Models of Subjectivity and Intentionality in Computational Architecture

From Centralized to Distributed Approach

Duygu Tüntaş Karaman

METU Department of Architecture
duygut@metu.edu.tr

Abstract. Triggered by the dominant criticisms on the formalism of current computational approaches and algorithmic modes to form generation, this paper challenges this view on computational design methods that are claimed to be incapable of embracing subjectivity and artistic expression, which in turn lead to data-driven forms as outcomes of pure calculations and rationalistic procedures. Providing a discussion and a framework on the disregarded dimensions of subjectivity in computational design processes, it proposes a tripartite model – centralized, partial and distributed approach to computational design – to understand and assess the condition of subjectivity and intentionality and reveal a possible shift from a centralized approach to a distributed one.

Keywords: Subjectivity, Design Intention, Computational Design

1 Introduction

The subjectivity of the architect and its related modes of design thinking repeatedly come to a point of question with advances in technology, which is further emphasized by the involvement of computation and associative methodologies in the design process that requires a rationalization of procedures both mentally and operationally. In the turn of the century, with the developments in computer science and increased capacity in information processing provided by the computational paradigm, the studies on computational design display great interest in complexity management. While, the aim is to cope with the intricacy of data, the ‘improved means and methods used in complexity management do not reduce but rather increase’ the complexity of design problems [1]. In order to respond to the rapidly changing status of tools and mindset, an epistemic choice in favor of rationalization with avoidance of subjectivity and intentionality has been made [1]. This condition leads to a determination in the dominant mode of computational thinking towards design confirmation in the form of
rationalization, optimization and efficiency, instead of exploring new ways to deal with subjectivity, intuition and artistic expressions of the designers.

The computational thinking requires an unambiguous translation of ideas into quantitative data. On the other hand, according to Zeynep Mennan, the qualitative notions of human thinking are claimed to be against the mathematical nature of computation, which creates a problem, as well as a challenge, of encoding subjectivity and design intention in computational processes [1]. As she points out, this condition originates from the epistemic opposition between subjectivity and rationalization, which repeatedly appears parallel to the gap between human and computational thinking [1]. Problematization this current approach to computation, the aim of this paper is to discuss the disregarded dimensions of designer’s subjectivity and intentionality in computational processes.

2 The Dominant Approach to Computational Design

In the book The Electronic Design Studio (1990), George Stiny distinguishes the creative act of designer from the role of computational procedures, which are problematic with their structured nature by stating that: ‘Designers do many things that computers don’t. Some of these are bad habits that the stringencies of computation will correct. But others are basic to design, and cannot be ignored if computation is to serve creation and invention.’ [2]. He then emphasizes the importance of ambiguity in design, ‘where it fosters imagination and creativity, and encourages multilayered expression and response,’ which computational procedures cannot incorporate due to the structured nature [2].

In the foreword of the book Expressive Form (2003), William Mitchell approaches the problem of dominant computational approach from a pragmatic-formal level. He associates the formal tendencies with which the software provides, an ‘economy of shapes’ that suggests the availability and ease in the creation of some forms with certain methods, while its expansion and restructuring has been made through the advancements in computer technology at the turn of the century. [3] In the book, Kostas Terzidis – who is the author – defines the problem from a methodological perspective [4]. He notes that:

‘What makes computation so problematic for design theorists is that it has maintained an ethos of rationalistic determinism -the theory that the exercise of reason provides the only valid basis for action or belief and that reason is the prime source of knowledge- in its field. Because of its clarity and efficiency, rationalistic determinism has traditionally been a dominant mode of thought in the world of computation. The problem with this approach is that it assumes that all computational activities abide by the same principles. In contrast, intuition, as defined in the arts and design, is based on quite different, if not opposing, principles [...] This mode of thought comes in contrast to the dominant computational model where methodical, predictable, and dividable processes exist.’ [4]
Terzidis reveals that the world of computation, in which a more rational, confined, organized, and methodical model exists, is resistant to such characterizations belonging to the human world, where ‘intuition has been an underlying assumption for many design activities.’ [4]. Elaborating on this division, he claims that the mathematical processes can easily be translated into quantitative methods, thereby, can be controlled through computation, whereas ‘manipulations, evaluations, and combinations of these processes are qualitative processes and as such can be handled by the architect.’ [4]. As he continues, on the point that we shift our design modes from manual to computerized, there occurs a necessity to ‘integrate the two seemingly contrasting worlds, that of intuition and that of computation.’ [4]. The outcome of such reconciliation may provide an alternative to the dissolution of subjectivity.

