T. Fischer, K. De Biswas, J.J. Ham, R. Naka, W.X. Huang, *Beyond Codes and Pixels: Proceedings of the 17th International Conference on Computer-Aided Architectural Design Research in Asia,* 679–688.

DESIGN ENIGMA

A typographical metaphor for enigmatic processes, including designing

THOMAS FISCHER
Xi'an Jiaotong-Liverpool University, Suzhou, China
thomas.fischer@xjtlu.edu.cn

Abstract. Presenting a hard-to-predict typography-varying system predicated on Nazi-era cryptography, this paper illustrates conditions under which unrepeatable phenomena can arise, even from straight-forward mechanisms. Such conditions arise where systems are observed from outside of boundaries that arise through their observation, and where such systems refer to themselves in a circular fashion. This illustration aims to show the dilemma of scientific design research: Objective outsiders are mystified while those subjectively involved understand.

Keywords. System boundaries; design; objectivity; subjectivity.

1. System boundaries, input, output, re-entry and difference

No man ever steps in the same river twice,
for it's not the same river and he's not the same man.

Heraclitus: On the Universe

Computer-aided architectural designing is an endeavour in which the boundaries of systems are crossed. "System" is understood here as whatever set of elements an observer considers to act together, following a common goal. An observer may choose to regard the components that make up a computer as a system. Similarly, an observer may choose to regard the organs making up the organism of a designer as a system, or consider the designer and the computer together as a system. With these different ways of looking (Weinberg 2001, pp. 51ff.), the imaginary boundary that circumscribes what is regarded as a system changes, and what is considered a system lies in the eyes of the observer.

Sometimes there are physical boundaries containing what is regarded as a system, such as the skin of a designer and the case of a computer – but this is coincidental. Designer and computer together may be regarded as one system contained by an imaginary, but without a physical boundary. Patterns in the widest sense crossing the imaginary boundaries of systems are, depending on perceived direction, called inputs and outputs. If the human is considered as a system, then making (the interior self affecting the exterior other) and learning (the exterior other affecting the interior self) constitute instances of outputs and inputs crossing boundaries. Illustrating the constraints of surprise and of observation in the context of such boundaries, this paper describes a simple technical system (a typography-generating system) which yields unpredictable output, provided its relatively simple inner workings remain hidden.

To justify this metaphor, it is necessary to acknowledge the human mind not as a static stimulus-response system, as a static translator between inputs and outputs, but as a system whose input channels are subjected to its own output. Contrary to the technologies it currently tends to develop, the human mind is subjected to what it itself produces and is thus changed by its own performance (see Figure 1).

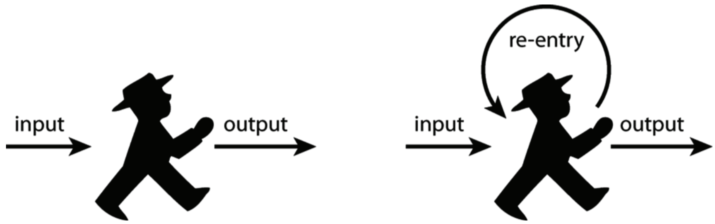

Figure 1. Human viewed as stimulus response system (left) and with the acknowledgement of human expressions re-entering as stimulations (right).

