

SYSTEM STORIES AND MODEL WORLDS:  
A CRITICAL APPROACH  
TO GENERATIVE ART  
MITCHELL WHITELAW

---

*In his article «System Stories and Model Worlds: A Critical Approach To Generative Art» Mitchell Whitelaw (CANBERRA, AUS) proposes to bridge what has been detected by various authors, the unproductive gap between «software formalism» and «software culturalism». While formalism tends to be visually abstract, and thus corresponds to the field of generative art, the culturalist approach, on the other side, suggests that software art is predominantly critical/political, focusing on and deconstructing software as cultural text. Refusing what he calls binary thinking, Mitchell Whitelaw instead proposes to overcome this split by calling for a «critical generativity». Such an approach would deconstruct the system stories contained in the formal objects used by generative art and thus would critically analyze their implications. To put it shortly, it would allow to «read generative software art according to the critical paradigm of the software culturalists» (Whitelaw).*

*But how exactly are these system stories to be deconstructed? Whitelaw hopes to find examples for «critical generativity» by analyzing generative artworks by Reas, Tarbell, Ngan, Capozzo, Masuda, Annunziato, Driessen and Verstappen (all of them, by the way, male artists). By asking what kind of narrative these projects convey, Whitelaw formulates poignant comments on generative art's, hm, let's call it basic level of imagination (paragraphs 2 and 3): «a*

*clone in a crowd, unchanging, with no traction on the space it inhabits, existing in an ongoing, perpetual present.» And he continues, criticizing the image of contemporary society that's being provided as naïve and utopian: «a mass of identical (or typed) individuals, each contributing equally to the collective dynamic, each equally connected with and affecting all the others.»*

*That's not what interests Mitchell Whitelaw. Instead, he is looking for «critical generativity»: Systems that sketch «possible worlds», imaginations of the systems we live in, revolutions cast in software so to speak. As generative art's basic material are systems themselves, Whitelaw predicts a «unique potential» for generative art: «unlike other forms of discourse, it can actually experiment with the emergent outcomes of particular ontologies, modes of being and relation.» Rather than reproducing known features and merely feeding these known features into «eye-candy machines» (as most generative art projects do, according to the author), he calls for prospective or utopian potential of generative ontologies that «might equally be ironic, critical, deconstructive or fantastic». Golan Levin's *Axis* applet is cited as an example: «Generative art can, and must, do more than make images of complex systems; it can tinker critically with the systems themselves, then set them running: possible worlds.»*

*Whitelaw's suggestion to read the implicit system stories and to decode the narratives and ontologies inherent in the systems employed in generative art is extremely interesting. It shows that the performativity of the program code is embedded in a system story, and that this system story or ontology is a text that is at the same time narrative, performative and prescriptive. However, Whitelaw's approach doesn't seem to be radical enough. Isn't the boringness of generative art projects all the more revealing in terms of uncovering system narratives contained within today's economic or a-life models than generative art projects that produce «possible worlds» as alternatives to the existing one and its narratives? In bringing*

*forth «possible worlds», wouldn't «critically generative» art projects rather conceal the system stories already at work in our contemporary world than uncover these narratives? And isn't Mitchell Whitelaw's counting on «critical generativity» (i.e. generative art producing alternatives to existing system stories) falling into the same trap of expecting generative art to produce the «unknown» or unexpected? Wouldn't this unexpected system story have to remain per se system immanent — precisely because the solutions it offers are software based?*

*Mitchell Whitelaw's postulation of «critical generativity» yet waits to be met by corresponding generative art projects. In the discussion following Mitchell's remote presentation it was suggested that one of the first projects that could be called «critically generative» possibly is Renate Wieser and Julian Rohrer's project «Invisible Hand Machine» realised for Readme100 (2005).*

INKE ARNS

## 1. Introduction

Writing in 2002, Florian Cramer draws a fundamental distinction in software art practice, between «software formalism» and «software culturalism.»<sup>1</sup> The former focuses on the generativity of code; the latter on software as a cultural text. Formalism is typically visually abstract, and focuses on the processual relations of coding and aesthetic output; culturalism is critical, discursive and reflexive, deconstructing the «mind control» techniques of software. For Cramer this split in software art practice is troublesome because neither approach, individually, seems promising. More recently, Troels Degn Johansson has taken up this split, pessimistically labelling it the «no future» of software art.<sup>2</sup> Inke Arns clearly announces the same distinction: «generative art ≠ software art.»<sup>3</sup>

