From the Poesy of Programming to Research as an Art Form

Laurent Mignonneau / Christa Sommerer

Conceptual Background

Since the early 90s we have been developing and programming interactive computer systems that consider the users' interaction input important components for self-generating software structures that are not predefined by us artists but instead constantly change, grow, develop, evolve and adapt to the environment. One of our main goals in creating these systems is to show how interaction is a key component to the creation of complexity, diversity and emergence, not only in real-life, but in artificial systems as well.
We have applied the principles of complex adaptive systems and artificial life to the creation of interactive software structures that constantly change when users interact with them. We have created dynamic interactive artworks that are not static, but instead process-based, open-ended, adaptable and environment-centered.
We call this "Art as a Living System,"[1] in reference to natural systems which are always dynamic, flexible and input-dependent. Before describing some of the research principles which we apply to the creating of these art works, let us consider a few basic questions on the role of programming, the function of the artist/programmer and the importance of research and development in our art works.

Research as an Art Form

For each of our interactive artworks we have developed custom-made software programs as well as special hardware interfaces that we specifically invented and designed for these systems.[2]
We concentrate a significant amount of energy on finding novel interfaces as well as novel programming algorithms for novel artistic concepts. Our artistic activity has become a research activity and the artworks we create have become research projects that expand and explore the status quo of what is known and commercially available.
One of our main motivations for writing our own software programs and developing our own hardware interfaces as opposed to using off-the-shelf software packages or commercially available interface technology is our wish to develop systems that investigate new questions and explore novel technical, conceptual and artistic approaches.
As media artists we have chosen to become artists/researchers or researcher/artists who define and shape new questions of creation, and set out to explore the forefront of creativity and digital technology, investigating the basic question of creation, invention and discovery.

Beyond End-user Art and the Art of Discovery

Having worked for around 10 years at advanced research centers in Germany, the US and Japan,[3,4,5] we have witnessed the usual time lag of a few years before a novel research prototype actually becomes part of the software or hardware market. This means, avail-

able software and hardware products usually lag a few years behind what is already scientifically and technically explored.
Working as an artist in research and development thus not only provides insight into today's visions and inventions which might shape the future, it also enables you to define new artistic research topics that might also shape the future of art, design, product and society as a whole.
While it is of course perfectly legitimate to be an "end-user artist" who develops her artwork through the currently available software or hardware packages, there is however always a certain kind of creative limitation and aesthetic resemblance of works produced with these packages.
The freedom to design your own software and create your own hardware could be compared to mixing your own colors out of pigments as opposed to using a paint-set with a predetermined number of premixed colors.
It is experimenting with details as well as sometimes accidentally discovering novel features that make programming itself a highly creative experience. During the programming process, essential new discoveries can happen by chance or even by misconception. The programming language itself has been designed to be used in a certain way (the grammar and the words, the syntax) but the content of what this language describes has not been determined, thus leaving a great deal of freedom to the programmer to use this language. Programming is a constant discovery, and asking others to create a program on your behalf means leaving out many creative details that can be essential to the final look and even to the final content of the work.

The Poesy of Programming

Programming can be compared to writing a novel: even though the language of the novel is defined (say French or German or English), the content of what is expressed is subject to the author's imagination and creative expression.
The same holds true for the art of programming: programmers each have their own style in writing programming code, and the result usually depends upon their skill and their experience. Especially in the area of interface programming, which can be quite complex as it needs to consider user input, there are great differences between programming styles and the personal creativity of the programmer.
Going back to the metaphor of the novel, you could for example imagine asking two writers to produce a novel on the same topic, using the same language. The resulting two novels would certainly differ greatly, even though both authors used the same language and perhaps even the same words. One of the novels might be more interesting than the other, the difference being how the authors managed to convey their flights of fancy and ideas.
It is the full control over a language combined with a complete openness towards discoveries and experiments that help an author to produce an outcome (be it a computer program or a novel) which expresses and transcends her creative vision.

