map_I.T.

// campus digital de l'Ecole Polytechnique Fédérale de Lausanne ----------//
// projet de master plan // fabric | ch -- www.fabric.ch -- // février 2003 //

DOC. C - ANNEXES

+++
.txt:
rhizoreality.mu / / paper ieee 2003 /
knowscape / / paper vecims 2003 /
datascapes & principes d'indexation /
gestion de niveaux et de zones /
+++
.proj:
i-weather /
i-light /
parisienne.ch /
norm sign generator /
electroscape 001 /
panorama 001 /
knowscape /
brillantmont international /

map_I.T.

// campus digital de l'Ecole Polytechnique Fédérale de Lausanne ----------//
// projet de master plan // fabric | ch -- www.fabric.ch -- // février 2003 //

+++

.txt:

2002 +++
rhizoreality.mu

rhizoreality.mu multi-user projects & architecture [description]
future extensions
paper for IEEE 2003 publication

RHIZOREALITIES IN 3D SHARED ENVIRONMENTS

Abstract

We will present a generic architecture called *rhizoreality.mu* that permits to set up 3D shared environments (based on *VRML* or not) but also to interact with reality (extension of the Web to the real world). This multi-user system is a part of a larger mechanism that allows to inter-connect different type of servers/services in order to share more vast set of data than only 3D or text. We are going to present our multi-user system and its concept. We will also demonstrate its ability to front a large set of various tasks through the description of real applications (artistic, conceptual and commercial).

1__CONCEPT

1.1__Definition

rhizoreality.mu: rhizo for rhizomic, .mu for multi-user. In other words, rhizomic realities in multi-user mode: realities that elaborate and inform themselves or each others, that develop, extend and/or fold up. But also hybrid, mix realities, in multi-user mode: single realities, multiple reality or plural reality. A paradoxical, re-qualified and recombinant space.

1.2__Approach and concepts

Today, *rhizoreality.mu* is a multi-user system directed towards live performances, interaction, real-virtual mixture/integration.
rhizoreality.mu is a technology resulting from a particular development specific to *fabric|ch*. It is a work in progress that first deals with aesthetic, philosophical, technological and conceptual considerations: a 'life-science' art.
It is only after two or three steps of development, when the project has acquired a certain level of maturity, that it is facing a market logic (commercial exploitation). What was more an experimentation becomes a commercial product or a service. Nevertheless, it preserves the specific marks of its development process. In this sense and considering our technological developments, we speak easily about 'concept-ware' or 'art-ware' due to the way we are producing it. We consider our applications in the same way we consider our contents: it must be meaningful and it must innovate (technically and/or conceptually, in their ways of approaching the technology development).

1.3__Development process

rhizoreality.mu is the result of a working process that intends to launch experimental approaches and projects at the technological level as well as at the aesthetical level. The idea is to create or to answer to other needs in setting up the initial work with the less possible external constraints. Then making the technology evolves on its own conceptual bases, according to the projects it arouses and the problems it meets. An almost 'neo-Darwinian' development process, which drives it to unexpected results and states. The generic idea that subtends our working process is to define conditions that make the unexpected use of technology possible. At the same time, we are also trying to find new ways of using electronic

media, under new paradigms. This experimental approach is considered for each project. It gives birth to aesthetic and/or technological prototypes. Some die out, others find a commercial or a conceptual existence. *rhizoreality.mu* is close to find both.

2__THE RHIZOREALITY.MU SYSTEM

rhizoreality.mu is based on a client-server architecture. In order to front classical problems linked to the connection of a huge number of clients, inter-servers communication is possible. Connected users can be shifted from one server to another depending on server's load or client's location. A world can be divided in a set of distinct zones ("*Area Of Interest – AOI*" in [8], "*Spatial Models of Interaction*" in [9]), each zone managed by a particular independent server.
We have tested the system on dedicated high-speed network lines, as well as through phone line (a normal modem). Some of our real-time performances (see section 0) were realised by using V90 56K modem. It permits to validate the protocol within a context that is closer to the one used by the major part of users. *Rhizoreality.mu* is able to handle *TCP/IP* as well as *UDP/IP* connection. Multi-user systems commonly use *UDP*, which generates less network traffic (but with a possible loss of quality). Unfortunately, *UDP* is not always supported by Internet service providers. It is also possible to remotely administrate servers. It is useful for managing connected clients as well as for configuring and managing servers by using a simple Internet browser.
The system is fully written in *JAVA* making possible to run servers on any kind of operating system and machine (*Windows*, *Linux*, *IRIX – PC*, *SGI*, *SUN*).
Functionalities we are presenting have to be seen as a particular use of the entire system. The system is able to receive information from a lot of different sources: it includes dedicated services that are producing and making available a particular information (such as sound server for example, delivering information on live music) but also particular devices such as movement or light detectors. In the same way, behaviours within *rhizoreality.mu* clients are not confined to the display of 3D worlds. A client can control a set of lights in order to produce a particular atmosphere in a room for example.

2.1__Clients

Writing a *rhizoreality.mu* compatible client program mainly consists in using a library. Through a couple of line of code, an application can be connected to the *rhizoreality.mu* system. When developing a *rhizoreality.mu* client, the main task resides in the implementation of a graphic layout. Existing clients include 2D or 3D based clients. 3D base clients are using *VRML* (*Blaxxun*[1]/*Cortona*[2] plug-ins) but also other Web 3D technology (see below). The library is existing in *JAVA*, in *Lingo* (*shockwave Director* [3]) and a *C++* version is under construction.

2.1.1__Shockwave client

In order to adapt *rhizoreality.mu* to new emerging 3D technology on the Web, a *Shockwave* [3] based client is now available. *Shockwave* is a well spread technology developed by *Macromedia* that allows the distribution of complex application on the Web. It needs the installation of a particular plug-in client side, but the integration of a particular programming language gives a real flexibility to the system. A large set of medias of different kinds can be manipulated through this programming language named *Lingo*. Recently, the 3D was introduced in *Shockwave* just like a new available media. In order to establish a connection between a *Shockwave*

based client and *rhizoreality.mu*, a set of *Lingo* libraries were written. The most obvious one is in charge of managing *rhizoreality.mu*-like communication. On the top of this library, another one was written, interfacing *rhizoreality.mu* events with 3D actions. This set of libraries can be included in any *Shockwave* based application, making a project part of a *rhizoreality.mu* network very easily through few lines of code. This is very similar to the way a simple *JAVA* applet can become a *rhizoreality.mu* client.
As *Shockwave3D* is using its own proprietary 3D file format, it is needed for a project that will allow heterogeneous set of clients (e.g. *VRML* based clients as well as *Shockwave3D* based clients) to have the 3D environment and objects available in both formats.
Dealing with a system that allows a heterogeneous set of clients based on different 3D technologies can be of a limited interest. Usually, it is better to choose one or another technology and then, realise the entire project by using the chosen one. In another hand, it is interesting to consider heterogeneous clients as a possible way to make available functionalities specific to a given system in another one. *Shockwave* is well known for its ability to be extended through external libraries called *XTra*. These libraries can be written by any third party companies and purchased independently. When setting up a project that will use particular libraries, additional libraries are downloaded seamlessly in real-time, client side, in order to enhance the client's *Shockwave* plug-in. For example, a very well known *Xtra* is a one made by *Havok* [4]. This library makes possible any physically based simulation within a 3D world, making possible to simulate a ball bouncing on surrounding obstacles just through few lines of *Lingo* code. We can then imaging having a particular *Shockwave3D* client connected to *rhizoreality.mu* system in charge of managing physically based behaviour objects and sending this information to all other *VRML* based client. You will obtain a *VRML* world, with physical behaviours that could have been difficult to include in all *VRML* clients. This simple example demonstrates that the ability to manage heterogeneous clients can be of some particular interests.

2.1.2 CAVE-like client: observers

CAVE systems [5] consist in rendering the same scene from several different point of views. Typically, these views are used to render what the user can see on its left side, right side, up side and bottom side from its current location. By projecting each view on a dedicated screen, it is possible to build a visualisation volume (usually a cube made of all projection screens). The user can stand and move freely within this volume, being surrounded by all rendered views. This creates a immersion experience without having to wear head mounted display.
We have set up a particular *rhizoreality.mu* client to be able to construct *CAVE*-like structure for any world managed by a *rhizoreality.mu* system. In this particular case, these clients do not need to be displayed (as avatar, like normal client/user) in the simulation. They are 'ghost' clients that just need to receive information from the simulation: we called them '*observers*'. It mainly implies that these *observers* do not have to send their location on the network. Basically, it is just a camera used to see a particular area of the scene.

We will distinguish *observers* of two types: *masters* and *servants*. They are both exactly the same kind of 'silent' client. They are just characterised by a parameter: an *observer* without another *observer* assigned to it will be a *master*, where *observers* invoked with another *observer* specified as a parameter will be a *servant* (Figure 1). A *servant* will follow exactly what its *master* is doing (reproducing remotely what its *master* is seeing).

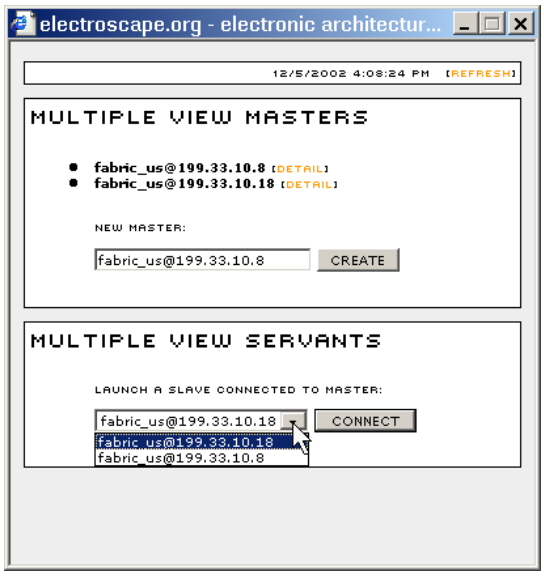


Figure 1: Creation of multiple view *masters* and *servants* through a simple Web page (*electroscape* projects [13] [15])

For each *servant*, it is possible to adjust manually the location and orientation of the camera. This is the way to make an *observer* to look at the right side or left side of the location they are in charge to follow (location of its *master*). This final manual adjustment permits to set up any kind of viewing volume (one *observer* = one projection screen) or even to create particular visual effect. It makes easier the adaptation of *observers* to any kind of physical installations.
Of course as location updates are sent to *servants* through the network, even within a local area network, a short delay will appear between the *master* movements and *servants* updates. Nevertheless, delay between *servants* linked to the same *master* is acceptable and most of the time imperceptible. This is mainly due to the fact all *servants* used to set up a visualisation volume in a given place are going to share the same network capabilities. So, from one *servant* to another, the delay observed to receive data from the *rhizoreality.mu* server is going to be identical. In the other hand, from one visualisation volume to another set up in a different place, it will not have any kind of synchronisation. As these installations are generally used as a part of particular events, all conditions (a fast and reliable network connection, good 3D enhanced computers) are generally ensured in order to have good visual results. In the worst case, when connecting *servants* through a phone line using a 56K modem, *UDP* can be used in order to obtain a more fluent simulation, making even possible the use of low bandwidth network connection.

