ARTICLE IN PRESS
+ MODEL
Frontiers of Architectural Research xxx (xxxx) xxx

Higher Education Press

Available online at www.sciencedirect.com
ScienceDirect
journal homepage: www.keaipublishing.com/foar

SOUTHEAST UNIVERSITY

Review

Computational design in architecture: Defining parametric, generative, and algorithmic design

Inês Caetano [a,*], Luís Santos [b], António Leitão [a]

[a] *INESC-ID/Instituto Superior Técnico, University of Lisbon, Lisbon, Portugal*
[b] *College of Architecture and Environmental Design, Kent State University, Kent, USA*

Received 19 September 2019; received in revised form 24 December 2019; accepted 25 December 2019

KEYWORDS
Algorithmic design;
Computer-aided design;
Computational design;
Generative design;
Parametric design

Abstract Computation-based approaches in design have emerged in the last decades and rapidly became popular among architects and other designers. Design professionals and researchers adopted different terminologies to address these approaches. However, some terms are used ambiguously and inconsistently, and different terms are commonly used to express the same concept. This paper discusses computational design (CD) and proposes an improved and sound taxonomy for a set of key CD terms, namely, parametric, generative, and algorithmic design, based on an extensive literature review from which different definitions by various authors were collected, analyzed, and compared.

Contents

1. Introduction . . . 00
2. Methodology . . . 00
3. CD in architecture . . . 00
4. CD terms . . . 00
5. Taxonomy of CD terms . . . 00
   5.1. Parametric design . . . 00
      5.1.1. Historical evolution . . . 00
      5.1.2. Parametric design definition . . . 00
   5.2. Generative design . . . 00

* Corresponding author.
E-mail address: ines.caetano@tecnico.ulisboa.pt (I. Caetano).
Peer review under responsibility of Southeast University.

https://doi.org/10.1016/j.foar.2019.12.008

5.2.1. Historical evolution . . . 00
5.2.2. Generative design definition . . . 00
5.3. Algorithmic design . . . 00
5.3.1. Historical evolution . . . 00
5.3.2. Algorithmic design definition . . . 00
6. Discussion . . . 00
7. Conclusions . . . 00
Conflict of interest . . . 00
Acknowledgments . . . 00
References . . . 00

1. Introduction

This work collects several terms that emerged from the increasing use of computational design (CD) methods in architecture, discusses the evolution of their definitions, and proposes a well-founded taxonomy. This section contextualizes CD within building design.

The growth and spread of CD marked a "computational turn" in building design that revolutionized traditional design processes, which were heavily based on manual drafting tasks. Currently, CD challenges and renovates previous architectural design conventions and praxis (Rocker, 2006).

Early implementations of CD include CRAFT (Armour and Buffa, 1963), an algorithm-based system that uses a heuristic to optimize spatial location patterns for physical facilities, such as manufacturing plants. CRAFT was among the first systems to automate design procedures using optimization techniques. However, it was confined to the design of the topological relationships among different programmatic parts of an industrial facility, neither addressing geometric descriptions nor presenting a graphical user interface (GUI) that designers could use. Meanwhile, Ivan Sutherland introduced the *Sketchpad* computer program (Sutherland, 1963), the ancestor of computer-aided design (CAD), which enabled not only the automation of drafting tasks but also the setting of parametric relationships among geometric entities using a GUI. In the same decade, General Motors developed the CAD-like system DAC-1 (1964), and some authors addressed the automated optimization of programmatic layouts of industrial buildings (Krejcirik, 1969; Seehof et al., 1966; Whitehead and Eldars, 1965).

However, CAD and building information modeling (BIM) tools only became commercially available in the early 1980s. Building-performance simulation tools also emerged, thereby empowering designers to analyze their designs in terms of different performance criteria.

Parallel to the development of CD tools, several scientific events on CD were critical for the adoption of computation-based approaches in architecture. The *Design Methods* (1962) conference was a pioneer event that mapped the early developments of CD in architecture. The 1st International Congress on *Performance* (1972) began the discussion on applying computation to simulate building performance. Other CD-related international conferences were consistently held in the following decades. Similarly, several scientific journals, such as *Automation in Construction* (1992), *International Journal of Architectural Computing* (2003), and *Journal of Building Performance Simulation* (2008), exclusively focused on CD research. Others gradually incorporated CD topics, such as *Architectural Design* and *Design Studies* (1979). Fig. 1 shows a timeline of CD-related conferences and journals.

As a result of the adoption of CD in architecture, several terms emerged for different approaches. However, the current literature shows inconsistencies in the definitions of some CD-related terms, mainly caused by their overlapping scopes. This work aims to propose a well-structured taxonomy for these terms, that is, a system for naming and organizing their definitions. Such taxonomy aspires for a consensus that prevents the inconsistent use of certain terms. To achieve this objective, the methodology of this work requires a statistical analysis of the use of these terms in the literature. Such analysis serves as a guide to the development of the proposed taxonomy that (1) confines the conceptual and operational scopes of each CD term, (2) avoids equivalent uses of different terms, and (3) clarifies the possible combinations among them. Thus, this work follows three main steps:

1. Identify the most relevant CD terms,
2. Collect and trace the evolution of their definitions, and
3. Propose a consistent and sound taxonomy for them.

We believe that the resulting taxonomy improves the use of CD-related terms in architecture. This research does not aim to propose a fully consolidated taxonomy of terms but rather to provide a starting point for the architectural community to promote further discussions that move toward the normalization of the terms' definitions in an improved and complete taxonomy of CD terms.

2. Methodology

To achieve the proposed research goals, our approach involved the following tasks:

1. Analyzing the existing CD-related literature. From this analysis, we selected the most used terms in the

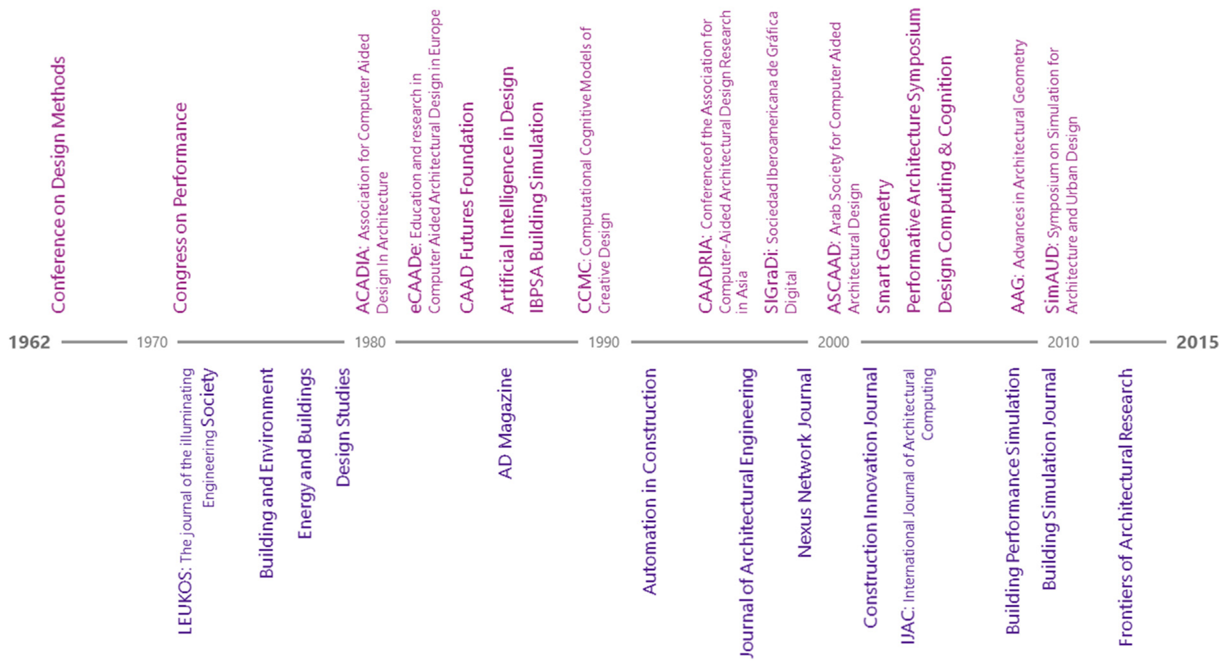
Conference on Design Methods
Congress on Performance
ACADIA: Association for Computer Aided Design in Architecture
eCAADe: Education and research in Computer Aided Architectural Design in Europe
CAAD Futures Foundation
Artificial Intelligence in Design
IBPSA Building Simulation
CCMC: Computational Cognitive Models of Creative Design
CAADRIA: Conference of the Association for Computer-Aided Architectural Design Research in Asia
SIGraDi: Sociedad Iberoamericana de Gráfica Digital
ASCAAD: Arab Society for Computer Aided Architectural Design
Smart Geometry
Performative Architecture Symposium
Design Computing & Cognition
AAG: Advances in Architectural Geometry
SimAUD: Symposium on Simulation for Architecture and Urban Design
1962
1970
1980
1990
2000
2010
2015
LEUKOS: The journal of the illuminating Engineering Society
Building and Environment
Energy and Buildings
Design Studies
AD Magazine
Automation in Construction
Journal of Architectural Engineering
Nexus Network Journal
Construction Innovation Journal
IJAC: International Journal of Architectural Computing
Building Performance Simulation
Building Simulation Journal
Frontiers of Architectural Research

Fig. 1 Timeline of scientific conferences focused on CD (top) approaches and journals that included CD research in their scope (bottom).

literature and those that deserve clarification, such as terms with equivalent use.

