xCoAx 2020
Computation,
Communication,
Aesthetics & X
2020.xCoAx.org
Graz, Austria

Allographic Drawing: Agency of Coding in Architectural Design

Corneel Cannaerts
corneel.cannaerts@kuleuven.be
KU Leuven, Ghent, Belgium

Keywords: Architecture, Drawing, Coding, Agency, Process.

Allographic Drawing explores the agency of coding in architectural design processes and its impact on architectural drawing and the allocation of authorship. The paper uses drawing as a lens to look at coding in architectural practice and argues that engaging with coding introduces novel ways of mediating between context, proposition and constructed artefact. Against the backdrop of the paradigm shift from drawing-based representation to model-based simulation, this paper argues that we can look at drawing as a means of understanding the mediation of coding in architectural design processes. The research looks specifically into algorithmic approaches to scanning and mapping environments, ideation and exploration of variation within architectural design processes and the translation between design proposition and material artefact through digital fabrication.

1. The Shifting Role of Architectural Drawing

Drawing has been the principal means for architecture to analyze and map contexts, to explore design propositions, develop architectural projects, to communicate and disseminate design ideas and anticipate processes of fabrication, construction and assembly. The emergence of the profession of architect, coincided with the formation of drawing as a means of claiming authorship in architectural design (Evans 1997, 160). In contrast to most other creative disciplines architects do not work with the objects they design directly, i.e. architects do not work on site or make buildings, but always work through some intervening medium[1]. Until recently the central medium in architectural practice was drawing, which comes in various forms, from sketches, diagrams, perspectives, over plans, elevations, sections to technical drawings and details. Architectural drawings are not autonomous objects; they are part of a multimodal mediation process that includes other documents such as models, descriptions, calculations... (Allen 2009, 41). The relationship between the architect, the site, the design proposition, building and the role of the drawing is complex (Sheil 2012), although instrumental in constructing buildings, architectural drawings do not completely determine the building, nor do all drawings anticipate processes of construction (Groak 1992, 150).

The widespread adoption of digital technologies has deeply affected architectural practice. All phases of the design process involve the use of computation in some form or another: from ideation, schematic design, design development, over fabrication and construction, to use, maintenance and occupancy. The most direct impact can be seen in the media, tools and procedures for designing and constructing architecture. In the early adoption of computer-aided design (CAD), architects approached digitalization of architectural media as merely digital versions of well-known analogue tools of drafting, modelling and rendering, this did not directly affect the role of the drawing. However, through recent developments in computational design, parametric modelling, digital and robotic fabrication and building information modelling (BIM), we see a shift away from an analogue drawing based approach to a digital model based approach (Kudless and Marcus 2018, 47). This is symptomatic of a deeper cultural shift from representation to simulation. Architectural drawings operate based on a clear understanding of the difference between the drawn and the made, the representation and the represented. Simulations behave, at least temporarily or partly, as the simulated, aiming to close the gap between the simulation and the building (Scheer, 2014).

The role of architectural drawing, as the disciplines foundational means of designing and producing architecture, is deeply affected by these shifts. On the one hand architectural drawing, both analogue and digital, have been dismissed as an anachronistic practice inhibiting architecture from

[1] While other disciplines also work preliminary sketches and drawings, in architecture the distance between the medium in which a design is developed and the artefact it represents is substantial. See Robin Evans, *Translations from Drawing to Building, and Other Essays* (1997), p. 161.

truly embracing the innovation provided by digital technologies (Silver 2006). On the other hand, digital technologies have liberated drawing from its purely instrumental and representational role, which has resulted in a renewed interest in architectural drawing in practice and academia, as is demonstrated by numerous publications, conferences and exhibitions[2]. The inspiration for this revival ranges from a nostalgia for the central role the drawing allowed architects to claim in processes of design and construction, over a reevaluation of the craft in architectural practice (Riedijk 2010), to embracing the potential of drawing to resist the deterministic nature of building information modelling (Kudless and Marcus 2018, 47). To some degree the revival of architectural drawing coincides with a rejection of digitalization as such, idealizing or even fetishizing the analogue practice of architectural drawing. However, more prominently we see the dissolution of the dichotomies of analogue and digital into post-digital drawing approaches (Leach 2018).

