Drawing to See / Drawn to Seeing: Multimodal Reinterpretation in an Autonomous Drawing Machine

PENMAN Scott
MIT Design Lab
scottdpenman@gmail.com
doi: 10.21606/drs.2018.687

Drawing is celebrated as a powerful tool in the designer’s toolbox. While it is generally understood to be a useful method in the visualization of abstract concepts and complex scenarios, it is equally important to understand drawing’s role in the pursuit of novel solutions. Drawing affords enough physical and conceptual space between modes of representation to permit the designer to reinterpret the design. With this creative act as inspiration, this paper reviews the topic of computational play as a framework for implementing autonomous, creative design in computational systems. The framework is demonstrated in a proof-of-concept drawing machine. Specifically, the machine uses shape grammars to exhibit multimodal reinterpretation in its drawing process: It physically draws each step of the design and then visually reinterprets the drawing in an iterative, playful process.

shape grammars; computational play; reinterpretation; drawing machine

1 Introduction

In The Thinking Hand, architect Juhani Pallasmaa depicts an elaborate relationship between hand, mind, and drawing. He describes the inherent spatiality of the act of drawing and celebrates its role in the design of space, stating:

When sketching an imagined space, or an object being designed, the hand is in a direct and delicate collaboration and interplay with mental imagery. The image arises simultaneously with an internal mental image and the sketch mediated by the hand...The second meaning of the word 'drawing' - to pull - points to this essential meaning of the drawing as a means of pulling out, revealing and concretising internal mental images and feelings as much as recording an external world. (Pallasmaa, 2009, p.91-92)

This is a sentiment echoed repeatedly by architects and designers. In Why Architects Still Draw, architect and poet Paolo Belardi celebrates drawing as an act that will never be fully replaced by digital tools. Belardi lauds the speed and ease of sketching, and the possibilities it affords. He writes:
Sketching is a notational system that is not only rapid and ready but also a mode of accessing information...Sketching, both because of its small dimensions and indeterminacy on paper as well as its independence from any code, is able to continuously regenerate itself, always offering new suggestions - sometimes ones that prove surprising even to their author. (Belardi, 2014, p. 30)

Such descriptions point to drawing not just as a means for communicating problems to other parties or visualizing complex scenarios, but as an act that holds almost mythical creative potential. In reality, there is much less mystery to drawing’s power, even if there is still near-limitless possibility. Drawing, as the intentional effort to translate a concept between different modes of representation, inherently embraces the vagaries and novelties that arise in an active engagement with a medium.

While aspects of performance, engagement, and making are finding their way into the discourse surrounding creative practices, there has been less effort put forth toward implementing these behaviours in computational models. Drawing provides an entrance point to this discussion. This paper reviews the topic of computational play as a framework for implementing autonomous, creative design in computational systems. The paper reviews four characteristics key to computational play, with special attention paid to the role of multimodality. While multimodality is defined simply as variation in representational form, its benefits are discussed with respect to creative design practices. This framework is implemented in an autonomous drawing machine that iteratively plots, views, reinterprets, and interacts with a drawing process. The drawing process itself is based on shape grammars and is structured so as to encourage the kind of exploratory curiosity that lies at the heart of creative design.

2 Background | Drawing to Make

2.1 Drawing as Cognitive Offloading

Design is full of what Professor Richard Buchanan calls wicked problems (Buchanan, 1992). Such problems are ill-defined, indeterminate, dynamic, and confusing. The problem space of an architectural design, for example, is a complex, fluctuating landscape that incorporates often-conflicting considerations of aesthetics, culture, structure, and budget, among a myriad of other factors. In dealing with such issues, designers explore with limited or no knowledge of what the end goal should be, or even if the considerations of today will be the same as those of tomorrow. Often, before the designer can even pursue the answer to a question, he or she must formulate the question itself, an activity that philosopher and urban planner Donald Schön calls problem-framing (Schön, 1990).