2. Collecting and comparing term definitions. We analyzed each term evolution in terms of frequency of use and definition by outlining the similarities and differences found.
3. Drawing a critical reflection based on the outcomes of previous tasks to develop the proposed taxonomy.

We discuss CD in architecture in the following section.

3. CD in architecture

Many terms, including digital, computational, and algorithmic, have been used to describe computers. When professionals began to apply computers in design, different uses were naturally named digital design (DD), CD, algorithmic design (AD), and so on. However, overlap and ambiguity ensued, which we now intend to reduce.

Thus, we begin by distinguishing CD from DD. We consider DD as the use of computer tools in the design process, whereas CD entails the use of computation to develop designs. In this perspective, CD is orthogonal to DD, that is, we can have CD without the use of digital tools, we can use digital tools without relying on CD, or we can have both. An example of CD that is not DD is Frei Otto's minimal surface experiments (Otto and Rasch, 1996), which were based on analog computation. Contrarily, the simple use of a CAD tool as a drafting device, not explicitly using computation, is an example of DD that is not CD. Finally, Mark Burry's work in *Sagrada Familia* (Burry, 1993) is both an example of DD and CD. In this paper, we focus on CD and some of its related terms. In this section, we develop a detailed definition of CD.

Before structuring a taxonomy of CD terms, the relevance of CD in architecture and its evolution must be analyzed to finally formulate an accurate definition for it. Fig. 2 assesses the relevance of CD in the architectural field by measuring the frequency of its use either as a main research topic or as a keyword in the literature.

Fig. 2 shows that the CD paradigm has been important to the architectural field, particularly in the last decade. The figure also indicates that CD-related topics will prevail or even flourish in the future.

Albeit CD only appeared as a research topic or keyword in the literature at the end of the 1990s, it emerged in the 60s under the influence of modernist thinking and technological explorations (Koutamanis, 2005). In fact, *CuminCAD*, a scientific repository of the CAD community, contains just one CD-related article that was published before 1960 (Schutzenberger, 1954). These first CD steps influenced other fields, such as artificial intelligence (Simon, 1969), cybernetics (Wiener, 1948), and mathematics, which have started applying CD approaches to architectural design, considering it an activity that handled design problems in a "thinking before acting" manner (Papamichael and Protzen, 1993). Ivan Sutherland's ideas of design variation, constraints, and parametric instances also influenced the shift in architectural theory toward CD (Ahlquist and Menges, 2011).

Initial attempts to consolidate CD in architecture were made in the 1970s. The first position papers (Eastman, 1975; March and Steadman, 1971; Mitchell, 1977) and Ph.D. theses on CD (Akin, 1979; Yessios, 1973) emerged during this period. In the 80s, CD became a recognizable and accepted field in architecture, mostly because of the commercialization of the first CAD tools (Koutamanis, 2005).

During the 1990s, CD was an established field with its own conferences and journals (Fig. 1). During this decade,

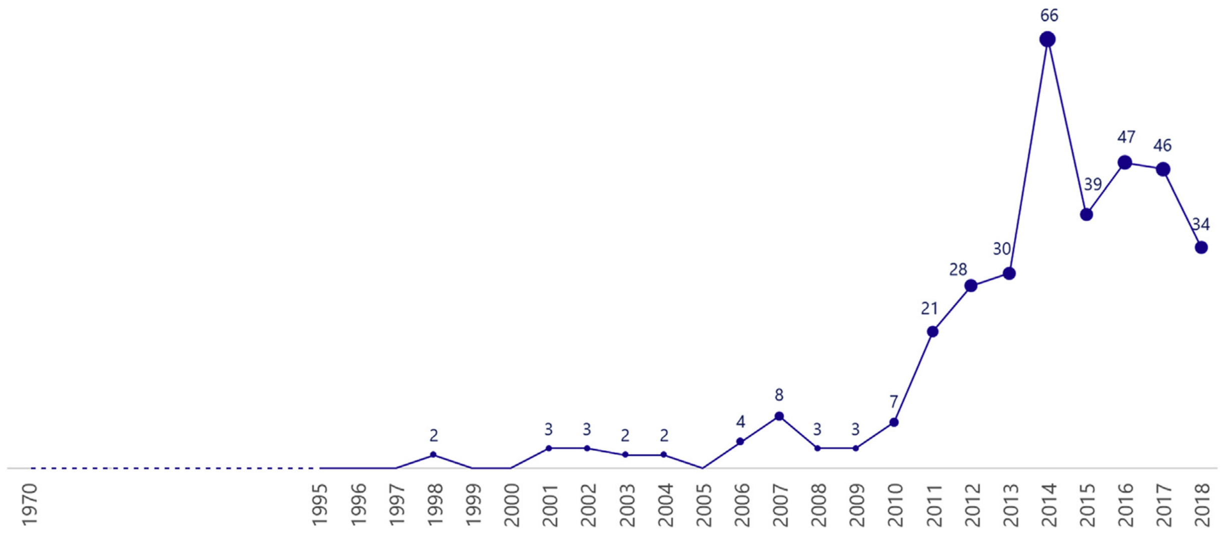

Fig. 2 Usage of the CD term through time: a timeline with the number of times CD appeared in the literature as keyword or main research topic (sources: CuminCAD, Science Direct, Scopus, Web of Science).

the popularity of the CAD software increased among architects, mostly because of the automation of repetitive tasks that led to increased productivity.

In the last two decades, the CD techniques applied in architectural design surpassed the automation of drafting tasks (Terzidis, 2004). Lately, emerging design approaches have been integrating different computation-based techniques, such as building simulation, evolutionary optimization, and novel fabrication methods, therefore originating new design approaches (Oxman, 2017) and terms.

Several authors define CD as an approach based on the use of digital tools, such as CAD programs, to develop design solutions (Alfaris, 2009; Knight and Stiny, 2015; Stiny and March 1981), and thus consider CD equivalent to DD. Although the use of computers is paramount in CD, an open discussion on which digital-design processes constitute CD exists. Some authors believe that CD requires exploiting computers' capabilities in the design process (Albayrak, 2011; Cagan et al., 2005; Humppi, 2015; Oxman, 2017; Peters, 2013; Terzidis, 2006). For example, Oxman (2006) proposed that the CD term applies to design processes that fully utilize computers because of their computational abilities instead of using them as electronic drawing boards. Terzidis (2006) extended the notion of CD by considering it as the entire process that leads to a final result through digital tools.

We agree that certain processes only use computers for drafting or other representational purposes and others truly take advantage of their computational capabilities, such as generating, providing information for, or steering the design process through algorithmic or computational-based procedures. Thus, CD is a design process that takes advantage of computational capabilities through the following activities:

1. Automating design procedures based on the following:
   a. A deduction, that is, applying a transformation to an element while knowing its outcome (Chokhachian, 2014);
   b. An induction, that is, extrapolating the required design process to obtain a specific result (Chokhachian, 2014);
   c. An abstraction that, understands the essential design features by removing irrelevant information.
2. Parallelizing design tasks and efficiently managing large amounts of information
3. Incorporating and propagating changes in a quick and flexible manner
4. Assisting designers in form-finding processes through automated feedback, such as mapping simulation results.

4. CD terms

In the last two decades, architects embraced the CD paradigm as a way of improving typical design workflows and exploring different research threads. The use of CD often requires specialized expertise, thereby forcing designers to acquire additional knowledge from other areas. The resulting field combinations produced novel design methods and paradigms from which new terms emerged. Some of these terms either have an ambiguous definition, embrace two or more conflicting ideas, or overlap with other terms. We address this problem by focusing on the most relevant and problematic CD terms based on two criteria: (1) frequency of use as a keyword in different scientific repositories and (2) scope overlap, particularly for terms that have an equivalent use in the literature. Fig. 3 maps the first criterion in *CuminCAD*, *Science Direct*, *Scopus*, and *Web of Science* between 1978 and 2018.

Fig. 3 shows that parametric design (PD) is the most popular term, followed by generative design (GD). The literature shows cases of equivalent use of PD, GD, and AD, albeit the lower frequency of the latter. These terms are frequently used in parallel and often confused with one another.