It is the reason why, only *servants* are used for projection, where the *master* is controlled by the user in order to move in the 3D scene (Figure 2).

Figure 2: A kind of multiple views installation set up for a private institution anniversary (*BMT* [7])

With such a mechanism, it is possible to control several visualisation volumes dispatched in different places with the same *master*. Even when sharing the same *master*, each visualisation volume can have its own characteristics, due to the fact that *servants* can be adjusted independently.
With such type of specific clients, it is also possible and even recommended to use UDP in order to minimize delay between *servants* and *master* as well as delay between all *servants* following the same *master*.
As a minimal case of use, a set of *servants* connected to a *master* can be used to share a dynamic walkthrough with users dispatched all around the world. Depending on the situation, plenty of different uses can be found to exploit this feature.

2.2__Client-server communication

Messages used for client to server communication are not linked to the *VRML* syntax. This choice can be justified by the fact that we do not use only *VRML* based application. As described in the previous section, we also have a 2D clients or *Shockwave3D* clients. By restricting the *VRML* dependency at the client level, the *rhizoreality.mu* system can use any kind of 3D language or graphics library for defining a new client. The entire system stays open to new emerging standard such as *X3D* [6].
All messages inherit a set of parameters from an initial message object. Each message can then define its own additional parameters. Each message is in charge of sending and receiving its data through a given kind of network link.
When implemented, a compression function can be used before sending a message. This compression function is defined in every message, giving the ability to precisely adapt the compression method to the type of data to transmit.
A generic message is also available. This message can be used while implementing a new client in order to transmit particular information from one client to another or to all of them independently of the *rhizoreality.mu* system. It gives a good flexibility for new clients development.

3__EXAMPLES OF CONCEPTUAL AND COMMERCIAL USES

3.1__3D chat for a newspaper: the "Digital Café"

This application of the *rhizoreality.mu* system was made for a Swiss newspaper. It is a 3D chat including a 3D server and four chat servers. The 3D world is organized in a huge "tower" that can contain an infinite number of rooms. In the current version, it includes a general room (entrance place) and a discussion room. In the discussion room, there is a set of three areas (tables) that are linked to particular topics. The left part of the main window contains the general chat client, the middle part contains the 3D or 2D client (choice of the user) and the left part of the window contains chat clients linked to specific discussion topics (Figure 3). This last part is empty by default (no chat activated). When a user comes closer to a table, the connection to the associated chat server is made. When moving away, the user is automatically disconnected from the specific chat. The user has to use the 3D or 2D graphical interface for accessing and participating to discussions.
Without having to manually edit any particular files, the newspaper's staff can partially modify the 3D world (rooms) and chat settings (server's remote administration feature). This system is linked to a database for maintaining customer's information.

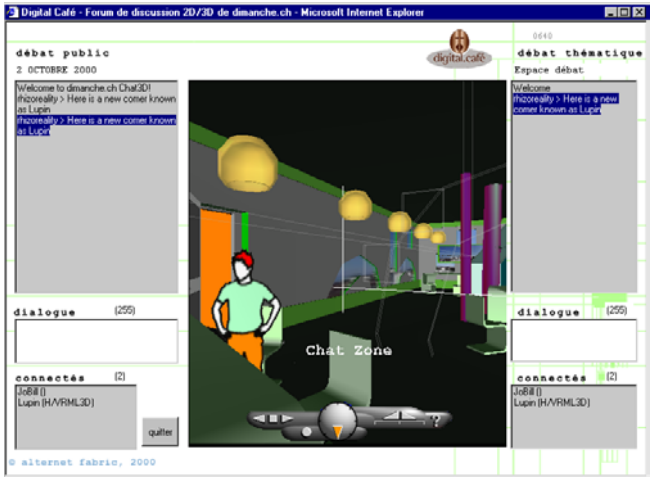


Figure 3: global view of the "digital café" application (discussion room)

One of the highest numbers of simultaneous connections recorded with this system was 53 connections, dispatched to three servers (two chat servers and one 3D server). For this project, *rhizoreality.mu* servers are running on a *Windows* based *PC*.

3.2__Live music show performances

The *rhizoreality.mu* system was used in a set of live music shows performed in nightclubs (2000). The first set of experiments was performed in one nightclub at a time using a local network (no external Web link). The *rhizoreality.mu* system was connected to a sound server and a *SMS* server (mobile phone short message system). The 3D world was animated according to the music played in the nightclub. To perform this task, the 3D server sends geometrical transformation generated according to information received from the sound server (analysing in real-time music beat, frequencies etc…). A *SMS* receiver (that has a cellular phone number associated to it) was installed on a *PC* on which was running the *SMS* server. The *rhizoreality.mu* system was connected to this *SMS* server

(as a service) in order to receive messages sent by people present in the nightclub or elsewhere (the cellular phone number was publicly distributed through flyers). *SMS* messages were displayed on the Web, as well as used as visuals within a dedicated 3D world. A set of *PCs* was dispatched in the nightclub to make possible the exploration of the 3D shared world by night clubbers.

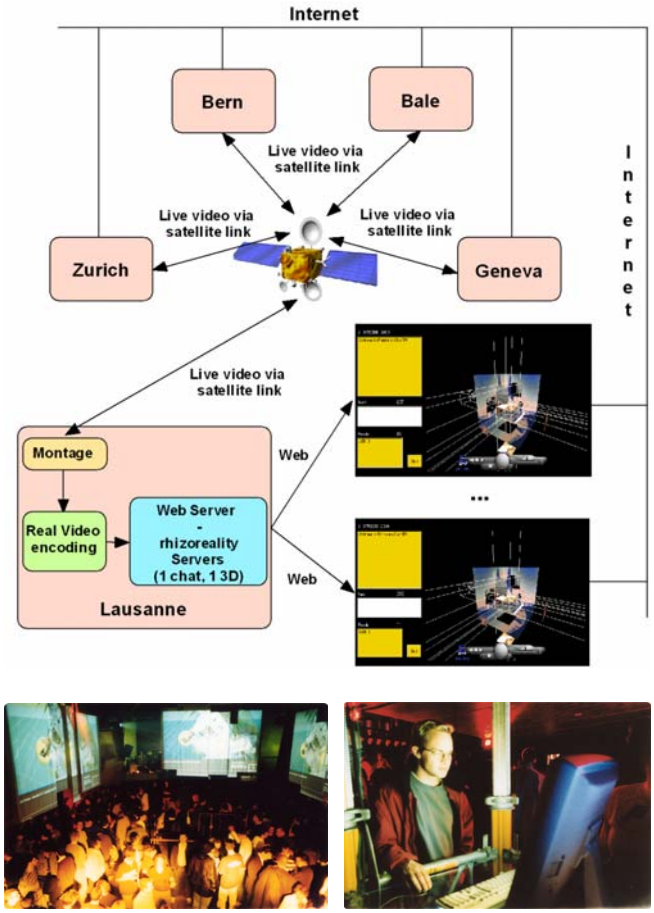


Figure 4: overview of the "*SensUAll*" night performed in five towns simultaneously

The latest use of the *rhizoreality.mu* system in such a context was performed for a party organised simultaneously in the five major cities in Switzerland (the "*SensUAll*" night in *Zurich*, *Basel*, *Bern*, *Geneva* and *Lausanne* - 2001). All places were connected via a satellite link (video and sound) and the Internet (phone line with a modem). The satellite was used for sending local video (fix and wireless cameras) to the main video editing premise located in the city of *Lausanne*. The source to be dispatched was chosen in *Lausanne*, locally encoded to *realvideo* format and made available in 3D within a *rhizoreality.mu* world through a Web server (Figure 4).
The 3D world set up for this event was geometrically simple and specifically designed to be used as a visual animation. Live video stream was mapped in the 3D world, enabling people connected to the *rhizoreality.mu* system to follow the real party. In this configuration, the *rhizoreality.mu* system included one 3D server and one chat server.

3.3__Electroscape

3.3.1__Concept

The work performed on *La Fabrique* (1999-2000) [12] has lead us to the set up of the *electroscape* concept. Basically, *La Fabrique* consists in setting up 3D shared world where a set of different artists have produced a given piece of art, added to the entire world. Each artist was given a subset of the entire world, and was free to modify it. Each *La Fabrique* was then defined by an initial state and a final state, generated by the sum of work produced by all involved artists. The creation process was completely hidden and major integration work was made manually. Of course the multi-user version of *La Fabrique* is using *rhizoreality.mu* system. Based on the experience of *La Fabrique*, the idea came out to set up a framework in order to organise and manage such shared creation project. The main idea was to make possible to follow the evolution of the work of each artist as well as the evolution of the entire 3D world at any time. Each *electroscape* project proposes the management of artists as well as the management of the produced 3D world. Once recorded as part of a *electroscape* project, an artist is able to add and modify data about himself, in the same way he is able to upload/download his work through the *electroscape* web site interface. It is then possible to access the on-going work by accessing directly an artist's current work or by visualising the actual state of the entire 3D world. We will see through the two following examples how the *electroscape* framework can be used.

3.3.2__Electroscape 001

Electroscape 001 [13] was submitted and accepted at *Siggraph* 2002. The project was selected to be part of the art exhibition where on-going art works were taking place during the whole week the conference last. This was an opportunity to test in one sitting major functionalities we want to include within *electroscape* projects. Within *electroscape 001*, it is possible for artists to upload/download 3D objects. Then, through a dedicated *rhizoreality.mu* client/interface, it is possible to add/remove and to manipulate these objects in the 3D shared world. For artists participating to the creation process, it gives the ability to work together in real-time, and from the end-user point of the view, it is possible to follow the entire creation process, live.

