Chapter 8

Softer side of art

Maciej Wisniewski

Around the middle of the 1990s, I recall posing the following questions: Do ways of communication change how we view others and ourselves? To what degree is our perception of the world dependent on technology, and specifically information technology?

Communication technologies shrink distances and expand the notion of time. In a typical communication situation the greater the distance between people, the higher the likelihood of technological intervention or manipulation. Having a conversation with a person from across the room generally doesn't require any technology. A video call with a space station requires a fair amount of technological infrastructure. A film clip from a movie situated in prehistoric times often demands extensive technological manipulation.

Recently I began to calculate the amount of information I receive daily that is software mediated, versus the amount of information I perceive through direct interaction with the outside world. Strikingly this process revealed that my view of the world outside daily routine is almost entirely technology dependent – being somewhere between 65% and 85% reliant on software. When considering the far-reaching consequences that arise in changes in human communication caused by new technologies, it does not seem unreasonable to state that artists should be involved in the early stages of their development. Further, if these technologies and new forms of communication do inform our views of the world then isn't it only logical that HTTP and TCP/IP[1] are equally as important for artists' studies as Baudrillard and Foucault?

Visual artist as programmer paradox

Focusing on software as a medium in visual arts might at first seem a bit awkward. By its very nature software is invisible and is not really a medium in the traditional sense of the word, as the boundaries between tools and content are blurred, i.e. software creates both tools and content, and in most cases they become interchangeable. Traditional media such as radio, television, photography and print rely on the separation of tools and content. A photograph cannot alter itself, but a software program is capable of reproducing, messaging and replicating itself repeatedly.

Imagine a piece of software that connects to a camera mounted on a top of a building. It takes a snapshot of the weather, manipulates it so that it appears to be raining and then sends it to a local weather station. Let's further speculate that the software also copies itself and sends itself to any other computer on the network connected to

a camera taking pictures of the weather. At first, it appears as if the program is being used as a tool in a traditional sense. However when the software starts replicating and repeatedly sending rainy pictures across the network is it a tool, content, or both? Let's for a moment compare the interface, the visual manifestation of this program, with the functioning of its invisible processes of replication, messaging and image manipulation. Without knowledge of these hidden operations our interpretation of the produced images would almost certainly lead us to believe that it has been raining. As I will argue in the following pages, building transparency and openness into software systems is crucial to an understanding of our mediated 'reality'.

Machine programming versus network programming

Programming and designing interfaces are only the tip of the iceberg. Designing and programming communication is the crux of any network-based software. For Julian Stallabrass:

> Internet art is caught in a productive paradox similar to that which sustained modernist photography: the more photographers strove to make an autonomous art that used purely photographic means, the more they immersed themselves in the subject in front of the lens. Likewise, the further online art moves into Net-specific realms, the more it is bound up with the instrumentality of the Net itself.[2]

Roughly 40 years ago programs ran on a single machine which multiple users could log onto to perform different tasks. When computers became networked, many machines could be connected to a single network and multitudes of networks were soon formed. Contemporary networked computing requires software capable of handling complex relationships between various machines on a given network, however what is little understood is the manner in which these software interfaces define the personal relationships of the people behind them. A process that inserts itself so intimately into the day-to-day activities of users, demands an ethical position in which network programming becomes secondary to the social interaction that it facilitates and enables.

Most of the software processes that structure our interactions are not visible and they do not manifest themselves on the surface. There is a need for us, as users and programmers, to know what is really happening with the information that is provided and processed. We might ask how and where it is being stored, and how is it being delivered and protected? We may also reflect on what this vast amount of collected data says about us in general and cultural terms, and most importantly who has access to it?

> Cultural anthropologists contend that humanity cannot be understood *except* as a species embroiled in a unique way of life totally dependent on the exchange of social information. Culture, they say, is something that surrounds us like the air we breathe, determining the ways in which we interpret our every experience.[3]

The way in which we define ourselves and view our world is directly connected to the social impact of these networks on society, politics and government. In Lev Manovich's words:

> Because new media is created on computers, distributed via computers, and stored and archived on computers, the logic of a computer can be expected to significantly influence the traditional cultural logic of media; that is, we may expect that the computer layer will affect the cultural layer.[4]

The social and political implications of software- and communication-based technologies are almost limitless. In Peter Weibel's preface to the exhibition catalogue for *Future Cinema* he speaks optimistically about their impact on society at large, 'this book offers evidence of a surprising fact: Even the technological and ideological apparatus of huge industries can be transformed by individuals'.[5] Software creates Weltanschauung.