The loop which is thus formed allows expressions of the mind to re-enter into the mind where they may leave increasingly stable traces (Glanville 1997), i.e. memory. This view was substantiated in Von Foerster's (1950) interpretation of a previous study of human memory. In that previous study subjects had been asked to memorise random, meaningless syllables and to re-count as many of them as they could afterwards at regular intervals. Memory and progressive forgetting were shown to follow an exponential decay curve, which did not approach zero, but a number of syllabi greater than zero that the subjects were increasingly more likely to remember permanently. Von Foerster explains this with the human being capable of both input (listening) and output (speaking), and hence circular closure and re-entry of articulations. Thus, every recounting of a remembered syllable (output) is also a new input which reinforces what is known. Repeated recalling thus leads to an eventually stable subset of remembered syllables. Von Foerster (2003) illustrated processes of this nature using his notion of the trivial machine, which he juxtaposed to his notion of the non-trivial machine. Both these models have an input and an output. The trivial machine predictably translates inputs into corresponding outputs, so that an external observer can, after a while of observation, establish a clear causal relationship between possible inputs and resulting outputs, for example in the form of a "truth table". A complete truth table is a reliable model for predicting the trivial machine's output response to a given input, irrespectively of how long the machine has been in operation. The non-trivial machine, in contrast, contains another input-output machine, with a feedback loop of its own generating internal states (labelled z in Figure 2) that affect the machine's behaviour and results in a vast number of possible outputs that cannot be reliably predicted by an external observer (Glanville 2003, p. 99). The non-trivial machine's operation, with changing internal states, can be said to leave traces in the machine through which the machine in effect turns into a different machine with every operation. An outside observer cannot easily establish a reliable truth table by which outputs resulting from given inputs can be predicted.

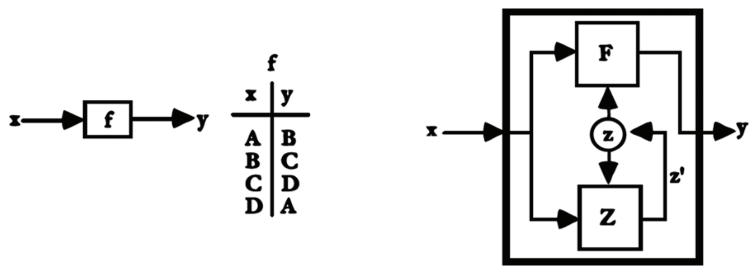

Figure 2. Trivial machine with truth table (left) and non-trivial machine (right), reproduced from Von Foerster (2003), 310–311.

The distinction between the trivial machine and the non-trivial machine deserves attention in general and in particular in our field because in every encounter the choice between both metaphors determines much of the ethical stance one takes – towards others, tools, buildings etc. Virtually all of our science and technology corresponds to the principle of the trivial machine in the sense that a given input is expected to always reliably lead to the same output. Multiple computers are, put simply, expected to always have the same response in the face of the same task or problem. Stereotypical engineers, managers and representatives of other professions, be they allied with architecture or not, are likewise expected to arrive at the same results when presented with the same input. In the education of these professions this aspiration to the ideal of the trivial machine is enforced with the principle of scientific repeatability. Multiple stereotypical engineers tasked with the same structural analysis problem, or the same engineer tasked with the same structural analysis problem twice should reliably arrive at the same results, i.e. fulfil expectations predictably and reliably. Briefing multiple architectural designers with the same project brief, in contrast, makes sense only if variety amongst multiple responses is desired. Briefing the same architectural designer with the same task twice will also lead to two different outcomes because the second time around, one would be facing a different architect – one who was subjected to her/his own first design process and outcome, which left traces in her/him and thus changed her/him. In this sense, and in the sense of the above quote of Heraclitus, one cannot design (or learn, for that matter) the same thing twice.