For *Siggraph* 2002 conference, three working groups were involved in the creation process. One group (*fabric|ch*) installed at the *Siggraph* conference, mainly in charge of coordinating artists interventions. In terms of creation, this group has tried to insert dynamically generated data in the 3D world by modifying *rhizoreality.mu* clients and to explore display limitation (stiffness limit of projectors, flat panel screens or classical monitors) (Figure 5 – bottom left). A second group constituted by *lab[au]* [14] was working from Brussels in Belgium. This work group was more dedicated to sounds within the 3D world (Figure 5 - right). The third group (*fabric|ch*) was working from Lausanne, in Switzerland. They were enhancing the 3D shared world with 3D dynamic objects reflecting the *electroscape 001* activity.

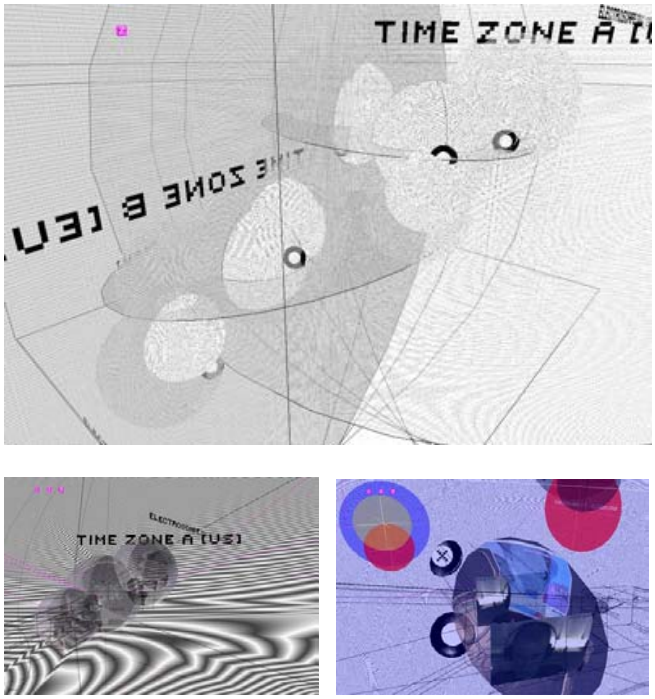


Figure 5: initial *electroscape* world (top), *fabric|ch* work (bottom left) and *lab[au]* work (bottom right)

Each group has a dedicated 3D space in the shared world, but all groups can potentially interfere everywhere, including in other groups reserved areas. All added 3D objects or sounds were managed by the persistency feature of *rhizoreality.mu*. Regular world backups (persistency safe keeping) were made automatically, making possible to visualise retrospectively the 3D world at any time/state during the creation process. It gives also the ability to go back to a previous state and then work again from this given state (classical backup function but applied to 3D persistency).
Due to the fact that working groups were dispatched all around the world, there was always an on-going work to follow in the 3D shared world. Except for a minor technical problem, this experiment was pretty successful. The major problem was linked to the management of the different artist's groups in order to obtain as much as possible a homogeneous result. The project is now closed (not evolving anymore) but both initial and final states can be accessed on the Web [13].

3.3.2__Electroscape 002

The second version of *electroscape* [15] was initiated in collaboration with modern art museum of Geneva (*mamco*) fall 2002. The creation process was much closer to the one used for *La Fabrique* series. As a part of a set of exhibitions (version A,B,C,D,E,F), it was proposed to set up the digital version of the *mamco*.
The digital version of the museum has to be seen as an extension of the real museum and not a 3D modelling of the real building (Figure 6 top). In order to inaugurate the digital *mamco*, it was proposed to an artist, *Heimo Zobernig*, to make the first piece of art within the digital space (Figure 6 bottom). A real space was also created in the real *mamco*, as a kind of intersection between the digital space and the real one… to be perceived as a space common to both worlds.

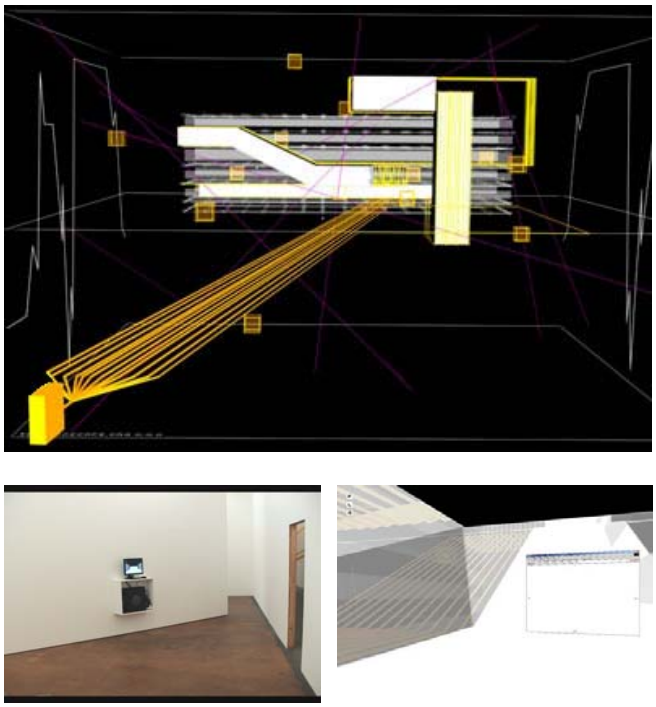


Figure 6: the digital *mamco* space (top), Heimo Zobernig's real space (left) and its digital clone (right)

In order to be used in the different types of *rhizoreality.mu* clients, a new service was developed. This service is in charge of analysing network load generated by all access made to *electroscape 002* data. It can deliver a regular data flow by providing the network load average or the exact network load when data are accessed. No local data (packet content) are sent by the service.
The analysis of the network load is made locally and resulting data are then sent to clients in the requested format. Of course, this service can monitor the activity of any server and is not linked to *electroscape* project. Within the *electroscape 002* context, these data were used to monitor in real-time 3D object colour and to generate 2D and 3D graph. Thus, the general aspect of the 3D world is evolving in time, according to connected user's activity. The structure can be perceived in different manner depending on the network load.

In addition to now classical 3D client, 2D based *rhizoreality.mu* clients were developed, linked to network load analysis. In the same way than in 3D, network load data were used to generated colour oscillation and graph drawing, full screen. These particular *rhizoreality.mu* clients were developed to be used during the project presentation and for setting up physical installation.

4__FUTUR WORK

The development of the system has now driven us to a point where we are considering a system that is truly and powerfully interactive, reactive and connective. Something that lets you map lots/heterogeneous inputs onto lots/heterogeneous outputs with the *rhizoreality.mu* layer in between our various adapted servers as data providers and our multiple clients as decoders. Real to real event and data mapping, virtual to real and vice versa, virtual to virtual, 2D to 3D, etc… all kind of playable architectures. This open a wide range of new real-virtual Internet applications for Web3D and multi-users in general.
The generic side of *rhizoreality.mu* permits to consider evolution of the system on new platforms. Even if the Web constitutes a very open context, it is still a very particular one, linked to computers. Game consoles is a new playground to explore. It is much more considered as a normal living-room extension, in the same way as television or DVD player are, where a computer is generally confines in another room. Consequently, it can permit to reach a new range of people with a new kinds of content, proposing an alternative to games.

5__ACKNOWLEDGEMENT

We would like to thanks the fabric|ch team linked to this project: Marc Escher, Stéphane Carion and Christophe Guignard.

6__REFERENCES

[1] *Blaxxun* VRML plug-in, http://www.blaxxun.com/.
[2] *Cortona* VRML plug-in, http://www.parallelgraphics.com/products/cortona/.
[3] *Macromedia Director*, *Shockwave* plug-in, http://www.macromedia.com/software/director/.
[4] *Havok* physics engine, http://www.havok.com/
[5] CAVE System, the Electronic Visualization Lab (EVL), 1992, http://www.evl.uic.edu/pape/CAVE/.
[6] *X3D* Specification, http://www.web3d.org/x3d.html.
[7] Brillantmont International College, http://www.brillantmont.ch/.
[8] Michael R. Macedonia, Donald P. Brutzman, Michael J. Zyda, David R. Pratt, Paul T. Barham, John Falby and John Locke. NPSNET: A Multi-Player 3D Virtual Environment Over the Internet. Symposium on Interactive 3D Graphics. 1995. New York. pages 93-94.
[9] Chris Greenhalg. Dynamic, embodiement, multicast groups in MASSIVE-2. Technical report NOTTCS-TR-96-8. Dept. Comp. Science. University of Nottingham. 1996.
[10] *Digital Café*, http://www.digitalcafe.ch/.
[11] "*SensUAll*" night, http://194.38.80.19/parisienne/main.html.
[12] La Fabrique series, http://www.fabric.ch/La_Fabrique/.
[13] electroscape 001, http://www.electroscape.org/001/.
[14] lab[au], Brussels, Belgium, http://www.lab-au.com/.
[15] electroscape 002, http://www.electroscape.org/002/.

2003 +++
knowscape

shared knowledge & profiles
projects & architecture [description]
future extensions
paper for vecims conference

_knowscape – a collective knowledge architecture

Abstract
In stead of focusing on negative sides of the freedom observe on the Web, it can be more interesting to consider the Web as a support for the establishment of a collective knowledge. Considering the Web through this kind of perspective makes obvious that tools are missing in order to enhance and develop the awareness of users on the Web as well as the ability to share Web experiences with others. In this paper we propose a 3D Web based tool called *_knowscape* that offers a mean to users to share their Web experiences. We will describe concepts that guides our project and technologies and features that are used to try to enhance the Web experience and bring Web browsing to another level of consideration.

