Representation and Mimesis in Generative Art: Creating Fifty Sisters

Jon McCormack
Jon.McCormack@monash.edu
Centre for Electronic Media Art, Monash Univeristy, Caulfield East, Australia

Keywords: Generative Art, Representation, Mimesis, Artificial Life.

Abstract: *Fifty Sisters* is a generative artwork commissioned for the Ars Electronica Museum in Linz. The work consists of fifty 1m × 1m images of computer-synthesized plant-forms, algorithmically 'grown' from computer code using artificial evolution and generative grammars. Each plant-like form is derived from the primitive graphic elements of oil company logos. The title of the work refers to the original 'Seven Sisters' — a cartel of seven oil companies that dominated the global petrochemical industry and Middle East oil production from the mid–1940s until the oil crisis of the 1970s.
In this paper I discuss the issue of representation in generative art and how dialogues in mimesis inform the production of a generative artwork, using *Fifty Sisters* as an example. I also provide information on how these concepts translate into the technical and how issues of representation necessarily pervade all computer-based generative art.

1. Introduction

In a recent paper, the author and several colleagues proposed what we considered to be the ten most important questions for generative art (McCormack et al. 2012). The fifth question on our list asked the following in relation to computational generative art:

> In what sense is generative art representational, and what is it representing?

In this paper, I will expand on this question and its implications. From the onset I should make clear that my topic relates to *computational* generative art. While generative art has many non-computational modes, they are not specifically addressed in this paper. Rather than discussing theoretical ideas, I will describe a recently completed generative art work, *Fifty Sisters*, and look at how representational issues come into play in almost every aspect of creating the work: conceptualization, implementation, and realization.

Representation and mimesis are some of the oldest issues in art, dating back at least to the ancient Greek philosophers (Scruton 2009). The idea of replicating naturalistic effects in painting came to the fore in renaissance aesthetics, where painters were concerned with a *truthful* representation of what they saw. Roughly corresponding with the mathematical formalization of perspective projections and with progressive advances in paint technologies (Ball 2002), artists' skills developed in portraying the 'real' in art. However, any art acting as a 'mirror of nature'—as was famously advocated by Leonardo—still requires 'interpretation and ordering' from the artist. Image reproduction technologies forever changed the idea of capturing the real in art, inviting the possibility for artists to focus on other kinds of 'truths'. By the time Hal Foster published *Return of the Real* (Foster 1996) mimesis had come full circle[1]. In more recent times, representation and semiotics arguably have been overtaken by other concerns, such as art as social exchange and dialogues concerning relational aesthetics. Representation and mimesis are old and well established discourses in art. Generative art reopens these dialogues in ways that other art forms cannot, because generative art brings something new to art: the idea of representing process.

In generative art, as with other forms of art, we should expect a range of representational styles, e.g. visual art ranges from abstract, non-objective mark making and vast sways of negative space, to highly figurative and photorealistic imagery. But unlike visual art, generative art has not been so extensively analyzed in terms of how it deals with mimesis and representation. Here we seek to begin to address that deficiency.

1. In fact a circle and a half.

1.1. Computers and Representation

It is almost impossible to write a computer program without—at least implicitly—considering representation. Digital computers use collections of bits (electrical signals or states standing for 1s and 0s, ON and OFF, etc.) to encode and represent data and instructions. At the level of software, programs generally represent things using atomic variables (integers, floating-point numbers, Booleans, characters, etc.) or compound collections of these variables (data structures, arrays, strings, objects), which may easily include other compound collections.

It is important to distinguish between a variable and its interpretation, i.e. it's *semantics*. I can give a variable any semantic interpretation I choose: it could represent happiness, my bank balance, or the text of this paper for example. The important point is that the programmer, not the computer, confers this meaning. The computer hardware does impose practical limitations on the kinds of interpretations that are possible. If I represent the concept of happiness as an integer, then the machine will manipulate it as an integer, not as an emotional state of being. I can interpret a variable called 'happiness' as the degree of happiness of an individual, but this representation is limited to the 32 or 64 bits typically used to represent integers.