In a similar way, Axel Kilian considers computation to be in many cases ‘an obstacle [...] in translating design intent’, since; ‘it lacks the fluidity of human thoughts.’ [5]. And he proceeds arguing against this dominant view by stating that:

‘Design should not be solely about the execution of established processes but about querying the understanding of the factors involved. This is a much more complex task and it goes far beyond the traditional geometric and numerical representation of current computational practices but it happens in designers minds regardless of the involvement of computation.’ [5]

By extension to this, Kilian proposes to see the critic of the dominant approach to computational design not as ‘a glorification of human designers’ but as ‘a reminder of the respective strengths and weaknesses of the different approaches,’ and not to perceive them as competing processes but ‘a potential collaboration between design in the mind and its externalized computational processes.’ [5].

2.1 Challenging the Nature of Computation

Computational methods are regarded as rational because of its ‘mechanistic nature,’ and parallelwise, they are considered as incapable of ‘artistic sensibility and intuitive playfulness in their practice.’ [4]. However, it is possible to claim their subjectivity, but since, subjectivity and objectivity are simultaneously emanated through the act of coding, it is a different king of subjective construction that we are no longer able to detect with the conventional cognition of form. This results in a highly embedded subjectivity and design intention existing within the act of code writing, which may or may not be expressed in the geometric definition of form. Thereby, the visual unreadability of subjectivity in the outcome of code and the condition of being too generic results in such claims that computational methods are incapable of including subjective design thought processes.

In her recent paper ‘Mind the Gap: Reconciling Formalism and Intuitionism In Computational Design Research’, Zeynep Mennan explains the dominant attitude towards this problem as an epistemic choice in favor of rationalization over subjectivity and intentionality, and she reports that the criticism of the dominant mode of computational design led to a re-appearance of subjectivity together and against an
increased formalism in the field of computation [1]. Rather than staying in the limits of optimization and rationalization in form control, designers started to explore other potentials by combining the mode of scientific reasoning and human reasoning, so that, ‘the gap’ between computational rationality and the expressivity and subjectivity of the designer could be joined [1].

Roland Snooks observes that based on the analytical and generative capacity of computational thinking, the algorithmic approach is an agent-based bottom-up approach where there is no pre-determined idea of form, and form is dependent on the capability of the architect to ‘encode architectural intent within the operation of the algorithm.’ [6]. As he explains, algorithms are used as generic templates for architects and they are ‘abstract formal generators operating on an appropriated logic, devoid of any recognition of the architectural problem or proposition.’ [6]. Recent attempts reveal that such reconciliations are possible, between creativity and reason, between the subjectivity of the architect and the rationality of computational procedures [1]. Therefore, a new model in which subjectivity and design intention is being encoded within the operation of algorithm is replacing the dominant computational model.

Although this research has been triggered by the dominant criticisms on the formalism of current computational approaches and algorithmic modes to form generation, it challenges this view on computational design methods that is claimed to be incapable of embracing subjectivity and artistic expression, which in turn leads to data-driven forms as outcomes of pure calculations and rationalistic procedures. Then, it aims to demonstrate the possible ways to achieve different methods that deal with this problem with the assumption that subjectivity and intentionality of the architect has not been replaced or dissolved in computational processes but rather has been embedded within the design process and become indirect and invisible.

3 A Tripartite Model for Computational Design

In order to assess the condition of subjectivity and intentionality in computational design processes, this research proposes a tripartite model as an interpretation of the terms centralized, decentralized and distributed models of network, which are initially diagrammatized by Paul Baran in 1964 [8] and defined in the field of computation, network and communication sciences for management and organization of information [9] (Fig.1).
increased formalism in the field of computation [1]. Rather than staying in the limits of optimization and rationalization in form control, designers started to explore other potentials by combining the mode of scientific reasoning and human reasoning, so that, ‘the gap’ between computational rationality and the expressivity and subjectivity of the designer could be joined [1].