Postal service technology ('snail mail'), has provided us with a different view of the world by making the geographical world smaller. Similarly, network-based technologies allow us, almost in real time, to connect shared experiences – we live in simultaneous technologically maintained worlds. Network-based electronic technologies have collapsed distances and in many cases taken away the real need to travel.

Communication software is by default an invasion of privacy. But questions still remain, how much privacy do we need to give up and how much are we willing to give up; does communication software imply a different notion of community; does it suggest a different notion of government?

> The Internet is out of control: It always has been. It threatens secrecy. It allows secrecy. It threatens privacy. It enables privacy. It links. It fractures. It allows sexual predators to predate and evade. It lets law enforcement track and trap sexual predators. It conveys offensive material. It allows a free space for people to explore the limits of their tolerance. It forces the use of English. It allows Chinese speakers to share music, stories, and news all over the world. There is a reason for this lack of control. The Internet works so well as an example of a very distributed information system because it was built without 'controls' but with 'protocols'.[6]

These 'protocols' define large parts of our culture today. Shaping these protocols is an important part of contemporary art practice and has been the principle guiding the development of the artworks discussed in the following sections.

netomat

In a publication from 2003, Edward Tufte figuratively discussed the ill effects created by relying on Power Point software by focusing on the concept of the 'un-education' or 'un-learning processes' induced by frequent use of it.[7] For Tufte, robotic expectations are created when we allow software to do our thinking for us. Similarly, around 1996, knowing that existing Web browsers limited our experience of the World Wide Web through their reliance on print metaphors to structure information, I began to develop ideas for an alternative browser(s) that would allow different ways to perceive and interact with information on the Internet.

The original *netomat* (1999) software proposed a way to explore the Internet that took a flexible and free-form multimedia approach to accessing and presenting information

Figure 8.1 netomat.

on the network. When first shown in 1999, viewers could search the Web for multimedia content which was then presented to the viewer as a floating stream of text, pictures and sounds. The viewer could navigate this material by changing the direction and speed of the stream giving a very different experience to that offered by the standard format of retrieval and presentation of data (Figures 8.1, 8.2 and 8.3).

The idea behind the original project was to design an audio-visual language specifically to explore the 'unexplored' Internet by revealing how the ever-expanding network interprets and re-interprets cultural concepts and themes.

In traditional browsers typing the word 'Microsoft' into the browser would result in either displaying Microsoft's official website or by providing a list of links to Microsoft-related sites. Submitting the same query to *netomat* would result in a stream of text, pictures, animation and sounds from Microsoft's web sites and juxtapose them with images, text and sounds from other publishers. The resulting structure might contain official Microsoft material alongside caricatures of Bill Gates, sound bites, jokes about Microsoft and other commentaries.

The original *netomat* software allowed the user to define what part(s) of the network they would like to access, the type of data they would like to retrieve and how those data could be presented. A 'netomatic mark-up language', (nml) was developed specifically for the purpose.