Manoeuvring in such scenarios requires an open mind and a readiness for change. Schön describes the designer’s ability to deal with surprise as reflection-in-action, a type of improvisational problem-solving and on-the-spot experimentation that allows designers to try previously untested ideas (Schön, 1990). Sociologist Richard Sennett reflects on a similar scenario at the heart of craftsmanship, stating that the experienced craftsman must be able to localize, to question, and to open up a problem (Sennett, 2008).

Navigating such a complicated terrain is a challenging task that necessitates some type of supportive tool, as demonstrated by the ubiquity of drawing and model-making in creative industries. Drawing serves as a powerful method of inquiry, allowing the designer to realize even the most complex or abstract concept with relative ease. Drawing enables designers to offload and test their ideas, shifting their focus from one part of a design to another (Suwa & Tversky, 2002). This can be a useful tool in complex scenarios, as drawing allows designers to study a concept visually, rather than cognitively. At the same time, it permits the designer to approach the issue from a different perspective and see the problem anew.

Even as technology has progressed and digital tools have come to replace analogue ones, the humble pen-on-paper sketch still occupies a primary spot in the designer’s studio. Sketching is fast
and, with practice, borderline effortless. The sketched drawing is an intentionally quick, loose effort to visualize an idea. Too quick, and it is illegible, but labour over it for too long, and the sketch becomes a full-fledged blueprint, wearing undesirable grooves into the mind of the designer.

Schön summarizes this rapid process as a cycle of seeing and moving. He states that:

...designers construct and impose a coherence of their own. Subsequently they discover consequences and implications of their constructions—some unintended—which they appreciate and evaluate...Their designing is a web of project moves and discovered consequences and implications, sometimes leading to reconstruction of the initial coherence—a reflective conversation with the materials of a situation. (Schön, 1990)

A conversation demands a partner, and for the solitary designer, the medium of drawing stands in as an alternate voice.

2.2 Drawing in Shape Grammars

Computation theorists George Stiny and James Gips capture many of the practices of creative design in their formulation of shape grammars (Stiny & Gips, 1972). In shape grammars, visual drawing rules are used to perform calculations on shapes. These rules can be additive and/or subtractive, permitting any kind of transformation that can be drawn. The rules supply some kind of shape or set of shapes to be recognized (on the left-hand side), and a resulting shape or set of shapes to be drawn (on the right-hand side). In the case of the diagram below (Figure 1), the rule involves rotating a triangle 180 degrees around its centroid, indicated by the light grey cross.

![Figure 1 Rule application: The left side of the rule specifies what shape to look for, while the right side specifies what shape to draw. In this case, once an equilateral triangle is found, it is rotated 180 degrees around its centroid (erased and redrawn).](image)

Multiple rules can then be strung together into a computation. Each step in the computation, indicated by a double arrow, involves the application of a chosen rule, indicated by a single arrow (Figure 2).

![Figure 2. Stringing rules together into a computation.](image)
Stiny describes design drawing as an iterative, calculating process of open-ended discovery (Stiny, 2011). It is up to the designer to decide when and where to apply the rule. Within this process, the designer identifies and embeds content into what he or she sees as a part of the drawing calculation. Once the drawing step is complete, the content fuses and loses any of its previous definitions, leaving the designer to see it anew (Stiny, 2006).

Although shape grammars is formalized as a computational process, it can seem counterintuitive in the context of computer science. Why intentionally ignore the steps leading up to the current position? The reason stems from shape grammars’ appeal to novelty: Each new step brings with it a multitude of new content, some of which the shape grammarist may never have predicted. Stiny draws together a number of descriptions for this phenomenon from a variety of artists, linguists, theorists, and philosophers, calling it:

- ambiguity, emergence, epiphany, eureka (aha) moments, figuration, flexible purposing, impression, intuition, invention, irony, negative capability, new perception, privileged moments, re-description, strong imagination, vitality - strange surprises (Stiny, 2017)

How the designer acknowledges and processes this content is critical to the design process. Shape grammarists welcome such content, actively encouraging the kind of sudden insight that is so critical to design. In so doing, shape grammars reverses the typical relationship between description and calculation: Instead of calculating with content that has already been described, the shape grammarist (visually) calculates first, only then providing a description as a result of the calculation. The description of what’s on the page takes a back seat to the process of discovery that occurs during design (Stiny, 2015). Dethroning the description in such a way allows — in fact, encourages — multiple descriptions of the same content, once again appealing to the variability at the heart of design. In this way, shape grammars faithfully represents not just the technique of the drawing process, but the ideology behind it as well.