At the base level, computers are symbol-processing machines — they transform patterns of bits that represent symbols. All symbols are subject to *interpretation*, either by the programmer, the user of the program, or the machine itself. As an added complication, symbols are often not interpreted directly (as bit patterns), but are transformed by some process.

As an example: two variables may represent the Cartesian coordinates of the center of a circle that is displayed graphically on a computer screen. As the variables are changed the circle 'moves on the screen'. An additional variable represents the circle's radius. We speak of a 'circle moving on the screen', but, like the world of *the Matrix*, there is no circle and it is not moving. Discrete patterns of bits are changing at regular intervals. We interpret the complex process of changing individual pixels seamlessly as a moving circle.

Perhaps this all seems obvious and even slightly trivial. Yet it is common for people working with computers to forget about these representational gaps. In observing a computer simulation of ant behavior, we might speak of 'ants foraging for food', but this interpretation (as it would be for a painting) is semantically loaded. They are only "ants" in as much as they homomorphically model an ant. 'Foraging' is a convenient anthropomorphic label we give to a series of discrete changes read as position, movement, behavior and so on. We use this shorthand because it is both convenient and necessary: speaking only in terms of bit patterns is not practical or enlightening (Dennet 1991), despite this being the basis of digital computer representations.

As Nietzsche reminds us, writing on a typewriter is different than writing with a pen (Kittler 1990). The tool affects our way of thinking. With the computer it is even more profound, because to translate ideas into code we must 'think algorithmically', which in turn influences how we think about the world and act in it.

In making a generative computer artwork, representation exists at many levels, not just the bit-pattern level of variables, data structures or screen graphics. Rather than expand on this in abstraction, let us look at a concrete example in generative art to see how these issues come into play.

2. Fifty Sisters

Fifty Sisters is a generative artwork commissioned for the Ars Electronica museum in Linz, Austria[2]. The work consists of fifty 1m x 1m digital images of computer-synthesized plant-forms, arranged in a 5 x 10 grid in the museum foyer. Each image is algorithmically 'grown' from computer code using artificial evolution and generative developmental grammars. The form is derived from the primitive graphic elements of oil company logos.

2. www.aec.at

The title of the work refers to the original 'Seven Sisters'—a cartel of seven oil companies that dominated the global petrochemical industry and Middle East oil production from the mid–1940s until the oil crisis of the 1970s. Fossil fuels began as plants that over millions of years were transformed by geological processes into the coal and oil that currently powers modern civilization. The images remind the viewer that the basis of an oil company's financial success is derived from plants and natural processes that operated over vast geological timescales. With 'peek oil' expected to be reached this century (if not already), we are expending this non-renewable resource in the relative blink of an eye. Two example images from *Fifty Sisters* are shown in Figure 1. More information on the work and its motivations can be found at http://jonmccormack.info/~jonmc/sa/artworks/fifty-sisters/.

The process to create each form involved a number of steps. Firstly, an oil company logo was chosen and 2D vector art created[3]. From the 2D vector art, the basic graphic elements were separated manually and then converted to 3D geometric primitives. To create each plant form custom software was developed by the artist. Technical details can be found in (McCormack 2005). In basic terms, the software simulates the growth and development of the form from a series of developmental rules, metaphorically similar to the way DNA 'encodes' the developmental plans of biological organisms.