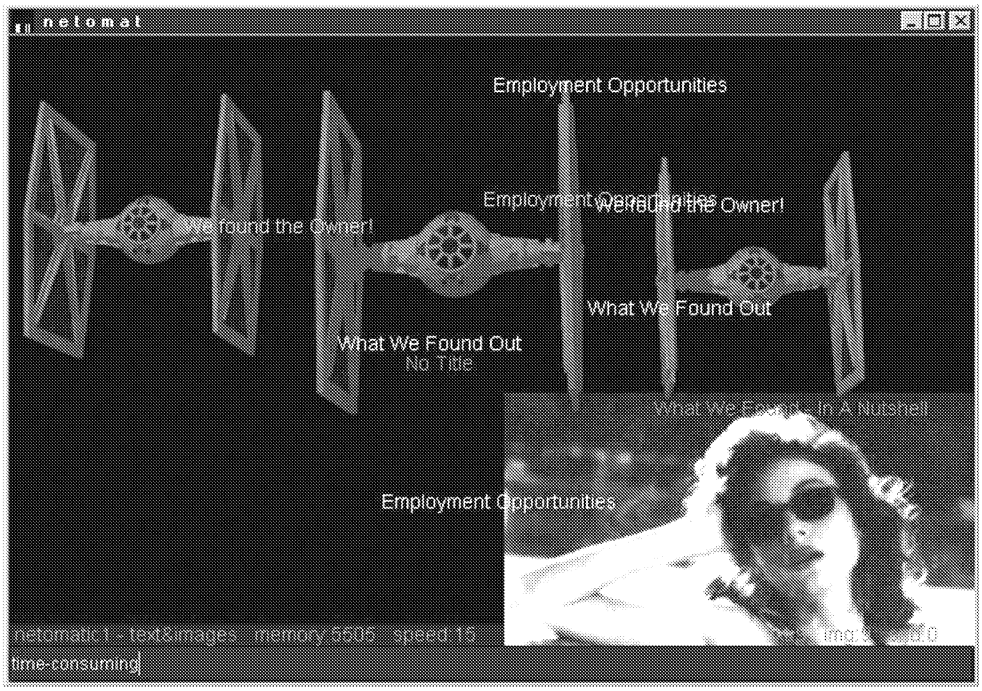

Figure 8.2 netomat.

The *netomat* software introduced both a new non-HTML-based Web but also worked seamlessly with existing Web content. The current iteration of the project is both a conceptual and technological extension of the original release and has evolved from a network 'viewer' into a network 'maker'. The most recent incarnation of *netomat* breaks down the barriers between the WWW and existing mobile phone networks. It allows the users to communicate freely between their mobile phones and their browsers and to create their own personal communication networks, again using nml.

These personal communication networks are user-driven, user-centric and user-managed. Unlike traditional networks such as Web (http network), email (smtp/pop/imap networks), IM (e.g. AIM, yahoo messenger, MSN, Jabber) and mobile phone networks (CDMA/TDMA) which are each separate incompatible silos of communication *netomat*'s networks focus on groups of individuals. From this perspective, each user has their own network of relationships that cuts across different domains of communication protocols and which can be fully controlled and administrated by the user. The strength of these networks lies in their ability to capture the diversity of a given social environment and serve the communication needs and the empowerment of individual users.

ScanLink

In 2000, a report from the National Academy of Science argued:

> For better or worse, we are inextricably interconnected and all must deal with the difficulties – and opportunities – that such interconnection brings.[8]