Introduction
The advent of computers and the Internet gives birth to a new kind of human collaboration and interaction. Where all information vectors have muted to reach pyramidal organisation (knowledge diffusion controlled only by few companies/persons), the Internet and computers gave back the power to the collective conscience for the management and the diffusion of its knowledge. We are pretty far from images and consideration concerning new medias and computers that can be commonly and regularly exposed in classical medias.
Major web experience does not include any awareness of other people browsing the same web page as you. Browsing the Web is usually a lonely experience. Few Web sites exposed the number of users accessing the same Web page at the same time, but it is generally Web sites link to computers and there is no way to communicate with these users. The Web gives the ability to everybody to make available information world wide generating an enormous amount of data.
Usually, a lot of time is wasted on looking for other people working in the same field as you. The classical framework is to look for information and documents, then, eventually trying to e-mail to authors. Through the use of multi-user based application, data mining and information search can be made at the same time than taking contact with people. Awareness of people browsing the same Web page as you combined with the ability to communicate with them can make possible very easily the ability to keep in touch with people having the same interests as you.

Concept
User profiling is now widely used on the Web. It is generally exploited in commercial Web site in order to surround user's habit. collected and generated Data are usually not directly available to the user, neither shareable with other users. An experiment was made in order to try to share a browsing experience by making available these profiling information to the user. The project was called *panorama* **[16]** (2001) and was linked to the activity report for year 2001 of the *Swiss federal institute of technology of Lausanne*. It was the Web version of the official report (Figure 7).
User's activity was tracked: which page was visited, how long the user remained on the same Web page, etc… All these rough data constitutes the profile of the user and was available to the user through different graphical representations. As a basic and classical use, the user can

exploit its profile data to retrieve faster a given information when coming back on the web site.

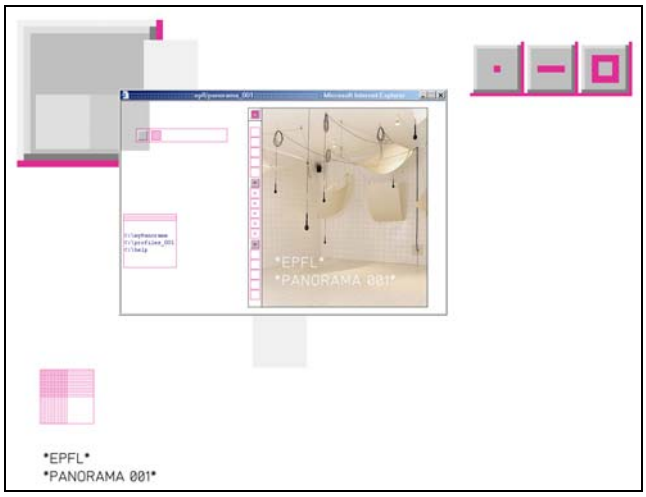


Figure 7 : *panorama* home page

It was also possible to share the profile by sending it to a friend by email or even by *SMS* (mobile phone short message system) using a dedicated web page (Figure 8).

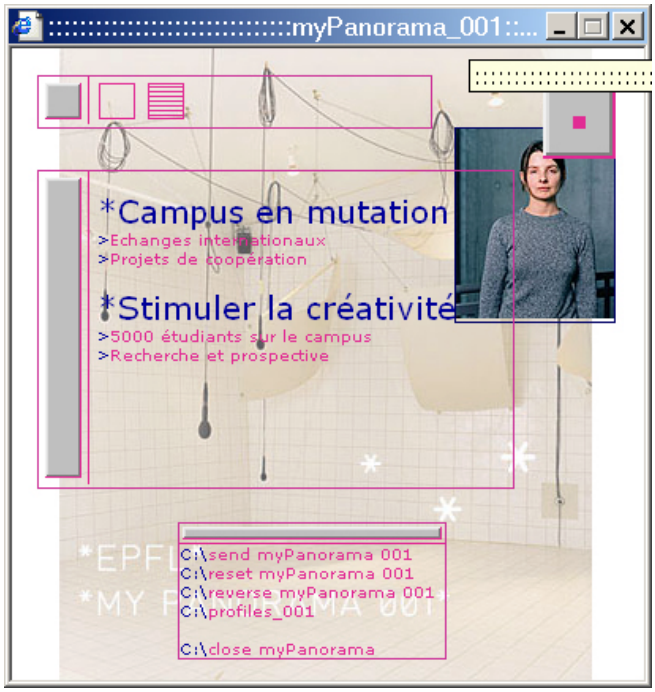


Figure 8 : a generated profile and sharing options (bottom)

Moreover, all profiles were accessible to everybody, organised by date (Figure 9). Of course it can be decided not to share all data, as privacy is a also a parameter to take in account. Within the particular context in this project, there was no reason why sharing all user's profiles could be a problem.

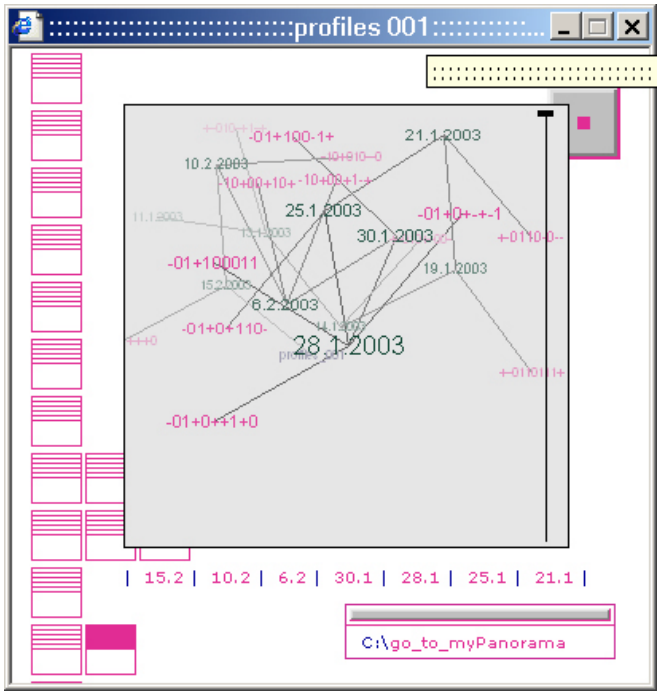


Figure 9 : all generated profiles

Panorama was a first experiment in sharing user's Web experience. The particular context of the project has brought an important set of constraints (multi web browsers compatibility including those on *Macintosh*), which makes difficult the use of some technologies like 3D and even restricts the use of some *HTML* functionalities. Nevertheless its user interface, this project was pretty tricky to realise.
The *_knowscape* project (2001) can be considered as an another reflection on sharing Web browsing experience and by extension on sharing the collective knowledge. Through its massive use of more or less 'heavy' technologies (3D, multi-user framework see section following section for details), it can be considered as the anti-*panorama* project. Basically, both projects are handling the same kind of data, but the use and interpretation (in 3D) of data in *_knowscape* makes more tangible the idea of a shared knowledge architecture: building an entire shared and navigable 3D world by exploiting the collective knowledge. These notions are existing because of the advent of the Internet and networks and are about to be considered by philosophers in order to set up experiments and methodologies **[17]**.

_knowscape

_knowscape is an experimental 3D browser accessible through the Web that let users browse online content, create 3D information environment and share it [or not] with other connected users thanks to its multi-user mode. Each user has a subjective 3D world, which belongs to a shared global world, and his own avatar [*IP* address or *GSM* number]. The main idea behind *_knowscape* is to build virtual spaces and avatars made out of users browsing choices and experience.

Figure 10 : avatar building mechanism, web addresses addon to user's avatar

As a multi-user based environment, a connected user is represented by an electronic body (an avatar) within the 3D shared world. Thus, all users are aware of each others. In its initial state, the user's avatar is made of the Internet address of the user's computer (Figure 10, top-left). When connecting to the world, a initial volume is assigned to the user. This volume (a wire frame cube) constitutes (and localises in space) the knowledge the user will build in 3D by browsing the web.

For the browsing experience, the user can use a normal web address (*URL* formatted) or a set of words in order to perform a search on the web (using the *google* search engine). Every input (URL) are used to modified the user's avatar. Browsed web page's *URL* becomes extension to the user's avatar, making possible for everybody connected to the world to be aware (in a basic way) of what every single user is browsing (Figure 10). It implies that avatar updates are shared with every connected users, e.g. all avatar modifications (adding or removing a web address) is relayed to all users.

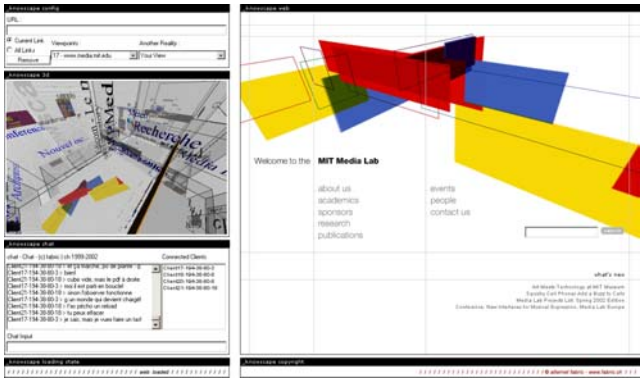


Figure 11 : a _knowscape experience

When entering a new web address the web page specified by the user is displayed, in a normal way (2D) on the right part of the *_knowscape* browser (Figure 11). Simultaneously, the user is automatically moved to a new location within its knowledge space. At this new location, a 3D transcription/translation of the 2D web page will be added, as new elements of the user's knowledge. To continue his browsing experience, the user uses the 3D translation to follow web links. Right part of the browser will be updated with the new web page and a new 3D element will be added, translation in 3D of the new followed link, and so on.

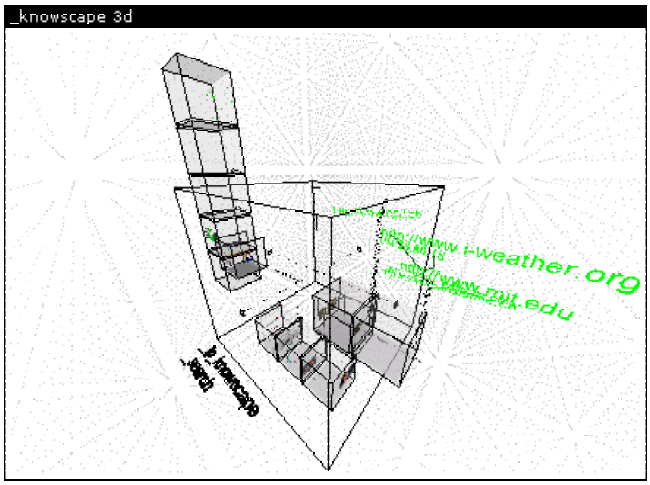


Figure 12 : the 3D space knowledge of a user

Step by step, the user will fill his space with its knowledge. Browsing links from a web site will make appear aggregated horizontal or vertical structures (Figure 12). It makes possible to distinguish one browsing session from another one. By navigating in 3D within these structures, a user can go through its knowledge space again and again. The user keep a tangible track of its experience and it will be possible to share it with other users.