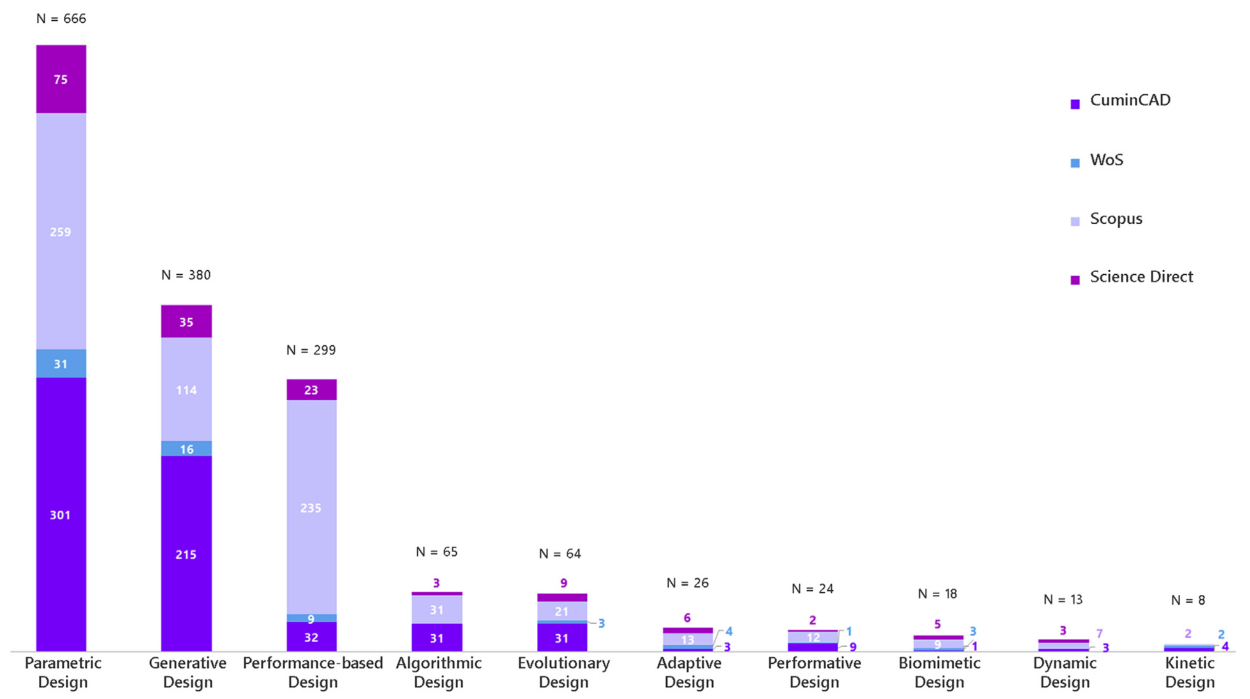

Fig. 3 Number of times each CD term appeared in the literature between 1978 and 2018.

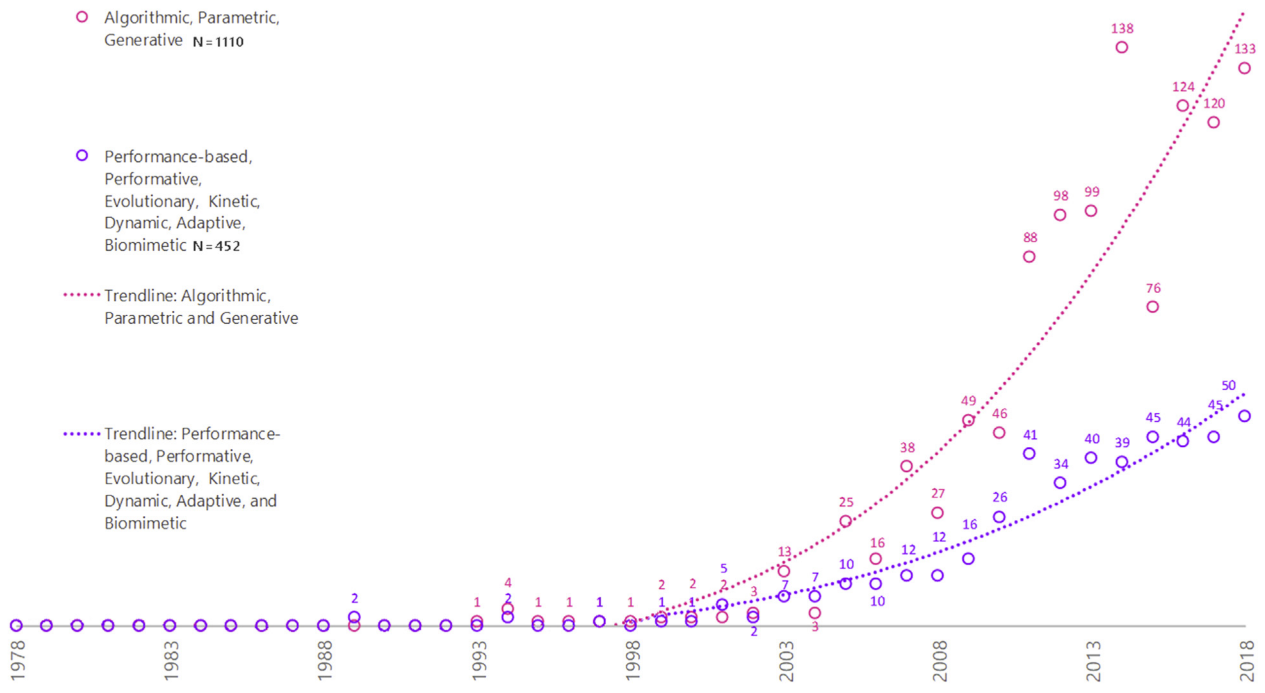

Fig. 4 Frequency of use of selected and non-selected CD-related key words between 1978 and 2017. For each group, we present a third-order polynomial trend line.

Performance-based design is the third most used term. However, it seldom replaces PD and GD and only overlaps with performative design, one of the least used terms.

To limit the scope of this paper, we focus on analyzing PD, GD, and AD, which are closely related. These terms account for 71% of the total occurrence of CD-related research keywords. Fig. 4 compares the usage of the selected terms to the excluded ones, showing that the trend of PD, GD, and AD applications significantly increased in the previous years. This phenomenon reinforces the need to provide precise definitions for them.

5. Taxonomy of CD terms

To clarify PD, GD, and AD, we analyzed each term's evolution and existing definitions to propose a consistent taxonomy. The following sections address each term individually.

5.1. Parametric design

This section analyzes PD, which is highly associated with other terms, including parametric model or modeling and parametric systems.

5.1.1. Historical evolution

Moretti (1971) defined parametric architecture as the study of "the relationships between the dimensions" of a design based on parameters. Kalay (1989) extended Moretti's definition by considering parametric modeling as a computational representation of geometric relationships that are "automatically updated and visualized on the screen" upon parameter change. Monedero (1997) shared a similar view, focusing on the relations between form parameters. Later, Szalapaj (2001) described it as "the use of geometric constraints as well as dimensional relations and data" to define form. Kolarevic (2003) defined PD as a process through which the "parameters of a particular design are declared and not its shape," thereby allowing the instantiation of several solutions (2003) while maintaining overall consistency. Other authors, including Burry (2003), Nassar et al. (2003), Barrios (2005), Aish and Woodbury (2005), Schodek et al. (2005), Roberto and Hernandez (2006), Menges (2006), Oxman (2008, 2006), Alfaris and Merello (2008), and Meridith and Sasaki (2008), proposed similar definitions.

Some authors have different perspectives, either by considering that PD involves some type of optimization to find a solution with an "acceptable performance and constraint satisfaction" (Eggert, 2004) or by claiming it to be a "contemporary architectural style that has achieved pervasive hegemony within the contemporary architectural avant-garde," *parametricism* (Schumacher, 2008). The use of these views in the literature is unusual, mostly because the former heavily focuses on the building engineering domain and the latter results from an architectural manifesto, thereby limiting its ability to generate a broad consensus.

The early 2000s' perspectives on PD inspired most of the current definitions. For Woodbury (2010), PD explores associative geometric relationships to support "the creation, management, and organization of complex digital design models." Eastman et al. (2008), Barrios (2011), Davis et al. (2011), Puusepp (2011), Chien and Yeh (2012), Jabi (2013), Davis (2013), Peters (2013), Queiroz and Vaz (2015), Kensek and Noble (2014), Oxman and Oxman (2014), Elghandour et al. (2014), Yu and Gero (2015), Humppi (2015), Zboinska (2015), Gerber and Pantazis (2016), Chaszar and Joyce (2016), Oxman (2017), Eltaweel and Su (2017), and Jabi et al. (2017) presented similar views.

Some authors proposed a categorization of different types of PD approaches. Zarei (2012) considered PD a set of techniques subdivided into two categories; one defines PD as "a method for conceptual modeling" that requires programming and scripting knowledge, and the other relates PD with "the idea of architectural construction" and manufacturing. Janssen and Stouffs (2015) also divided PD into categories according to the types of modeling, such as object, associative, data-flow, and procedural modeling.

Other views narrowed the scope of PD. Zboinska (2015) restricted PD to a design approach that exclusively resorts to algorithmic processes, thereby considering it a subtype of AD. Similarly, Elghandour et al. (2014) stated that PD is a code-based design approach that facilitates the generation of several design instances without manually recreating the models.

Fig. 5 reports the use of PD as a keyword in the literature between 1978 and 2018. The popularity of the term in the last two decades explains the large number of definitions of PD found in that period and indicates that architects are increasingly adopting PD approaches, thereby recognizing the advantages of using computation in design.