Based on current practice, it seems that this split is persistent. If there has been a recent shift in the balance, it has been towards

the abstract, formal or generative approach; a sign of its currency is that one of its favoured tools, Casey Reas and Ben Fry's *Processing*, won a Golden Nica at the 2005 Prix Ars Electronica. Here I want to set out a critique that focuses on abstract generative works, while ultimately attempting to overcome the split diagnosed by Cramer and Johansson, and the «no future» it implies. Instead the future for software art practice could lie in }a fusion of formalism and culturalism: what we might think of as a critical generativity.

Another way to position this argument is in terms of abstraction and complexity. For Lev Manovich, contemporary generative art is distinctively concerned with complexity, unlike the paradigm of reduction that characterised abstraction in the visual arts in the first half of the twentieth century. Here, following a scientific paradigm shift, the visual arts pursue «*new types of representations adequate to the needs of a global information society, characterised by ... new levels of complexity*».<sup>4</sup> Yet Manovich goes on:

*This still leaves open the question of representing the new social complexity symbolically. While software abstraction usually makes more direct references to the physical and biological than the social, it maybe also appropriate to think of many works in this paradigm as such symbolic representations. For they seem to quite accurately and at the same time poetically capture our new image of the world — world as the dynamic networks of relations, oscillating between order and disorder — always vulnerable ready to change with a single click of the user.*

This paper proposes another possible answer to the initial question, of how we might represent our «new social complexity.»

Software art does, as Manovich recognises, have a particular ability to address that situation, because it adopts complex (formal) systems as a basic generative tool. As such it can present not only an «image» of our situation, but more powerfully, a systemic abstraction, a model.

## 2. A Critical Approach: System Stories

So far the discourse around software and generative art has focused largely on defining and contextualising the field, and reflecting on its particular processes and materials—for example the nature of «code»,<sup>5</sup> or the question of software / process as art.<sup>6</sup> In order to come to grips with the works themselves, I would argue that any critique must be able to address the specifics of their generative systems; that the systems, not their outputs or residues, are the core of the work. System can be distinguished from code: code is the language-specific text that implements the abstract, formal structure that I will call system. So a code-literate reader can interpret system from code, but systems can also be described in other forms, either «natural» or other languages.

Software art systems are concrete collections of objects, relations, actions and processes. In part they are formal but constructed ontologies, describing entities and their interrelations. These ontologies are partly metaphorical or figurative—constructing for example «agents» in an «environment.» They are also partly technical / textual, in the sense that the implementation of these figures occurs within the structures of a formal language with particular representational and computational limits.

How do we read such systems, critically? They are literally texts, in their source code, but also in a critical sense, in that they involve specific figurations, relations, decisions, values and ideologies.

We can draw on the ways critics from the humanities have approached similar systems, from artificial life. Stefan Helmreich<sup>7</sup> and Katherine Hayles<sup>8</sup> have made strong analyses of a-life science, pursuing a basically deconstructive approach and arguing that a-life systems are fundamentally narrative in their operation. Moreover for these critics a-life's narratives themselves «re-inscribe» particular assumptions about embodiment, subjectivity, gender, family and theology. These narratives are

decoded in part from the discourse around the software system—Hayles for example makes use of a video representing Tom Ray’s *Tierra* system, where Ray’s biological and theological analogies are spelled out in the narration and the construction of the visualisation. However when Stefan Helmreich analyses John Holland’s *Echo*, a platform for creating agent-based a-life simulations, he does so based on conversations with a programmer and inspection of the code; Helmreich’s observations come as much from the defined formal structures of the software, as they do from the discourse around those structures. These analyses suggest a way of reading systems as stories; they in turn create new, critical stories based on that interpretation.

So, a «system story» is a translation or narration of the processual structures, ontology, entities and relations in a software system. Such stories are useful devices for opening up these systems to discussion and critique. System stories are not singular or objective; each one is a particular and situated reading. Nor are they floating signifiers though, since they draw on the concrete, formal object that is the software system. What generative art criticism needs are system stories that engage, in detail, with that formal object, and draw out its implications.