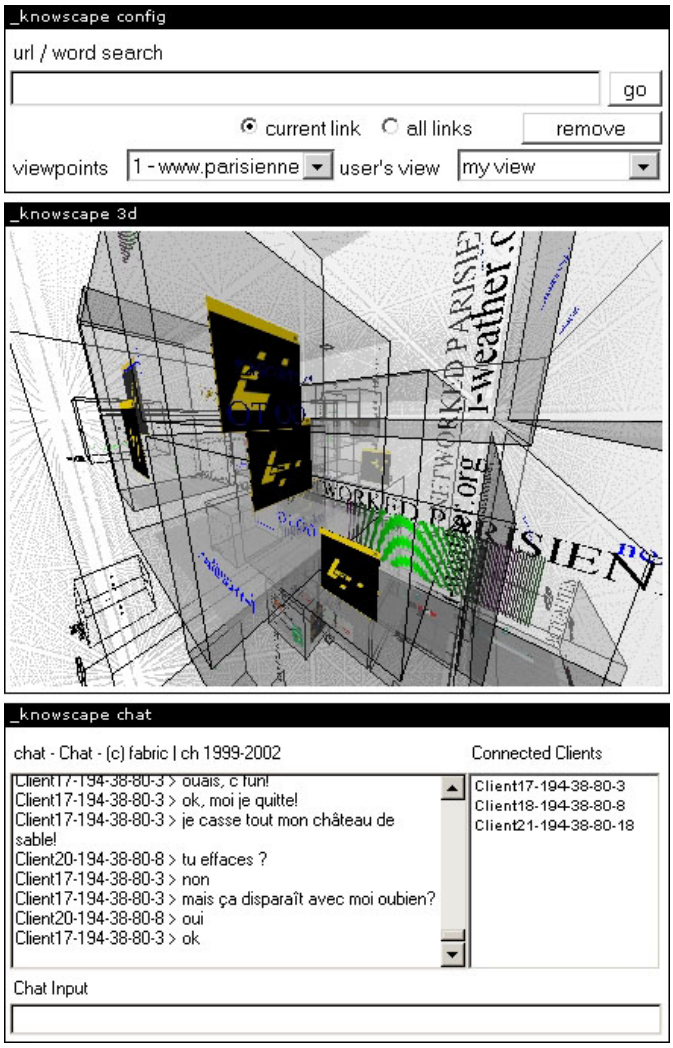


Figure 13 : _knowscape interface (text input and subjective view management – top, chat interface - bottom)

By default, 3D structures built by the user while browsing the web are private. It means that these 3D objects, translation of 2D data, are used to build the subjective view of a user. It is not possible to other users to see what a given user is building (exception made of links added to the user's avatars). In order to do so, a user have to switch from its knowledge space to the one belonging to the user he wants to reach. This is done through the *_knowscape* browser interface by selecting the desired user in the connected users list (Figure 14, top). Once switched, the user is able to see the 3D knowledge-scape constructed by the selected user (Figure 14 bottom). Through 3D, it is possible to explore this knowledge and even using it for his own knowledge-scape. Meanwhile, you can have a chat with the user, making possible to have an explanation on a given link or on a given topic.

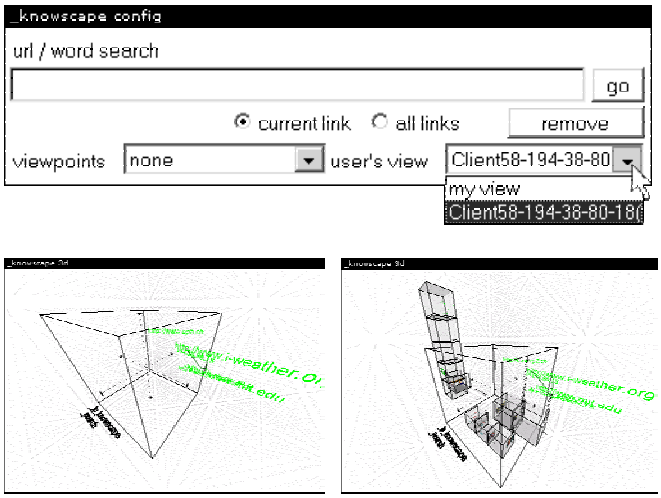


Figure 14 : switching to another user's subjective view.

Technology

_knowscape, which is based on our own multi-user system *rhizoreality.mu*, demonstrates the ability for a user to modify and to make evolve the content of a 3D shared world. *rhizoreality.mu* has also been used in our *La_Fabrique* **[12]** project and for our *electroscape.org* **[13] [15]** framework. *rhizoreality.mu* system has follow a particular development way, trying to merge technologies (network, 3D) with concepts.

rhizoreality.mu

rhizoreality.mu **[31]** is based on a client-server architecture. The system is fully written in *JAVA* making possible to run servers on any kind of operating system and computer (*Windows, Linux, IRIX - PC, SGI, SUN, XBox game console*).

In its very last version, *rhizoreality.mu* is able to handle connection request through *TCP/IP* as well as through *UDP/IP* using a circular distribution system [CDS]. Both protocols can be available at the same time. For web clients, it is possible to have a *JAVA* based client as well as *Shockwave* based clients. 3D shared world can be modelled by using *VRML* **[18]** or *Shockwave3D* **[3]** or even both at same time. 3D visualisation is done through the use of plugins **[19] [2]**.This flexibility permits to face a large set of different needs and applications **[26] [12] [13] [15]**. Functionalities linked to the *_knowscape* project have to be seen as a particular use of the entire *rhizoreality.mu* system. It is able to receive information from a lot of different sources: it includes

dedicated services that are producing and making available a particular information (such as sound server for example, delivering information on live music) but also particular devices such as movement or light detectors. In the same way, behaviours within *rhizoreality.mu* clients are not confined to the display of 3D worlds. A client can control a set of lights in order to produce a particular atmosphere in a room for example. As *rhizoreality.mu* was also used in commercial application, it includes features like connected users management as well as remote administration (through a dedicated *JAVA* applet, on the Web). Usually, a *rhizoreality.mu* based system generally deals with a set of servers (3D servers and/or chat servers). Each server is aware of the existence of others, making possible the exchange of information between them. In addition to these servers linked to the core system, an unlimited set of services can be used to enhanced basic *rhizoreality.mu* functionalities. It relays on the integration of *rhizoreality.mu* servers in a meta network based system in charge of maintaining a clear view of the state of the network, including a list of available services. When a server receives a request that concerns a given service, the server will ask to the meta-system where to find such a service and how to establish a basic connection with it. A direct connection to the service is then established, information about the protocol are exchanged, making possible to the server to finally perform the request received from the client. This system make possible to include a lot of functionalities within *rhizoreality.mu* system (making them available client side) just by the set up of new services (no need to modify the core implementation of the system).

_knowscape related services

HTML *to 3D translation*
This service is invoked by a client when a web address is specified. In its actual version, it converts a *HTML* web page in *VRML*. Followed rules to ensure the conversion can be changed at any time. Basically links and images are selected to be reproduced in 3D in order to construct the volume associated to the Web page. This task consists in retrieving the Web page content, translating in *VRML* and sending back the result to the requesting client.

SMS login
As a demonstration on how mobile phone world and the Web can be connected, a login system using *SMS* is proposed. Users that connect this way to *_knowscape* are represented in 3D by their mobile phone number in place of the *IP* address of their computer. By clicking on avatar, it is possible to send them *SMS*. Of course, it can be decided not to display real mobile phone number while keeping the functionality (all relevant information are kept safe server side). When logging in, the user is asked to send to a fixed mobile phone number a *SMS* containing a given code. The user will then receive back a final code that have to be entered as a password on the Web page. If the matching mobile phone number/entered code works, the user will be allowed to access *_knowscape* system. The *SMS* login system will then generate relevant information to be passed to the multi-user application (which will determine user’s avatar and linked functionalities).

Discussion

In *_knowscape*, we have tried to emphasize the use of 3D as a real advantage in comparison with more classical 2D based interface. If 3D is not that much used on the Web, it is mainly because of the lack of killer applications, proving that 3D can be of a real interest if used correctly and at the right place. Browsing a Web with *_knowscape* gives the ability to build in real-time a 3D visualisation of the Web site structure. In particular, it keeps visual tracks of links between browsed Web pages. When coming back to a previous volume and clicking on a new link, the structure will fork in order to make possible to distinguish this new browsing session.

The simple fact of browsing the Web can bring to the apparition of a new online communities based on 3D. Then it is possible to users to connect to these communities without having to use the *_knowscape* browser, as 'passive guests' (as they will not participate to the raise of the world, but they will be able to take advantage of it). Overload of the graphic hardware can be avoided by the use of level of details on objects. The world is apt to increase a lot in terms of complexity and volume. It is important to optimise objects to render in 3D by minimising their weight (in terms of geometry). This is especially true with text rendered in 3D. This problem will rise as soon as we authorise to visualise all subjective views at the same time, in order to perceive the world in its entire complexity.

The main idea is not to replace the Web. It is obvious that 3D is not well adapted to visualise text, which constitutes major part of content on the Web. It is the reason why we have kept a classical 2D view in the *_knowscape* browser. The goal is much more linked to the establishment of a knowledge communities through the use of 3D technology (to visualise the knowledge) and multi-user features.

Evolution

One of the most obvious thing to do is to enhance the conversion process that translate *HTML* to *VRML* (or any 3D language). *HTML* is now a very complete language that includes many dynamic functions. Structure *TAGs* can be much more exploited in order to generate a much more complex structure in 3D. It is also important to keep in mind that each converted Web page should stay connected with each other in a way or another to make possible to identify one browsing session. We can imagine setting up a set of conversion rules based on a set of generic 3D objects that can be used as containers for converted data. Then, the generated world will be much more complex than in the present version. We can imagine to use these structures to build the entire world (and not only the subjective world of each user). Through the conversion rules mentioned previously, it is possible to make this world following a given logic, avoiding to dive in a chaotic set of data where user just get lost.

