Florian J. Gruber

Paper: **CONCEPTUAL FRAMEWORK FOR GENERATIVE DESIGN**

Abstract:

Designing programmers or programming designers are a rarity. It is usually assumed that one thinks either rational or intuitive. A combination is the exception. The technological hurdle of 'programming for artists' is already overcome by a number of freely available, well documented development environments. Also a number of articles and books have been published, dealing with issues of craftsmanship and the technical aspects of generative design and processual computer art. What is missing, is a conceptual analysis of work practices and the development of specific design methodologies for artists and designers, who understand code as tool and material for their artistic creations.

This paper outlines a conceptual framework by identifying essential aspects of generative design processes, derived from literature and the author s practical experience as media artist. These aspects are presented as process models and a deck of method cards, which help you plan and assess complex projects, by quickly sketching out different scenarios and applications for generative works.

Topic: Applied Art,

Design Theory

Authors:

Florian J. Gruber
Institute for Design &
Assessment of
Technology,
Vienna University of
Technology, Austria

References:
[1] Georg Trogemann, Jochen Viehoff, “Code@Art”, Springer, Vienna, 2005

[2] Holger Schulze, “The Aleatoric Game”, Fink, Munich, 2000
[3] www.ambientartlab.com

Contact:
florian.gruber@ambientartlab.at

Keywords:
generative design, conceptual framework, method cards

Introducing a Conceptual Framework for Generative Design Processes

Florian J. Gruber, Dipl.-Ing.(FH)
Institute for Design and Assessment of Technology,
Vienna University of Technology, Austria
http://igw.tuwien.ac.at/hci
http://www.ambientartlab.com
e-mail: florian.gruber@ambientartlab.at

Abstract

Designing programmers or programming designers are a rarity. It is usually assumed that one thinks either rational or intuitive. A combination is the exception. The technological hurdle of 'programming for artists' is already overcome by a number of freely available, well documented development environments. Also a number of articles and books have been published, dealing with issues of craftsmanship and the technical aspects of generative design and processual computer art. What is missing, is a conceptual analysis of work practices and the development of specific design methodologies for artists and designers, who understand code as tool and material for their artistic creations.

This paper outlines a conceptual framework by identifying essential aspects of generative design processes, derived from literature and the author´s practical experience as media artist. These aspects are presented as process models and a deck of method cards, which help you plan and assess complex projects, by quickly sketching out different scenarios and applications for generative works.

1. Introduction

All computer-based generative design techniques can be simplified to a theoretically straightforward action – the selection and organization of elements from a chosen repertoire, according to a set of rules programmed by the author. The repertoire consists of digitally representable media objects, whose properties can be modulated, varied, automated and transcoded [1]. The generative software runs in real-time with a certain degree of autonomy and self-organization. Generative designers use the principles of circular systems to produce artworks by constantly repeating and modulating a set of computer operations. The designs are ‘classes of artworks’ [2], which are self-similar and vary like natural or organic systems within certain minima and maxima. There are uncertainties, but the sense of cause and effect remains [3]. Designing with algorithms is contiguous to the principle of an aesthetic theory [4], which inextricably link the formulation of procedures and the production of aesthetic objects.

2. Method

Based on several media art projects [5] created with the ambientartlab collective [6] and the author´s PhD thesis research, common aspects of generative design processes were identified. A synthesis of computer art history, design theory and personal experiences in collaborative, interdisciplinary teams led to the development of a Generative Design Model (GDM) and Generative Design Method cards (GDM cards).

The findings are based on an arts-based research method, in which artistic practices are reflected intersubjectively [7] and are transferred to a standardized system to investigate essential elements of the creation process [8]. The results are intended to advance the art discourse and help designers, artists and technicians to plan, discuss and evaluate generative procedures not only from a technical or aesthetic point of view, but aspect by aspect within a conceptual framework.