### 2.3 Drawing as Making

Schön’s seeing and moving and Stiny’s embedding and fusing bear a striking resemblance to one another. Both point to the agential act of isolating content in an externalized representation, and then transforming that content through the act of drawing. Additionally, both insist on design as an ongoing, cyclical construction of meaning (Figure 3).

![Figure 3. Comparing Schön’s description of design as a cyclical conversation of seeing and moving to Stiny's description of embedding and fusion in shape grammars.](image)

Perhaps another way to understand the similarities of these two approaches, as well as the potential that drawing holds for design, is through the idea of making. Anthropologist Tim Ingold uses this term to depict the creative process, describing it as “an ongoing generative movement that is at once itinerant, improvisatory, and rhythmic” (Ingold, 2010, p. 91). He later states that practitioners “are wanderers, wayfarers, whose skill lies in their ability to find the grain of the world’s becoming.
and to follow its course while bending it to their evolving purposes” (Ibid., p. 92). In a recent issue of Design Studies dedicated to this topic, professor of design and computation Terry Knight and co-editor Theodora Vardouli define making as:

...a process that is time-based (unfolding in real-time, in-the-moment), dynamic (changing), improvisational (dealing with uncertainty, ambiguity, and emergence), contingent (subject to chance and the unique), situated (within a social, cultural, physical environment), and embodied (engaging the (maker’s) active body and sensorimotor capabilities). (Knight & Vardouli, 2015, p. 2)

Characterized this way, design is an iterative, meandering search for interesting questions and possible values in an ambiguous and dynamic space of possibilities. Designers make their own way through this indeterminate space, often with very little guidance. In order to do so, designers cultivate certain behaviours—of curiosity, open-mindedness, and freedom from external constraint. These behaviours indicate a willingness to change that stands contrary to traditional notions of problem-solving.

All of this contributes to an understanding of the role that drawing plays in the design process. These curious, exploratory behaviours are facilitated by drawing, an added modality that forces the designer to translate and reinterpret the design at each step of the process. The act of drawing allows designers to roam, discovering both interesting questions and potential creative solutions.

3 Methods | Making the Machine

3.1 Computational Play

The creative process described above is fluid and dynamic, and providing some kind of all-encompassing description of it is a famously controversial effort. That being said, the activities embedded in design practice can serve as excellent references for the development of creative computational agents. The discussion of the multimodality of drawing fits into a larger framework that I call computational play. The framework draws from research into the nature of play and its role within the design process to create a set of characteristics typically observed in a playful designer. This research is summarized below.

Play is the autotelic behaviour of a subject temporarily exploring a system of rules (Penman, 2017). This definition succinctly captures many of the qualities of Dutch historian and early play theorist Johan Huizinga’s original description:

...a voluntary activity or occupation executed within certain fixed limits of time and place, according to rules freely accepted but absolutely binding, having its aim in itself and accompanied by a feeling of tension, joy and the consciousness that it is ‘different’ from ‘ordinary life.’ (Huizinga, 2014, p. 28)

It also bears a strong resemblance to game designer Brian Upton’s depiction of play as “free movement within a system of constraints” (Upton, 2015, p. 15). Playful behaviour is meandering and exploratory, but also very orderly, as it is based on rules (Gadamer, 2004). The rules form the playground, or setting, for the play. Huizinga refers to this as the magic circle of play (Huizinga, 2014). Players are driven by an inner curiosity and motivation, often eschewing external guidance in favour of their internal interests (Bruner, 1979). This characteristic is captured in play’s autotelism, which specifies that play has no goal other than self-perpetuation (Hein, 1968).

