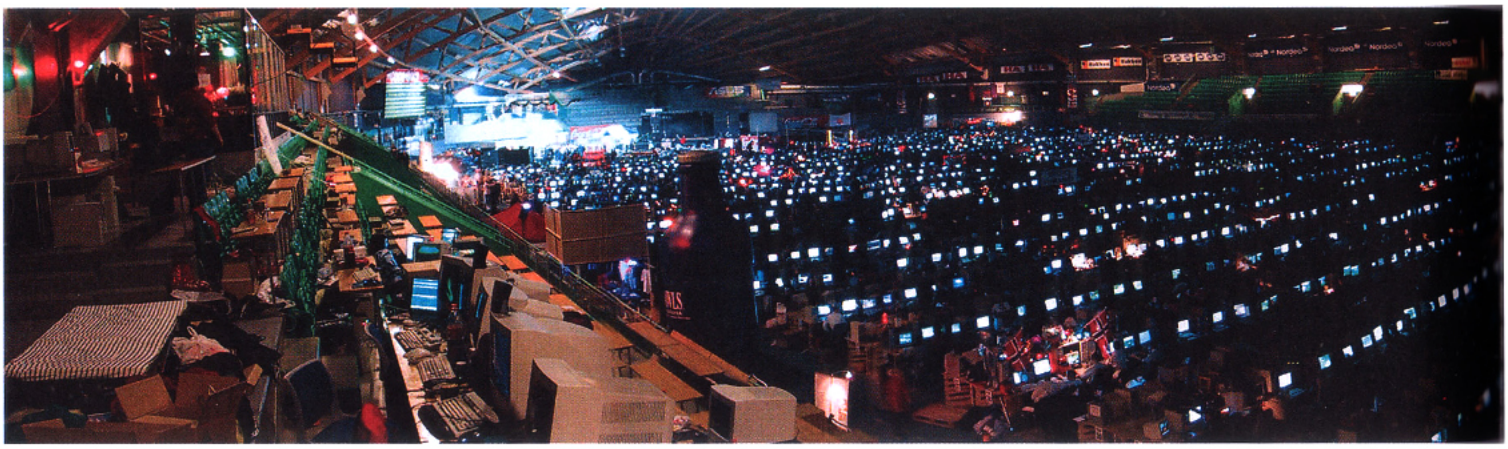
The writer is an artist, curator, organizer, and a founding member and chairman of katastro.fi, based in Helsinki. *Language editing Tomi Snellman.*

Below: *The Gathering*, 2002. The Vikingskipet Olympic Arena, Hamar, Norway, 27.3.–31.3. 2002. Photo by Espen Lyngaas. © *The Gathering.*



Juha Huuskonen

The Art of Defining Software Culture:

"If you are using software made by someone else, you are living in that person's dream"(1). The limitations of available software tools often set the limits of how we use current technologies, or even what possible uses we can imagine for them. Open source community, media activists, VJ community, demo scene and many other global movements have broken free from these limitations by actively developing new software concepts and tools.

The Read_me festival for software cultures is bringing together many of these communities. This is not an easy task, since even though all of them are experimenting and developing software, they have radically different motivations for their activities. The festival organisers have developed methods which allow these communities to be involved in defining the content of the festival and meaning of the term 'software cultures'.

The Hot Stuff

At first glance, the communities experimenting with software development can seem very different from the world of media art. But if one makes the effort to learn the basic jargon, similarities with the art world start to emerge. Exhibitions are replaced by festivals or laptop jams, catalogues by websites and discussions forums, curators by competitions and online polls.

A good example of a community with a sophisticated self-organised structure is 'demo scene', a movement that developed around the early home computers (Commodore, Atari, early PCs) in the late 80's. The main activity of the demo scene is to create 'demos', real-time software shows with graphics, music and visual software effects. A demo is the collective effort of a group, and the goal is to beat the other groups by coming up with original demo concepts and maintaining high quality in design. The collaborative work process to produce a major demo can often take months or even years to complete. Various charts (such as Eurochart) follow the demo scene on a longer time span, allowing people to vote for their all-time favourite demos, groups and individual creators. The most respected groups form an 'elite', at the bottom of the hierarchy are beginners with only few skills.