2. Variety and automated production

Surprising variety (in the cybernetic sense: number of choices available) and reliable predictability are, paradoxically both for better and for worse, essential human needs and human characteristics (Fischer 2010, p. 611). We experience this paradox in numerous contexts in which we enjoy both stimulating variety in expression as well as economic and organisational benefits of uniformity. In shaping our products and environments, the advantages offered by predictably uniform (hence interchangeable) prefabricated components famously gave rise to assembly-line based production since the early days of industrial production; and it is part and parcel of architectural construction today. Having been introduced to architecture with uniform building elements, prefabrication brought along with it sameness at the scales of component repetition. At small scales of component repetition, such as that of clay bricks (Fischer 2007, p. 45), interchangeability may be appreciated for allowing flexibility and subtle texture. At larger scales such as that of floor plans or whole buildings as found in *Platenbau* developments (Hopf and Meier 2011), repetitive sameness is criticised for being monotonously boring or even socially detrimental. Aiming at repeatable input-output relationships, early computer applications in our field focused on predictable input-output relationships, leading to criticisms of applying the computer as a "fancy drawing board" (Dantas 2010, p. 161) and of valuing it as an equivalent to "an army of clerks" (Alexander 1965). In architecture and in other industries, there are now tendencies acting against monotonous sameness. Referred to as customisation approaches (Gilmore and Pine 1997), these tendencies are increasingly aided by computational (generative, parametric etc.) techniques that allow increasing of variety via circular feedback. The development of typography follows a similar pattern. Moveable type introduced economic benefits along with monotonous sameness to book printing. Using type wheels and the like, typewriters, teletypes and computer printers achieved similar predictable sameness and cost-efficiency also in documents produced in small numbers. Contextual variations such as ligatures have been introduced to mechanical typesetting. Some contemporary computer typefaces go further and achieve "organic", "random" or "handwritten" appearances by introducing randomness to curve paths or by providing sets of alternative glyphs for the same characters.

3. Enigma cipher machine

While most technical devices – in particular digital ones (Fischer 2011) – are designed to function in the fashion of the trivial machine, there are a few that were designed to approximate the characteristics of the non-trivial machine. One of them is the Enigma cipher machine used by Nazi Germany before and during World War II. The machine is used to encrypt and to decrypt text messages by substituting letters with a replacement mechanism that changes systematically as the machine is used. It looks somewhat akin to a typewriter, with a qwertz keyboard (label 1 in Figure 3), but, located just above the keyboard, is a grid of alphabetically labelled lamps, also arranged in a qwertz layout (label 12 in Figure 3). Keys close electrical circuits that are routed into a sequence of 3 to 5 (depending on the model) exchangeable and rotatable cylinders (labelled 6, 7, 8, 9 in Figure 3), each with an "arbitrary" wiring between usually 26 inputs and 26 outputs (one per each key and lamp; see labels Fig. 1 and Fig. 2 in Figure3), then returned back through these cylinders and to the set of lamps.

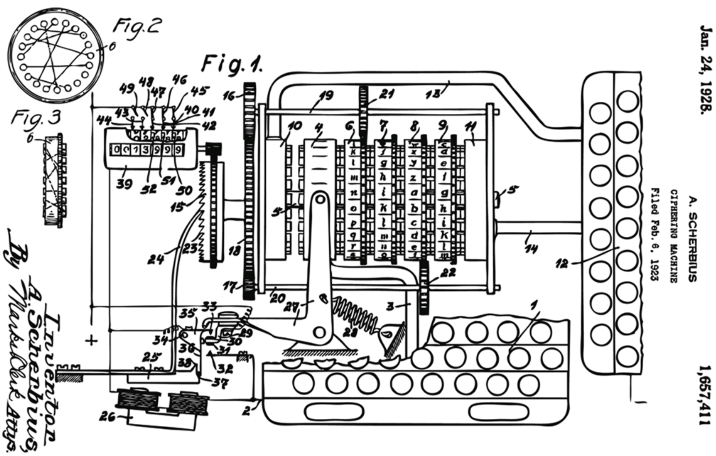

Figure 3. Schematic diagram of Enigma machine (from Scheribus 1928).