The framework for implementing play in a computational setup involves four characteristics. First, as elaborated in this paper, computational play is facilitated by multimodal representation conducted by the agent. This can take a variety of forms, but the process must be significant enough that it requires the agent to translate the work between different representations, reinterpreting the content along the way. As discussed, an example is the designer’s tendency to sketch. This externalizing process is key to the designer’s ability to shed a cognitive bias and approach a topic
from a different perspective (Suwa & Tversky, 2002). Insisting on multiple modes of representation acknowledges that the process of translation often ignites our subconscious, causing the kind of insight that often proves so fruitful to the design process (Stiny, 2015).

The second characteristic of computational play is its focus on generative techniques that do not rely upon results for justification. Designers—as well as many other artisans, craftspeople, and creative professionals—improvise. Rule-based behaviours work well in this regard, as they specify behaviour without relying upon efficiency or other heuristics for guidance.

Third, computational play is iterative. Combined with the insistence upon multimodal representation, this lends play a cyclical structure reminiscent of the design processes outlined independently by both Schön and Stiny (Schön, 1990; Stiny, 2011). Iteration opens the design process up to shifting contexts and changing requirements. Iteration grants the same level of importance to both question and problem. At the same time, it also results in a natural discretization of the creative process that is conveniently applicable to computation.

Finally, computational play should be autotelic, or internally driven. While the three traits described up until this point provide a strong framework for observable playful activity, autotelism serves as the inferred component to the process. Autotelism provides a strategy for untethering the designer and encouraging the kind of curious exploration at the centre of play.

3.2 Hardware
The framework discussed above might be difficult to reconcile with a traditional interpretation of computation, but it is not inherently difficult to implement in a computational machine. In order to test this framework, including the insistence on multimodal drawing, I have developed a computational drawing machine that is capable of drawing, viewing its own drawing, and making decisions based on what it sees. This machine has provided me with a way to explore my ideas, and with it I have attempted to demonstrate that computational play can be implemented relatively simply. This section will discuss the construction of the machine, which demonstrates computationally playful behaviour through a multimodal, generative, iterative, autotelic drawing process.

The machine hardware is a Makeblock XY Plotter (V2.0) resting on a 2 x 2 ft acrylic base (Figure 4). The Z-axis of the machine is outfitted with both a dry-erase marker and an eraser. The stepper motor for this axis is fixed with an elliptical rotor. Resting on this rotor are two separate 3D-printed mechanisms that individually hold the marker and the eraser. As the motor turns, the elliptical rotor causes one mechanism to rise while the other falls; turning in the other direction reverses the motion, yielding both “Draw” and “Erase” modes (Figure 4, upper-right corner). When the rotor is positioned so that the long axis of the ellipse is horizontal, both marker and eraser are lifted above the acrylic base, yielding “PenUp” mode (Figures 4, 5). In this way, the machine is capable of both additive and subtractive drawing procedures.
Figure 4. Exploded axonometric of drawing machine. The diagram in the upper-right corner shows how the motion of the elliptical rotor on the Z-axis allows the machine to both draw and erase.
A webcam sits near the centre of the machine, approximately 250 mm above the surface of the acrylic base. In order to ensure an evenly lit drawing surface, LED lights line the underside of the machine around the entire perimeter (Figure 6).

A connected computer acts as the digital "brain" to this plotter "hand" and webcam "eye." All of the computational processes—the webcam feed, the digital drawing manipulations, and the G-code processing—are managed in Grasshopper for Rhino. The components for receiving the video feed and communicating with the Arduino are provided through Firefly. Once the G-code has been determined, it is communicated from Grasshopper through Firefly to the Arduino, which then uses pre-loaded Grbl software to translate the G-code into commands for the plotter’s motors.
3.3 Software