3. Aspects

3.1. Combinatorics (Repertoire, Selection, Organization)

Compared to traditional design projects in generative design we have to develop processes which continuously vary and transform along a time axis. Possible and particularly interesting makes this one ‘constant’: coincidence. Every person has an individual idea of coincidence and can term the probability or likeliness of certain events to come true. One important aspect of generative design is to articulate those individual perceptions of chance as mathematical probabilities and computable conditions. Pure chance does not exist. There has to be always a certain intention or necessity to make chance happen. Generative art refers to this ‘objective’ or ‘deterministic’ coincidence. While producing random and unpredictable events, the combinatorial process is always deliberately designed and executed with certain intent in mind.

Schulze [9] identified three basic elements of so called 'aleatoric games' in his analysis of nonintentional artforms of the 20th century, which can be easily transferred to generative design processes. The **repertoire** of an aleatoric game consists of single elements or groups (like words, sound samples, acoustic envelopes, geometric shapes, color palettes, photo or video material), which are selected or activated by a certain set of rules (**selection filters**). The selected elements are temporally and spatially distributed by organizational rules. The **organization** can occur serially or according to a prearranged pattern or again determined by chance.

3.2. Programmability (Function, System)

Programming in an artistic context means a very tight intertwining of form and function. In contrast to commercial software engineering, artistic software is not developed in large teams and to serve a mass of users. Sometimes there is only one user – the author him/herself. Programming as an art form is not about solving a defined problem, but to approach problems and to continuously line up questions [10]. A generative designer has a dual role as a programmer and as an aesthete. Aesthetically we formulate the artistic criteria and in practice we develop algorithms that transmit the aesthetic qualities from the creative into the binary language [11]. This transfer is not a single task, but a continuous approximation and revision of the original ideas.

Generative designers often develop self-referential, cybernetic systems or so called nontrivial machines [12]. By changing its internal rule set a nontrivial machine continuously produces a different output when the same input is applied. These machines are analytically undeterminable and unpredictable. They are synthetically determined but in a functional analysis trans-computational. One challenging aspect of generative designers is to describe non static, dynamic forms with computational code and defining the **function** and the aesthetic **system** of the machine.

3.3. Processuality (Openness, Intention)

Generative artists develop systems. Systems are nested processes between the two poles of order and randomness. The most inspiring and complex results are obtained by systems that implement a mix of surprise and redundancy. In our human perception something completely arbitrary and something highly organized holds very few meaningful content. Both poles have a very low effective complexity level, as Galanter [13] showed.

The situation is similar with the artistic concepts of work and process. A work of art in the emphatic sense is untouchable, though open for different interpretations, it will never be altered or questioned [14]. On the opposite side, a process is vague, its form is loose and **open**. A process cannot be objectified, it seems every time different, every time new. Both poles – the pure work and the pure process – are idealized and do not exist as such. More likely in daily practice is a convergence, a hybrid of processual and structural elements to achieve a compelling work of art.

There are two ways of designing with **intent**. In classic form a certain idea is fixed to a semantic relation (coherence) and the subsequent process of enriching, detailing and puzzling is always related to the initial idea [15]. With the technique of ‘heuristic fiction’ [16] an intention, a meaning or an interpretation evolve retrospectively while developing and working with the generative program.

3.4. Interactivity (Communication, Interaction, Dramaturgy)

Delinear processes, self-referentiality, bidirectional communication and networked systems have become paradigms of our time. Instead of objects with static properties, we research, study and analyze dynamic relations in sciences and arts.

Today it is less about the exploration of the essence of things, to question is how the processes evolve, how they connect and how they interact. Generative designers not only rely on these paradigms, they work with it.

Interactivity and participation are not a must-have of a generative design piece, but especially with computer-based art the ability to participate and interact with an ongoing process, is an important aspect. Compared to the batch programs of the early computer artists nowadays playing and improvising with a generative program in real-time is possible in many different ways. Through interaction computer art becomes more tangible and loses part of its per se inherent virtuality and detachment. By creating experimental interfaces or control units a generative program becomes a creative tool.

Developing a **communication model** and anticipating possible or desirable **user interactions** connects a generative artifact to its environment. By creating an interactive **dramaturgy** a generative piece of work can be set up as an instrument or installation.