Pressing a key will activate one of the lamps, apparently at random, according to selection of cylinders and their current orientation. Additionally, each keystroke results in the rotation of the first cylinder by one of 26 rotation positions, after 26 keystrokes, the second cylinder will also rotate by one position and so forth, somewhat in the fashion of the digit cylinders in a mechanical odometer. Thus, each keystroke results in a new wiring between keyboard and lamps coming into effect for the subsequent keystroke. In other words: Use of the machine leaves a trace in it, changing the wiring of the machine, and hence the cipher, progressively. (Due to the symmetrical setup of the wiring going into the cylinders and back out through the same cylinders, the same machine setup can be used both to cipher and to decipher. The identical setup is achieved by referring to a secret timetable based code book of which both ends must hold a copy.) To an outside observer the input-to-output mapping of the Enigma machine is extremely difficult to determine, while it is perfectly determinable to those who developed it and who have a good understanding of its setup and inner workings. With inner workings of this kind the Enigma machine shares key characteristics of designing, making it a useful metaphor for the purpose of showing how designing is a relatively straight-forward process when viewed from the inside perspective but mysterious and wonderful when viewed from the outside perspective.

4. A typographical metaphor

The illustration presented here is a piece of software predicated on the Enigma machine and implemented as a VBA script controlling Rhino3D. Glyph renderings of characters input via keyboard are distorted dynamically and individually, with the use of a "private key" string stored inside the system. Somewhat akin to the (de)ciphering process of the Enigma machine, each key that is typed, modelled, transformed and rendered changes the internal state of the system (leaves a trace in it) to change the way the following glyph is distorted. In contrast to common computer typefaces, glyphs of same characters are unpredictably variant. As a point of departure, the system uses the typeface Helvetica to derive initial glyph outline curves for each typed character. The system then applies a combination of six Thompson (1992, pp. 1053-1090) transformations (see Figure 4) to these outline curves. Thompson transformations (parametric "warping" based on quadratic functions, Wilkinson 2005, pp. 223-224) as illustrated in Figure 3 above allow positive and negative transformations based on parameters P1 through P10. Of these, the presented generative system uses the six parameters P1, P2, P3, P6, P7 and P8. Parameters P4, P9 and P10 are ignored while P5 can be toggled manually, providing an "italics" option. Parametric input for the six Thompson transformations per-formed by the system is derived from the ASCII bit patterns of characters of a "private key" string, which is "rotated" by three characters with each input key stroke. Any ASCII string of any length can be used for this purpose.

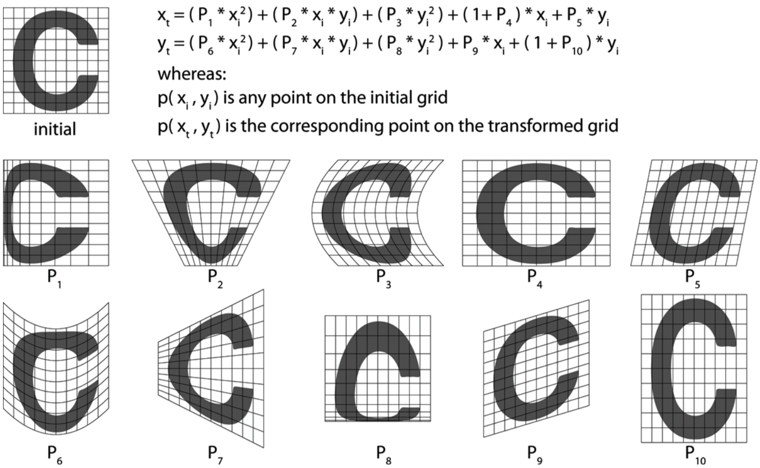

Figure 4. Quadratic functions and Thompson transformations based on parametric variation.

Once a key is pressed, the first three characters of the "private key" string are converted to their respective ASCII bit patterns and the six last bits of each are used to produce two factors out of the set {-4, -3, -2, -1, 1, 2, 3, 4}, which are multiplied with a scaling factor (0.075 gives a good effect) to produce a total of six parameters (see Figure 5).