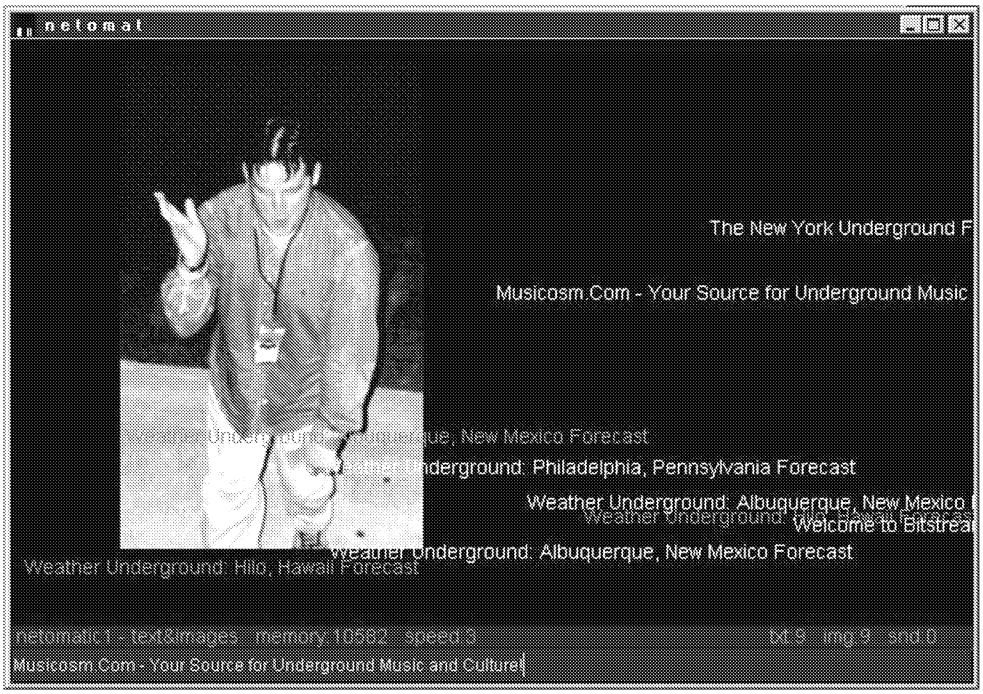

Figure 8.3 *netomat.*

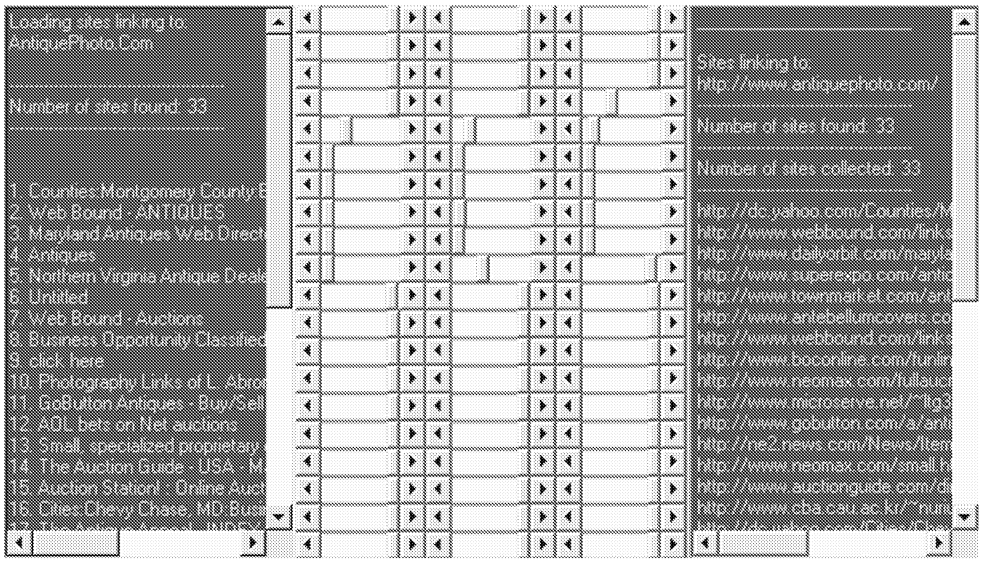

Figure 8.4 ScanLink.

Figure 8.5 *Netomatheque*, installation shot courtesy of the Postmasters Gallery.

A year or two before the original *netomat* was published, I began working on a project, which could be described as 'walking the Web backwards'. I wanted to see what a reverse traversal of links on any given web page would look like. In other words, I wanted to know who linked to whom and who was popular. At the time several major 'link hubs' such as Yahoo were already well established. However, these portals dealt with mainstream Web usage, and my interest leant towards the fringe. Using *ScanLink* (1998) the user could select a website and within seconds get a list of all sites that linked to it. The resultant underlying topology of connectedness of the WWW was not what I had expected, disconnectedness and isolation was very common.

Netomatheque

Installed in a faux living-room setting, with comfortable couch, fireplace and floor to ceiling projections, *Netomatheque* (2001) allowed the user to talk directly to the WWW using a regular phone, i.e. it re-formatted their encounter with the Net as a conversation.

Netomatheque consisted of custom-designed software that employed voice recognition, a text-to-speech engine and a search and display engine. The voice recognition engine translated the visitor's voice input into search queries, which then located the appropriate resources on the network. Once located *Netomatheque* (Figure 8.5) displayed information in the form of streaming sound, music, text and imagery floating through space. The projected walls changed before the viewer's eyes like living digital wallpaper.