2. Allographic and Autographic Practices

The central role of drawing in processes of designing and making has led to different understandings of where authorship can be located in architectural design. One could locate authorship in the drawing, as it captures the design intent of the architect without being constrained by the contingencies of building practice (Till 2009, 44–56); or you could locate the authorship in the building, reducing the drawing to a mere instrument to arrive at its construction. In practice, these extreme positions are hard to maintain and authorship lies somewhere in-between, and is partly collaborative: The drawing only partially reflecting design intent, introduces its own qualities, and cannot completely control the process of building that depends on many parameters and external influences. Or, as Stan Allen states: "*architectural drawing is in some basic way impure, unclassifiable. Its link to the reality it designates is complex and changeable*" (Allen 2009, 41).

In his classification of different art forms, Nelson Goodman makes a distinction between autographic and allographic art practices (Goodman 1976) in autographic arts, such as painting and sculpture, the authenticity of the work depends on it being executed by the artist; in other words, it bears the traces of the hand of the artist[3]. In allographic arts, such as music or poetry, the work can be executed without the direct presence of the author. Where autographic arts work directly with the matter at hand, allographic arts work through notation, usually leaving execution to others. Allographic arts are often temporal and ephemeral or need coordinated execution by many people, as in a theatre or in an orchestra. Nelson Goodman considers architecture to be a "curious mixture" of autographic and allographic practices. Like all arts, it started out as the autographic practice of making and building but acquired allographic elements through the introduction

[2] Most notable: *Drawing Futures* conference at the Bartlett School of Architecture, University College London in November 2016, *Between Paper and Pixels: Transmedial traffic in architectural drawing*, Jaap Bakema Study Centre & TU Delft Oktober 2016, *Drawing Millions of Plans*, KADK Kopenhagen, 2018, The Drawing Show, A + D Museum, Los Angeles, 2018, to name a few.

[3] This is obviously a generalization, many examples of sculpture and painting can be found that are partly executed by other hands than that of the artist.

of notation in the form of the drawing. Unlike other allographic practices, architecture deals with concrete material and is not purely ephemeral, but its construction needs the coordinated execution by many people. Architectural drawings cannot be reduced to "pictures" of a future building, according to Goodman; he compares architectural drawing with a musical score, an instruction that combines graphic notations with texts and symbols. The instructions captured in an architectural drawing are not complete and need to be complemented through other documents, the process of building involves many decisions to be made, often requiring the architect to visit the construction site (Allen 2009, 48).

The appropriation of digital technologies urges us to rethink the divide between autographic and allographic practices in architecture. Carpo argues that since the introduction of computational design and digital fabrication, architecture has become a completely allographic practice since exact copies of digital files can be reproduced and fabricated as material artefacts regardless of the architect's presence (Carpo 2011, 71). Kolarevic on the other hand proposes that digital technologies such as digital fabrication and building information modelling have the potential for architects to reclaim its autographic past, going back to the idea of the building master pre dating the drawing as means of claiming authorship (Kolarevic 2005, 55). Striking in these positions is that they both express the belief in reclaiming a more central role for the architect and that they both approach digital technologies as being neutral means for designing and constructing architecture. In contrast, this paper looks into digital technologies not as a means of exerting control over design and making, but for the agency these bring to the design process. We are particularly interested in the *allo*[4], the otherness present in digital technologies and the other forms of drawing we can explore through engaging with coding. Coding is used not for closing the gap between design intent and materialised artefact, but for extending the journey of design exploration.

3. Appropriating Coding

Architects have appropriated coding in avant-garde architectural practice and research: from the pioneering work of the 60s and 70s using university mainframe computers and pen plotters, over the theoretical discussions and formal experimentation of the 80s and 90s to the experimentation with parametric modelling, creative coding and digital fabrication in the 00s and 10s. Throughout these lineages the impact of digital technologies in architecture, the appropriation of coding in architectural design processes was linked with the practices of drawing. This is revealed through the recent interest in the history of the impact of digital technologies on architectural design as demonstrated by the exhibitions and publications *Archeology of the Digital*[5] and *When is the Digital in Architecture* (Goodhouse et al 2017).

[4] Kostas Terzidis sees code as an extension of human thought, which is fundamentally different, or what Terzidis calls allo, derived from Greek, meaning other. Terzidis sees algorithmic computation not as an extension of human cognition, but as a fundamentally different form of cognition, see Terzidis, Kostas, *Algorithmic Architecture* (2006), p. 27.