Roland Snooks observes that based on the analytical and generative capacity of computational thinking, the algorithmic approach is an agent-based bottom-up approach where there is no pre-determined idea of form, and form is dependent on the capability of the architect to ‘encode architectural intent within the operation of the algorithm.’ [6]. As he explains, algorithms are used as generic templates for architects and they are ‘abstract formal generators operating on an appropriated logic, devoid of any recognition of the architectural problem or proposition.’ [6]. Recent attempts reveal that such reconciliations are possible, between creativity and reason, between the subjectivity of the architect and the rationality of computational procedures [1]. Therefore, a new model in which subjectivity and design intention is being encoded within the operation of algorithm is replacing the dominant computational model. Although this research has been triggered by the dominant criticisms on the formalism of current computational approaches and algorithmic modes to form generation, it challenges this view on computational design methods that is claimed to be incapable of embracing subjectivity and artistic expression, which in turn leads to data-driven forms as outcomes of pure calculations and rationalistic procedures. Then, it aims to demonstrate the possible ways to achieve different methods that deal with this problem with the assumption that subjectivity and intentionality of the architect has not been replaced or dissolved in computational processes but rather has been embedded within the design process and become indirect and invisible.

3 A Tripartite Model for Computational Design

In order to assess the condition of subjectivity and intentionality in computational design processes, this research proposes a tripartite model as an interpretation of the terms centralized, partial¹ and distributed, which are initially diagrammatized by Paul Baran in 1964 [8] and defined in the field of computation, network and communication sciences for management and organization [9] (Fig. 1).

Fig. 1. Diagrammatization of the tripartite model based on the three types of network proposed by Paul Baran [8]. Redrawn by the author.

Translating these terms from network sciences to the field of computational design as centralized, partial¹ and distributed, constitutes a platform for a possible mapping of computational approaches, and it is instrumental to understand and assess the condition of subjectivity and intentionality, and then, reveal a possible shift from centralized to distributed approach.

3.1 Centralized Approach

The centralized approach can be described as the model where all data is sent to one central node, which then directs the data to the intended recipient [10]. This definition can be translated to the field of computational design where the central node corresponds to the main algorithm that determines and forms the whole.

In this approach, there exists an underlying idea about the formal logic rather than a pre-determinate idea of final form, and an algorithm can be designed or customized to write a specific code for form exploration [4]. According to Alejandro Zaera-Polo, such approach to computation can be interpreted as a centrally organized algorithmic system ‘that tries to articulate everything at once.’ [11]. Therefore, in this model, computation is essential to formal content and generation of form, and correspondingly, this condition makes this approach more vulnerable to the changes

¹ As an indirect translation from Baran’s diagram, I propose the term “partial” instead of “decentralized” for a better definition.
in computational methods and theory, as the structuring of the code directly affects the resultant configuration due to the organizational capacity of the algorithm [11].

The comprehension and assessment of this approach is substantially bounded to the recognition the potentials and limitations of algorithmic methods. As an example to this, work of Roland Snooks - whose approach falls into a rather experimental and innovative design field - reflects such centralized model, yet, according to Mennan, he manages to embed his design intentions and subjectivity within the quantitative logic of computation by participating actively in the automatic evaluation processes through ‘strange feedback’ which attempts to hybridize emergent characteristics of bottom-up algorithmic processes with the architect’s top-down subjective decision mechanisms [1].

Such interference to centralized algorithmic systems is a great difficulty, but similar attempts will extend the formal and organizational capacity of computational design processes, this time enabling more of subjective domain from the architect as well.

### 3.2 Partial Approach

*Partial* computation is a method used in computer science to evaluate and optimize partial programs with the given parameter values [12]. If we borrow and apply this definition to the domain of computational architecture, it suggests the application of computational methods to evaluate and optimize partial phases of form generation with the given parameter values. In this approach, algorithmic methods are applied to certain phases or parts of design, which then cause the formal content to become partially dependent on the computational content.

Studies based on optimization and efficiency can be positioned under this approach, where the main reasoning in the use of computation is rather *explanatory* through optimization Pre-rationalization and post-rationalization can be interpreted as the two dominating uses of this approach.

The intentionality in the use of computation is similar in both processes of pre-rationalization and post-rationalization, however in pre-rationalization, the rationalization process is superior to form generation process, whereas in the post-rationalization, the subjectivity is superior and rationalization is inferior to form generation. Therefore, it is possible to state that, in the former approach, rationalistic-determinism is the dominant approach, on the contrary, for the second approach; the formal logic is subjective and intuitive, yet partially rationalized.

As a critique of this performance or optimization approach to computation of form, David Benjamin examines efficiency and creativity, the two contrasting yet complementary concepts, and their implications in the field of architectural design. He names them, *exploitation* and *exploration*, meaning in sequence, ‘utilizing existing’ and ‘searching for new.’ [7]. He states that:

‘Designers interested in exploitation prefer a narrow, continuous design space, such as a slanted plane or a topological surface with one or two bumps. In this case, it is possible to quickly hone in on the region of best performance and to locate the single
global maximum. The simpler the design space is, the faster they can find the optimal design.