The original system for distributing work within the demo scene tells about the level of commitment and persistence of the sceners. On the darker side of the demo scene, 'crackers' were making games and commercial software available for the community by breaking their copy protections. Pirate copies of software and demos were spread by 'swappers' and 'BBS hosts'. The swappers would frequently send out disks by regular mail to dozens of contacts around the world. The disks would contain the latest piece of software they had in their hands. Within a week or two, the contacts would return the disks, with copies of the 'hottest stuff' they had themselves. This relatively random system functioned as a surprisingly efficient data distribution network long before the days of the Internet.

The most important arenas for evaluating demos are 'parties', events which bring together hundreds or even thousands of sceners. People show up with their computers and sleeping bags for a weekend of heavy Coke drinking, pizza eating and competing with demos. Some of the oldest and well-known parties are *Assembly* in Finland, *The Party* in Denmark and *The Gathering* in Norway. Similar events are also regularly organised by the VJ community, media activists and hackers. For the VJs, there are festivals and laptop jams such as *AVIT* and *SHARE*, hackers and activists come together in events like *HAL 2001* (Hackers At Large 2001) and *CCC* (Chaos Computer Camp). These events are important for the social ties within the network, and they offer an opportunity for people to get feedback and recognition for their work.

Today the demo scene still thrives by integrating itself to other subcultures. The demo scene parties are now also known as LAN parties, events where large groups of people join the same local area network to play games in teams. After getting bored with playing games, some of these people will eventually join the ranks of sceners and start experimenting with creating graphics, music or software.

A Jungle of Software

The VJ scene has attracted thousands of new adherents in the past few years, ranging from visual artists to filmmakers and clubbers. The sudden development has a lot to do with the fact that home computers finally have the processing power to properly handle real-time video processing. The situation is similar to the early days of home VCRs, when artists such as David Rokeby, Myron Krueger and Erkki Kurenniemi created interactive audiovisual installations and performances with their custom hardware instruments. Now the standard tool of the VJ has changed from a video mixer to a laptop and custom hardware has been replaced by custom soft-

The Benevolent Dictators of the Read_me Festival

ware. Around one hundred new tools for VJs have been released during the past couple of years, mostly made by VJs themselves.

In a similar way, media activists are actively developing tools for their own use. One impressive example is the indymedia network, a global network of organisations which are all hosting local versions of an early self-publishing software. This software was an answer to the urge of the activists to create a channel for uncensored and rapid commentary on current events. It was a pioneering tool in an area that has now grown into a vast universe of weblogs. Indymedia organisations are still active today and have an important role as alternative news resources.

Sceners, VJs, hackers and activists who are involved in software development often openly borrow and steal ideas from others. This is possible since, because there are few if any financial interests involved, one does not have to be afraid of lawsuits if one takes an idea and tries to develop it further. Quite the contrary, this is one of the basic principles of collective development. One can start learning by imitating what others are doing, and only after a certain time of practising is one expected to bring in something new. The fact that software developers are just like any other people in the community makes it easier for others to contribute ideas or even start developing software on their own.

Each software tool presents a different set of concepts and working methods, based on different visions of what the community is about. The constant development of software keeps the community on its toes, constantly questioning its values and goals.

The Invisible Barriers around Self-organised Communities

Sceners, VJs, hackers and activists have built their own alternative universes outside the traditional disciplines of art and science. This creates a strong empowering effect, the community itself is seen to be in charge and there is space for creativity and experimentation. At the same time, this disconnection can be very limiting for the longer term development of the community.

Software cultures bring together people of all ages, nationalities and educational or professional backgrounds. One can imagine that this would make the communities very open-minded and tolerant of ideas coming from other contexts. The reality is different – the fact that the members of the community do not share the same background makes it difficult for the community to step outside its barriers. The discussions within the community tend to focus on the community itself simply because that is the only common subject to talk about.