4. Generative Design Model (GDM)

Generative design is procedural. Not the development of a completed and closed work of art is the focus of the design process, but the creation and formalization of an aesthetic system. Combinatorics and improvisation are two possible approaches to create an aleatoric game. In computer-based arts, both ways are possible through real-time interaction during the combinatorial calculation process. A generative design process can be either combinatorial or interactive or can combine both techniques.

The GD model introduced in the following, shows 1) the procedural scheme of generative design processes, (2) the structural composition and (3) the progression over time. At the intersection of the combinatorial and improvisational part is at any time the current state of the generative system (state S).

The GD model is divided into two halves – combinatorial processes and interaction processes. Analogous to Laurels ‘flying wedge model’ of interactive narratives [17] at the beginning of a generative sequence the potential of its development is completely open (**Possible**), but with the programs progressive course a specific state becomes more likely (**Probable**) and is ultimately required by the previously made decisions (**Necessary**).

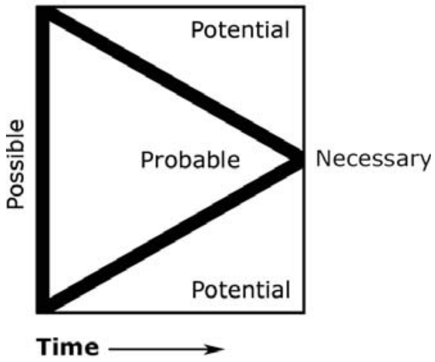


Fig.1 - Laurel´s Flying Wedge Model (sketch by the author)

4.1. GDM Infinity

The development potential at the beginning of an iterative sequence can be compared to the sample space or universe Ω of an experiment in probability theory and is the set of all possible outcomes. The universe of a generative project is the formal and aesthetic framework created by the designer. Every outcome within the scope of the generative program is possible.

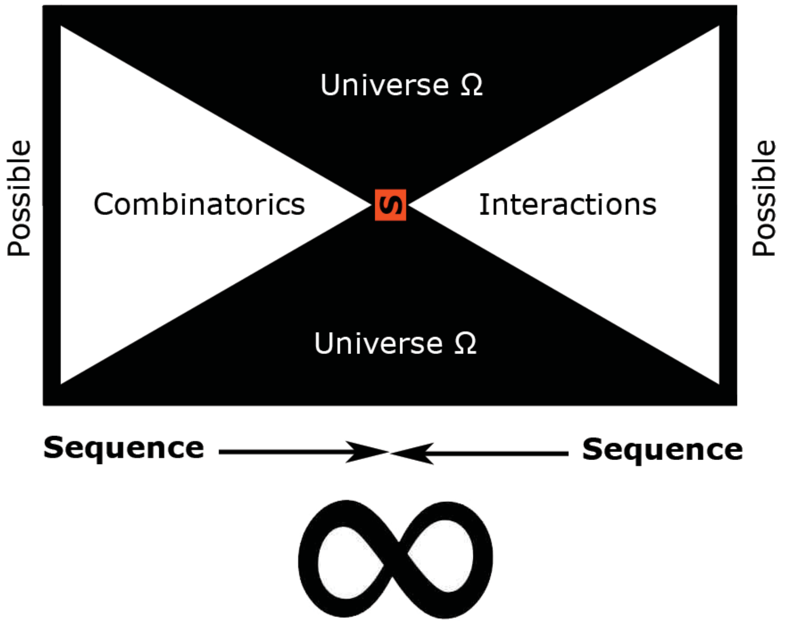


Fig.2 - GDM Infinity

With each loop, if-construct and case differentiation a certain state (**S**) becomes more likely through the interplay of combinatorics and interactions. At the end of a sequence the process restarts. Theoretically for infinity, in practice until a certain termination criteria is met (GDM Hourglass).

4.2. GDM Hourglass

The task of the generative designer is to unify technical issues and aesthetic ideas. Both as generative sequence of combinatorics and interactions frame by frame, as well as a procedural genesis and narrative structure over time until the program finally terminates.