[5] Archaeology of the Digital Exhibition at Canadian Centre for Architecture, Montreal 2013.

The recent developments in the impact of digital technologies in architecture, since the development of computational design, building information modelling and digital fabrication technologies has been mainly associated with pushing the agenda of a model-based approach of simulation. However, with the emergence of the post-digital and the renewed interest in drawing within architectural practice, some architects and researchers are explicitly exploring coding as a way to rethink processes and procedures of architectural drawing. Related work includes the *Machinic Protocols*[6] project by Edouard Cabay, the work and writing of Carlo Lostritto (Lostritto 2016) and drawings of Andrew Heuman and Andrew Kudless[7]. While there is a substantial amount of relevant related work in visual arts, graphic design and creative coding, the drawings in those practices have a different status within architecture. A prominent project is *Drawing Codes: Experimental Protocols of Architectural Representation*[8], resulting in an exhibition displaying newly commissioned drawings by prominent architectural practitioners and researchers (Kudless and Marcus 2018). Works explore code as a generative constraint to work with or against, code as means of rethinking the language of architecture, code as means of encrypting or obfuscating information in architectural drawings and code as a script or recipe. The drawing explorations discussed below are mainly dealing with the latter: code as a set of instructions resulting in architectural drawings.

4. Drawing Explorations

Rather than dismissing architectural drawing in favor of a model-based approach, the drawing explorations discussed below use coding as an extension of architectural drawing practices. The drawings explore the agency coding brings to processes of mapping, design ideation and translation into material artefacts, and asks how coding affects these aspects of architectural drawing. The drawings discussed below where produced by the author in the context of architectural design practice, design courses and research projects. Instead of extensively describing the context in which these drawings where produced we will reflect on the agencies of coding they demonstrate.

4.1. Sketching with Code

Architectural drawings play an important role in the ideation of architectural design, propositions are developed through sketching out design ideas. Drawings mainly aimed at design ideation rather than presentation or communication are called *process drawings.* While programming has its origins in computer science and engineering, today architectural practitioners have access to coding through graphic programming add-ons, text based scripting interfaces for design software to programming languages and environments specifically aimed at designers, architects and artists. A number of practitioners and researchers have embraced coding as a major

[6] See https://machinicprotocols.com/, consulted on 01/02/2020.

[7] See http://andrewheumann.com/ and https://www.matsys.design/, consulted on 01/02/2020.

[8] See http://digitalcraft.cca.edu/research/drawing-codes, consulted on 01/02/2020.

part of their design process, establishing an architectural culture of coding (Burry 2011). While there are many reasons for architects to engage with coding, the foremost reason is design ideation, or developing ideas through sketching with code.

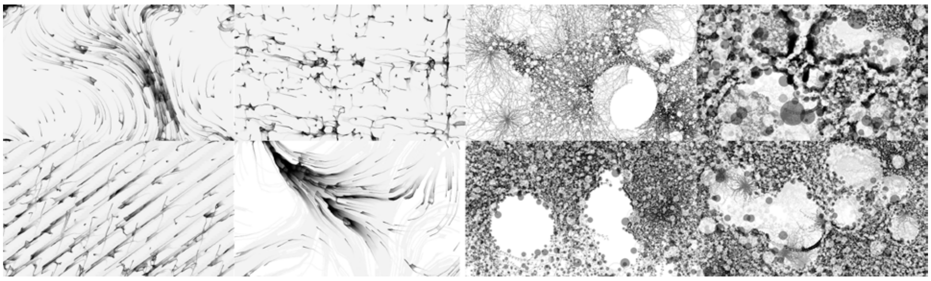

Fig. 1. Tinkering with Code: exploring generative systems as inspiration for design.

Tinkering with Code is a sketchbook of coding experiments, ranging from quick sketches, design experimentation, coding tools for other architects, designers and artists, writing code for specific fabrication machines. The sketches shown can be compared to doodles, or explorations of specific generative systems. Moments of playing, interacting with the graphical representation of code and tweaking values are alternated with changes to the code itself. Sketching through coding shifts the attention of the architect from working on a singular design solution, to exploring the logic and systems at play in design processes, it introduces a nonlinear way of exploring design variation.