5.1.2. Parametric design definition

The Oxford Dictionary defines parameter as "a numerical or other measurable factor forming one of a set that defines a system or sets the conditions of its operation," or as "a limit [...] which defines the scope of a particular process or

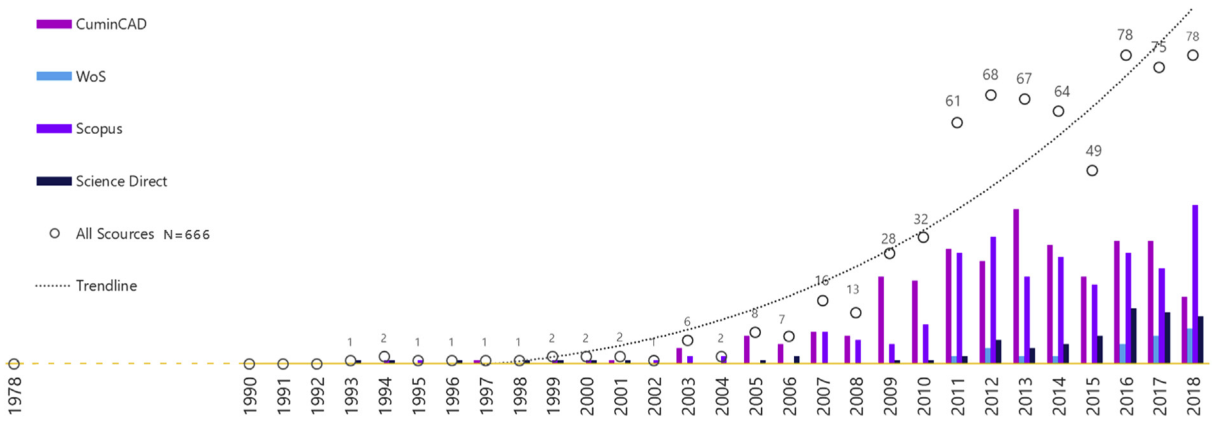

Fig. 5 Frequency of the use of the term PD as a keyword in the different scientific repositories between 1978 and 2018 (in a sample of 666 collected keywords).

activity," and the word parametric as "relating to or expressed in terms of a parameter or parameters."

On the basis of the literature, we can synthesize PD into a design process based on algorithmic thinking (Jabi, 2013; Jabi et al., 2017) that uses parameters and rules to constrain them (Barrios, 2005; Eastman et al., 2008; Jabi, 2013; Kensek and Noble, 2014; Kolarevic, 2003; Marin et al., 2015; Monedero, 1997; Moretti, 1971; Nassar et al., 2003; Roberto and Hernandez, 2006; Schodek et al., 2005; Szalapaj, 2001; Woodbury, 2010; Yu and Gero, 2015). PD also relates to the BIM paradigm as the latter uses PD's concepts of associative geometry and topological relationships (Gerber and Pantazis, 2016; Oxman, 2017, 2006; Qian, 2009) that establish dependencies among different design elements.

Therefore, PD is an approach that describes a design symbolically based on the use of parameters.

As an example, instead of designing walls using exact positions, lengths, heights, and thicknesses, these properties are replaced by symbolic parameters that have specific domains. The result is a symbolic representation of a set of walls. This approach is commonly used in BIM tools and is expressed in the concept of a family/object that describes sets of building elements. For example, in the case of a wall family, each combination of parameter values corresponds to a different wall. In this example, a direct relation exists between the parameters and the resulting design, but in other cases, this relation might not be evident because the parameters can be used in an intricate way to produce complex designs.

Some architectural examples of PD applications include the *Hangzhou Olympic Sport Center* by NBBJ Architects and the *Qatar Integrated Railway Project* by UNStudio. In both cases, the design studios developed parametric programs that allowed them to generate variations of the buildings by modifying the design parameters.

The proposed definition encompasses all the previous ones presented by other authors without unnecessarily constraining its applicability. In this regard, associative geometry is not a requirement for PD, although it emerges from its practical use. In sum, if a design depends on parameters, it is PD.

5.2. Generative design

This section analyzes the term GD, which tends to appear alongside PD. The literature also considers generative systems and models in the scope of GD.

5.2.1. Historical evolution

In the late 1970's, Mitchell described generative design systems (1975) as devices capable of generating potential solutions to a given problem. In the following two decades, the literature poorly addressed GD (see Fig. 6). At the beginning of the 21st century, Fischer and Herr (2001) defined GD as a design approach where "during the design process the designer does not interact with materials and products in a direct way, but via a generative system of some sort." For Herr (2002), a generative system refers to computer-aided generative systems that are typically developed by architects, thus reflecting the uniqueness of architectural design problems.

Frazer et al. (2002) described GD as the use of the "virtual space of the computer in a manner analogous to evolutionary processes in nature." Similarly, Krause (2003) defined GD as the development of "systems which can develop, evolve, or design architectural structures, objects, or spaces more or less autonomously (...)." For Caldas (2008), these systems are evolutionary-based and search a design space for solutions that meet formal and performance requirements.

McCormack et al. (2004) defined GD as the exploration of "the principle of generating complex forms and patterns from a simple specification." Malkawi (2005), Chase (2005), Shea et al. (2005), Oxman (2008, 2006), Fasoulaki (2008), Karzel and Matcha (2009), Puusepp (2011), Larsen (2012), Granadeiro et al. (2013), Garber (2014), and Roggema and Nikolay (2015) provided similar definitions.

Zee and Vrie (2008) went further, stating that GD can "find solutions to complex problems" that can hardly be

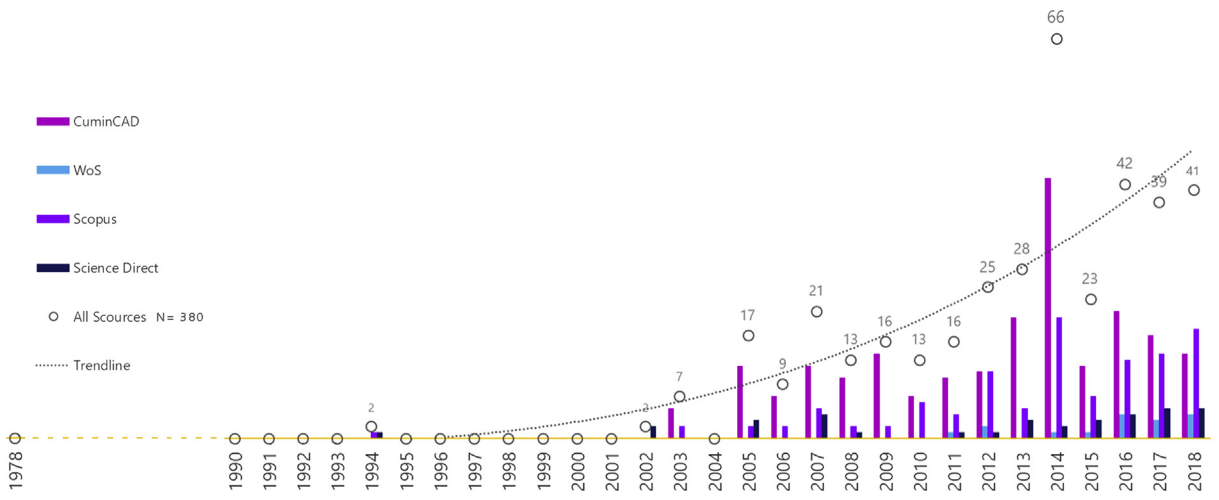

Fig. 6 Number of times the term GD appeared in the different scientific sources between 1978 and 2018 (in a sample of 380 collected keywords).

found using traditional problem-solving procedures. According to Bernal et al. (2015), GD allows for "apparently creative outcomes since every new combination of parameters brings the opportunity to look for the emergence of new properties or affordances from the resulting composition." For Chaszar and Joyce (2016), GD overcomes the shortcomings of traditional manual design methods by "harnessing computational power to address issues of speed and accuracy, as well as complexity," and augmenting inventiveness as it increases "the number of design variations" and "the range of variations," which include "happy accidents," that is, unexpected results that positively impact the design process.

Abrishami et al. (2014) have a broader perspective, describing GD as the use of a system, "such as a computer program, to produce the solution to the design problem with some level of autonomy."

Bukhari (2011) narrowed the scope of GD to the use of "algorithms to generate a host of different solutions from a given set of design goals and constraints," considering it as a type of AD. Similarly, Humppi and Österlund (2016) defined GD as "design utilizing algorithmic logic and generative processes."

Fig. 6 presents the use of the term from 1978 to 2018 and indicates that before 2002, GD was nearly irrelevant in the design field, but after 2004, its popularity greatly increased.