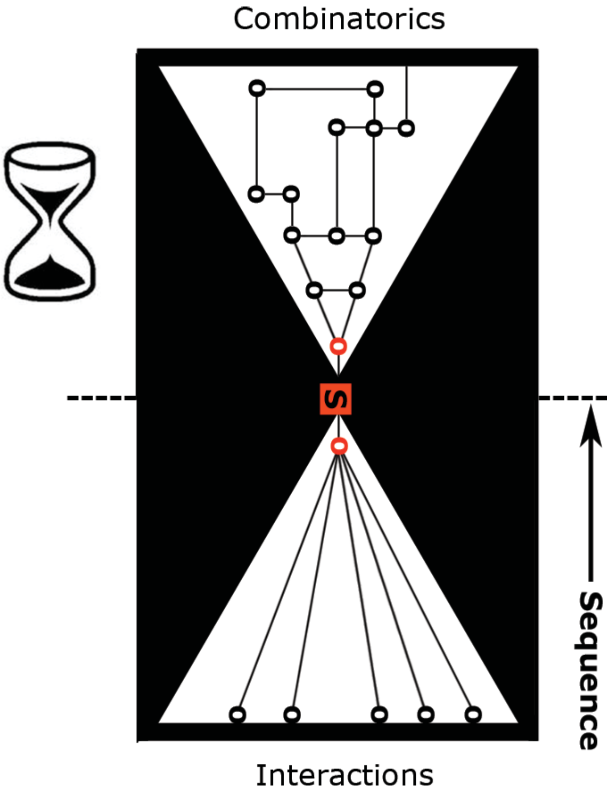


Fig. 3 - GDM Hourglass

The GDM Hourglass reads as a structural model of generative programs. Each node at the combinatorial part represents a different internal state of the machine that modifies the interaction sequence. In return the interactions influences the current internal state (node) of the combinatorics.

Placed on a time axis the GDM Hourglass depicts the principle of a nontrivial machine. It will continuously change its state, respond differently to the same inputs and become analytically unpredictable.

4.3. GDM Time-based

A continuous change over time is a characteristic feature of generative design processes. Depending on the work concept, these changes can be past-dependent or past-independent. Past-dependent processes take into account the current state of the system (**S**) for the next transformation sequence, past-independent processes do not include the current state and restart anew with each iteration. Both concepts can be visualized with the GD Time-based model.

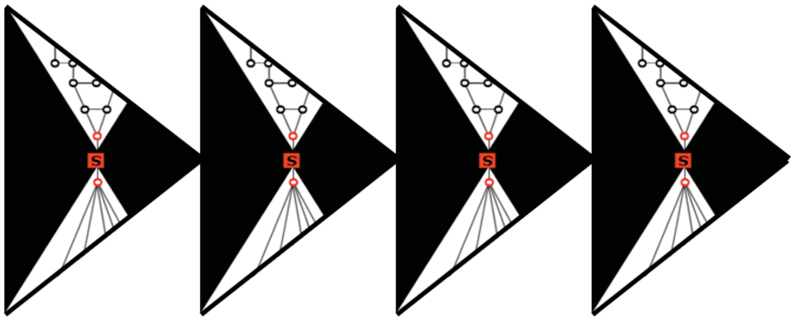


Fig.4 - GDM Time-based (past-independent)

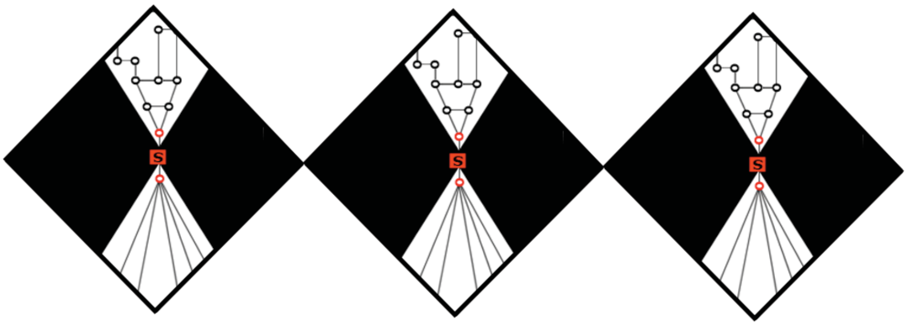


Fig.5 - GDM Time-based (past-dependent)