Hayles and Helmreich also provide an argument as to the importance of system stories. In their analyses, the narratives of artificial life are tacit, built-in assumptions which inform software models and simulations. In the case of a-life, there is an obvious relationship with the world «outside» the simulation—with life as we know and live it. The critics warn us against mistaking these assumptions for «the rules» of life—confusing the made with the given, or culture with nature. Similarly the value of system stories for generative art is in their ability to connect—critically, prospectively, speculatively—entities and relations within the system, with entities and relations outside it.

A cultural critique of software art systems is the bridge spanning Cramer's formalist / culturalist duality. It seems to offer a way to read generative software art according to the critical paradigm of the software culturalists. Yet how can this approach be compatible with the paradigm of abstraction that characterises this work? As Manovich has shown, abstraction is recognised as a hallmark of contemporary generative art; for Brad Borevitz this software has been uncoupled from instrumentality or referentiality—it «*serves nothing save its own play, display and critique.*»<sup>9</sup> Even when it uses the modality of simulation, its «*simulations ... may refer only generally to real-world physics, since they borrow the formulations of Newtonian rule merely to abstract them and play with them according to the demands of an aesthetic production...*» My argument is complementary, but not contradictory, to Borevitz. Simulation techniques are used in these works as generative devices, not as tools for modelling; but nonetheless the work is entirely shaped by the construction of its underlying system, its configuration of entities and relations. That configuration, what Borevitz calls its «logic» or «systemacity,» is revealed to the user through a process of dynamic interaction; as Borevitz says there is a kind of experiential reverse-engineering at play, as we map back from residue or output to system. Once again however, the system is core, and therefore surely the structure of that system is crucial. Especially in works using simulation and related techniques, abstract generative art performs *cosmogeny*: it brings forth a whole artificial world, saying, *here is my world*, and *here's how it works*. Once again, I will argue that this practice is in a unique position to explore and critique «how it works.» Borevitz quotes Greenberg on abstract painting and sculpture: «like functional architecture and the machine, they *look* what they *do*.» So, what do they do?

### 3. System Stories: Some Examples

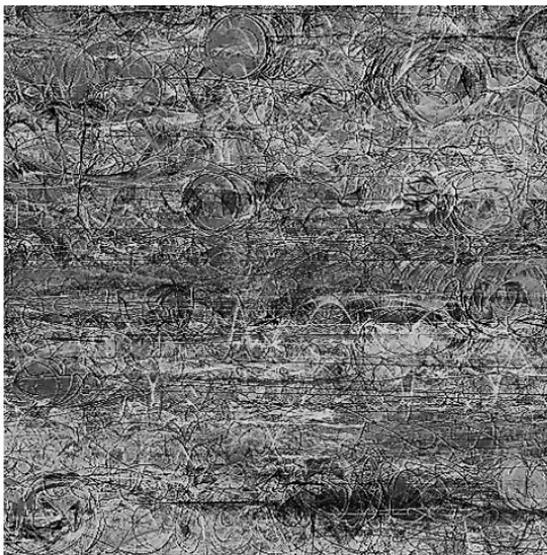
Engaged as it is in the pragmatics of generativity—of making something make something—generative software art turns to computationally expedient techniques. The simplest of these is combinatorics or the playing out of permutations. Some recent visual generative art follows this approach, setting a simple system in motion and observing its outcomes. The results are visually complex, but the underlying system is surprisingly simple, as in some of the pieces in Casey Reas' *Software {Structures}*: Reas' #002 and #003, Tarbell's #003A and #003B, and Ngan's #003B.<sup>10</sup> In this project the artist's focus was reflexive and processual: considering the «natural language» specification of a structure, and its varied implementation. Removed from that context, however, we are faced once again with the shape of the system, and the question of interpreting, or responding to that configuration of entities and relations. The model worlds in these instances are pure machines, clockwork constellations. They transform determinism into aesthetic complexity using scale of population and a kind of analytic or integrative visualisation—displaying spatiotemporal relations rather than the entities themselves. What is extraordinary here are the forms and patterns generated by that derived visualisation: deterministic but impossible to predict, as if the LeWitt-inspired procedural structure was being viewed through some strange high-dimensional lens (see especially Tarbell's #003B). Yet the underlying systems themselves are crystalline and impervious, and this character underpins our experience of these works.