Spatial distribution can also take in account proximity in terms of content between different Web pages. Through content analysis, a 'distance' and other parameters (content, type of content…) can be generated for each pages. These parameters can be used for determining where translated contents are located in space. Depending on parameters that are considered, it can build the same world in completely different spatial configurations. By making available to the user the ability to choose one parameter or another, he can decide what configuration to use in order to visualise and navigate the 3D world (according to its interests).

With this kind of functionality, the spatial location of a given user will give instantaneously information on its actual interest: its 3D coordinates will reveal what kind of content the user is considering (relatively to the kind of spatial distribution of the world the user has chosen).

The actual version of *_knowscape* does not include any recording of what users are doing. When a user quit the application, all his browsing experience is lost. In the same way as in the *panorama* project, user's action can be recorded and saved. These data can be used in many different manners. As a very obvious one, it can be possible to rebuild the world of a given user when this one is coming back to *_knowscape*. This ability can easily be handled by the persistence feature ensured by the *rhizoreality.mu* system. Moreover, user's profile can be also used elsewhere than in the *_knowscape* browser. The profile can be sent to a mobile phone. New mobile phone generation includes graphics capabilities (colour display, graphic libraries (*JAVA* based with *J2ME* and *MIDP* **[32]**, *Mophun* **[33]**)) which gives the ability to develop software for mobile phone. The advent of new types of connection (*GPRS*, *UMTS*) even allows the establishment of network connections from a cell phone. It brings new capabilities for online communities as it will not be an obligation anymore to be in front of a computer in order to be connected and access content.

Conclusion

We have pointed out the fact that the Web can be considered as the advent of a collective knowledge structure/architecture. *_knowscape* browser gives the ability to users to be aware of each other and to communicate at the same time they are browsing the Web. Through a mixed of 2D and 3D interfaces *_knowscape* permits to build and to visualise every user's Web experiences, result of a subjective selection of all information available world wide. Such interface can make easier for a user to contact people with the same interest while browsing/discovering a content.

References

[16] *panorama*, shared profiling example, http://001.epfl.ch, 2001.
[17] The Canada Research Chair on Collective Intelligence, University of Ottawa, http://www.collectiveintelligence.info/
[18] *VRML*, Virtual Reality Modelling Language, http://www.web3d.org/.
[19] *Blaxxun* VRML plug-in, http://www.blaxxun.com/.
[20] *Cortona* VRML plug-in, http://www.parallelgraphics.com/products/cortona/.
[21] *Macromedia Director*, *Shockwave* plug-in, http://www.macromedia.com/software/director/.
[22] CAVE System, the Electronic Visualization Lab (EVL), 1992, http://www.evl.uic.edu/pape/CAVE/.
[23] *X3D* Specification, http://www.web3d.org/x3d.html.
[24] Michael R. Macedonia, Donald P. Brutzman, Michael J. Zyda, David R. Pratt, Paul T. Barham, John Falby and John Locke. NPSNET: A Multi-Player 3D Virtual Environment Over the Internet. Symposium on Interactive 3D Graphics. 1995. New York. pages 93-94.
[25] Chris Greenhalg. Dynamic, embodiement, multicast groups in MASSIVE-2. Technical report NOTTCS-TR-96-8. Dept. Comp. Science. University of Nottingham. 1996.

[26] *Digital Café*, http://db.fabric.ch/digital_cafe/.
[27] "*SensUAll*" night, http://194.38.80.19/parisienne/main.html.

[28] La Fabrique series, http://www.fabric.ch/La_Fabrique/, 1999, 2000, 2001.
[29] electroscape 001, http://www.electroscape.org/001/, 2002.
[30] electroscape 002, http://www.electroscape.org/002/, 2002.
[31] Christian Babski, Patrick Keller, rhizoreality.mu, Web3D symposium, 2001, Paderborn, pages 109-116.
[32] Java micro edition, http://java.sun.com/j2me/.
[33] Mophun(TM), http://www.mophun.com/.

// map_IT -- //
// projet de campus digital pour l'EPFL // fabric | ch // www.fabric.ch // mai 2003

2003 +++
datascapes & indexation

descriptions de datascapes
principes d'indexation
extensions

Génération de datascapes

Les datascapes peuvent être perçus comme une vue conceptuelle d'un ensemble de données, sans qu'une interprétation évidente puisse obligatoirement en être déduite. Utilisant la 2D ou la 3D, les datascapes peuvent avoir un coté purement fonctionnel, aidant à la perception et l'analyse rapide de certains paramètres.

Dans un premier temps, la génération de datascapes peut se baser sur des paramètres directement associés à une zone, à un serveur ou à une couche. Typiquement, les données spatiales d'une zone, location (XYZ ou données GPS), surface, peuvent servir par exemple à la génération de cartes. Ce type de datascapes constitue une interprétation, une représentation de données directement disponibles, sans aucun traitement particulier des ces dernières. L'utilisation brutes de ces données ne peut générer que des datascapes très simples, évident à lire et qui sont souvent produits dans un but purement fonctionnel.

Dans un second temps, il est possible de considérer le fait que la mise en place ainsi que le suivi d'un système du type map_I.T. va générer un nombre important de données qui vont venir s'ajouter aux données initiales principalement liées au contenu diffusé par le système. Ces données additionnelles peuvent s'avérer être également intéressantes pour la génération et la diffusion de contenu. Au delà de la simple utilisation fonctionnelle (analyse du système), ces données que nous qualifierons de 'brutes' puisque résultant directement du suivi en temps réel du système sont porteuses d'information qu'il peut être utile de mettre à disposition des utilisateurs. Dans le cadre d'un environnement multi-utilisateurs par exemple, il est souvent intéressant de connaître la répartition des personnes connectées à l'intérieur du monde partagé. Ce type de datascapes permets non seulement de visualiser l'activité du système mais également d'analyser le comportement des utilisateurs. Avec une approche conceptuelle appliquée au traitement des données prises en compte, cela peut aboutir à des représentations plus ou moins évidentes à lire, dont l'intérêt est plus artistique que fonctionnel. Les données sont perçues de manière abstraite, le flux (évolution dynamique) prenant le pas sur le sens qui reste cependant aisément accessible.

Les datascapes décrits jusqu'ici sont limités par le fait que ces derniers sont souvent générés à partir de quelques paramètres seulement (1, 2, 3, voir éventuellement 4 dimensions). Ces paramètres étant directement extractibles des bases de données permettant de définir les zones et les couches de map_I.T. ainsi que leur suivi. Le volume important de ces données empêche généralement de distinguer toute corrélation éventuelle, certaines corrélations étant purement et simplement indiscernables sans une inspection poussée (algorithmes) des données. En analysant ces bases de données, des liens vont apparaître, qui à leur tour peuvent servir de sources d'information pour générer des datascapes. Ces liens, autrement appelés '*règles d'association*' [34], peuvent être déduits automatiquement des bases de données. Des algorithmes existent [35] [36] qui peuvent extraire différent type de '*règles d'association*'. L'avantage particulier qu'apporte la considération de ces '*règles d'association*' est que cela va permettre de générer des datascapes à partir de n-tuples d'information (n dimensions, avec n > 4) très facilement. Les datascapes produits à partir de ces règles vont permettre de visualiser certaines caractéristiques

(principalement liées au comportement du système et des utilisateurs) imperceptible sans une analyse en profondeur des données. De tels systèmes sont utilisés par exemple pour l'analyse du comportement des consommateurs dans les grandes surfaces. Cela peut conduire par exemple à l'optimisation de la répartition des marchandises dans les rayons. Mais ce sont ici des applications très terre-à-terre par rapport à l'usage qu'il pourrait être fait des ces mêmes techniques dans le cadre de map_I.T, avec une approche conceptuelle.

La prise en compte de n dimensions peut rendre les datascapes complètement abstraits dans ce sens qu'il devient normalement impossible de suivre ou de deviner un quelconque processus de calcul (traitement de données) juste en visualisant l'évolution de ces datascapes. Ils bénéficient alors d'une existence propre, là où les datascapes plus simple n'existent qu'a travers le flux de données (devinable) qui les nourrit.

Appliqué à la 3D, cela peut donner naissance à des mondes avec leurs caractéristiques propres, nécessitant un investissement de l'utilisateur afin d'en comprendre le fonctionnement et l'intégrité. Jusqu'aux modes d'interactions (et de navigations), tout dans ces datascapes peut faire l'objet d'une évolution particulière, obligeant l'utilisateur à s'adapter en permanence. On s'oppose ainsi aux procédures commerciales classiques où tout est fait pour contenter les besoins/envies/requêtes de l'utilisateur, de peur de le perdre. Dans ce contexte particulier, il s'agit plus particulièrement de proposer de nouvelles expériences (dont la véracités reste à étudier) aux utilisateurs par la découverte de nouveaux espaces à n dimensions.

Il s'agit alors d'appliquer ces notions générales au datascape particulier décrit dans map_I.T. Les containtes sont clairement d'ordre géographique, le datascape map_I.T. s'approchant par mimétisme au volume englobant du site de l'epfl. Les données formant le datascape seront dans un premier temps directement cantonées spatialement. La position de ces documents fera partie des paramètres initiaux caractéristiques. De manière évidente le temps peut être pris également en compte (interprété comme un paramètre de proximité par exemple). D'autres paramètres fixes peuvent être ajoutés à la liste : une personne, une discipline, une fonction, etc.

Viennent ensuite s'ajouter des paramètres de second niveau correspondant aux données additionnelles évoquées plus haut. Ces paramètres vont venir mettre en relief des relations entre les documents, permettant ainsi une restructuration du datascape et/ou la création d'associations hors contrainte. Des textes peuvent être 'rapprochés' par exemple si leur contenu concerne un même sujet par l'intermédiaire de paramètres de proximité, système basé sur la classification floue [37]. Ces paramètres sont mis en place/calculés à l'insertion du contenu dans le système map_I.T. et peuvent faire l'objet de mise à jour à tout instant (dynamique). Par l'intermédiaire d'un choix de mode de visualisation du datascape, tout ou une partie de ces paramètres vont être considérés afin de déterminer l'agencement des données ainsi qu'un mode de navigation (typiquement 2D ou 3D). Des données dynamiques peuvent venir grossir la liste : la fréquence et le temps de consultation définissant un taux d'occupation, la manière elle même via laquelle les données ont été accédés (réseau classique ou sans fil, ordinateur, portable ou assistant personnel). Autant de notions qui peuvent faire apparaître une proximité

entre des données et donc définissent des possibilités éventuelles de ré-aménagement du datascape.
Comme évoqué précédemment de manière générale, le comportement même de l'utilisateur peut générer des paramètres qui vont avoir un effet sur ce qu'il percevra du datascape et comment il le percevra (les sujets côtoyés, le temps passé dans un système, la quantité de documents consultés, le type des données consultées, etc…). Le suivi des habitudes de l'utilisateur peut permettre de modéliser un profile. Ce profile peut alors être usité comme filtre terminal, ultime traitement des données avant d'être présentées au requérant. Ce formatage va ainsi mettre en avant les données le plus en adéquation avec la personnalité de l'utilisateur. L'avantage réside ici dans le fait que ce biaisement des données est indépendant du mode choisi de représentation du datascape.