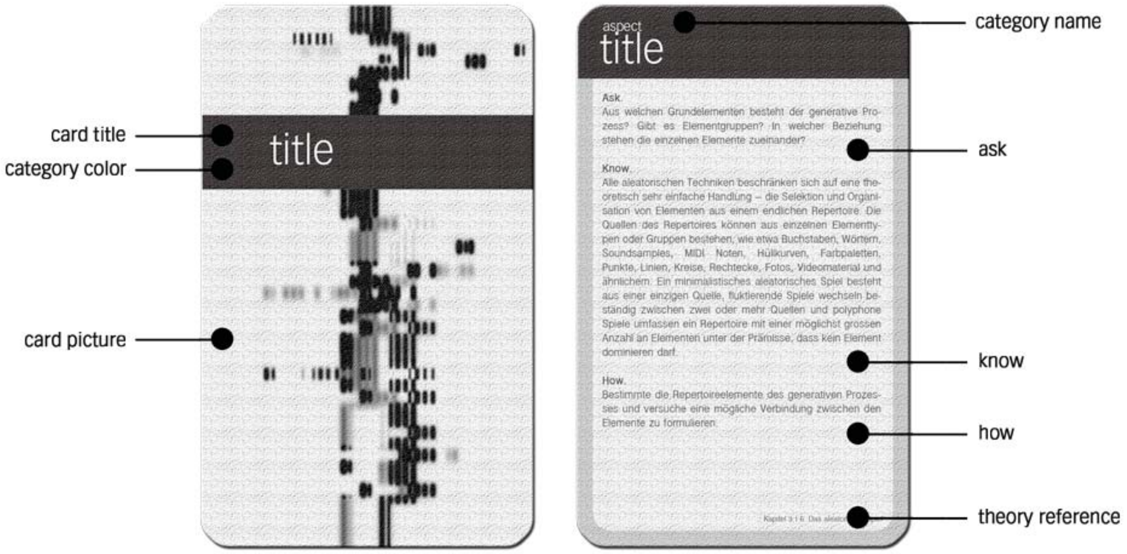
5. Generative Design Method Cards (GDM cards)

Working with tangible objects in the conception and planning phase of a project can have advantages over purely digital documents. With physical artifacts different scenarios can be sketched out and discussed on the fly by connecting, arranging and combining artifacts on a table or wall. Photographs, post-its and stickers can be used in design and project management teams to collaboratively comprehend complex issues and develop a common understanding [18]. In software engineering, agile development methods with ‘story cards’ and ‘estimation cards’ [19] representing use-cases and complex functions, have become more popular in recent years with developers and clients alike. Using metaphors and defining user stories help to establish a basis of discussion and evaluation, and promote object oriented and module based software design and development.

Programming as an art practice is inherently agile. There is rarely an exact specification or completely worked out plan. Generative designers are continuously refining, testing and exploring their algorithms. ‘Working software’ as it states in the agile software development manifesto [20], is a prerequisite. Generative design is ‘responding to change’ per se, as is the development process by constantly validating, retaining or discarding different options to achieve specific aesthetic goals. The project itself is often in constant transformation and never completely finished. It remains open and contains an invitation for further development and change.

The GDM cards are a set of cards to foster and encourage conceptual thinking in generative design projects and to establish a discursive basis in collaborative environments. Any number of cards from the deck can be used in a project. The point is, to single out specific aspects and elements of the overall process and give an impetus for discussion and reflection. The more cards are used, the better a possible scenario can be analyzed. The GDM cards can be used in any stage of the work. For example at kick-off meetings to work out and evaluate ideas, in the middle of a development process to help designers make decisions, and at the end for reflection and analysis.

The structure of the cards is similar to the IDEO method cards [21], which report a very positive feedback from different areas of applications and from groups that are not necessarily engaged in design initiatives. Each of the ten GDM cards contains a brief description and theory reference of the identified aspect (**know**), a set of questions to **ask** and instructions **how** to continue and what to do.



Ask.
Aus welchen Grundelementen besteht der generative Prozess? Gibt es Elementgruppen? In welcher Beziehung stehen die einzelnen Elemente zueinander?