5.2.2. Generative design definition

The Cambridge Dictionary defines *generative* as the "capacity to produce or create something." Some authors define GD as a design process that mainly refers to evolutionary techniques in both the creation and production processes of design solutions (Fischer and Herr, 2001; Frazer et al., 2002; Zhang and Xu, 2018), whereas others do not restrict GD to evolutionary processes, considering it a design approach based on algorithmic or ruled-based processes that generate multiple and, possibly, complex solutions (Bernal et al., 2015; Bukhari, 2011; Chase, 2005; Fasoulaki, 2008; Humppi and Österlund, 2016; Leitão et al., 2014; McCormack et al., 2004; Mitchell, 1975; Oxman, 2008; Shea et al., 2005). Moreover, several authors consider approaches, such as algorithmic generation, cellular automata, evolutionary methods, generative and shape grammars, L-systems, self-organization, agent-based models, and swarm systems, as part of GD (Abdelmohsen, 2013; Caldas, 2008; Chase, 2005; Fasoulaki, 2008; McCormack et al., 2004; Oxman, 2008; Puusepp, 2011; Zee and Vrie, 2008).

Considering these two perspectives, we believe that the first—confining GD to evolutionary processes—is narrow because it excludes other methods that also generate design. Moreover, GD must be differentiated from other terms, such as PD. Thus, we define GD as a design paradigm that employs algorithmic descriptions that are more autonomous than PD. In GD approaches, after starting the generative process, the system executes encoded instructions until the stop criterion is satisfied. Consequently, GD-based methods can generate complex outputs even from simple algorithmic descriptions. In many cases, the algorithm is difficult to correlate with the generated output, thus making the outcome difficult to predict by merely reading the algorithmic description. Performance-based generative design systems (PGDS) are good examples of the application of GD methods. In these systems, the designer sets a performance target, and an algorithm finds design solutions that best approximate the desired goal. As examples, in *AudiOptimization* (Monks et al., 2000) (a PGDS for acoustic-based design), *EifForm* (Shea, 2000) and *Paragen* (von Buelow, 2012) (both computational tools for structural design), *GENE_ARCH* (Caldas, 2006) (a generative design system for green buildings), and Autodesk's *Dreamcatcher*, the non-traceability between the algorithms that compose the design system and the results it provides is mostly due to the probabilistic or non-deterministic nature of their search procedures.

The non-traceability between GD programs and the generated designs is one of the main reasons GD methods produce unexpected results, such as the "happy accidents" mentioned by Chaszar and Joyce (2016).

5.3. Algorithmic design

In the literature, the scope of the term AD overlaps with those of PD and GD, generating some inconsistencies in the definition of AD. For example, some authors state that AD includes GD (Bukhari, 2011), others mention that AD and GD are the same approach (Garber, 2014; Humppi, 2015), whereas others regard AD as an approach dependent on PD tools (Zboinska, 2015). This section proposes a clearer AD definition.

5.3.1. Historical evolution

Terzidis (2004, 2003) defined AD as an approach based on describing computer programs that "generate space and form from the rule-based logic inherent in architectural programs, typologies, building code, and language itself." Thus, AD allows designers to incorporate the "computational complexity and creative use of computers" (Terzidis, 2003) within the design workflow.

Bukhari (2011) considered that AD includes GD and evolutionary design approaches. The latter uses fitness functions that measure different performance factors to steer the search process (Caldas, 2008).

For Queiroz and Vaz (2015), AD allows "the user to design directly through code manipulation," therefore reducing the limitations of existing modeling applications. Similarly, Humppi and Österlund (2016) described AD as the process of controlling the building form through user-tailored scripts. Oxman (2017) defined AD as the coding "of explicit instructions" to generate "digital forms."

For Zboinska (2015), AD is a design paradigm based on PD tools to produce complex geometries "using relatively simple rules and relationships."

Fig. 7 shows that the use of AD in the literature is less frequent than PD and GD. This is due to its narrower scope and steeper learning curve. Nevertheless, as the figure shows, its use has been increasing in the last decade.

5.3.2. Algorithmic design definition

AD is a design process based on algorithms. According to the Cambridge Dictionary, an *algorithm* is a "set of mathematical instructions or rules that [...] will help calculate an answer to a problem." Thus, AD becomes difficult to distinguish from GD. In fact, some authors consider the

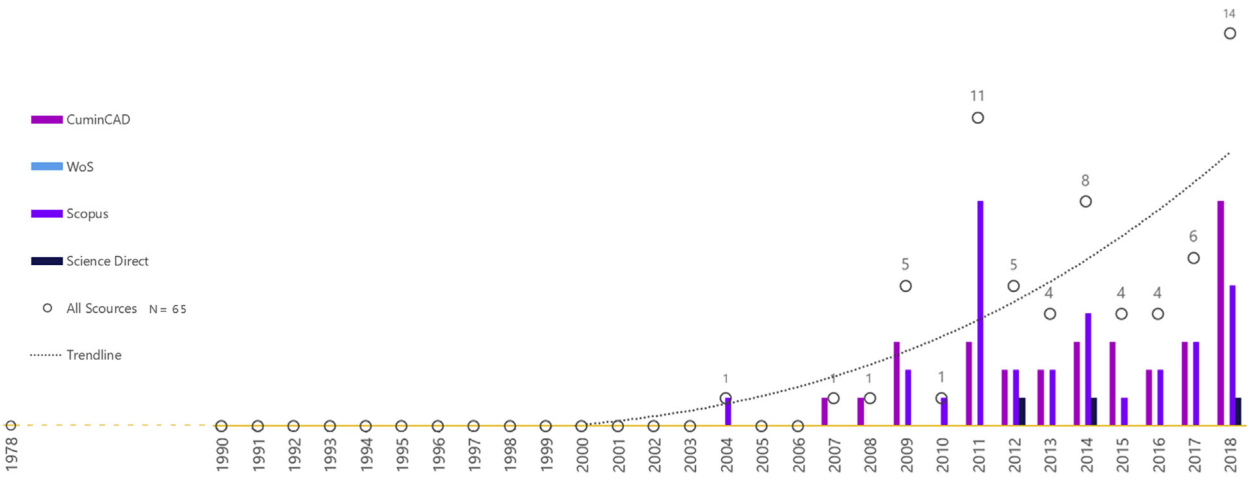

Fig. 7 Number of times that the term AD appeared in different scientific sources between 1978 and 2018 (in a sample of 65 collected keywords).

terms synonymous (Garber, 2014; Humppi, 2015), stating that typical GD examples, such as stochastic search, L-systems, cellular automata, genetic algorithms, and evolutionary design, are also AD (Bukhari, 2011; Terzidis, 2003, 2004).

However, to reflect the relevance of the term in the literature, we argue that it should have a stricter definition within the boundaries of GD.

We consider AD a design paradigm that uses algorithms to generate models and, therefore, we also consider it generative. Nevertheless, in AD, a correlation between the algorithm and the generated model exists, thus providing traceability and allowing the user to identify the parts of the algorithm that generated a given part of the model. In a sense, in AD, the algorithm is isomorphic to the model.

According to this definition, AD is a subset of GD, where the algorithmic development focuses on the envisioned design at the expense of producing fewer surprising results. Nonetheless, it provides a finer degree of control, facilitating debugging and maintenance tasks.

According to our proposal, despite being examples of GD, L-systems and cellular automata are not AD examples as they do not provide a strong correlation between the algorithmic descriptions and the generated outputs. On the contrary, a program that produces a model of a building by separately creating its slabs, columns, beams, walls, windows, and so on should be considered an example of AD because tracking the parts of the code that produce a given part of the model is easy.

Some architectural examples of AD applications include the recent work on *Sagrada Familia* done by Burry and Burry (2006) and the *Morpheus Hotel* by Zaha Hadid Architects. In both cases, the implemented algorithms preserved the traceability properties between the program and the product of its execution and produced consistent results. The work of Burry and Burry (2006) in *Sagrada Familia*'s Passion Façade used an algorithm that creates structural members using a "mathematically driven fully parametricized design model" (Burry, 2011) where different operations are responsible for producing specific effects. In the *Morpheus Hotel* case, the AD program that generated the building is composed of specific modules that were responsible for modeling different aspects of the building. In both cases, a strong correlation clearly exists between the program and the generated model.

6. Discussion

The previous sections present definitions for AD, GD, and PD that clarify the confusion between the terms. The proposed definitions consider that GD requires the explicit use of an algorithm that generates a design. Additionally, if the algorithm satisfies the traceability property, that is, an identifiable correlation between the algorithm and the generated design, then it is also considered AD. Finally, if the design is dependent on a set of parameters, then it is PD.

Albeit we can define each term, a conceptual overlap exists, which is the main cause of their inconsistent use. Fig. 8 illustrates this overlap through a Venn diagram that shows that AD is a subset of GD and has a non-empty intersection with PD. Additionally, PD is orthogonal to the other terms. For example, an AD approach that uses algorithms with parameters is also a PD instance. Consider the

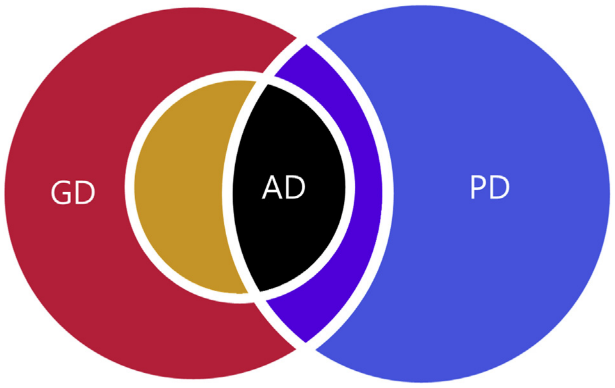

Fig. 8 Conceptual representation of the terms' extension regarding the CD paradigm.

case of an algorithm that generates a façade based on a set of parameters, such as its overall dimensions and the size and distribution of the different façade elements (e.g., windows, balconies, and shades). The black fill in Fig. 8 represents this level of overlap.