Figure 8.6 *3 Seconds in the History of the Internet*, installation shot courtesy of the Postmasters Gallery.

Netomatheque employed associative semantics. The voice input triggered constantly evolving streams of information from the network's 'memory'. For Steve Dietz the project provides 'an alternative browsing experience [...] which will take you on a journey deep into the Internet's subconscious'.[9]

3 seconds in the memory of the internet

3 seconds in the memory of the internet (2002) premiered at the Postmasters Gallery in New York in January 2002. The project consisted of material gleaned from arbitrary time samples selected from the three different decades of the Internet's development (1980s, 1990s and 2000s). For a period of a month, a custom-designed 'spider' software agent crawled the Internet to retrieve material which was displayed on three simultaneous live feeds and rendered using a specially created image and text synthesizer, and animation engine (Figures 8.6 and 8.7). Found material was surprisingly rich and included: images, news, email and other communication messages, and log and error files. Even material from the 1980s – prior to the invention of the World Wide Web – proved extraordinary in depth of content and media.

The specific slices of time were organized linearly on their own walls, so that the viewer could walk through a 'time tunnel' consisting of the represented changes in the Internet's development.

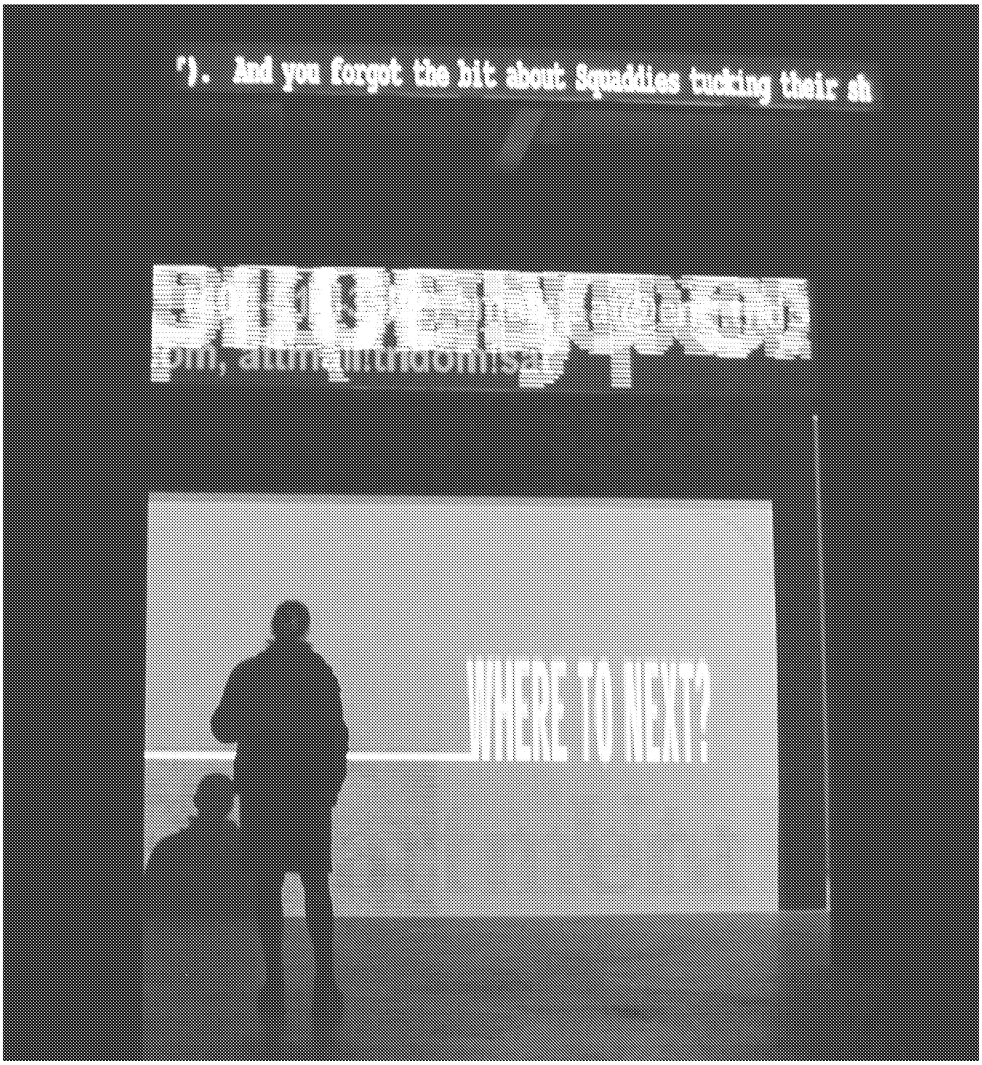

Figure 8.7 3 Seconds in the History of the Internet, installation shot courtesy of the Postmasters Gallery.