Another factor contributing to the closed nature of the communities is the widely spread practice of using aliases. It is impossible to guess the real identities of the demo sceners Yolk, Tsunami and Carebear or the VJs Hello World and Bionicpope. This makes the community more democratic – it is easier for anyone to slip from the role of a passive observer to that of an active contributor – but it also increases the community's distance from the outside world.

It can also be said that the sheer volume and chaotic nature of the communities make it impossible to develop them in a certain direction. The phenomenon can be described as 'mass-amateurization', to borrow a term that has been used to compare blogging with professional journalism. Mainstream media have to compete with thousands of alternative versions of the same story, written by people with varying motivations and levels of expertise. This makes it more difficult

[1] A quote from John Maeda, from Golan Levin's presentation at Ars Electronica 2003.

[2] A quote from Judith Butler. Mika Hannula, Kolmas tila. Väärin ymmärtäminen eettisenä lähtökohtana. Kuvataideakatemia, 2001, page 19.

First *Placard Event* in Finland, 22.5.-23.5.2004, Fred's apartment, Meritullinkatu in Helsinki. Photo by Juha Huuskonen.

PixelACHE festival in New York, 31.5.–3.6. 2003, Gershwin Hotel, SHARE event at Open Air club, Galapagos art space. Photo by Antti Ahonen. © Antti Ahonen and Piknik Frequency ry.

Links: www.scene.org (Demo scene community site); www.vjcentral.com (VJ community website): www.runme.org (Runme software art repository and Read_Me software art festival); www.katastro.fi (Electronic art and subcultures network)

to define what should be considered expertise. Having an understanding of a special subject can be more valuable than knowledge of the principles of journalism. Instead of understanding the big picture, one should have a good understanding of the interests of a certain small community.

The closed nature of software cultures can also prevent fruitful exchange of knowledge from taking place. For example, some of the new VJ tools could be useful for people working with cinema or theatre. And vice versa, the VJs could definitely make use of lessons learned in the history of cinematography and story-telling. Unfortunately there are currently only a few projects trying to make such crossover communication happen.

The Benevolent Dictators of the *Read_Me Festival*

One successful example of how these communities can be brought together is the *Read_me* festival. Maybe we could learn something from the way it evolved into its current format.

When Alexei Shulgin (one of the *Read_Me* festival initiators and curators) gave a presentation at the *D.I.N.A* festival in Bologna, he focused on proving there is nothing new in what Netochka Nezvanova (creator of the Nato.0+55 video performance software) is doing. His proof was based on his own definition of net art a few years back. The presentation was especially surprising, because the next presenter on the programme was Netochka herself, or rather one of her alter egos. Netochka Nezvanova (which can be roughly translated as "nameless nobody") is a character created by the programmers of the Nato software, a young aggressive female who is actively spamming several popular discussion forums using her own language (appl!ng 0+1 kaoz teor!e 2 0+1 evakuaz!on rout!n).

The *Read_Me 1.2* festival in Moscow continued in a similar vein, with curators coming up with definitions for software art and selecting works for the festival on the basis of the definition. Fortunately, a sharp u-turn was made when the festival was organised again the following year in Helsinki. An online software art repository (www.runme.org) was created, where categories and definitions are continuously evolving based on the submissions. Currently there is a substantial amount of work in several categories (audiovisual artistic tools, for example) which were originally excluded from the festival. The term software art has been replaced by software culture, and an online tool opens the process of defining the meaning of the term to the communities themselves. The power is in the hands of those who actually contribute something.

The curators of the project still have a chance to leave out work, and they often do (for example, so far all submissions from the demo scene have been rejected). To borrow a term from the open source community, the curators of *Runme* are the 'benevolent dictators' of the project. In theory they have absolute power to decide what happens within the project, but the only way for them to succeed is to keep the community happy. If enough resistance builds up, the project will split into two camps (referred to as 'forks', to borrow another term from open source software development), which will continue in different directions. Should such a situation occur, it would actually be a sign of the success of the *Runme* website and the *Read_Me* festival. It would show that they are important for the creative community, not merely an exhibition showing work to curious tourists.

"We should all try to affect the meaning of the terms which are important for us."(2). A year ago I registered the domain www.dontrunme.org, just in case :)