Echoing the design process espoused by both Stiny and Schön described earlier, the machine works through an iterative cycle of embedding/seeing and fusing/moving. The computational framework is divided into three stages: machine vision, rule application, and plotting. These stages roughly align with the seeing-moving cycle of design, with the intermediate rule application section representing the conscious processes that designers use to choose what to draw. The resulting cycle is see-choose-move (Figure 8).
In the first stage, the machine views its drawing. The webcam feed that is passed to the computer is immediately processed into a more usable format through brightness and threshold controls. Next, the computer uses a set of custom machine vision scripts to extract edges from the image and collect them into digital curves. This rough geometry is then smoothed into a cleaner interpretation. A separate data tree is constructed that describes the connectivity of the curves: Each branch of the tree is an intersection of two or more curves, and each leaf is a neighbouring intersection. Using this data tree, the computer uses classic search algorithms to determine every possible closed, non-self-intersecting path that exists in the drawing. After removing duplicates, the computer uses the unique paths to build closed shapes, which are passed to the next stage in the computation.

In the second stage, the machine determines all of its possible next moves, decides whether or not it will continue, and chooses a move to make. The moves are determined through the use of the drawing rules described earlier: The machine takes as input a certain shape or drawing and outputs a modified shape or drawing. While any number of drawing rules can be utilized, even the use of a single rule can afford a vast space of possible trajectories.

The machine matches the curves supplied from the previous stage against all of its internal rules. Having discovered all of the shapes that match a particular description, the actual rule applications are carried out, resulting in a number of possible next moves. One of these shapes is then selected to be drawn. The rule application portion of the computation also includes the implementation of autotelism. This component takes as input both objective values (the number of possible next moves) as well as subjective values (a measurement of the machine’s own internal interest in continuing the activity) in order to determine whether or not to continue. While the subjectivity of the machine is simplified to a stochastic value, the importance of its inclusion lies in the unpredictable-yet-attributable behaviours that result. For more on this, please see Penman, S. (2017). Toward Computational Play.

In the third and final stage, the computer instructs the plotter to draw the new geometry. First, the digital lines are connected together into a toolpath, complete with initial travel distance, intermediate travel steps, draw and erase information, and concluding travel distance. This toolpath is then broken into individual points, which are converted into corresponding G-code and sent to the
machine. At the end of the drawing process, the plotter returns to the home position, at which point the entire computation iterates, returning to the machine vision stage.

3.4 Multimodality

While this drawing machine has been constructed to demonstrate all four characteristics of computational play, the very aspect of building the machine is an ode to the importance of multimodality. The process could potentially have been left in digital format without any externalized drawing, but to do so would be to miss the opportunities available in varying the mode of representation. The software “brain” of the machine stands as one representation of the design, while the drawing stands as another. Pulling the two apart creates the physical and conceptual space for productive exploration to occur.

The insistence on externalizing the drawing through the plotter and reinternalizing it through the webcam is inefficient by most computational standards, but this only highlights the gap between computational optimization strategies and creative design practices. As the multimodality of drawing is critical to design, so it has been deemed critical to the drawing machine.

4 Results | The First Drawings

The machine demonstrates a cyclical drawing process that is entirely reminiscent of a designer iteratively constructing a drawing. It begins with an initial shape, views the shape, and cycles through possible next steps based on its drawing rules. By overlaying footage of the digital drawing process with the physical one, we see the rule application process in action: After using machine vision to recognize the current geometry, the machine cycles quickly through its possible options, before choosing one and proceeding to the plotting stage. All three computational stages blend into one continuous process.

In an early drawing experiment, before the ability to erase had been added, the machine handily demonstrated the ability to deal with novelty imposed by environmental accidents. In this case, the only rule supplied was a triangle inscription rule visualized in Figure 9.

![Figure 9. Drawing rule involving the inscription of a triangle in another triangle, using the centerpoints of the outer triangle’s sides for the vertices of the inner triangle.](image)