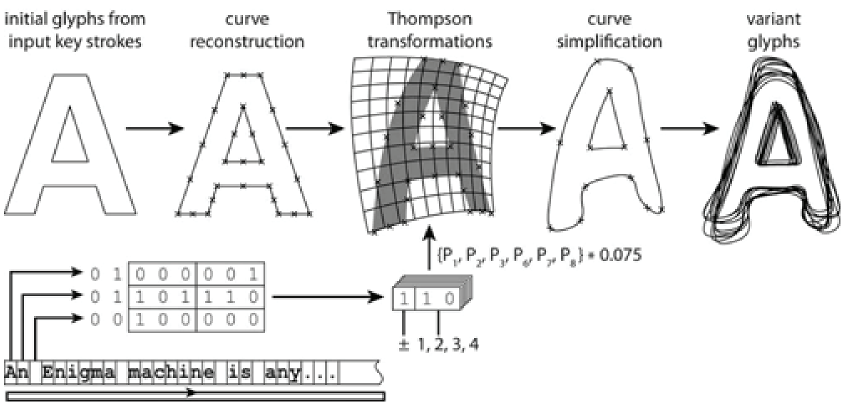

Figure 5. Key operations of glyph-variation based on "private key" string.

Following parametric Thompson transformations, the resulting glyph outline curves are simplified by reducing their numbers of control points, resulting in casual looking, “blobby” glyphs. These are variant with identical characters being shown as different glyphs (see the bottom line of Figure 6). Overall, the resulting types are nevertheless largely consistent and recognisable as members of the same typographical style, which I call Polymorph. The generative system includes a rudimentary automatic kerning function, the performance of which is also visible in the bottom line of Figure 6. The top line of Figure 6 shows the typeface Helvetica. The middle line shows the hand-drawn, Helvetica-inspired typeface YWFT HLLVTKA Round, each character of which looks irregular, while identical characters are rendered with the same glyphs.

Figure 6. Helvetica (top), YWFT HLLVTKA Round (middle)and Polymorph (bottom).

Just as the Enigma machine’s cipher output is enigmatic and unpredictable to outsiders such as wartime enemies, the glyph transformations generated by the described system are unlikely to be predictable to outsiders of the system. Nevertheless, both systems are perfectly determinable and appear straight-forward to those aware of both systems’ setups and of the ways in which the performances of both systems leave traces within them, changing their internal states, thus in effect leading to new inner workings with each operation. Designers, articulating and (re-)considering ideas can be seen as embodying a similar, non-trivial re-entry structure which, similarly, can appear either surprising and unpredictable or straight-forward and traceable depending on an observer’s inside or outside perspective.

5. Observations

Some processes are linear, predictable and seem causal while others are circular and unpredictable. The difference can be shown with the distinction between the trivial machine and the non-trivial machine. Like the Enigma machine, instances of designing can be viewed as circular systems (Fischer 2010, Gänshirt 2011, p. 79) which display the structure and quality of the non-trivial machine. This principle can be implemented technically and used to generate design outcomes (Glanville 1992). Numerous examples have been presented in our field and the typography-transforming system presented here is another one. The wonder and surprise offered by such systems exists for as long as their interior workings evade observation (Fischer 2008). Design is, on the inside, concerned with what is unpredictable to outsiders. It hence corresponds to the non-trivial machine. Science, always on the outside, is concerned with prediction and hence corresponds to the trivial machine. This poses a challenge to scientific researchers aiming to research into designing objectively, somewhat comparable to the challenge of cryptography that leaves outsiders mystified while insiders understand. Design processes can be appreciated and understood on the subjective inside. Objective scientific description, though, is required to approach design from the outside.

It is difficult to view both from the inside and from the outside of the boundaries of a system one observes at the same time. Hence it is difficult to be mystified by what one understands (and vice versa), and to appreciate what one is not a part of (and vice versa). While this state of affairs has been pointed out for decades (Glanville 1982) it is not yet addressed sufficiently in our field. Postgraduate students whose work is scrutinised closely under both designerly and scientific criteria are particularly affected.