The Role of the Code in our Artworks

Let us now look at the function of code in our own artworks.
One of our main goals in creating interactive systems is to show how dynamic and complex life is. Or as Dennett described it, "William Paley was right about one thing: our need to explain how it can be that the universe contains many wonderful designed things."[6]
Deeply fascinated by the workings of nature, we created several artificial systems that reflect on real life systems by re-creating and interpreting some of its beauty and complex-

ity of design. From analyzing some principles of nature, such as the emergence of life, the emergence of order and complexity as well as the dynamics of interactions, we were inspired to create artificial systems that model some of these processes.
To do this, we needed to create software structures which themselves are dynamic and open-ended. Over the years we have been instrumental in developing and establishing a research field called Artificial Life Art, Generative Art and Art that creates Complex Adaptive Systems through Interactivity.[7]
Let us briefly summarize some of the underlying research fields.

Complex Dynamic Systems

Complex System Sciences studies how parts of a system give rise to the collective behaviors of the system and how the system interacts with its environment. Social systems made up, in part, of people, the brain made up of neurons, molecules made up of atoms, and the weather made up of air currents are all examples of complex systems.
Complex System Sciences, as a field of research, has emerged in the past decade. By searching for inherent structures in living systems and trying to define common patterns within these structures, it approaches the question of how life on earth could have appeared. A whole branch of research—not only within biology but also across its borders to physics and computer science—deals with complex dynamic systems and can be seen as the attempt to find basic organizing principles.
Complex Systems Sciences focuses on certain questions about parts, wholes and relationships. Efforts to describe and define the notions of complexity have been made by many scholars including Aschby,[8] Baas,[9] Bennett,[10] Cariani,[11] Casti,[12] Chaitin,[13] Jantsch,[14] Kauffman,[15] Landauer,[16] Langton,[17] Pagels,[18] Wicken,[19] Wolfram[20] and Yates.[21]
Although there is no exact definition of what a Complex System is, there is now an understanding that when a set of evolving autonomous particles or agents interact, the resulting global system displays emergent collective properties, evolution, and critical behavior that exhibits universal characteristics.
Such a system is fundamentally novel and not deducible into its mere parts. These agents or particles may be complex molecules, cells, living organisms, animal groups, human societies, industrial firms, competing technologies, etc. All of them are aggregates of matter, energy, and information that display the following characteristics. They

- couple with each other
- learn, adapt and organize
- mutate and evolve
- expand their diversity
- react to their neighbors and to external control
- explore their options
- replicate
- organize a hierarchy of higher-order structures

Emergence

In the study of complex systems, the idea of emergence is used to indicate the arising of patterns, structures, or properties that do not seem adequately explained by referring only to the system's pre-existing components and their interaction. Emergence becomes increasingly important as an explanatory construct when the system is characterized by the following features:

- when the organization of the system, i.e., its global order, appears to be more salient and of a different kind than the components alone
- when the components can be replaced without an accompanying decommissioning of the whole system
- when the new global patterns or properties are radically novel compared to the pre-existing components; thus, the emergent patterns seem to be unpredictable and non-deducible from the components as well as irreducible to those components.

Interactivity

One of the central roles in the creation of complexity and emergence is interactivity. By coupling with each other and by exchanging salient information that in turn can trigger the creation of new information, interactivity can be described as a key principle in the organization and transformation of components within a complex dynamic system.