*Software {Structures}* also shows examples of another common world-system, using techniques of physical simulation. Hodgkin's implementations of #003, and Ngan's #003A, both introduce simulations of momentum and gravity (disobeying the «structure» in the process). Among the many other uses of this technique are

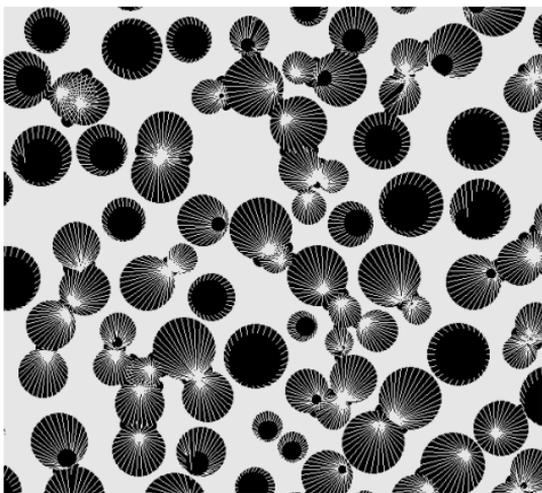
Mark Napier<sup>11</sup> and Scott Snibbe's<sup>12</sup> works in the CODEDOC project. These techniques are pragmatic and effective, in generative terms: they create complex, dynamic interactions between elements, at a low computational cost. They also bring with them an immediate physical resonance, as we recognise these physical dynamics and infer the properties of the entities (their relative masses, the strength of gravity). As Borevitz says these techniques are generative, but they are also inherently narrative and metaphorical, they create model worlds and characteristic patterns of relations. It's striking to observe how a strictly physical simulation provides a basis for the artists' organic and even social analogies: Ngan writes of trying to imbue a «sense of life» into the entities in his beautiful #003A; Hodgkin describes the results as «organic» and «cellular»; Tarbell goes further, imagining the circle entities «experiencing» and «choosing» intersections, «analogous to daily life.»<sup>13</sup> This critique is not intended to discourage or overinterpret these narratives, but rather to imagine the consequences of taking them more seriously, especially in their potential relationship with the «outside» world.

This unfulfilled potential is especially clear in the way generative art uses multi-agent systems. In this ubiquitous technique, entities are explicitly defined and visualised, often literally traced as they move around a cosmos/canvas. Their relations with each other can be more complex than in a physical simulation, including «flocking» behaviour, where individuals modify their motion based on that of their neighbours (see for example Alessandro Capozzo's *Relations* series<sup>14</sup>). Casey Reas has used this technique extensively, in systems including *Tissue*, *Microimage*, *Articulate*, *TI* and *Cells*.<sup>15</sup> Reas' systems show the organic multiplicity of the flock, but also add mobile «attractors» that draw in swarming elements.

Here too, the generative technique is effective in creating visual complexity, and emergent dynamic form; but again each multi-agent



JARED TARBELL, STRUCTURE#003B



WILLIAM NGAN, STRUCTURE #003A

system encodes an ontology, a structure of entities and relations, which must be read as the core of the work. The entities themselves have characteristic properties: they are identical, or belong to a set of pre-defined types, and their properties and behaviour are static over time. The systems have a particular relation to time: they tend to be a series of instantaneous slices. The state of the system at one moment is a function of its state in the moment just passed (this is also true of physical simulations). In other words, history is all but absent. This is reflected in the construction of «agent» and «environment» in these systems. The environment here is (literally) a blank canvas, inert, empty space. Agents tend not to have a means of influencing that environment—even when they leave «traces» in that space, the traces have no impact on the agents. The traces are visualisation devices, not entities in the formal ontology. What kind of narrative is this? All these attributes can be explained as computationally pragmatic—the simplest or most efficient way to achieve the generative payoff of the swarm aesthetic. Again any referentiality of this system can be downplayed in favour of pure generative instrumentality. And again I would argue that in fact these works are fundamentally determined by this ontology, and that in a basic way we *see* it in the works (cf. Greenberg, above). The works visualise their structure of entities and relations. They model a world. My concern is not for realism or to oppose the necessary abstraction that any simulation or agent-based system involves. Rather it is to point out that these systems encode, for whatever reason, specific ontologies, and that those ontologies in turn, especially in agent-based systems, present specific attributes: modes of being and relation, relationships between individual and group, morphology of groups, relations of individual and environment, models of being-in-time. Manovich sees in such work an image of «world as the dynamic networks of relations, oscillating