Glanville's (2003, pp. 98-99) view of the non-trivial machine goes beyond that of Von Foerster (2003, pp. 310-311), who originated it. Von Foerster on the one hand describes it as two trivial machines, one nested inside and changing the internal state of the other, resulting in unpredictable selections from a finite number of choices available (as the Enigma machine does). Glanville on the other hand does not explain the non-trivial machine's inner workings, for doing so would turn it into a trivial machine. Instead, Glanville grants the non-trivial machine the capability of increasing the number of choices available. This can be done, in principle, by differentiating fewer available choices into more available choices, or by charting uncharted territory to make new choices available there. Both views of the non-trivial machine describe processes that offer us wonder and surprise. Processes of both kinds suffer in the presence of scientific description: When hidden mechanisms making apparently random selections from given sets are revealed, their mystery is destroyed (an aesthetic loss). But worse: The capability of making more choices from fewer choices, where it still exists, is denied (an ethical loss).

Acknowledgements

I gratefully acknowledge the feedback and support I received from Nancy Diniz, Theodorus Dounas, Ranulph Glanville and Timothy Jachna.

References

Alexander, C.: 1965, The question of computers in design, *Landscape*, **14**(3), 6–8.
Dantas, J. R.: 2010, The end of Euclidean geometry or its alternate uses in computer design, *Proc. SIGraDi*. Bogotá, Colombia, 161–164.
Fischer, T.: 2007, Rationalising bubble trusses for batch production, *Automation in Construction*, **16**, 45–53.
Fischer, T.: 2008, Obstructed magic, *in* Nakpan, W. et al. (eds.), *Proc. CAADRIA2008*, Pimniyom Press, Chiang Mai, 609–618.
Fischer, T.: 2010, The interdependence of linear and circular causality in CAAD research: a unified model, *in* Dave, B. et al. (eds.), *Proc. CAADRIA2010*, CUHK, 609–618.
Fischer, T.: 2011, When is analog? When is digital? *Kybernetes*, **40**(7,8), 1004–1014.
Gage, S.: 2005, The wonder of trivial machines, *Architectural Design*, **78**(4), 12–21.
Gänshirt, C.: 2011, *Werkzeuge für Ideen*, Birkhäuser, Basel.
Gilmore, H. G. and Pine, B. J., II: 1997, The four faces of mass customization, *Harvard Business Review*, Jan–Feb, 91–101.
Glanville, R.: 1982, Inside every white box there are two black boxes trying to get out, *Behavioural Science*, **27**(1), 1–11.
Glanville, R.: 1992, CAD abusing computing, *in* Martens, B. et al. (eds.), *CAAD Instruction: The New Teaching of an Architect?* eCAADe Proceedings, Barcelona, 213–224.
Glanville, R.: 1997, A ship without a rudder, *in* Glanville, R. and de Zeeuw, G. (eds.), *Problems of Excavating Cybernetics and Systems*, BKS+, Southsea.
Glanville, R.: 2003, Machines of wonder, *Cybernetics & Human Knowing*, **10**(3,4), 91–105.
Hopf, S. and Meier, N.: 2011, *Plattenbau privat. 60 Interieurs*, Nicolai, Belin.
Scheribus. A.: 1928, *Ciphering Machine*, US Patent No. 1.657.411.
Thompson, D. W.: 1992, *On Growth and Form*, Dover Publications, New York.
Wilkinson, L.: 2005, *The Grammar of Graphics*, 2nd ed., Springer, New York.
Von Foerster, H.: 1950, Quantum mechanical theory of memory, *in* von Foerster H. (ed.), *Cybernetics: Transaction of the Sixth Conference*, Josiah Macy Jr. Foundation, NY.
Von Foerster, H.: 2003, *Understanding Understanding*, Springer, New York.
Weinberg, G. M.: 2001, *An Introduction to General Systems Theory*, Dorset House, NY.