Dynamic Programming, Emergent Design and User Interaction

Intrigued by the idea that order, structure and design can emerge through the interaction of particles or agents in a system, we have explored (since 1992) complex dynamic systems that are open-ended, process-based, adaptable and environment centered.
From an artistic point of view we aim to create artworks that are like dynamical living systems themselves ("Art as a Living System"),[1] as they constantly change, adapt and diversify according to their environmental input parameters.
The idea of creating a dynamic and emergent software structure also requires a new programming approach: instead of asking the computer only to execute a given set of instructions, the code itself should re-organize itself "on-the-wing," while the dynamic input parameters are processed. Just as in complex dynamic systems, all components of the software code and all the input parameters from the interaction are coupled to each other: this leads to a system that adapts and organizes itself, mutates and evolves, expands its diversity, reacts to its neighbors and to external control, explores its options and replicate and finally organize a hierarchy of higher-order structures.
It's artistically interesting to see how creation in this process becomes an emergent property which can produce unexpected and novel results. Through the dynamic software structure and a linked user interaction, novel content can emerge and new forms of expressions can emerge. The final outcome is not so much a pre-determined "result" but instead a dynamic process of constant re-configuration and adaptation.
Figures 1 and 2 show the work *Life Spacies* created in 1997. Here language is used as genetic code to create artificial on-line creatures that live, mate, evolve, feed on text and die. Users can create these creatures by simply writing text and by feeding the creatures with text characters. In this work the idea of the code of language is used literally, as the genetic code for artificial life forms. An in-depth description of this system is provided in.[22] Some of the earlier generative artworks we have created since 1992 using dynamic and generative image processes are given in literature. [1, 23, 24, 25]

Generating and Interacting with Complexity on the Internet

The Internet is an ever-expanding database of images, text and sound files, currently containing several billion documents. These data and their internal organization are constantly

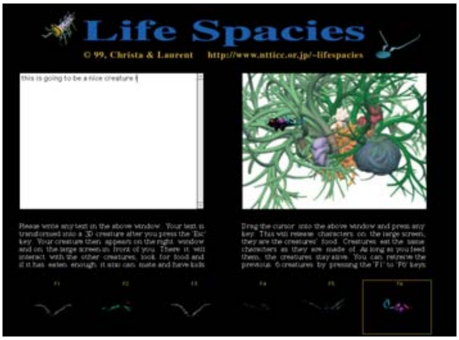

Fig. 1 *Life Spacies II*—graphical user interface. Written text is used as the genetic code and food for artificial life creatures.

Fig. 2 *Life Spacies II*—user as she creates and feeds various artificial creatures that mate, eat, die, interact and evolve, creating an open-ended, complex dynamic system.

changing, as new documents are being uploaded, new web sites created, and old links deleted. New connections between various sites are also constantly being built, and the Internet itself has basically become an evolving, re-connecting and reconfiguring network of user-driven data input and output. Since 1999 we have created various interactive systems that directly tap into this complexity and link it to multi-modal interaction experiences.

The first system we created is called *Riding the Net,* built in 1999.[26] Here users can use speech communication to retrieve images from the Internet, watch these images as they stream by, and interact with them by touching them. Two users can interact in this system simultaneously, and as they communicate, their conversation will be supported and visualized in real-time through images as well as sounds streamed from the Internet. Figure 3 shows an example of this interaction, shown at Siggraph 2001.

In 2001, we adapted the *Riding the Net* image retrieval software for an interactive information environment, called *The Living Room.* The system was developed for the "Bo01-Living in the Future" architecture exhibition held in Malmoe, Sweden, in May 2001. Users in this system enter a 6 x 6 meter space that consists of four 4 x 3 meter screens and as they talk, microphones placed on the ceiling of the space detect their conversations. Detected keywords are then used to generate word icons, which start to appear and float on the four screens. Users can touch any of these word icons and their touch will trigger the downloading of corresponding images from the Internet. Up to 30 users in the system can choose to touch the various word icons, which will generate constantly changing image and sound downloads from the Internet. As a result of these multi-user interactions, a dynamic, self-organizing, and constantly changing information space emerges. It represents the users' individual conversations, their individual interests in certain topics, and their collective interaction with the shared information.

Fig. 3 *Riding the Net*—multi-modal interaction with complex data on the Internet.

In May 2002, we adapted *The Living Room* software to the 3D immersive environment of

Fig. 4 shows two users as they interact with the *The Living Room* data environment.

Fig. 5 User as she interacts with the complex 3D data environment of *The Living Room* inside a CAVE™ system.

the CAVE™ system. In this system, called *The Living Web,* users can actually "enter the Internet" and interact with the available image and sound information in three dimensions. When users talk into their headset microphones, images that relate to their conversations are streamed from the Internet and displayed in 3D in such a way as to surround them. By grabbing one of the floating images, the user can retrieve more information about this specific image (for example its URL), place the icon in a 3D space for bookmarking, and sort the various selected icons as 3D bookmarks to create further links, main interests, and connections between the various selected topics.