3. Wikipedia has vector versions of many oil company logos.

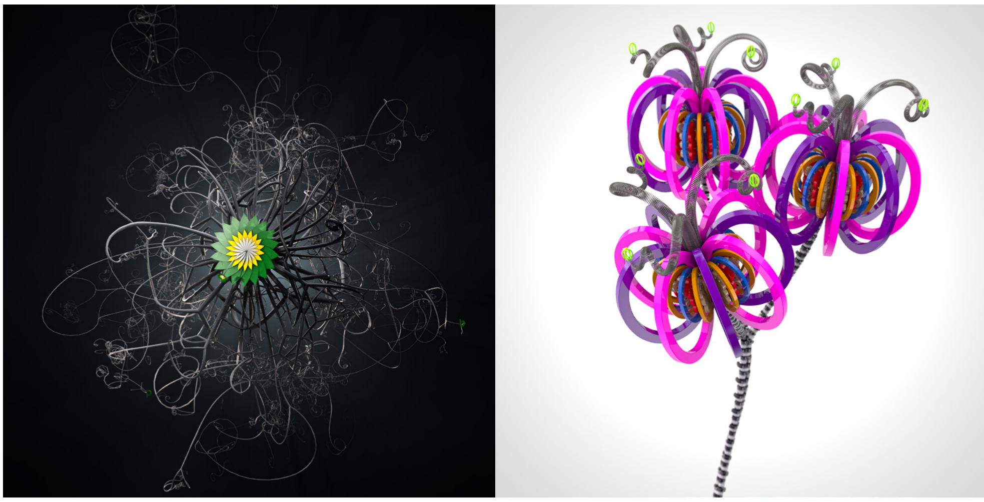


Fig. 1. Synthesized plant forms based on the BP logo (left) and ESSO logo (right).

Rules consist of any number of *developmental symbols* that represent individual or collective elements of the growing form. Symbols include continuous data, such as size, age, chemical concentrations, etc., that change over the lifetime of the developmental simulation. If certain conditions are met (e.g. size becomes greater than some fixed value), the symbol may subdivide, be replaced by another symbol, or die. This method is somewhat analogous to cell division in biology, but with far greater abstraction and simplification.

As the developmental rules are a machine-representable code, they can be subject to genetic manipulation, including mixing of rules from other forms (a kind of 'gene splicing') or guided evolution using a variant of the *Interactive Genetic Algorithm* or *IGA* (McCormack 2004). The terminal symbols of any rule can be interpreted as instructions that encode the geometric construction of form. These symbols include instantiations of the geometric elements of the oil company logos. Thus when the form is constructed, its geometry includes geometric elements of the original logo. The final form depends on how the rules have evolved and mutated. The results are often surprising; in some cases the original logoform is clearly visible, in others it is almost impossible to recognize as it has become highly abstracted. Figure 2 shows an example form (using elements from the Shell logo) and the rules, or 'digital DNA' used to generate it.

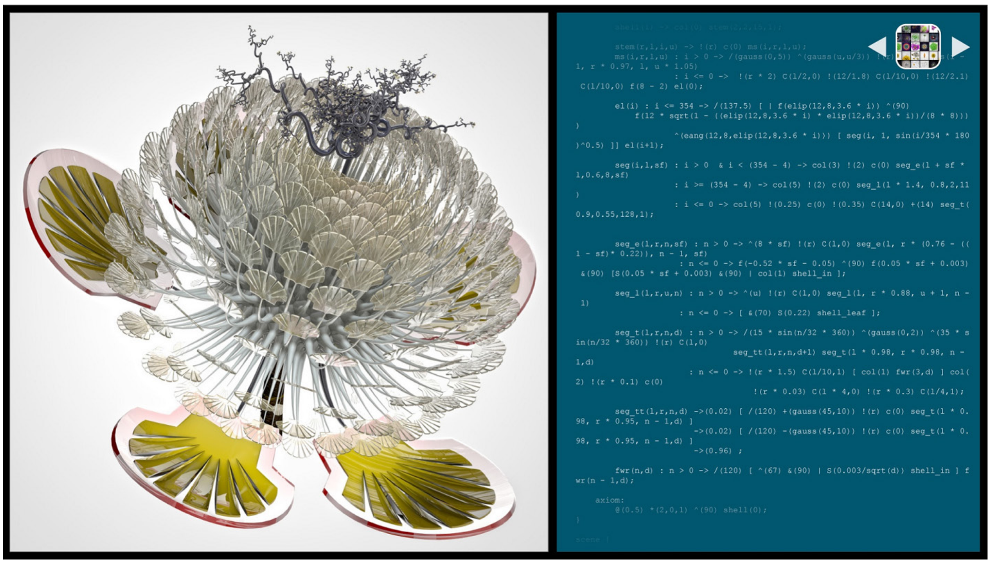


Fig. 2. Synthesized plant forms (left) and the developmental rules from which it was generated (right).