However, if tracing an explicit correlation between the algorithm and the generated design is difficult, then we do not consider it as AD, despite it being a GD example. A good example is a black box optimization procedure. By being agnostic to the meaning of the parameters to optimize and the objective function, identifying a correlation between the parts of the generated output and the optimization mechanism is difficult, thus making it a GD approach that is PD but not AD. In Fig. 8, this case belongs to the purple area.

Additionally, cases of GD that are neither AD nor PD exist and are represented in Fig. 8 by the red area. As an example, consider a cellular automaton with specific cell state rules to design a façade. Fig. 9 illustrates this scenario by showing the application of Wolfram's rule 135 (Wolfram, 1983) in Cambridge North's train station, a design authored by Atkins Design Consultancy. Given that cellular automata rules are not parametric, we cannot consider this approach PD. Moreover, the outcome is nearly impossible to directly infer from the rules of the automaton. Therefore, we do not consider this approach AD, although it is generative.

An AD approach that is not parametric is another possibility and a case that often occurs in digital fabrication. For example, when a computer numerical control machine operates, it executes a program that is often automatically generated and entirely non-parametric. This case belongs to the orange area of Fig. 8.

Finally, some cases are exclusively parametric (blue area in Fig. 8). For example, consider a BIM object, such as a wall, that allows the user to change its parameters, such as length, thickness, and materials. This ability makes the object parametric. However, because changing the parameters does not require the explicit use of an algorithm, it is neither an example of GD nor of AD.

The terms' overlap justifies the need for the clearer definitions presented in this paper. Moreover, this section maps the scope of each term and any possible overlap. The following paragraphs debate the relationship between the proposed definitions and those in the existing literature.

Regarding PD, some definitions require associative geometry as a necessary condition (Oxman, 2017, 2012; 2008; Turrin et al., 2011). Associative geometry establishes that different parts of a model must be interdependent so that a change in one part propagates to other parts of the model. Although we consider that associative geometry is a product of PD, it does not exhaust it, as suggested by Woodbury (2010), Turrin et al. (2011), and Jabi et al. (2017). For example, consider a façade with uniformly distributed perforations whose radii are randomly selected based on a range of values. The range is a parameter of the algorithm, but it does not entail an associative constraint between the

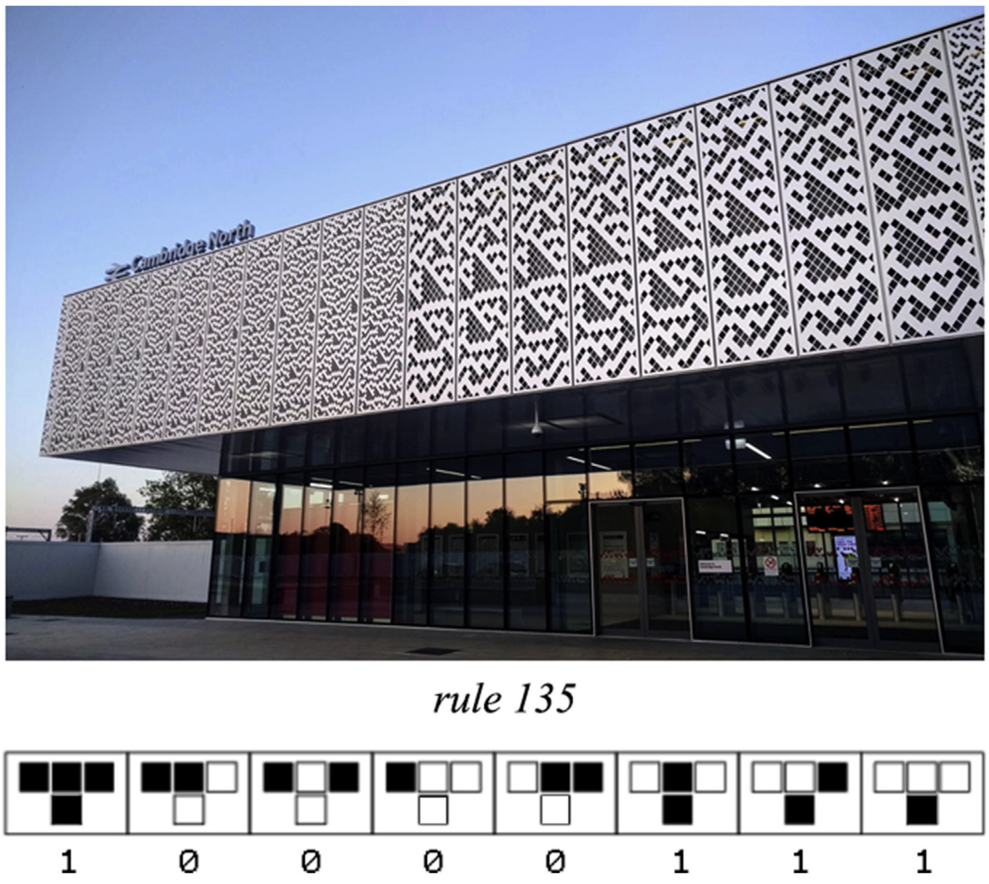

Fig. 9 Application of Stephen Wolfram's rule 135 in the design of Cambridge Railway station, Cambridge, UK. Top: façade of the station. Bottom: rule 135 that were used to generate the façade pattern.

perforations. Other authors (Zarei, 2012) claim that PD has direct links to digital fabrication and requires programming. In our view, neither of these conditions is necessary. An example is a parametric building model developed in BIM software. The design entails parametric elements that users can instantiate without the use of programming and does not involve any kind of manufacturing. Our PD definition aligns with the one proposed by Hudson (2010), that is, the use of computers to modify a design by changing the values of its parameters.

Several authors associated GD with the use of evolutionary-based processes (Fischer and Herr, 2001; Frazer et al., 2002; Zhang and Xu, 2018), which is extremely narrow in our opinion. GD includes such processes and other algorithmic approaches, as mentioned by (Bernal et al., 2015; Bukhari, 2011; Chase, 2005; Fasoulaki, 2008; Humppi and Österlund, 2016; Leitão et al., 2014; McCormack et al., 2004; Mitchell, 1975; Oxman, 2008; Shea et al., 2005). Considering the vast scope of the term, whenever a GD process satisfies the traceability requirement, one should employ the more rigorous term AD for the sake of accuracy.

Finally, Queiroz and Vaz (2015) considered that AD refers to designing through the manipulation of programs. Although vague, this description is consistent with the proposed definition because it suggests a correlation between the manipulations of the program and their corresponding effects on the design. Similarly, the AD descriptions provided by Humppi and Österlund (2016) and Oxman (2017) also align with our definition. By contrast, other authors, such as Zboinska (2015), focused on the parametric features of AD, which are not distinguishing features because the majority of AD approaches are also parametric. Additionally, some authors (Bukhari, 2011) considered that GD is a subset of AD, and AD includes evolutionary approaches. However, we present the opposite, that is, AD is a subset of GD and excludes evolutionary approaches because they hardly support the traceability requirement.

7. Conclusions

The increased computational capacity of tools and the diversity of the available CD methods have enabled architects to enhance the design process, either by making it more efficient or by expanding its conceptual boundaries. These methods empower architects to (i) explore and evaluate other complex solutions, (ii) create and deploy advanced fabrication techniques, and (iii) control the design process at different stages remarkably. Thus, the question is no longer whether CD is good or bad for architecture (Picon, 2010) but rather how can the discipline benefit from it.

The gradual appropriation of CD-related terms by different authors has led to different definitions depending on the context and time period. Consequently, a considerable scope overlap exists between terms, which is the root of the ambiguous use of some CD-related terminologies by the design community. This work first identified relevant but inconsistent CD terms and then established an initial taxonomy that is consistent and promotes discussion toward a generalized consensus. The research revealed that the critical terms are PD, GD, and AD. We systematically discussed these terms and proposed a clear and consistent definition for them. To ground such definitions, this work traced their historical evolution, mapped the different perspectives found in the literature, and discussed the existent contradictions, ambiguities, and common intersections between those perspectives. The outcome is a CD taxonomy that provides a clear definition for each term, explains how they relate to each other, delimits their specific scopes, and maps their possible interactions.

The discussion conducted in this paper contextualizes the proposed taxonomy regarding the divergent and convergent existing definitions. Rather than providing definitive and closed definitions, the resulting taxonomy draws the attention to the problem of misuse of fundamental CD-related terms and lays the foundations for further discussion. Thus, this work proposes a theoretical basis for a CD-related taxonomy, which can in turn be further improved and completed by the design community until it achieves a mature point of generalized consensus.