L'ensemble du système peut être perçu comme un rayon d'électrons (le flux de données) passant à travers des grilles (filtres amplifiant ou diminuant l'intensité du rayon) et des champs magnétiques (déviation des données, attribution spatiale) successifs. Le datascape map_I.T. permet l'utilisation de tous les systèmes d'indexation possibles sans perdre ses caractéristiques, préservées par l'énonciation de contraintes initiales.

[34] G.Piatetsky-Shapiro, *Knowledge Discovery in Databases*, AAAI/MIT Press, 1991.
[35] T.Elomaa, H.Mannila, H.Toivonen, *Principles of Data Mining and Knowledge Discovery*, 6th European Conference, PKDD 2002, LNCS 2431, (Eds.). Springer 2002.
[36] Takashi Okada, *Datascape Survey Using the Cascade Model*, Discovery Science, 2002.
[37] Alain Lelu, *De l'émergence des concepts*, "Les sciences de l'information : bibliométrie, scientométrie, infométrie". In *Solaris*, n° 2, Presses Universitaires de Rennes, 1995.

// map_IT -- //
// projet de campus digital pour l'EPFL // fabric | ch // www.fabric.ch // mai 2003

2002 +++
gestion de niveaux et zones

map_I.T. gestion
niveaux et zones, description
niveaux et zones, gestion
net services

map_I.T, gestion de niveaux et zones

map_I.T. est organisé sous forme de couches successives. Suivant le contexte, une couche peut être résumée à un thème, une fonctionnalité, etc… Nous allons maintenant détailler comment peuvent être créées et contrôlées ces couches.

Une couche est contrôlée par une base de données relationnelles. La création ainsi que la configuration initiale d'une couche se résume à entrer un ensemble de paramètres dans la base de données. Cette phase est clairement réservée au service informatique de l'EPFL. Par la suite, une fois la nouvelle couche totalement spécifiée, il sera donné à d'autres unités (responsable informatique d'une unité de recherche, une secrétaire etc…) la possibilité de contrôler une partie de la nouvelle couche en intervenant sur des paramètres de plus haut niveau via l'application de simples filtres d'accès.

L'addition d'une nouvelle couche à map_I.T. peut être résumée en 2 phases principales. La première phase illustrée par la Figure 15 permet de réaliser une correspondance entre la couche immatérielle et la couche physique de map_I.T. Une couche va être constituée d'un ensemble de zones. Chaque zone est définie par l'ensemble des serveurs qui vont participer à la diffusion (via le réseau) de cette zone. Une fois la ou les différentes zones créées, il s'agit d'attribuer à chaque serveur une ou plusieurs tâches. Une tâche peut être identifiée à un service qu'elle sera capable de rendre. Evidemment, un serveur peut être en charge de plusieurs services, de même qu'un serveur peut appartenir à plusieurs zones simultanément (dans le cas d'un service peu coûteux en ressources et qui peut être commun à plusieurs zones par exemple). Cette phase d'attribution des serveurs aux zones, de même que l'attribution des services aux serveurs peut se faire par l'intermédiaire d'une interface graphique de haut niveau. Elle permet de mettre en place et d'initialiser un ensemble de bases de données relationnelles qui vont permettrent la gestion de cette nouvelle couche.
De manière évidente, une zone peut être constituée d'espaces physiquement non contigus. Dans le cas de définition de zones à vocations administratives, les zones pourront correspondrent exactement aux unités de recherche existantes, cette configuration se rapprochant probablement du découpage et du management actuel des ressources.

Une fois cette première phase validée, elle peut enclencher un système de vérification des serveurs sélectionnés, de même qu'engendrer des processus de mise à jour ou d'installation de services. Il s'agit ici de faire en sorte que les serveurs sélectionnés pour effectuer certaines tâches soient à même de remplir cette tâche lorsque la nouvelle couche sera fonctionnelle et accessible par les utilisateurs. La mise en place d'un système de mise à jour et d'installation des services est importante dans la mesure ou elle peut éviter ou permettre de minimiser les interventions physiques sur chaque serveur.

La base de données relationnelles liée à une couche peut également être utilisée par les services eux même. Chaque service peut être conscient de l'existence des autres services, et à partir de là, un service peut également devenir requérant auprès d'un autre service. Grâce à un méta-protocole de communication, chaque service peut entrer en contact avec un autre service dont il n'a aucune connaissance particulière. Les premiers

échanges inter-services permettent au service requérant de prendre connaissance du protocole de communication du service qu'il veut utiliser. Une fois cette phase écoulée, le service requérant apparaîtra comme un client classique pour le service cible. Chaque service peut gérer un cache local dans lequel il peut mémoriser les services avec lesquels il a déjà établi un contact et leurs protocoles associés. Ce cache optimisera une future requête avec un service déjà connu.

Il est possible à chaque instant de connaître l'état de chaque service. Un service peut être à même d'analyser son état courant. En cas de surcharge, de dysfonctionnement, le service peut décider de manière autonome des actions à initier afin de résoudre le problème. En cas de surcharge par exemple, le service peut faire une demande de duplication sur un autre serveur. Cette autonomie des services peut simplifier la maintenance du système de même que lui insuffler une vie propre.

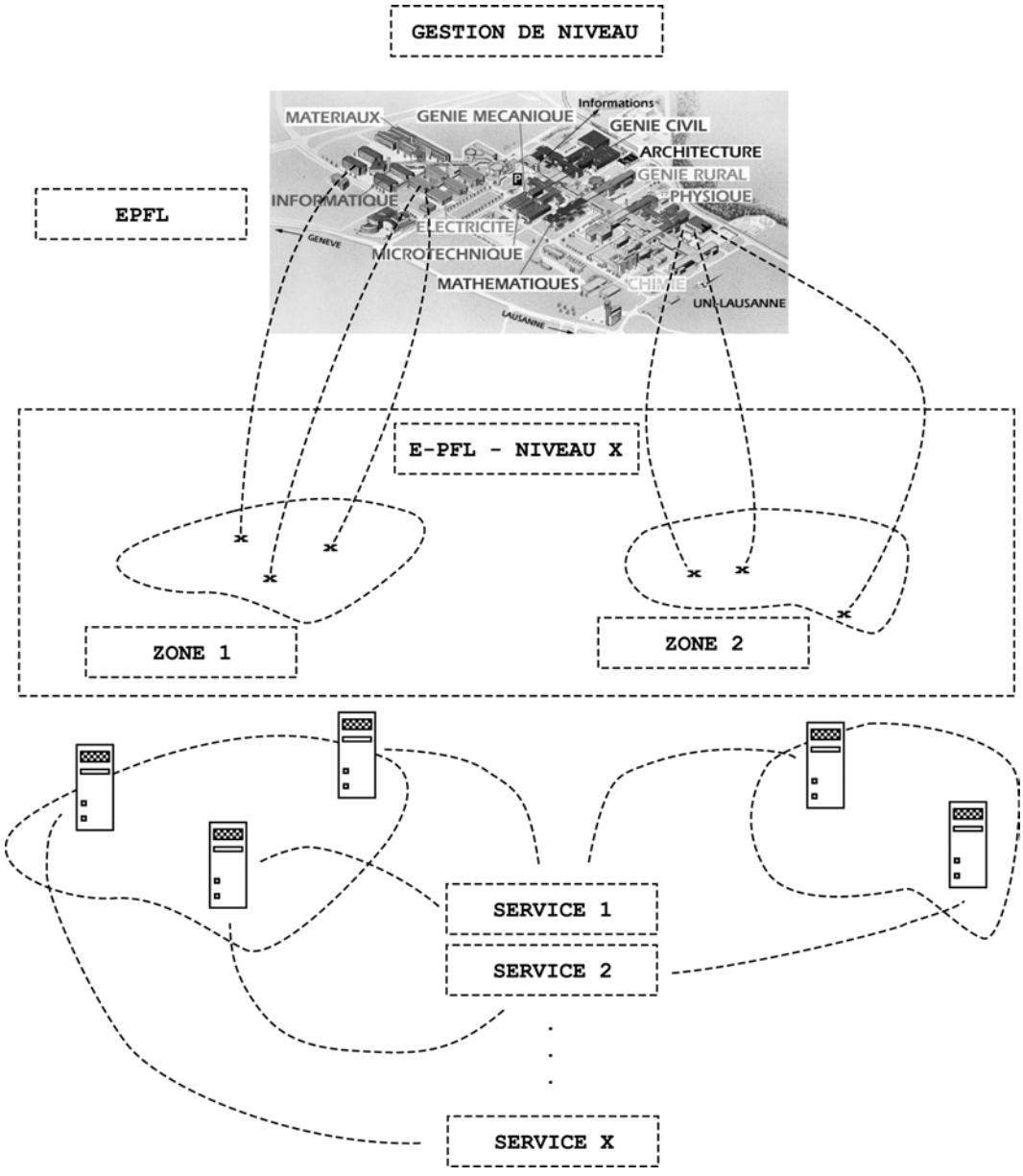


Figure 15 – création d'une nouvelle couche (niveau) en spécifiant ses zones, puis attribution des services aux machines constituant les zones.