The forms generated by this developmental/genetic process are output as 3D geometric models. Most plant forms are easily expressed in only a page or two of information (a few hundred bytes), yet they generate geometric models many orders of magnitude greater (~10^7 – 10^9 bytes). The models are read into a 3D renderer, which renders an image using "photorealistic" rendering techniques.

2.1. Representation in Fifty Sisters

Fifty Sisters is a useful example of representation in generative art, because it deals with representation and mimesis at multiple levels. As images, each plant form has several representations: that of a real plant, a computer graphic, and a corporate logotype. The generative code ('digital DNA') from which each image is generated (Figure 2) is also exhibited in a separate touch screen application that forms part of the exhibition of the work. This allows the viewer to see a different representation of the form: as code that through a process mimetic to biology generates that form.

Beyond the visual and textural representations, there is an additional layer of representation to contend with, that of the generative process. What is represented in this process? The process represents another process: biological development and evolution. The artist-developed computer program simulates and abstracts the process of biological development and evolution. It is the *personal expression* of a biological process in software. So in this sense the software program, when run, represents these natural processes in a somewhat similar way to that in which a landscape painting represents a landscape.

The difference, of course, is that the viewer of the work cannot see or otherwise experience this process directly.

This idea of one dynamic process representing another is new to art, and is what best distinguishes this kind of generative art from other practices. Certainly process was often of interest in modern art. One only has to think of Sol Le Witt, Cornelius Cardew or Jackson Pollack for example. But in these cases the process of generating the art was not representing another process: Le Witt's drawing instructions were not representing anything other than instructions to draw. A computer process being mimetic to another process is different, because it involves choices about sign, signifier and what is signified. Moreover the complexity and unpredictability of a computer process (vis. Emergence (McCormack & Dorin 2001)) introduces additional properties not directly represented in the generative process itself.

2.2. Mimesis in Fifty Sisters

As alluded to in the introduction, mirroring nature involves interpretation and ordering by the artist. As simulacra or simulation, a computer process is not the same as what it seeks to mirror. This is well known in the simulation sciences, were formal methods are used to verify and validate simulations to models, and models to reality. The experimenter selects those aspects to model and those to ignore. Naturally, aspects or mechanisms that the experimenter is unaware of cannot be in the model, although through experimentation she or he may become aware of them, and then subsequently incorporate them into the model. The aspects of a phenomena or system that are modeled are subject to varying degrees of abstraction necessary for them to be practically simulated.