between order and disorder—always vulnerable ready to change with a single click of the user.» This is true, the swarm aesthetic is enchanted with dynamic multiplicity, with shifting networks of relation, with coalescence and dispersal. But consider the subject or agent modelled here, if that's the story we want to tell: a clone in a crowd, unchanging, with no traction on the space it inhabits, existing in an ongoing, perpetual present. If these systems provide images of contemporary society then they are, at best, naïve and utopian: a mass of identical (or typed) individuals, each contributing equally to the collective dynamic, each equally connected with and affecting all the others. As a social model this is a kind of idealised, frozen anarcho-democracy, where power relations (unequal causal connections) can never emerge.

#### 4. Possible Worlds

This critique is simply a starting point; its flipside is more positive and important. If generative software art communicates system stories, particularly in the form of model worlds or ontologies, then it is potentially a platform for telling system stories that are more sophisticated, critical or experimental; it could take seriously the prospect that Manovich proposes, the potential of software and generative technique to provide images of, or rather *imaginations of*, the (social, cultural, personal, material...) systems we live in. Generative art has a unique potential here, because unlike other art forms its basic materials are systems themselves.

I will use a handful of works here to illustrate this (mostly unstated) potential as it appears in contemporary generative works. While many multi-agent systems are ontologically awkward, the genre can tell more interesting stories. Casey Reas' works *Tissue* and *Microimage* begin to develop the homogeneous swarm, creating distinct «species» of agent with distinctive (but again fixed) rela-

tionships. The added complexity of the interaction within the system is revealed in the images, as tangled clouds resolve into dark loops and braids. Similarly Ichitaro Masuda's recent work *Haohao*<sup>16</sup> has multiple species of agent, differentiated in size and colour, and attracted to and repelled from each other to varying (randomised) degrees. While Masuda's code reveals that the parameter for attraction is «love», this is no agent-meets-agent story. Individuals form pseudo-stable clusters of five or more where forces of attraction and repulsion are in equilibrium; these clusters might in turn orbit other groups, and are readily disrupted if another agent approaches. If there is a social story here, it is one of pursuit, desire and loss, but above all the delicate negotiation of local collectives or cliques. Once more this dynamic informs the aesthetic of the trail-paintings which the system produces, with tight gnarls and knots, as well as dense circular orbits and linear vectors.

These examples retain the usual disconnection between agent and environment—agents interact with each other, but have no functional impact on their world. However this feature is not computationally or formally necessary, and in fact there seems to be a generative and aesthetic payoff for linking agent and environment more tightly. Mauro Annunziato's *Artificial Societies* drawings are an excellent example of this.<sup>17</sup> Their character arises from a simple feature of his system in which agents' paths are drawn in to the environment; agents «die» when they intersect another's path. Equipped with a simple genetic / evolutionary mechanism, the agents progressively divide their environment into isolated «habitats», each applying a particular selection pressure to the agents within it. Annunziato shows that the environment need not be a blank space, but can be a powerful generative constraint that also brings a system's history to bear on its present and future (for a further discussion see<sup>18</sup>). Another beautiful example comes



MAURO ANNUNZIATO, CONTAMINAZIONE (DETAIL)

from Dutch artists Driessens and Verstappen, whose *E-volver* software generates images using a diverse «ecosystem» of pixel-manipulating agents. Each individual agent has a (deterministic) rule-set for moving around the image and altering pixel values; yet the interaction between agents, especially *through* their shared environment, gives rise to image surfaces which are strikingly unified and organic.<sup>19</sup> Environment here is a dynamic terrain, a developing residue which again shapes agents' behaviour in an ongoing co-formation.