Fragments and Figuration. Is a series of drawings made in preparation for a computational drawing workshop that looked into computational techniques for fragmenting and assembling those fragments into novel compositions. Rather than geometric primitives that come standard with most design software, the starting point for these algorithms where found objects in the form of downloaded 3D models. Existing elements are stripped of their familiarity, fragmenting them to be reassembled in estranged composition.

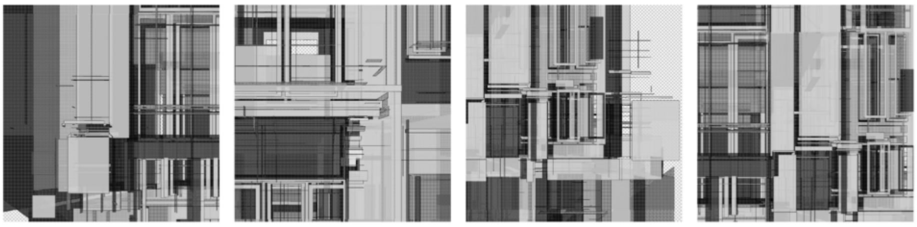

Fig. 2. Fragments & Figuration.

Working with code as a design medium provides the designer with different kinds of feedback on the screen: a graphical window showing the result of the running code, a textual one showing the actual code itself and possibly textual feedback through the console. Design through coding progresses through altering between working on the code itself and influencing the running software through various inputs. This alteration between making a using a custom design tool, between control and play can produce results that would not be possible with a standard design tool without access to the code that runs it. Text-based coding is an unforgiving medium, forgetting even one character will lead to a syntax error, and it is often hard to tell from the visual feedback alone what is exactly going on in an algorithm. These limitations can be overcome by continuously testing the code, incrementally building on working versions of the code and using the console to provide textual feedback, or by developing a debug mode that renders certain information on the screen.

4.2. Code as Lens

Architects have used drawing as a means of studying, analysing, mapping and understanding contexts, site conditions, spatial phenomena, material properties and existing constructions. Drawing is a selective and hierarchical, lens for observation, some aspects are drawn and emphasised while some are not. As such, there is no clear distinction between drawing as documenting observations and drawing as interpretation, developing a starting point for a design process. Coding allows architects to integrate various input devices for observation, as well as explicitly defining algorithms for selecting, mapping and visualising data.

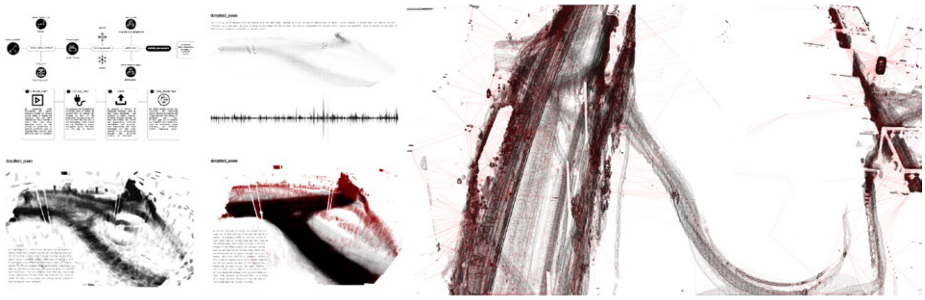

Fig. 3. Chrono Drawing: tracking motion through computer vision.

Chrono Drawings. This series of drawings emerged out of an interest in mapping time-based, dynamic and ephemeral phenomena. By processing video footage through a custom-made computer, vision algorithm various series of experimental drawings were produced that collapse a certain duration of time into a single drawing. One series of drawings[9] used the custom software to capture occupancy and flows, by tracking the motion of various users in a public space. The data was translated in a drawing accompanied by a number of other diagrams quantifying and visualising the data. The same software was also used to develop more experiential drawings using a moving camera or tracing the changing light conditions in both exterior and interior spaces.