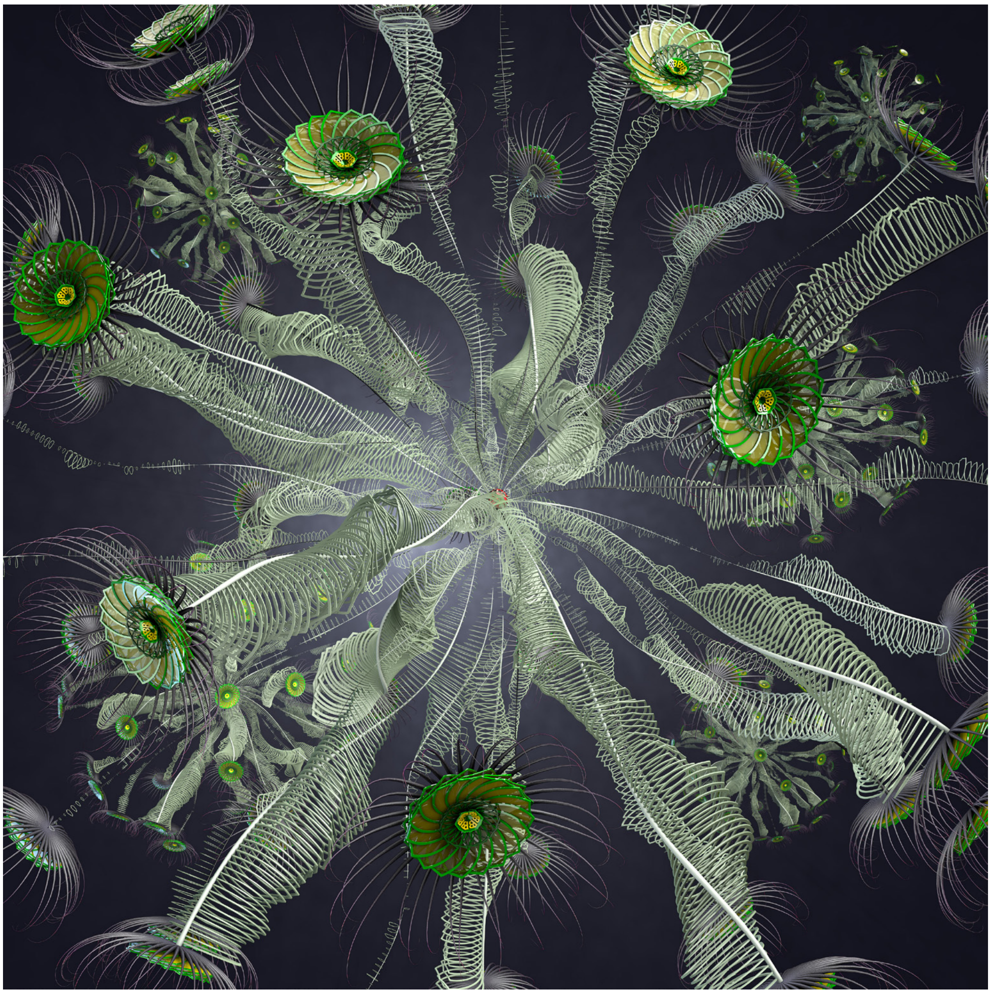
Art allows a difference license, where the most interesting works can abstract from the world of the imagination rather than the world of the real. In *Fifty Sisters* mimesis plays many roles. The plants themselves are in some way mimetic to real plants, yet no such forms could ever exist in reality. This is not really surprising; such issues have been endlessly explored in painting.

Things become more interesting in relation to process however. The generative process is mimetic to real biological development and evolution. The work speaks of 'digital DNA', 'evolution' and 'development' as signifiers to the interpretation of their biological parallels. While this simulated biology is grossly abstracted and simplified, it still exhibits some of the features of its real-world counterpart. Moreover, its conceptualization and intent as artistic concepts originates from interpretations of biological development and evolutionary process.

This analysis reveals some curious aspects about the work. For example the choice of using standard 3D rendering techniques, which focus on a Cartesian, photographic-like visual realism, whereas the biological processes focus on a somewhat different kind of 'realism'. This is partly explained by the technical constraints in developing works like this, but more importantly the aesthetic language of modern corporate communication is similarly derived from these techniques. Corporate logos are visually presented using the purity of glittering computer graphics, with its clean and sleek mathematical veneer. *Fifty Sisters* deliberately borrows from this vernacular, presenting developmentally mangled corporate logo forms using their native visual language.