As in the "Riding the Net" and *The Living Room* systems, the imprecision of the speech recognition system and the randomized choice of images from the various search results are used intentionally to create a dynamic system that is unpredictable, full of surprise, and compliant with some of the definitions of a complex system. While users have some control over what kind of image and sound downloads are triggered, the sheer quantity of available information makes straightforward selection impossible. For each keyword, typically several hundred or at times several thousand image and sound documents are available and users can typically only perceive a fraction of the available data. To manage this complex and constantly changing database of images and sounds and to allow intuitive as well as creative data browsing, these systems were designed to deal with randomness and order, allowing partly directed and partly undirected searches. [27] Again, like in the principles of complex systems, it is precisely this notion of order and randomness, predictability and surprise that make dynamic complex systems interesting, emergent and full of discoveries.

Transcending the Code

Writing computer programs is a complicated task, involving extended knowledge of the ever-changing programming languages and versions, their capacity and their inner structures. In addition, knowledge of the computer's internal hardware architecture as well as its resources and infrastructure can be of great advantage if you want to extend the already known and explored.

Knowing both hardware and software structures is important if you want to explore new forms of expressions and become less dependent on today's pre-design and the limitations of computer hardware and software.

It is extra freedom that in-depth knowledge provides, as you can change, modify and extend any part and use both hardware and software as flexible materials to express and shape your imagination and artistic vision.
Only when all the components of the materials are known, can you begin to transcend the actual technology and create outputs that go beyond the purely technical and materialistic.
As in biological systems where the phenotype differs greatly from the genotype, programming as an art form is not purely a question of the code for its own sake, but rather a question of how this code is expressed, how it is linked to other environmental influences and what it actually means.
Instead of focusing only on the technical details and getting lost in materialistic questions of the code, artists who work with this technology have to transcend the software code and hardware constraints and present us with intellectually as well as emotionally challenging ideas and questions.
The most difficult part in creating artworks with computers is thus not so much to acquire technical skills or to learn the software languages, but to evaluate and estimate the technical possibilities of software and hardware constraints, as well as to explore new technical and intellectual ideas by balancing their conceptual and technical capacity and value.
As with any mastery in creative fields of expression (may it be dance, theater, music, film or fashion—think of the dancers' mastery over their bodies as essential component for the final artwork of the dance performance), computer-dependent art requires a certain mastery over the material before it can express itself in a higher and more transcended form.
The quality of a media artist, then, lies in her sensitivity to create new visions and explore new tools and structures that support these visions and finally present us with content and experiences that transcend time and material by touching deeper emotional qualities that are not readily explained through code or numbers alone.