In sum, the proposed taxonomy introduces the following core definitions:

- GD is a design approach that uses algorithms to generate designs.
- AD is a GD approach characterized by an identifiable correlation between the algorithm and its outcome.
- PD is a design approach based on the use of parameters to describe sets of designs.

Regarding the scope and relationship among terms, Fig. 8 shows that AD is a subset of GD, that is, an AD approach is always generative but also requires a direct correlation between the algorithm and the generated design. When this correlation is difficult to establish, the GD approach should not be considered AD. PD is orthogonal to AD and GD. Therefore, GD or AD approaches can be parametric or use parametric modeling techniques, but there are instances of PD that do not rely on generative approaches.

As a future work, we plan to expand the scope of this taxonomy and include other relevant CD-related terms, such as evolutionary architecture, performative design, and performance-based design. We also intend to monitor the scope of the terms by continuously investigating their evolution.

Conflict of interest

None.

Acknowledgments

This work was supported by national funds through *Fundação para a Ciência e a Tecnologia* (FCT) with references UID/CEC/50021/2019 and PTDC/ART-DAQ/31061/2017, by the PhD grants under contract of FCT with references SFRH/BD/128628/2017 and SFRH/BD/98658/2013, and by the PhD grant under contract of University of Lisbon (UL), *Instituto Superior Técnico* (IST) and the research unit *Investigação e Inovação em Engenharia Civil para a Sustentabilidade* (CERIS).

References

Abdelmohsen, S.M., 2013. 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.

Abrishami, S., Goulding, J.S., Rahimian, F.P., Ganah, A., 2014. Integration of BIM and generative design to exploit aec conceptual design innovation. J. Inf. Technol. Constr. 19, 350—359.

Ahlquist, S., Menges, A., 2011. Introduction: computational design thinking. In: Menges, A., Ahlquist, S. (Eds.), AD Reader: Computational Design Thinking. John Wiley & Sons Ltd, United Kingdom, pp. 10—29.

Aish, R., Woodbury, R., 2005. Multi-level interaction in parametric design. In: Lecture Notes in Computer Science, vol. 3638, pp. 151—162.

Akin, O., 1979. Models of Architectural Knowledge - an Information Processing Model of Design. PhD thesis. Carnegie-Mellon University, Pittsburgh.

Albayrak, C., 2011. Performative Architecture as a Guideline for Transformation of the Defence Line of Amsterdam. Master thesis. Middle East Technical University and Delft University of Technology.

Alfaris, A., 2009. Emergence through Conflict the Multi-Disciplinary Design System (MDDS). PhD thesis. Massachusetts Institute of Technology.

Alfaris, A., Merello, R., 2008. The generative multi-performance design system. In: Proceedings of the Annual International ACADIA Conference, pp. 448—457.

Armour, G., Buffa, E., 1963. A heuristic algorithm and simulation approach to relative location of facilities. Manag. Sci. 9, 294—309.

Barrios, C., 2005. Transformations on parametric design models: a case study on the sagrada familia columns instances of a parametric model. In: Proceedings of the 11th International Conference on Computer Aided Architectural Design Futures, pp. 393—400.

Barrios, C., 2011. Parametric affordances: what, when, how. In: Proceedings of the ACADIA Regional Conference, pp. 203—207.

Bernal, M., Haymaker, J.R., Eastman, C., 2015. On the role of computational support for designers in action. Des. Stud. 41, 163—182.

Bukhari, F. a, 2011. A Hierarchical Evolutionary Algorithmic Design (HEAD) System for Generating and Evolving Building Design Models. PhD Thesis. Queensland University of Technology.

Burry, M., 1993. The Expiatory Church of the Sagrada Família. Phaidon Press, London.

Burry, M., 2003. Between intuition and process: parametric design and rapid prototyping. In: Kolarevic, B. (Ed.), Architecture in the Digital Age: Design and Manufacturing. Taylor&Francis, New York, pp. 149—162.

Burry, M., 2011. Dimensions. In: Scripting Cultures: Architectural Design and Programming. Architectural Design Primer, pp. 86—125.

Burry, J.R., Burry, M.C., 2006. Gaudí and CAD. J. Inf. Technol. Constr. 11, 437—446.

Cagan, J., Campbell, M.I., Finger, S., Tomiyama, T., 2005. A framework for computational design synthesis: model and applications. J. Comput. Inf. Sci. Eng. 5, 171.

Caldas, L., 2006. GENE_ARCH: an evolution-based generative design system for sustainable architecture. In: Smith, I.F.C. (Ed.), Intelligent Computing in Engineering and Architecture. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 109—118.

Caldas, L., 2008. Generation of energy-efficient architecture solutions applying GENE_ARCH: an evolution-based generative design system. Adv. Eng. Inf. 22, 59—70.

Chase, S.C., 2005. Generative design tools for novice designers: issues for selection. Autom. ConStruct. 14, 689—698.

Chaszar, A., Joyce, S.C., 2016. Generating freedom: questions of flexibility in digital design and architectural computation. Int. J. Archit. Comput. 14, 167—181.

Chien, S., Yeh, Y., 2012. On creativity and parametric design. In: Proceedings of the 30th International eCAADe Conference, pp. 245—254.

Chokhachian, A., 2014. Studies on Architecture Design Procedure: A Framework for Parametric Design Thinking. Master thesis. Eastern Mediterranean University.

Davis, D., 2013. Modelled on Software Engineering: Flexible Parametric Models in the Practice of Architecture. PhD Thesis. RMIT University.

Davis, D., Burry, J., Burry, M., 2011. The flexiblity of logic programming: parametrically regenerating the Sagrada Família. In: Proceedings of the 16th International Conference on Computer-Aided Architectural Design Research in Asia, pp. 29—38.

Eastman, C. (Ed.), 1975. Spatial Synthesis in Computer-Aided Building Design. Applied Science, London.

Eastman, C., Teicholz, P., Sacks, R., Liston, K., 2008. BIM Handbook: A Guide to Building Information Modeling for Owners, Managers, Designers, Engineers, and Contractors, first ed. John Wiley & Sons, Inc., New Jersey & Canada.

Eggert, R.J., 2004. Engineering Design. Prentice Hall, Upper Saddle River, New Jersey.

Elghandour, A., Saleh, A., Aboeineen, O., Elmokadem, A., 2014. Using Parametric Design to Optimize Building's Façade Skin to Improve Indoor Daylighting Performance, pp. 353—361.

Eltaweel, A., Su, Y., 2017. Parametric design and daylighting: a literature review. Renew. Sustain. Energy Rev. 73, 1086—1103.

Fasoulaki, E., 2008. Integrated Design: A Generative Multi-Performative Design Approach. Master thesis. Massachusetts Institute of Technology.

Fischer, T., Herr, C.M., 2001. Teaching generative design. In: Proceedings of the 4th International Conference on Generative Art. Milan, Italy.

Frazer, John, Frazer, Julia, Xiyu, L., Mingxi, T., Janssen, P., 2002. Generative and evolutionary techniques for building envelope design. In: Proceedings of the 5th International Conference on Generative Art. Milan, Italy, pp. 1—16.

Garber, R., 2014. The digital states and information modelling. In: Garber, R. (Ed.), BIM Design: Realising the Creative Potential of Building Information Modelling. John Wiley & Sons Ltd, Chichester, West Sussex, United Kingdom, pp. 122—131.

Gerber, D.J., Pantazis, E., 2016. A multi-agent system for facade design: a design methodology for design exploration, analysis and simulated robotic fabrication. In: Proceedings of the 36th Annual Conference of the Association for Computer Aided Design in Architecture, pp. 12—23.

Granadeiro, V., Duarte, J.P., Correia, J., Leal, V.M.S., 2013. Building envelope shape design in early stages of the design process: integrating architectural design systems and energy simulation. Autom. ConStruct. 32, 196—209.

Herr, C.M., 2002. Generative architectural design and complexity theory. In: Proceedings of the 5th International Conference on Generative Art. Milan, Italy, 1—13.

Hudson, R., 2010. Strategies for Parametric Design in Architecture: an Application of Practice Led Research. PhD thesis. University of Bath.

Humppi, H., 2015. Algorithm-Aided Building Information Modeling: Connecting Algorithm-Aided Design and Object-Oriented Design. Master thesis. Tampere University of Technology.

Humppi, H., Österlund, T., 2016. Algorithm-aided BIM. In: Proceedings of the 34th eCAADe Conference, pp. 601—609.

Jabi, W., 2013. Parametric Design for Architecture. Laurence King Publishing Ltd, London, UK.

Jabi, W., Soe, S., Theobald, P., Aish, R., Lannon, S., 2017. Enhancing parametric design through non-manifold topology. Des. Stud. 1—19.