## 5. Critical Generative Systems

Narrative critiques reading software and generative art have a significant limitation, or rather a kind of grain or directionality. They can decompose a system, analyse the modes of being and relation that it encodes, but they have little to say about how those encodings play out, how they operate in a generative process. The emergence of complex, dynamic forms and behaviours from these local encodings is central to artists' interests in complex systems<sup>20</sup>; this is the moment of emergent generativity or the «computational sublime.»<sup>21</sup> Once again this is where generative art is in a unique and powerful position, in that unlike other forms of discourse, it can actually experiment with the emergent outcomes of particular ontologies, modes of being and relation. Christopher Langton inaugurated artificial life under the banner of «life as it could be»; Stefan Helmreich has argued instead that a-life systems reinscribe social conventions of «life as we know it.» However Helmreich ultimately recognises the potential of a-life in undertaking not increasingly-accurate simulations of an authorised «life», but experimental, reflexive performances of possible lives.<sup>22</sup>

So too for generative art, though its scope should be wider. In the critiques above I have focused on social narratives and ontologies,



ERWIN DRIESESS AND MARIA VERSTAPPEN,  
IMAGE FROM E-VOLVER / E-VOLVED CULTURES

but generative art's models might move across (and especially between) domains—physical, chemical, personal, social, cultural, technological, economic. So far I have also emphasised the prospective or utopian potential of generative ontologies, but this is only one of several possible modes or registers for these narratives; they might equally be ironic, critical, deconstructive or fantastic. Golan Levin's *Axis* applet abstracts political rhetoric into a database-driven combinatoric.<sup>23</sup> It's not difficult to imagine a generative process that draws its algorithms from the same source, extrapolating, diverting or visualising rhetorical entity/relation structures. Once again we should reconsider the distinction between critical, reflexive, cultural software art, and utilitarian, unreflexive, result-oriented generative art.

One of the further implications here is a reconsideration of the context for generative art. If it is fundamentally concerned with creating model ontologies, then we can imagine it in relation to other practices of formal modelling and simulation. These techniques have a long history in military strategy and geopolitics, but in recent years they have become more widespread. For example, a new branch of social science has emerged which uses simulation as a basic tool for testing «explicit models of social phenomena.»<sup>24</sup> One recent paper from this field claims to model the «dynamics of youth subculture,» creating a multi-agent simulation and discovering that «only a few assumptions of the individual's behaviour are necessary to regenerate known features of youth culture.»<sup>25</sup> In other words: we are already being modeled, in artificial worlds that can fold back powerfully into the real. Like Helmreich I would be very concerned if social modeling was used only to entrench our «known features». Unknown features must be more promising, and here again generative art can step in. Borevitz writes: «If there is a chance that software will contribute significantly to a new politically relevant aesthetics, it lies in the

way software shows us a way out of order, in and through order.»<sup>26</sup> Yes, but what's required is attention to the specifics of that order, its structures and properties. Generative art can, and must, do more than make images of complex systems; it can tinker critically with the systems themselves, then set them running: possible worlds.

If abstract or generative software art can, and sometimes does, work this way, where does this leave the binary of formalism / culturalism, or generative / software art? Perhaps the relation could be one of complementarity. «Culturalist» software art has often focused on intervening critically, and practically, in existing software systems, reconfiguring them from the inside. In the process it shows up the latent cultural agency of software, but also its potential transformation. For Johansson however the critical specificity of this approach is also a limit to its potential; following Cramer he worries that it might become merely a «critical footnote» to mass software culture.<sup>27</sup> Johansson calls instead for «an alternative» to «established formats.» By «formats» I understand cultural and social, as well as technical constructs. As I have argued we can think of abstract software art, or generative art, as potentially exploring alternative modes of being and relation, telling stories but also literally toying with complex, dynamic systems, exploring them prospectively, and not (merely) as eye-candy machines, but as model worlds. To re-state the binary: perhaps generative formalism can be prospective and exploratory, where culturalism is more local, situated, concrete, interventionist. The two strands might in fact be complementary, and their critical potential might be far greater if we think them together, instead of apart.