Designers interested in exploration prefer a wide, discontinuous design space, such as a jagged mountain range with multiple peaks. In this case, there are many distinct regions of good performance, and it is often possible to find multiple local maximums that are both interesting and high-performing, even if they are not the global maximum. The more complex the design space is, the more likely it is that they will make an unpredictable discovery.’ [7]

Benjamin also suggests introducing ‘subjective criteria’ into the optimization processes in order to integrate the seemingly separate qualities of human intuition and creativity with computational thinking. Even though such method is under-utilized, it enables to incorporate subjective criteria, such as atmosphere, aesthetics and program, with objective technical criteria, like structural performance and circulation efficiency, in the same optimization process [7]. In such a process, he argues that the subjectivity of the architect is translated into objectives and value judgment, and the creativity of the designer comes from ‘designing objectives and designing experiments rather than simply designing solutions,’ which bring about the role of the architect more engaged in designing the problem and focused on potential design space, ‘the complex topological surface’. [7]. About the subjectivity in these processes, he claims, ‘although they might be buried and hidden, they are there.’ [7].

It is possible to name pre-rationalization, post-rationalization and reverse engineering under this model.

**Pre-rationalization**

As the name clearly expresses, in this approach, the rationalization process is at the early stages of form generation, consequently the formal logic is dependent on the initial-factual data and therefore, it can be argued as highly objective. This approach can also be defined as a data-centric approach since the form is optimized from the beginning, and efficiency is the major decisive factor in form generation [13]. Thomas Fischer explains this approach by mentioning Buckminster Fuller: ‘[His] approach of addressing design challenges before they become acute, which he referred to as “comprehensive anticipatory design science” is largely based on the concept of pre-rationalization.’ [13].

As a result of the dependency of form on the data, the freedom and subjectivity of the architect in form generation can be evaluated as low, but since the construction of design problem and the intention to use such methods belong to the architect, it still embodies some degree of subjectivity but in a highly rationalized form.

**Post-rationalization**

In this approach, the use of computation is partial and rationalization process is placed at the final stages of form generation. The formal logic is dependent on the intuitive
and artistic decision making of the architect and therefore, it can be argued as belonging to the ‘subjective world of states of consciousness, or of mental states-with intentions, feelings, thoughts, dreams, memories.’ This approach can also be referred as the intuitive approach, since intuition is the major decisive factor in form generation. Although the dominant mode of formal logic is intuitive and therefore subjective, it still requires some sort of rationalization at the final stage in order to calculate structure and to construct and fabricate the final form (Such as surface tessellations of facades, some parts of structural design etc.).

Based on the intuitive decision-making of the architect, this approach mostly denotes a traditional top-down approach where the creator relies on her background knowledge and former experiences. As William Mitchell defines it, this ‘knowledge-based’ [14] approach can be problematized, as its design mechanism is a closed system or a ‘black box’ where the idea of form is in the designer’s mind and is pre-determinate.

Neil Leach remarks that Frank Gehry is a seminal name and a representative of this formal approach in which architect is considered as the creator, who ‘imposes form on the world in a top-down process’, while engineers are responsible for the making of form as close as possible to the architect’s initial expression [15]. Although, there exists some degree of rationalization in Gehry’s projects, the formal logic is purely subjective in which intuition is considered as the source of inspiration.

Reverse Engineering

There exist a more generative version of post-rationalization, which deals with reverse engineering in order to translate not just the subjectively constructed form in a computable representation, but also to extract the underlying logic and geometry of the form to an algorithm. By this way, this approach enables more than just post-rationalization but breeding new variations based on the initial form and extracted information [16].

The biggest challenge here is the involvement of a secondary designer and his/her ability to encode the subjectivity in the architect’s mind and to rationalize this highly intuitive and subjective formal logic to the algorithmic logic. Mark Burry’s research on Gaudi’s design of Sagrada Familia can be an appropriate example of this type of an approach [17]. Here, according to Neil Leach, Burry explores ‘digital techniques for understanding the logic of Gaudi’s own highly sophisticated understanding of natural forces.’ [15].

As a remark on the partial approach to computation, it can be stated that it is highly practical based on the design intention. However, it fails in rendering a totalitarian architectural approach, and thereby becomes a limited application within a larger content. It stays within the borders of validation by being explanatory rather than exploring the potentials brought along with computational paradigm. The

---

condition of this approach that doesn’t fully test the limits of computing subjective design thinking, it becomes inapplicable both to rationalization of subjectivity and subjectification of computation.