1 Sommerer, C. and Mignonneau, L. "Art as a Living System", in *Art @ Science,* C. Sommerer & L. Mignonneau, (Eds.). pp. 148-161. Springer Verlag, Wien / New York, 1998
2 Mignonneau, L. and Sommerer, C. "Designing Interfaces for Interactive Artworks," in *KES 2000 Knowledge Based Engineering Systems Conference Proceedings,* pp. 80-84, University of Brighton, UK,. 2000
3 ATR Advanced Telecommunications Research Center, Kyoto, Japan: http://www.atr.co.jp
4 Fraunhofer Gesellschaft, Bonn, Germany: http://www.fraunhofer.de
5 Beckmann Institute, NSCA National Center for Super Computing Applications, Champain / Urbana, IL, USA: http://www.beckman.uiuc.edu
6 Dennett, D. *Darwin's Dangerous Idea: Evolution and the Meanings of Life,* Simon and Schuster, New York,1995.
7 Sommerer, C. "A Life in Art, Design, Edutainment, Game and Research", in *LEONARDO Journal.* Issue 34. pp. 297-298. MIT Press, Cambridge / MA, August 2001,
8 Ashby, W. Ross. "Principles of the self-organizing system", in Heinz Von Foerster & George W. Zopf, eds: *Principles of Self-Organization,* pp. 255-278. Pergamon Press, Oxford,1962
9 Baas, N. A. "Emergence, Hierarchies, and Hyperstructures", in C. G. Langton, ed., *Alife III, Santa Fe Studies in the Sciences of Complexity, Proc. Volume XVII,* pp. 515-537, Addison-Wesley, Redwood City, 1994
10 Bennett, CH. "Logical depth and physical complexity", in Rolf Herken, ed.: *The Universal Turing Machine,* pp. 227–257. Oxford University Press, Oxford, 1988
11 Cariani, P. "Emergence and Artificial Life", In Christopher G. Langton, Charles Taylor, J. Doyne Farmer and Steen Rasmussen, eds.: *Artificial Life II. Santa Fe Institute Studies in the Sciences of Complexity, Proc. Vol. X,* pp. 775-797, Addison-Wesley, Redwood City, Calif., 1992
12 Casti, J.L. *Complexification.* Abacus, London 1994.
13 Chaitin, G.J. *Information Theoretic Incompleteness.* World Scientific, Singapore, 1992
14 Jantsch, Erich. *The Self-Organizing Universe.* Pergamon. Oxford and New York, 1980

15 Kauffman, St. *The Origins of Order. Self-organization and Selection in Evolution.* Oxford University Press, Oxford, 1993
16 Landauer, R. “A simple measure of complexity”, in *Nature* 336, pp. 306–307. 1988
17 Langton, C. 1989. “Artificial Life”, in C. Langton, ed., *Artificial Life.* pp. 1–47. Addison-Wesley, Redwood City
18 Pagels, H. *The Dreams of Reason.* Simon & Shuster (Bantam ed. 1989 N.Y.),1988
19 Wicken, J. S. *Evolution, Thermodynamics, and Information.* Oxford University Press, Oxford, 1987
20 Wolfram, S. “Cellular automata as models of complexity”, in *Nature* 311. pp. 419–424. 1984.
21 Yates, F.E., ed. 1987. *Self-Organizing Systems. The Emergence of Order.* Plenum Press, New York, 1987
22 Sommerer, C. and Mignonneau, L. and Lopez-Gulliver, R. “LIFE SPACIES II: from text to form on the Internet using language as genetic code”, in *Proceedings ICAT’99 9th International Conference on Artificial Reality and Tele-Existence.* pp. 215-220. (Tokyo: Virtual Reality Society), 1999e
23 Sommerer, C. and Mignonneau, *“Interacting with Artificial Life: A-Volve”,* in *Complexity Journal.* Vol. 2, No. 6. pp. 13-21. Wiley, New York, 1997
24 Sommerer, C. and Mignonneau, L. “The application of artificial life to interactive computer installations”, in *Artificial Life and Robotics Journal. Vol.2, No.4.* pp. 151–156. Springer Verlag, Tokyo, 1998
25 Sommerer, C. and Mignonneau, L. *“Modeling Emergence of Complexity: the Application of Complex System and Origin of Life Theory to Interactive Art on the Internet”,* in *Artificial Life VII,* edited by M.A. Bedau, J.S. McCaskill, N. H. Packard, and St. Rasmussen. pp. 547-554. MIT Press, Boston, 2000
26 Mignonneau, L., Sommerer, C., Lopez-Gulliver, R. and Jones, S. “Riding the Net: a Novel, Intuitive and Entertaining Tool to Browse the Internet”, in *SCI 2001—5th World Multiconference on Systemics, Cybernetics and Informatics Conference Proceedings.* pp. 57–63. Orlando, Florida, International Institute of Informatics and Systemics, 2001
27 R. Lopez-Gulliver, C. Sommerer, and Mignonneau L. “Interfacing the Web: Multi-modal and Immersive Interaction with the Internet”, in *VSMM2002 Proceedings of the Eight International Conference on Virtual Systems and MultiMedia.* pp. 753-764. Gyeongju, Korea, 2002