Janssen, P., Stouffs, R., 2015. Types of parametric modelling. In: Proceedings of the 20th International Conference of Association for Computer-Aided Architectural Design Research in Asia CAADRIA 2015, pp. 157—166.
Kalay, Y.E., 1989. Modeling Objects and Environments (Principles of Computer Aided Design). Wiley-Academy, New York.
Karzel, R., Matcha, H., 2009. Experimental design-build: teaching parameter-based design. In: Proceedings of the 27th eCAADe Conference, pp. 153—158.
Kensek, K.M., Noble, D.E. (Eds.), 2014. Building Information Modeling: BIM in Current and Future Practice, first ed. John Wiley & Sons, Hoboken, New Jersey.
Knight, T., Stiny, G., 2015. Making grammars: from computing with shapes to computing with things. Des. Stud. 41, 8—28.
Kolarevic, B. (Ed.), 2003. Architecture in the Digital Age: Design and Manufacturing, first ed. Spon Press.
Koutamanis, A., 2005. A biased history of CAAD. In: Proceedings of the 23th eCAADe Conference, pp. 629—637.
Krause, J., 2003. Reflections: the creative process of generative design in architecture. In: Proceedings of the 6th International Conference on Generative Art.
Krejcirik, M., 1969. Computer-aided plant layout. Comput. Aided Des. 2, 7—19.
Larsen, N.M., 2012. Generative Algorithmic Techniques for Architectural Design. PhD thesis. Aarhus School of Architecture.
Leitão, A., Lopes, J., Santos, L., 2014. Illustrated programming. In: Proceedings of the ACADIA 2014 International Conference, pp. 291—300.
Malkawi, A.M., 2005. Performance simulation: research and tools. In: Kolarevic, B., Malkawi, A. (Eds.), Performative Architecture: beyond Instrumentality. Spon Press, New York, pp. 86—95.
March, L., Steadman, P., 1971. The Geometry of Environment : an Introduction to Spatial Organization in Design. MIT Press, Cambridge, Massachusetts.
Marin, P., Blanchi, Y., Janda, M., 2015. Cost analysis and data based design for supporting programmatic phase. In: Proceedings of the 33rd International Conference on Education and Research in Computer Aided Architectural Design in Europe, pp. 613—618.
McCormack, J., Dorin, A., Innocent, T., 2004. Generative design: a paradigm for design research. In: Proceedings of Futureground. Design Research Society, Melbourne.
Menges, A., 2006. Polymorphism. In: Architectural Design, vol. 76. John Wiley & Sons Ltd, pp. 78—87, 02.
Meridith, M., Sasaki, M., 2008. From Control to Design: Parametric/algorithmic Architecture. Actar-D, Barcelona and New York.
Mitchell, W.J., 1975. The theoretical foundation of computer-aided architectural design. Environ. Plan. Plan. Des. 2 (2), 127—150.
Mitchell, W.J., 1977. Computer-Aided Architectural Design. Van Nostrand Reinhold, New York.
Monedero, J., 1997. Parametric design. A review and some experiences. In: Proceedings of the 15th eCAADe Conference. Vienna University of Technology, Vienna, Austria.
Monks, M., Oh, B.M., Dorsey, J., 2000. Audioptimization: goal-based acoustic design. IEEE Comput. Graph. Appl. 30, 76—91.
Moretti, L., 1971. Ricerca Matematica in Architettura e Urbanisticâ. Moebius IV (1), 30—53.
Nassar, K., Thabet, W., Beliveau, Y., 2003. Building assembly detailing using constraint-based modeling. Autom. ConStruct. 12, 365—379.
Otto, F., Rasch, B., 1996. Finding Form: towards an Architecture of the Minimal. Edition Axel Menges.
Oxman, R., 2006. Theory and design in the first digital age. Des. Stud. 27, 229—265.
Oxman, R., 2008. Digital architecture as a challenge for design pedagogy: theory, knowledge, models and medium. Des. Stud. 29, 99—120.
Oxman, R., 2012. Novel concepts in digital design. In: Gu, N., Wang, X. (Eds.), Computational Design Methods and Technologies: Applications in CAD, CAM and CAE Education. Information Science reference, United States, pp. 18—33.
Oxman, R., 2017. Thinking difference: theories and models of parametric design thinking. Des. Stud. 1—36.
Oxman, Rivka, Oxman, Robert, 2014. Theories of the Digital in Architecture. Routledge, New York.
Papamichael, K., Protzen, J.P., 1993. The limits of intelligence in design. In: Proceedings of the 4th International Symposium on System Research, Informatics and Cybernetics. Baden-Baden, Germany, pp. 1—10.
Peters, B., 2013. Introduction: computation works: the building of algorithmic thought. Architect. Des 83 (2), 8—15.
Picon, A., 2010. Introduction: architecture and the virtual: towards a new materiality. In: Sykes, A.K. (Ed.), Constructing a New Agenda: Architectural Theory 1993-2009. Princeton Architectural Press, New York, pp. 268—269.
Puusepp, R., 2011. Generating Circulation Diagrams for Architecture and Urban Design Using Multi-Agent Systems. PhD thesis. University of East London.
Qian, Z.C., 2009. Design Patterns: Augmenting Design Practice in Parametric CAD Systems. PhD thesis. Simon Fraser University.
Queiroz, N., 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.
Roberto, C., Hernandez, B., 2006. Thinking parametric design: introducing parametric gaudi. Des. Stud. 27, 309—324.
Rocker, I.M., 2006. When code matters. Architect. Des 76 (4), 16—25.
Roggema, R., Nikolay, P., 2015. Swarm planning: development of generative spatial planning tool for resilient cities. In: Proceedings of the 33rd eCAADe Conference - Volume 1. Vienna, Austria, pp. 519—527.
Schodek, D., Bechthold, M., Griggs, J.K., Kao, K., Steinberg, M., 2005. Digital Design and Manufacturing: CAD/CAM Applications in Architecture and Design. John Wiley & Sons, Inc., Hoboken, New Jersey.
Schumacher, P., 2008. Parametricism as style: parametricist manifesto. In: The Darkside Club, 11th Architecture Biennale, Venice, 11 September.
Schutzenberger, M.P., 1954. A tentative classification of goal-seeking behaviors. J. Ment. Sci. 100, 97—102.
Seehof, J.M., Evans, W.O., Friederichs, J.W., Quigley, J.J., 1966. Automated facilities Layout Programs. In: Proceedings of the 1966 21st National Conference, pp. 191—199. https://doi.org/10.1145/800256.810696.
Shea, K., 2000. eifForm: a generative structural design system. In: Proceedings of the 2000 ACSA Technology Conference: Emerging Technologies and Design: the Intersection of Design and Technology, pp. 87—91.
Shea, K., Aish, R., Gourtovaia, M., 2005. Towards integrated performance-driven generative design tools. Autom. ConStruct. 14, 253—264.
Simon, H.A., 1969. The Sciences of the Artificial. MIT Press, Cambridge, Massachusetts.
Stiny, G., March, L., 1981. Design machines. Environ. Plan. Plan. Des. 8, 245—255.
Sutherland, I.E., 1963. Sketchpad: A Man-Machine Graphical Communication System. PhD thesis. Massachusetts Institute of Technology.
Szalapaj, P., 2001. CAD Principles for Architectural Design. Architectural Press, Oxford.
Terzidis, K., 2003. Expressive Form: A Conceptual Approach to Computational Design. Spon Press, New York.
Terzidis, K., 2004. Algorithmic design: a paradigm shift in Architecture ?. In: Proceedings of the 22nd eCAADe Conference, pp. 201—207.

Terzidis, K., 2006. Algorithmic Architecture, first ed. Elsevier Ltd.

Turrin, M., Von Buelow, P., Stouffs, R., 2011. Design explorations of performance driven geometry in architectural design using parametric modeling and genetic algorithms. Adv. Eng. Inf. 25 (4), 656—675.

von Buelow, P., 2012. Paragen: performative exploration of generative systems. J. Int. Assoc. Shell and Spatial Struct. 53 (4), 271—284.

Whitehead, B., Eldars, M.Z., 1965. The planning of single-storey layouts. Build. Sci. 1, 127—139.

Wiener, N., 1948. Cybernetics: or Control and Communication in the Animal and the Machine, first ed. Hermann & Cie) & Cambridge (MIT Press, Paris.

Wolfram, S., 1983. Statistical mechanics of cellular automata. Rev. Mod. Phys. 55, 601—644.

Woodbury, R., 2010. Elements of Parametric Design. Routledge, New York.

Yessios, C.L., 1973. Syntactic Structures and Procedures for Computable Site Planning. PhD thesis. Carnegie-Mellon University, Pittsburgh.

Yu, R., Gero, J.S., 2015. An empirical foundation for design patterns in parametric design. In: Proceedings of the 20th International Conference of the Association for Computer-Aided Architectural Design Research in Asia, pp. 551—560.

Zarei, Y., 2012. The Challenges of Parametric Design in Architecture Today: Mapping the Design Practice. Master Thesis. University of Manchester.

Zboinska, M.A., 2015. Hybrid CAD/E platform supporting exploratory architectural design. CAD Comput Aided Des 59, 64—84.

Zee, A.v. d., Vrie, B.d., 2008. Design by computation. In: Proceedings of the Generative Art Conference. Milan, Italy.

Zhang, P., Xu, W., 2018. Quasicrystal structure inspired spatial tessellation in generative design. In: Proceedings of the 23rd CAADRIA Conference, pp. 143—152.