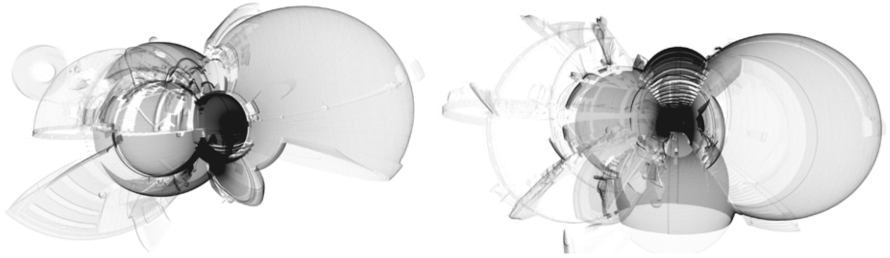

Fig. 4. Sphere Inversion: Environmental Scans

[9] Developed together with master students Toon Geukens, Martha Samyn, Maarten Moens & Amir Malakouti for the research elective Computation & Materiality, KU Leuven Faculty of Architecture in 2018

Sphere Inversions. This series of site-specific drawings and 3D printed artefacts that collapse a surrounding room into a spherical object. A spherical inversion is a transformation on coordinates in space: points on the sphere are not transformed, all points between infinity and the sphere are transformed into the sphere, with infinity collapsed in the centre of the sphere. This was done through modelling the context as a mesh, transforming the vertices and reconstructing the mesh after transformation. However, as planar geometry describing the room is transformed in spherical geometry, the mesh needed to be refined at an acceptable resolution for the model to be 3D printable. Several refining techniques were tested, instead of a uniform meshing of the room, we settled on casting rays from the centre of the sphere and refining the mesh where the rays intersected. This results in a mesh where the resolution depends on the distance from the inversion point, which proved to be an efficient way of improving the algorithm. In later versions, a lidar scanner was used to scan the room, the scanning process physically analogous to casting rays from a point, and as such, the lidar scanner is a suiting input device for the specific algorithm used.

When using computation in architectural design, material and spatial entities are captured or encoded into the quantifiable language of code. Code functions as a specific lens for looking at and describing material and spatial entities. In architecture, these descriptions are often geometric in nature and limited to describing the form of spaces and artefacts, although they can be extended to incorporate other quantifiable aspects. Code as a lens relies on data, which are essentially discrete and finite. In order to capture continuous phenomena, which can be spatial, material or experiential, they are sampled at discrete intervals, digital data always has a resolution: dots per inch, bit depth, sample rate, frame rate… As the Chrono Drawings and Sphere Inversions demonstrates the discrete nature and resolution of digital media is not merely a technicality, it introduces its own qualities that can become part of the design process.

4.3. Encoding Translation

A substantial amount of architectural drawings anticipate processes of fabrication, construction and assembly: from sections and plan drawings to details and diagrams. Architectural drawing has established a language combining orthographical projection, graphical conventions, textual and symbolic annotations that describe the fabrication and construction of a material artefact. The drawing annotates the construct to a degree of detail and clarity that it becomes practically feasible to construct by contractors able to understanding the language of an architectural plan. However, through developments in building information modelling, and digital and robotic fabrication, the drawing is increasingly being replaced by exchanging digital data. The design experiments described below explore digital fabrication, not for closing the gap between design idea and material artefact, but carefully examines what happens in the encoding of design propositions in digital files and the translation between digital files and material artifacts.

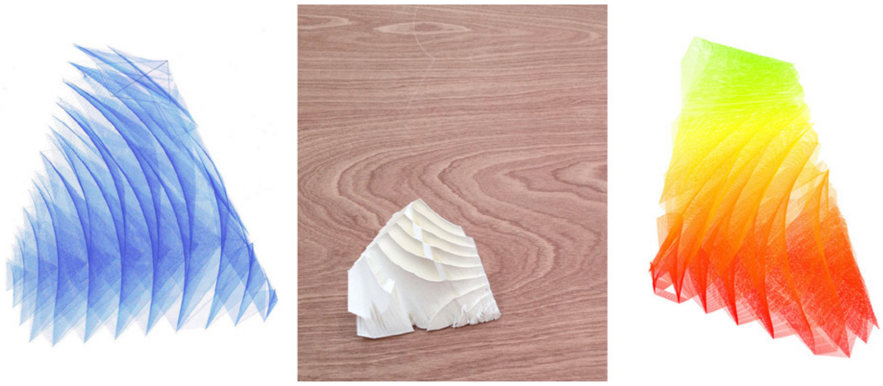

Fig. 5. Objects Without Skins: plotter drawing, 3D print and screenshot of the software.