3.3 Distributed Approach

In the simplest version, distributed computation can be defined as the condition where the use of algorithms and codes are multiple and distributed to the different and particular stages of design. Different from partial computation, this approach includes both design exploration and exploitation, and furthermore, it is flexible and intention-oriented [11]. As Alejandro Zaera-Polo explains, it is ‘the co-evolution and optimization of relationships between multiple routines, mediated through the mainframe, which is able to produce real innovation, rather than the heaviness of a centrally organized system that tries to articulate everything at once.’ [11].

This approach can also be referred as a non-linear workflow approach in which computational methods are used and customized in a certain degree to adopt the subjectivity and intentionality of the architects. It includes employing custom and disposable codes, which are ‘intentionally purpose-built for the task at hand’, to respond to specific problems or spontaneous needs at certain phases of the design process in order to encourage creative thinking rather than perfect the code itself [18].

Accordingly, this dismantling and distributing computation throughout the design process makes it more efficient and flexible to encode subjectivity and intentionality, therefore subjectivity of the architect becomes more solid due to the radical increase in his/her freedom and control over form. Furthermore, it enables instantaneous design experimentation and rationalization within the same system by integrating two seemingly contrasting worlds - that of computation and intuition - with more freedom in subjective criteria.

The challenging part of this approach is the management of the complexity created by the involvement of different parts (specialized design territories) and parties (vast amount of specialized information) distributed within the same process. However, it allows design collaboration and inclusive design practices in order to respond the increased amount of information and complex design problems. This leads a great extension in the capacity of the architect.

4 Conclusion

Along with the methodological and epistemological changes in design thinking and modes of reasoning, this ongoing research expects a shift in the condition of subjectivity and intentionality in comparison to the dominant computational design approach and its modes of reasoning. It is argued in the paper that such a change is indicated through definitions of new modes of subjectivity within computational design that are impure, distributed and embedded. The gap created by the shift towards the language of computation and its associated rationality requires reestablishing the modes of subjectivity and intentionality, since “[c]alculation leaves
an incomplete space that cannot be saturated with information alone and waits to be filled with meaning and interpretation.” [1]. This gap emerging from the epistemological opposition of human and computational rationality constitutes the point of departure in the search for alternative approaches restating the subjectivity and intentionality of human agency in the field of computation.

In 2009, Neil Leach points out a shift prior to the introduction and multiplication of computational methods in architectural design that ‘[…]the architectural imagination has been displaced into a different arena – into the imaginative use of various processes.’ [15]. In 2012, the editor of the book Digital Workflows in Architecture, Scott Marble, identifies a further shift that the dominant computational approach experiences: ‘from process to workflow.’ [19]. Marble states that ‘the identity of the architect is largely built upon her or his ability to author design solutions’ and the challenge is in the ‘capturing the full range of architectural design intent within digital workflows,’ and he suggests the proper formation and expanded use of these workflows, which has the potential to transform the subjectivity and intentionality with freedom and control that have been ‘increasingly displaced by technologically mediated processes over a long time’. [19].

This study proposes a further shift in tandem with these alterations in the definition of designer(s)’ subjectivity and intentionality within computational design processes. The centrally posited subjectivity and intentionality is moving towards a distributed and embedded model as the use of computational methods becomes multiplex, distributed and intention-oriented.

Acknowledgements. This paper is stemmed from the author’s ongoing PhD study at METU Department of Architecture under the supervision of Prof. Dr. Zeynep Mennan.

References

8. Baran, P.: On Distributed Communications, RAND (1964)
Acknowledgements.

This study proposes a further shift in tandem with these alterations in the computational design designer(s) subjectivity and intentionality within computational design workflows, which has the potential to transform the subjectivity and intention within digital workflows, 'capturing the full range of architectural design imagination has been displaced into a different arena of computational methods in architectural design that are largely built upon her or his ability to author design solutions' and the challenge is in the identity of the architect.

In 2009, Neil Leach points out a shift prior to the introduction and multiplication of digital workflows, which has the potential to transform the subjectivity and intention within digital workflows, 'capturing the full range of architectural design imagination has been displaced into a different arena of computational methods in architectural design that are largely built upon her or his ability to author design solutions' and the challenge is in the identity of the architect.

This paper is stemmed from the author's ongoing PhD study at METU Department of Architecture under the supervision of Prof. Dr. Zeynep Mennan.

References