The retrieved traces reflected a subjective collective memory of the Internet's use and emphasized its scale, organic and social disposition and persistence as a cultural formation.

Instant messaging

Following these explorations, I became interested in making networks that could be created spontaneously, *ad hoc,* on top of already-existing networks and which were also capable of autonomous growth and retraction.

One of the first of these experiments, *Instant Messaging* (2002), consisted of a network of software-generated 'flies'. The software worked in a similar manner to any

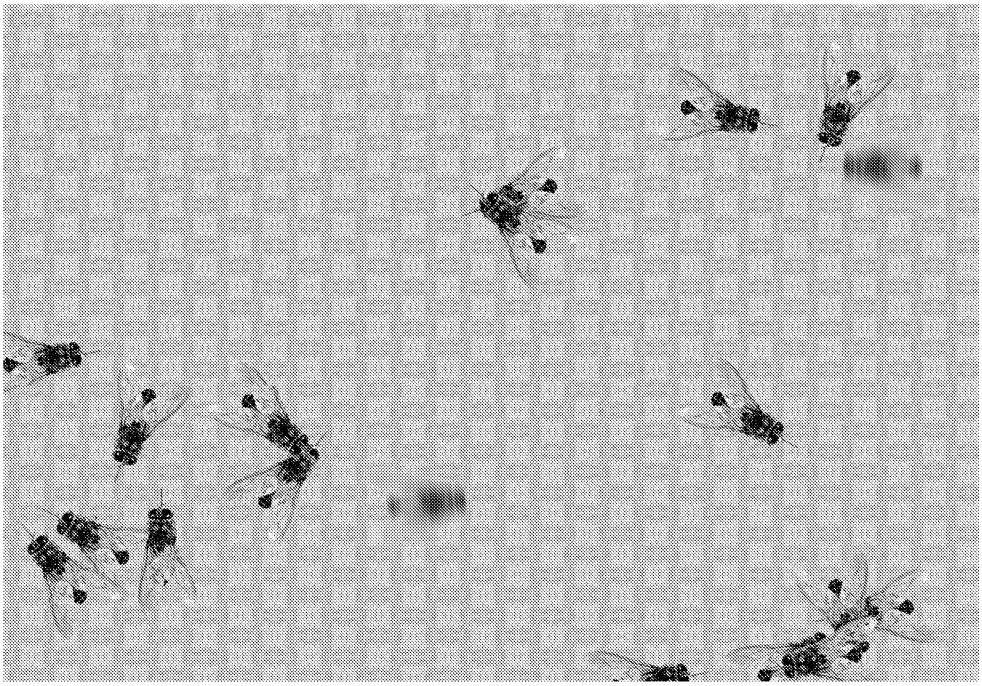

Figure 8.8 Instant Messaging.

real-time peer-to-peer messaging software – the 'flies' communicate, send and receive messages, seek each other out, mate, and travel back and forth between computers on a local network (Figure 8.8). Once the software has been installed, flies quickly propagate from computer to computer on any Local Area Network.

Instant places

The second project, *Instant Places* (2002), consisted of a software 'fiction' populated with inhabitants, that creates a network to connect dispersed data places. Once *Instant Places* has been installed, it connects to the Internet and begins to search for other installed versions of itself on the network. It makes no difference whether this is the computer next door or on the other side of the world, the routing time is virtually the same. Once located, the software allows its inhabitants to travel uninhibited place to place.