Objects Without Skins. The Encoded Matter project, used an open-source self-build 3D printer, and explored the process of fabricating with this printer. Going from file to artefact requires processing a digital model through a machine specific software, in additive manufacturing this means going from a digital model describing the outer shape of an artefact – as a triangulated mesh – to a file that encoded the movement of the tool head and describes the fabrication process and how it unfolds in time – a g-code file. The starting point of this series was to go beyond the volumetric description of the artefact as a mesh and directly generate the g-code from a custom developed design tool, written in Processing and Grasshopper[10]. This results in material qualities that are radically different from what the fabrication technology normally produces, and actually exploits the difference between the encoded file and the materialized artefact, not as a failure but as design potential.

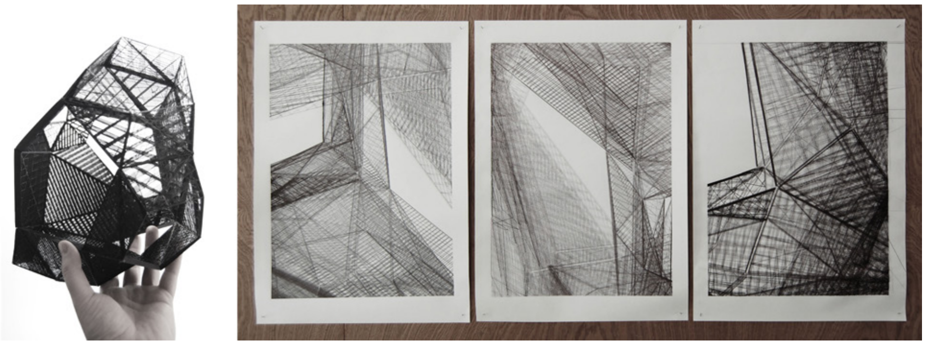

Fig. 6. Hatching With Matter: 3D print and plotter drawings.

Hatching with Matter. In architectural drawing, hatches and line-weights operate as symbolic notations of materiality, some properties of the represented material are reflected in how the hatches and lines are drawn, e.g. thicker lines and denser hatches tend to represent heavier and denser materials. The series of artefacts explore digital fabrication as drawing with matter. Instead of approaching hatches and lines as a symbolic notation, they are used directly to control translucency and density of the material.

[10] See https://processing.org/ and https://www.grasshopper3d.com/ consulted on 01/02/2020.

The same algorithms used for generating the fabrication files are used for making drawings through a pen plotter. These drawings are themselves outcomes of the same processes, they are not made as notations prior to fabrication, and the process of making them enacts a performance similar to the movement of the tool head of the fabrication machine.

Material Obfuscation. Obfuscation is the process in software development of making the source code hard to interpret for a human reader, while remaining executable for a computer. This experiment[11] looked into the allographic qualities introduced by photogrammetry and 3D scanning on the one hand and a d.i.y. filament printer on the other. Through various iterations of modelling, printing, scanning and repeating that process, the accumulative particular qualities or errors of both these technologies become amplified. The feedback loop accelerates the question of authorship, allowing the algorithms, the machinic fabrication processes and the material to actively contribute to the resulting fabricated artefact. The final iteration shows traces of the code, the triangulation introduced by photogrammetry, the layered and linear qualities of the slicing algorithm that controls the motion of the 3D printer, as well as material properties and limits.

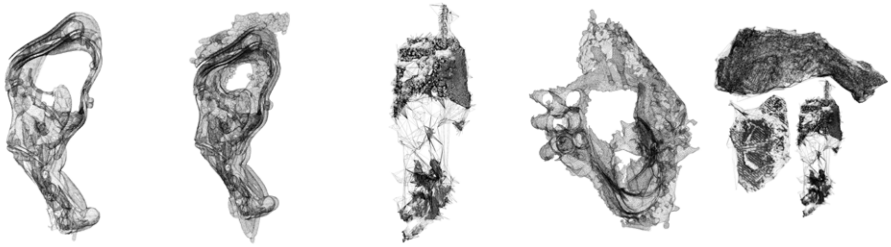

Fig. 7. Material Obfuscation: digital drawing sequence.

Since the algorithms are highly attuned to the fabrication and material nature of the 3D printed artefacts, their 2D counterparts can be seen as representational: they represent a material reality outside themselves, but the way they refer to this materiality is not symbolic, but rather an enactment of the same movements that can be made by a different machine to produce material artefacts. When drawing an architectural section, line-weights and hatches are used as a symbolic notation of materiality: the thicker the line, the denser the material. The hatches in the Hatching with Matter drawings might be reminiscent of hatches in architectural drawings, but they operate in a non-symbolic manner. As such, these drawings also acquire an experiential quality, and become non-representational. The nature of the drawing is altered through digital fabrication: The drawing loses its projective connotations and becomes an unfolded trace for the fabrication process of cutting or adding material.