Know.
Alle aleatorischen Techniken beschränken sich auf eine theoretisch sehr einfache Handlung – die Selektion und Organisation von Elementen aus einem endlichen Repertoire. Die Quellen des Repertoires können aus einzelnen Elementtypen oder Gruppen bestehen, wie etwa Buchstaben, Wörtern, Soundsamples, MIDI Noten, Hüllkurven, Farbpaletten, Punkte, Linien, Kreise, Rechtecke, Fotos, Videomaterial und ähnlichem. Ein minimalistisches aleatorisches Spiel besteht aus einer einzigen Quelle, fluktierende Spiele wechseln beständig zwischen zwei oder mehr Quellen und polyphone Spiele umfassen ein Repertoire mit einer möglichst grossen Anzahl an Elementen unter der Prämisse, dass kein Element dominieren darf.

How.
Bestimmte die Repertoireelemente des generativen Prozesses und versuche eine mögliche Verbindung zwischen den Elemente zu formulieren.

Fig. 6 - Sample GDM card

The GDM cards are basically designed to be used en suite from 1 to 10 and thereby specifying the generative process from its basic repertoire elements to its dramaturgical concept.

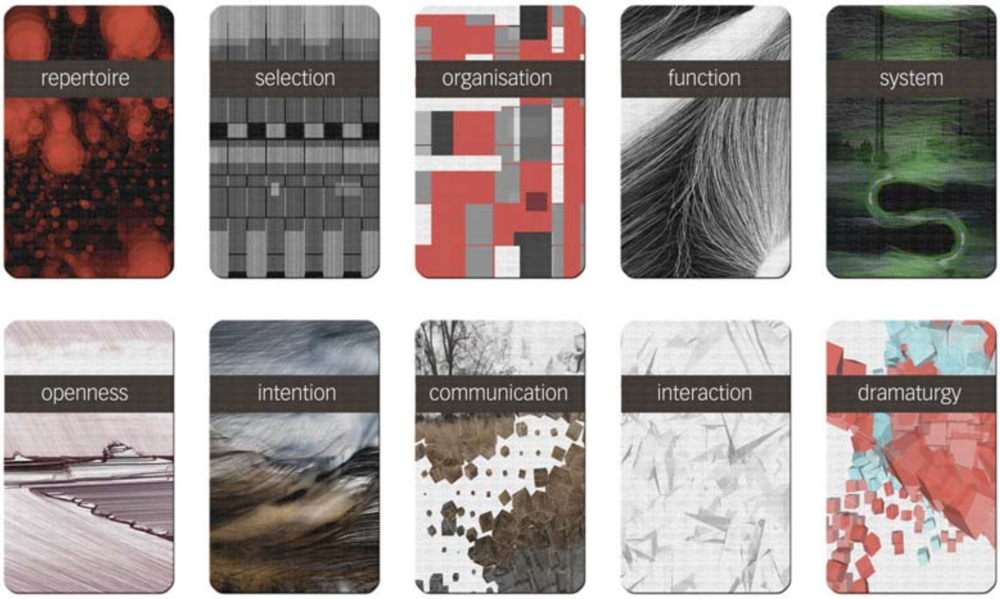


Fig. 7 - GDM cards (front view)

Not every project must go or can go through every aspect. The GDM cards can also be used individually to more accurately define one aspect or specifically work out a connection of two or three aspects in a project. The GDM toolkit is intented to explore different approaches, to solve a problem, to gain perspectives and to adapt and develop own methods.

6. Conclusion

The paper has outlined a conceptual framework for generative design processes based on a review of computer art, design theory and the author´s experiences as media artist in collaborative, interdisciplinary design teams.

All identified aspects refer to computer-based generative design processes. The GD models are visual schemes that depict the cyclical development process, the recursive program structure and the transformative characteristics of generative design programs. The GDM cards are an introduction to a conceptual toolkit for generative design to assist a common understanding and holistic project development. The cards prime intent is to inspire the design process, act as decision-solver and lead to new approaches and agile development processes.