Using this, the machine demonstrated the ability to see multiple options at once (Figure 10, [1-3]), including new options that showed up in subsequent stages [4-6]. In an interesting twist, the machine accidentally trailed the pen across the drawing as it was resetting during one iteration. Unfazed, it incorporated this additional line into the digital interpretation of what it saw [7], and then was able to recognize a new triangle as a result of the accident [8]. It deviated from the space of possibilities projected by the drawing rule and instead dealt directly with the content at hand.
After installing the erasing mechanism, the machine completed a reconstruction of a triangle computation described in Stiny’s *Shape: Talking About Seeing and Doing* (2006, p. 296) and reproduced below (Figure 11). In this computation, the only drawing rule is a triangle rotation rule, depicted in Figure 1. In each step, the vivid red outline indicates the shape that is chosen for rule application, and the light pink outline indicates the result of the previous step’s transformation. Critical to this computation is the emergence of a new, larger triangle: In Figure 11 [4], this triangle is visibly composed of the outer edges of the three rotated triangles. This triangle is only available via active engagement with the drawing rules and iterative reinterpretation of the content in the drawing.
In the images below (Figure 12), the lower red shape indicates the shape that the machine intends to erase (the offset is due to the physical separation between the pen and the eraser). The upper red shape is the shape that the machine intends to draw. After rotating (erasing and redrawing) the three initial triangles [1-3], the machine recognized the emergent outer triangle, which it proceeded to rotate [4]. It also recognized the emergent inner triangle, which it rotated [5]. This provided the machine with three new triangles to rotate [6-8]. In the final step, the machine accidentally trailed both pen and eraser through the drawing, resulting in the strange-looking result shown [9].
Figure 12. Drawing sequence using the triangle rotation rule in Figure 1. In each image, the lower red shape is the shape to erase (the offset is due to the physical separation between the pen and the eraser), and the upper red shape is the shape to draw. The drawing sequence largely follows the same progression as Figure 11.

In a subsequent drawing, the machine was supplied solely with the triangle rotation drawing rule in Figure 1. Due to a slight misalignment of the eraser, the machine repeatedly failed to fully get rid of the triangles it was trying to erase (Figure 13, [1-2]). Once again, the machine took these residual marks in stride. The lines resulted in additional, small triangles appearing in the drawing, and the machine repeatedly chose these accidental shapes as its targets for new drawing rule applications [3-4].
5 Conclusion

This machine is a proof-of-concept of the proposed computational play framework. It demonstrates an autonomous, playfully creative drawing process that is multimodal, generative, iterative, and autotelic. In early drawing experiments, the physical setup of the machine resulted in messy drawings that did not reflect the possibility space projected by the idealized computational process. While these outcomes could be written off as mistakes, I contend that it is more important for us to consider the opportunities they open up. As the machine began to make errors, I found myself moving to correct the process. It is that very reflexive action, however, that must often be stymied in order to make use of productive failure and discover novel opportunities. Where I might have quit the drawing process, the machine didn’t; in fact, it handily incorporated the leftover lines, resulting in unpredictable drawings (Figure 14). This open-ended pursuit of emergent values begins to approximate aspects of the design process that can be difficult to model.
With the invention of Sketchpad, Ivan Sutherland highlighted the usefulness of the digital drawing’s inherent geometrical structure. He compared this to the unstructured “dirty marks on paper” (or, in this case, acrylic) made by the analogue draftsman (Sutherland, 1975, p. 75). It is perhaps just such a lack of structure that has proven to be invaluable to the fast-paced, creative environment of design. The fact that sketching is still central to design indicates that a drawing’s true value is not always in its accuracy of representation, but rather in its ability to be reinterpreted. Drawing enables the designer to shed the weight of an idea such that he or she might shoulder a new concept; in fact, designers sometimes draw in order to see new things. Drawings are “not merely a static medium for externalizing internal visions, but rather a physical environment from which ideas are generated on the fly” (Suwa & Tversky, 2002, p. 342).

The machine is meant to provoke discussion as much as it is meant to provide useful technical insight. As computation and creative design become increasingly entwined, it will be imperative to consider how we model creativity. The act of drawing, as a method for designers to reinterpret their design concepts, encourages open-ended exploration through playful engagement with possibilities. Drawing enables the design process to be truly multimodal and provides a rich field of inquiry for autonomous, creative computational processes.

6 References

About the Author:

**Scott Penman** is a design + computation researcher interested in creativity, artificial intelligence, drawing, and the playful attitudes exhibited during design. He currently works as a research associate for the MIT Design Lab.
Section 14.

Experiential Knowledge