[11] Developed together with master students Jari Jacquet, Joris Putteneers & Olaf Mitka for the research elective Computation & Materiality, KU Leuven Faculty of Architecture in 2017.

5. Discussion

Recent developments in digital technologies are affecting the role of drawing in architectural practice and research. Instead of rejecting the drawing in favour of a more model-based simulation approach this paper proposes using coding as a means of rethinking how architectural drawings can operate as means for mapping architectural qualities, developing design ideas and translating these ideas in artefacts and drawings. Situated within a renewed interest in drawing in architectural practice and research, *allographic drawing is* a specific form of post-digital drawing. The drawing explorations: *sketching with code*, *code as lens* and *encoding translation* are particular ways of understanding the agency of coding in architectural drawing. This understanding of agency of code emerges out of the specific work; there are probably other ways of framing the agency of coding in architectural drawing. These drawing explorations operate in-between the poles set out in the framework: negotiating between being authored and allowing technology to introduce allographic qualities, between model-based simulations and drawing based representation. The research proposes *allographic drawing* as a way describing the agencies introduced by coding in architectural design processes.

References

Allen, Stan.
2009. *Mapping the Un-mappable, in Practice: Architecture, Technique + Representation.* London ; New York: Routledge.

Carpo, Mario.
2011. *The Alphabet and the Algorithm, Writing Architecture.* Cambridge, MA: The MIT Press.

Cook, Peter.
2014. Drawing: *The Motive Force of Architecture.* AD Primers. London: Wiley.

Evans, Robin.
1997. Translations from Drawing to Building, and Other Essays. London: Architectural Association Documents.

Goodhouse, Andrew, et al, eds.
2017. When Is the Digital in Architecture? Montréal: Canadian Center for Architecture,

Goodman, Nelson.
1976, *Languages of Art.* Indianapolis/ Cambridge: Hackett Publishing Company

Groák, Steven.
1992. *The Idea of Building: Thought and Action in the Design and Production of Buildings*. London; New York: E & FN Spon.

Kolarevic, Branko.
2005. Information Master Builders. In Branko Kolarevic, Architecture in the Digital Age: Design and Manufacturing. London: Taylor & Francis

Leach, Neil.
2018. *We Have Never Been Digital,* In ACADIA // 2018: Recalibration. On imprecisionand infidelity. [Proceedings of the 38th Annual Conference of the Association for Computer Aided Design in Architecture (ACADIA) ISBN 978-0-692-17729-7] Mexico City, Mexico 18-20 October, pp. 20-29

Lostritto, Carl.
2016. *Computational Hatching*, Journal of Architectural Education, 70:1, 83-90, DOI: 10.1080/10464883.2016.1122475

Marcus, Adam; Kudless, Andrew.
2018. *Drawing Codes. Experimental protocols of architectural representation*. In ACADIA // 2018: Recalibration. On imprecisionand infidelity. [Proceedings of the 38th Annual Conference of the Association for Computer Aided Design in Architecture (ACADIA) ISBN 978-0-692-17729-7] Mexico City, Mexico 18-20 October, 2018, pp. 46-55

Murray, Janet H.
1997. *Hamlet on the Holodeck: The Future of Narrative in Cyberspace.* Cambridge, MA: The MIT Press.

Riedijk, Michiel ed.
2010. *Architecture as a Craft: Architecture, Drawing, Model and Position*. Amsterdam: SUN.

Scheer, David R.
2014. The Death of Drawing: Architecture in the Age of Simulation. London ; New York: Routledge.

Sheil, Bob.
2012. *Distinguishing Between the Drawn and the Made* in Achim Menges, Material Computation: Higher Integration in Morphogenetic Design Architectural Design. London: Wiley.

Silver, Michael.
2006. *Buildings Without Drawing: Automason Ver. 1.0.* In Programming Cultures: Architecture, Art and Science in the Age of Software Development, ed. Michael Silver, London: Wiley.

Terzidis, Kostas.
2006. Algorithmic Architecture. Amsterdam ; Boston: Architectural Press

Till, Jeremy.
2009. *Architecture Depends.* Cambridge, MA: The MIT Press.