Instant Places (Figures 8.9 and 8.10) inhabitants communicate with each other over the network and consist of both predator and prey in the form of birds and mice. When the animals (whose population can dwindle or multiply) travel across networks, viewers are able to track their point of origin, their behaviour (fleeing, attacking, surveying, defending, etc.) and their precise longitude, latitude and altitude.

Instant Places cannot be influenced or altered by any factors outside of the software fiction itself. The narratives do not repeat but instead follow 'social' rules outlined by

Figure 8.9 Instant Places.

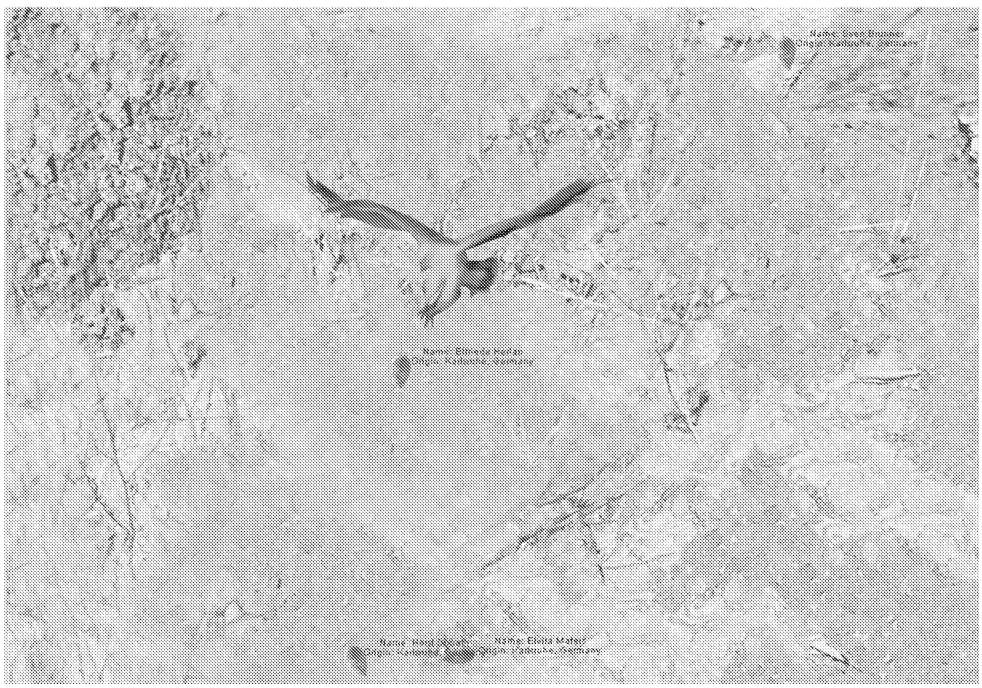

Figure 8.10 Instant Places.

the custom software. The inhabitant's actions are defined by real-time messaging. If there is a drop in the messaging, the system is programmed to generate activity so as to avoid lulls in the narrative.

Four stories with a twist

My experimentation with *ad hoc* networks led me to believe that the propagation of content through spontaneously interconnected nodes and resultant *ad hoc* connections provided a powerful alternative to existing channels of communication. However, an issue remained in how to extend these environments into the realm of human to human communication.

Four Stories with a Twist (2004) was my first attempt at combining messaging networks with a literary narrative. The project consists of an algorithmically generated 'music video' with an original score and script based on Albert Camus' *The Stranger*, a story set in Paris.

The project functions using a custom-designed software application that allows image and text capture for a mobile camera phone. This material is then re-mixed in real-time and programmatically animated by the musical rhythm of an original composition recorded using a synthesizer and two connected laptops.

