecture*. Routledge. London. DOI: <https://doi.org/10.4324/9780080461298> [in English].

[17] Terzidis, K., Isorna, J.M., & Srinivasan, V. (2004). Algorithmic Design: A Paradigm Shift in Architecture? *Proceedings of the 22nd International Conference on Education and Research in Computer Aided Architectural Design in Europe (eCAADe)*. DOI: <https://doi.org/10.52842/conf.ecaade.2004.201> [in English].

[18] Jabi, W. (2013). *Parametric design for architecture*. Laurence King Publishing. Retrieved from <https://archive.org/details/cvrt-ave-jabi-wassim-parametric-design-for-architecture-laurence-king-publishing-2013/page/207/mode/2up> [in English].

[19] Woodbury, R., 2010. *Elements of Parametric Design*, 1st ed. New York: Routledge. Retrieved from

https://edisciplinas.usp.br/pluginfile.php/7366617/mod_resource/content/1/woodbury%20elements%20of%20parametric%20design.pdf [in English].

[20] Zboinska, M. A. (2015). Hybrid CAD/E platform supporting exploratory architectural design. *Computer-Aided Design*, 59, 64–84. DOI: <https://doi.org/10.1016/j.cad.2014.08.029> [in English].

ABSTRACT

Otkydach A. Terminologies of Computational Design: Issues in Term Definition

The article addresses the issue of ambiguity and lack of unification in the terminology of computational design, in particular the concepts of "generative design", "algorithmic design" and "parametric design" in the Ukrainian academic environment. An analysis of English-language studies revealed key interpretations of these concepts, the characteristics of each approach, their differences and similarities, and the specifics of their application.

The purpose of the work is to review terminology in the English-speaking scientific context, identify differences between methods, and propose the unification of computational design terminology in Ukraine.

Methodology. To achieve the research objectives, an analysis of the most commonly used terms in computational design was conducted. The study identified the most frequently used terms in the literature and those requiring clarification – specifically, terms with equivalent usage, such as "computational design", «generative design», «parametric design» and «algorithmic design». This analysis serves as a basis for creating proposed terminology within the Ukrainian context of research and practice in design and architecture. It aims to clearly define the conceptual and operational scope of each term, prevent the interchangeable use of different terms, and clarify potential combinations between them.

The research consists of three stages: 1. Identifying the most relevant terms in computational design; 2. Collecting and comparing definitions of these terms. 3. Proposing a consistent and well-grounded terminology for computational design.

Results. The concepts of "computational design", "generative design", "parametric design" and "algorithmic design" are described and unified, a hierarchical model is described which demonstrates the relationship between the terms, and the unique features and limitations of these methods are summarized in a matrix.

Scientific novelty. For the first time in Ukrainian scientific research, such concepts as computational design, generative design, algorithmic design, and parametric design are distinguished.

Practical relevance. The results of the study and the proposed unification of terms can contribute to the unification of terminology, which will facilitate scientific communication between researchers and practitioners, the proposed model of relationships between terms and the matrix of features and limitations of these methods can be used to create educational materials on computational design.

Keywords: computational design, generative design, algorithmic design, parametric design, digital design, design methods, terminology, design automation.

AUTHOR'S NOTE:

Otkydach Andrii, Postgraduate Student at the Department of Design and Architecture Fundamentals, Lviv Polytechnic National University, Lviv, Ukraine, e-mail: andrii.i.otkydach@lpnu.ua, orcid: 0009-0008-1847-872X.

Стаття подана до редакції 23.01.2025 р.