- 1 Florian Cramer, «*Concepts, Notations, Software, Art*», 2002. Available:  
[http://userpage.fu-berlin.de/~cantsin/homepage/writings/software\\_art/concept\\_notations//concepts\\_notations\\_software\\_art.pdf](http://userpage.fu-berlin.de/~cantsin/homepage/writings/software_art/concept_notations//concepts_notations_software_art.pdf)
- 2 Troels Degn Johansson, «*Mise en Abyme in Software Art: A Comment to Florian Cramer*», in *read\_me Software Art & Cultures Edition*, ed. Olga Goriunova & Alexei Shulgin, Center for Digital Æstetik-forskning, 2004, p.151. Also available:  
<http://runme.org/project/+miseenabyme/>
- 3 Inke Arns, «*Read\_me Run\_me, Execute\_me : Software and its Discontents, or: It's the Performativity of Code, Stupid*», in *read\_me Software Art & Cultures Edition*, ed. Olga Goriunova & Alexei Shulgin, Center for Digital Æstetik-forskning, 2004, p.178. Also available: <http://runme.org/project/+executeme/>
- 4 Lev Manovich, «*Abstraction and Complexity*», 2004. Available  
[http://www.manovich.net/DOCS/abstraction\\_complexity.doc](http://www.manovich.net/DOCS/abstraction_complexity.doc)
- 5 Geoff Cox, Alex McLean and Adrian Ward, «*The Aesthetics of Generative Code*». Available: <http://www.generative.net/papers/aesthetics/>  
 See also Inke Arns, «*Read\_me ...*
- 6 Tjark Ihmels and Julia Riedel, «*The Methodology of Generative Art*», 2004. Available:  
<http://www.medienkunstnetz.de/themes/generative-tools/generative-art/> See also Florian Cramer, «*Concepts...*
- 7 Stefan Helmreich, *Silicon Second Nature: Culturing Artificial Life in a Digital World*, University of California Press, 1998, pp.163-164.
- 8 N. Katherine Hayles, *How We Became Posthuman*, University of Chicago Press, 1999, pp.227-231.
- 9 Brad Borevitz, «*Super-Abstract: Software Art and a Redefinition of Abstraction*», in *read\_me Software Art & Cultures Edition*, ed. Olga Goriunova & Alexei Shulgin, Center for Digital Æstetik-forskning, 2004, pp.310- 311. Also available:  
<http://runme.org/project/+super-abstract/>
- 10 Casey Reas et al, *Software {Structures}*, 2004. Available: <http://artport.whitney.org/commissions/softwarestructures>
- 11 Mark Napier, *3 dots*. Available:  
<http://artport.whitney.org/commissions/codedoc/Napier/app/wave.html>
- 12 Scott Snibbe, *Tripolar*. Available:  
<http://artport.whitney.org/commissions/codedoc/Snibbe/Tripolar.html>
- 13 William Ngan, Robert Hodgkin and Jared Tarbell, 2004. Available: [http://artport.whitney.org/commissions/softwarestructures/s3\\_william\\_1/comments.html](http://artport.whitney.org/commissions/softwarestructures/s3_william_1/comments.html)

- 14 Alessandro Capozzo, *Relations*. Available: <http://www.abstract-codex.net/relations/relations/index.html>
- 15 Casey Reas, «*Software*», available: <http://www.groupc.net/category.php?section=software>
- 16 Ichitaro Masuda, *Haobao*, 2004–5. Available: <http://www.iamas.ac.jp/~madmano3/haobao/>
- 17 Mauro Annunziato, *Artificial Societies*, 1998–2002. See: <http://www.plancton.com/art-soc/asociety.htm>
- 18 Mitchell Whitelaw, *Metacreation: Art and Artificial Life*, MIT Press, 2004, pp.169–173, 178–179.
- 19 Erwin Driessens and Maria Verstappen, *E-volver / E-volved Cultures*, 2004. Available: <http://www.xs4all.nl/~notnot/e-volver/e-volver.html>
- 20 Casey Reas, «*Microimage*», 2003. Available: <http://reas.com/texts/microimage.html>
- 21 Jon McCormack and Alan Dorin, «*Art, Emergence and the Computational Sublimes*», in *Proceedings of Second Iteration: a conference on generative systems in the electronic arts*, ed. A. Dorin, CEMA, 2001, pp.67–81.
- 22 Stefan Helmreich, *Silicon Second Nature...*, p.244
- 23 Golan Levin, *Axis Applet*, 2002. Available: <http://artport.whitney.org/commissions/codedoc/Levin/axis.html>
- 24 Nigel Gilbert, «*Simulation: An Emergent Perspective*», 1995–6. Available: <http://cress.soc.surrey.ac.uk/resources/emergent.html>
- 25 Petter Holme and Andreas Grönlund, «*Modelling the Dynamics of Youth Subcultures*», *Journal of Artificial Societies and Social Simulation* vol. 8 no. 3 (2005). Available: <http://jasss.soc.surrey.ac.uk/8/3/3.html>
- 26 Brad Borevitz, «*Super-Abstract...* p.311
- 27 Troels Degn Johansson, «*Mise en Abyme...* p.154