3. Conclusion

In order to understand a generative artwork we must examine the process alongside what that process produces. It is important to look at what the process is representing, and how it performs this representation. For generative computer art, there always exist multiple levels of representation and it is easy to forget about how these representational structures are formed when they are so easily taken for granted. One of art's roles can be to reveal what is normally hidden or taken for granted, bringing it into awareness (or even sub-consciousness). Computer representations and processes are typically hidden from direct perception, so by bringing them into perception we reveal their most unique and interesting aspects as symbol processing machines.



```
# 1 “bpOldZoom.dna”
# 1 “<built-in>”
# 1 “<command-line>”
```

```
object spine {
    << noises >> efn;
    << math >> efm;
    < efn.gauss > gauss;
    < efm.acos > arccos;
    < efm.sin > sin;

    surface BP_old_hood;
    surface BP_old_back;
    surface BP_old_BP;
    surface sphere10;

    equiv ‘ col;

 rules:
     el(i) : i <= 47 -> /(137.5) [ ^(arccos(1 - i * 0.04000) )
                             [ f(90) elem(i) ]] el(i+1);

    elem(i) -> sph seg(100,40,5,1,10,0.1,0.1);

    sph -> !(15) c(0) !(7) C(1,0) !(3) C(1,0) !(1) C(1,0);
    seg(n,l,t,u,r,sc,v) : n > 0 -> /(t) ^(u) !(r * sc) C(l,0) [col(1)
S(sc) BP_old_hood ] seg(n - 1, l + gauss(0,v), t + gauss(0,v), u
+ gauss(0,v), r, sc + gauss(0,v), v)
                      : n <= 0 -> /(t) ^(u) !(r * sc) C(l, 1) [ col(9)
tusks(36)
 ] f(14) S(1.0) ll(18, 15, 10);

    tusks(n) : n > 0 -> [ ^(90) !(14) c(0) C(100, 0)
tusk_s(20,gauss(90,5),14,5)
 ] /(10) tusks(n - 1);

    tusk_s(n,l,r,u) : n > 0 -> !(r) /(gauss(0,5)) &(u) C(l,0) tusk_s(n
- 1, l *
0.90, r * 0.78, u * gauss(1.08, 0.067))
                    : n <= 0 -> !(r * 0.1) ^(u) C(l,0);

    ll(i,u,t) : i > 0 -> /(20) [ &(u) +(t) col(2) BP_old_hood col(1)
BP_old_back
 ] ll(i - 1, u, t)
          : i <= 0 -> [ f(5.0) col(1) lm(18,u+10,t - 4.5) ];
    lm(i,u,t) : i > 0 -> /(20) [ &(u) +(t) S(0.5) col(3) BP_old_hood ]
lm(i - 1,
 u, t)
          : i <= 0 -> [ f(5.0) col(1) ls(6,u + 10,t - 10) ];
```

```
    ls(i,u,t) : i > 0 -> /(20 * 3) [ &(u) +(t) S(0.25) col(3) BP_old_
hood col(1)
 BP_old_back col(1) BP_old_BP ] ls(i - 1, u, t);

 axiom:
    @(0.2) *(2,0,1) col(0) [col(8) S(9) sphere10] el(0);
}

scene {
    spine(time * 50);
}
```

Fig. 3. Synthesized plant form based on the BP logo (top) and the developmental rules from which it was generated (bottom).

Acknowledgements: I am grateful for discussion with Gordon Monro who first raised the question of representation in generative art with me. This research was supported by an Australian Research Council Discovery Grant, DP1094064.

References

Ball, P. *Bright earth: art and the invention of color.* New York, Farrar Straus and Giroux. 2002.

Dennett, D. C. Real Patterns. *Journal of Philosophy* 88: 27–51. 1991.

Foster, H. *The return of the real: the avant-garde at the end of the century.* MIT Press, Cambridge, Mass.1996.

Kittler, F. Discourse Networks 1800/1900, with a Foreword by David E. Wellbery. Stanford. 1990.

McCormack, J. Aesthetic Evolution of L-systems Revisited. *Applications of Evolutionary Computing (EvoWorkshops 2004).* G. R. Raidl, S. Cagnnoni, J. Brankeet al. Berlin, Heidelberg, Springer-Verlag. LNCS 3005: 477–488. 2004.

McCormack, J. A Developmental Model for Generative Media. *Advances in Artificial Life (8th European Conference, ECAL 2005).* M. Capcarrere, A. A. Freitas, P. J. Bentley, C. G. Johnson and J. Timmis (eds). Berlin; Heidelberg, Springer-Verlag. LNAI 3630: 88–97. 2005.

McCormack, J., O. Bown, A. Dorin, J. McCabe, G. Monro and M. Whitelaw. Ten Questions Concerning Generative Computer Art. *Leonardo* (to appear, accepted July 2012).[preprint available at: http://www.csse.monash.edu.au/~jonmc/research/Papers/TenQuestionsLJ-Preprint.pdf] 2012.

McCormack, J. and A. Dorin. *Art, Emergence and the Computational Sublime.* Second Iteration: a conference on generative systems in the electronic arts, Melbourne, Australia, CEMA. 2001.

Scruton, R. *Beauty.* Oxford University Press, Oxford; New York. 2009.