Four Stories with a Twist was first shown at the *Villette Numérique* festival in Paris in 2004 (Figure 8.11). During the exhibition local participants equipped with mobile

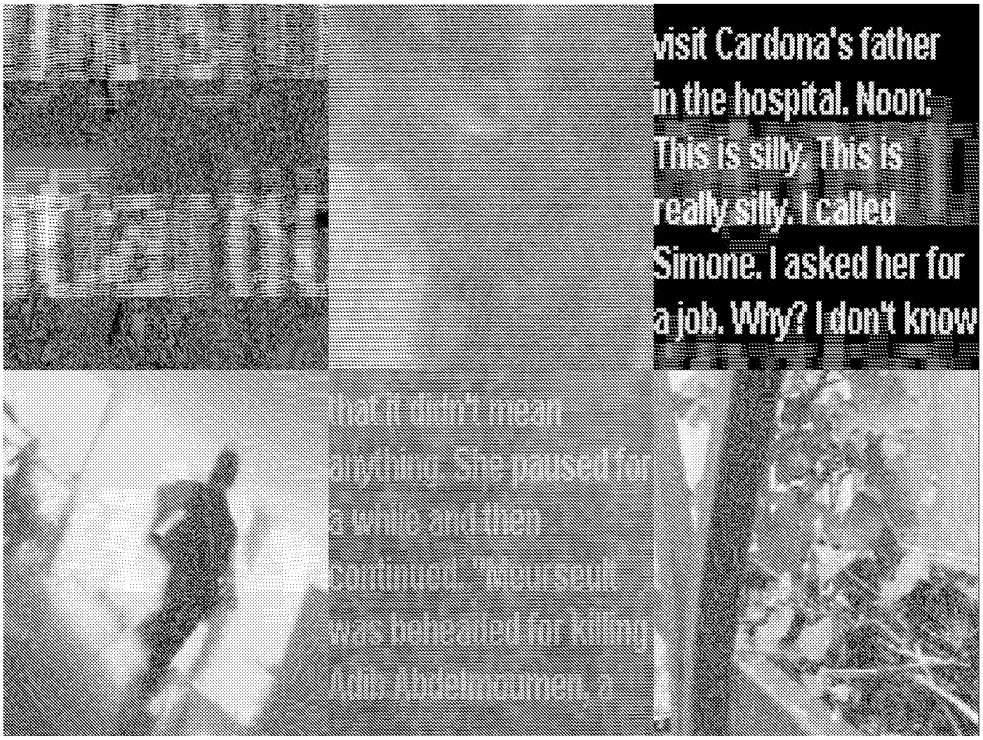

Figure 8.11 Four Stories with a Twist.

camera phones could send picture messages from around the city to the live generated 'music video' situated in the exhibition space. Throughout the period of the festival, a story would evolve based on the incoming messages, their volume, and the interconnectedness of the participants. The feedback loop established between the generated narrative and unpredictable nature of the incoming messages functioned to destabilize the original story, whose form increasingly fragmented.

Notes

1. TCIP is short for Transmission Control Protocol/Internet Protocol, and is a set of instructions used to connect computers on the Internet. HTTP stands for Hypertext Transfer Protocol, the communication protocol used by the World Wide Web that controls how information is formatted, accessed and transmitted.
2. J. Stallabrass, *Internet Art the Online Clash of Culture and Commerce*, London: Tate, p. 36, 2004.
3. R. Aunger, *The Electric Meme a Network Theory of How We Think*, New York: The Free Press, p. 46, 2002.
4. L. Manovich, *The Language of New Media*, Cambridge, MA: MIT Press, p. 46, 2001.
5. J. Shaw and P. Weibel (eds) *Future Cinema the Cinematic Imaginary after Film (Electronic Culture: History, Theory, and Practice)*, Cambridge, MA: ZKM, Center for Art and Media Karlsruhe, Germany and MIT Press, p. 16, 2003.
6. S. Vaidhyanathan, *The Anarchist in the Library: How the Clash Between Freedom and Control is Hacking the Real World and Crashing the System*, New York: Basic Books, p. 32, 2004.
7. E.R. Tufte, *The Cognitive Style of PowerPoint*, Chester, Connecticut: Graphic Press LLC, 2003.
8. *The Digital Dilemma Intellectual Property in the Information Age*, report produced by the National Academy of Science, Washington, D.C: National Academy Press, p. 58, 2000.
9. Steve Dietz used this phrase in describing *netomatheque* for the travelling Independent Curators International (ICI) exhibition, *Telematic Connections: The Virtual Embrace* (2001–2002).