It is my belief that a semantic naming of aspects, a meaningful visual representation and practically usable objects can benefit and enrich a design process for all participants. A study of the usage and effect of the framework in projects with different sizes and complexity would give further insight on how to improve its utilization and design. A fully translated and edited version of the GDM cards will be available soon with the publication of the author´s PhD thesis (spring 2012). Meanwhile I welcome artists, engineers and designers in generative projects to use, peer-review and contribute to the proposed framework.

References

[1] Manovich, L. (2001). *The Language of New Media*, Cambridge: MIT Press.

[2] Nake, F. (2010). *Paragraphs on Computer Art, Past and Present*, in *Proceedings of CAT 2010 London Conference*. London: CAT 2010.

[3] Galanter, P. (2003). *What is Generative Art? Complexity theory as a context for art theory*, in *Proceedings of the 6th Generative Art Conference (GA2003)*. Milan: Generative Design Lab.

[4] Klütsch, C. (2007). *Computer Grafik. Ästhetische Experimente zwischen zwei Kulturen. Die Anfänge der Computerkunst in den 1960er Jahren*. Vienna: Springer.

[5] Gruber, F. J., Hartmann, N., Lorenz T., Simmel C. (2008). *Strøem [exhibition]*, in *3rd International Design Cinema Conference*. Istanbul: ITU Taskisla.

Gruber, F. J., Hartmann, N., Lorenz T., Simmel C. (2009). *Strøem [exhibition]*, in *5th International Symposium on Computational Aesthetics in Graphics, Visualization, and Imaging*. Victoria, Canada: CAe.

Gruber, F. J., Hartmann, N., Lorenz, T. (2010). *Infection [exhibition]*, in *13th International Conference on Generative Art (GA2010)*. Milan: Generative Design Lab.

[6] 'ambientartlab' is a Vienna based collective of artists and designers from media arts, architecture, and experimental music. For detailed project documentations see http://www.ambientartlab.com.

[7] Damianisch, A. (2009). *Program for Arts-based Research (PEEK)*, Vienna: FWF.

[8] Schmidt-Wulffen, S. (2007). *Research in the Arts in Europe*, in *die bildende. die zeitung der akademie*. Nr. 02 (June 2007). Vienna: Akademie der Bildenden Künste.

[9] Schulze, H. (2000). *The Aleatoric Game*, Munich: Fink.

[10] Trogemann, G., Viehoff, J. (2005). *Code@Art*. Vienna: Springer.

[11] Giannetti, C. (2004). *Cybernetic Aesthetics and Communication.* Media Art Net: http://www.mediaartnet.org/themes/aesthetics_of_the_digital/cybernetic_aesthetics/.

[12] Foerster, H. von (1983). *Observing Systems*. Stanford, California: Intersystems.

[13] Galanter, P. (2010). *Complexity, Neuroaesthetics, and Computational Aesthetic Evaluation*, in *13th International Conference on Generative Art (GA2010)*. Milan: Generative Design Lab.

[14] Essl, K. (2007). *Algorithmic Composition*, in *Cambridge Companion to Electronic Music*. Cambridge: Cambridge University Press.

[15] Bohnacker, H., Groß, B., Laub, J., Lazzeroni, C. (2009). *Generative Gestaltung. Designing, Programming, Visualization with processing*. Mainz: Verlag Hermann Schmidt.

[16] Schulze, H. (2005). *Heuristik. Theorie der intentionalen Werkgenese*, Bielefeld: Transcript.

[17] Laurel, B. (1991). *Computers as Theatre*. Reading, MA: Addison-Wesley Publishing.

[18] Blevis, E., Lim, Y., Ozakca, M., Aneja, S. (2005). *Designing interactivity for the specific context of designerly collaborations*, in *Proceedings of CHI '05 extended abstracts on Human factors in computing systems*. New York: ACM

[19] Cohn, M. (2005). *Agile Estimating and Planning*. Prentice Hall.

[20] Beck, K., Beedle, M., Bennekum, A. V., Cockburn, A., Cunningham et al. (2001). *Manifesto for Agile Software Development*, http://www.agilemanifesto.org/.

[21] IDEO (2003). *IDEO method cards: 51 ways to inspire design*. http://www.ideo.com/work/method-cards/.