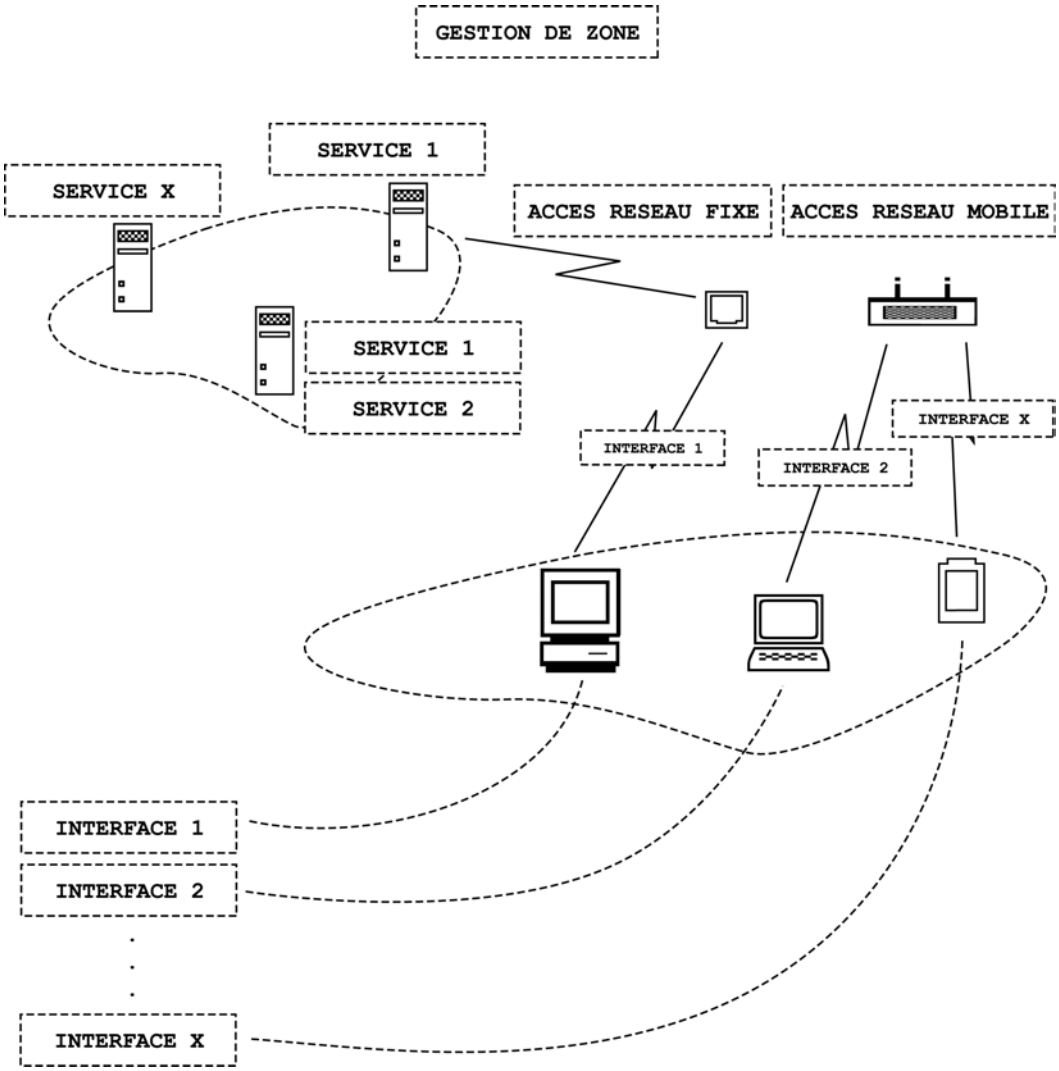


Figure 16 – Gestion des zones

La gestion des zones d'une couche peut être déléguée à un ensemble de responsables de zone. Il est alors nécessaire de configurer les services de chaque serveur appartenant à la zone. Cette configuration inclus un aspect purement fonctionnel (configuration de bas niveau) et un aspect graphique. Ce dernier va définir la manière dont les services sont visibles par les utilisateurs. L'aspect graphique ou interface (Figure 16) va dépendre de la plate-forme utilisée pour accéder au service, du type de service (service sur demande, service automatique et continu, service à accès restreint, etc…), du type d'utilisation souhaité, etc…

Les interfaces vont interpréter les informations délivrées par les services afin de les transformer en données visualisables. Ces interfaces peuvent faire l'objet de développements particuliers ou même de recherches. De même que les services, chaque interface peut être soumise à un système de validation (test de conformité au système). Si ce test s'avère concluant, l'interface peut alors faire partie d'une base de données, rendant disponible toutes les interfaces développées à tous les responsables de zones. Là où les services demandent un contrôle plus poussé (éviter les problèmes de sécurité, etc…), les interfaces peuvent

faire l'objet d'un processus de développement beaucoup plus ouvert. Une interface mise au point pour un contexte précis peut s'avérer utile dans un autre contexte, pour une autre unité de recherche. L'interface apparaît alors comme un véritable filtre appliqué à un service. Par exemple, un service de recherche de livres lié à une bibliothèque peut apparaître sous la forme d'un ensemble de requêtes à remplir et à soumettre. Ce même service peut venir « nourrire » un monde en trois dimensions permettant de visualiser le volume de requêtes traitées par le service et ce sous des formes plus ou moins abstraites, sans que le rapport soit forcément explicite.

Services based systems

Microsoft .Net:

It is a development environment and desktop applications. Microsoft integrated the .Net functionalities in all its "XP" products line. They allow the end-user to access and to interact with web services. The VisualStudio .Net is the development environment that allows, in a simplified manner and by using different programming languages, to develop .Net web. Microsoft proposes a set of servers dedicated to different types of web services (business, database, Hailstorm), one of them being specifically dedicated to mobile devices. .Net seems generic enough to generate any kind of web services, with a predilection for the e-business. A .Net application can only interact with services that exist during its creation. In other words an existing service can not connect to a service that is younger than himself without updating its source code (e.g. download of the younger service protocol through web enabled functions). The references to communicate with associated web services are integrated in the code of each web services. A service cannot communicate with another one if it doesn't contain/know the communication mode/protocol. It implies an inability of the system to react in real-time to the system evolution. All situations have to have been planed when the web service was created.

http://www.microsoft.com/net/

WebSphere IBM:

WebSphere is a development environment to create web sites with dynamic contents. By encapsulating one or more database interfaces and a development language (JAVA), it allows the creation and the installation of web sites through a graphical interface. Updates and maintenance are made easily without having to shutdown the entire web site. The notion of web-services is here strictly commercial. The idea is not to simplify the life of the user, but to ease the conception of the website. WebSphere is absolutely not dedicated to new devices.

http://www.ibm.com/websphere/

SunOne / J2EE (Java 2 Enterprise Edition):

It is a development environment with a suite of application (SunOne) dedicated to specific web service tasks (Messaging, calendar, database ...). The whole architecture is based on classic standards like **XML, SOAP**. The development environment is called SunOne Studio and relies essentially on the javabeans technology (linked to JAVA programmation language) from Sun. Wireless devices can be integrated through dedicated javabeans. The J2EE is powerful enough to produce any kind of web services, and is optimised to create e-business web services. Based on our knowledge of the system, SunOne has the same restrictions on dynamic interactions between servers than .Net from Microsoft.

http://wwws.sun.com/software/sunone/

http://www.xml.org/
http://www.w3.org/TR/SOAP/

// map_IT -- //
// projet de campus digital pour l'EPFL // fabric | ch // www.fabric.ch // mai 2003

map_I.T.

// campus digital de l'Ecole Polytechnique Fédérale de Lausanne ----------//
// projet de master plan // fabric | ch -- www.fabric.ch -- // février 2003 //

+++

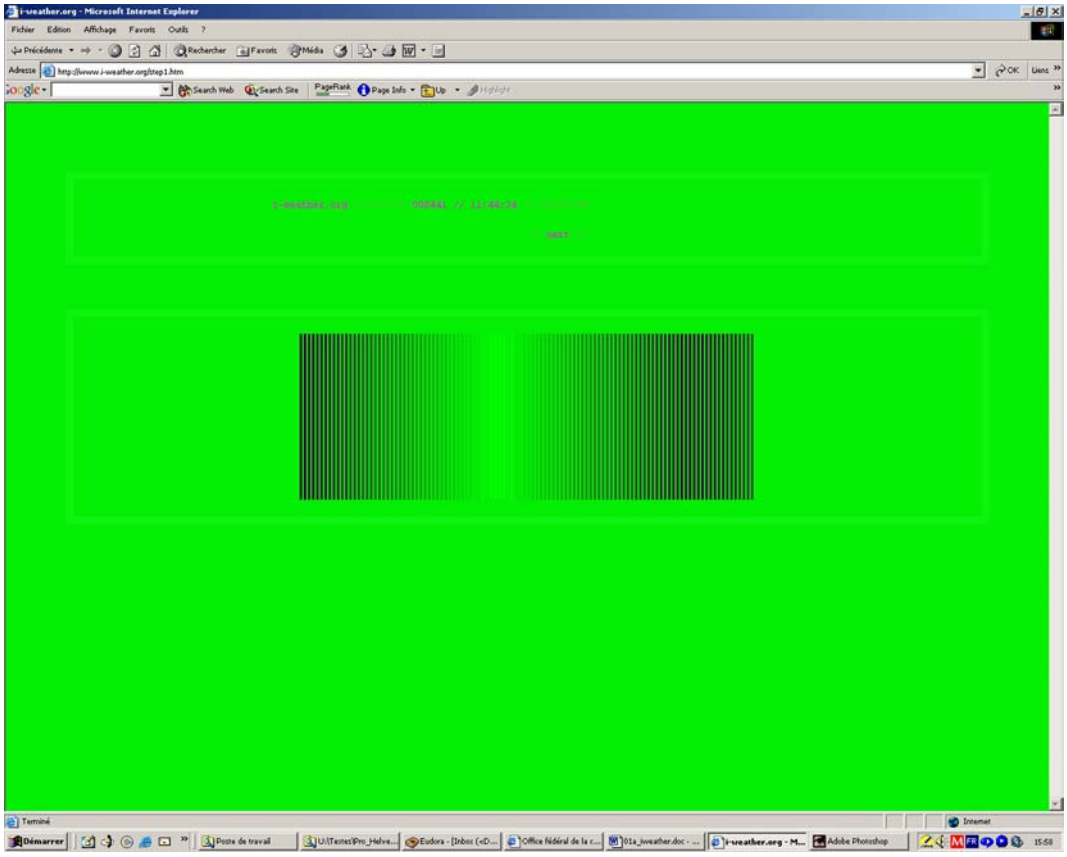
.proj:

2002 +++

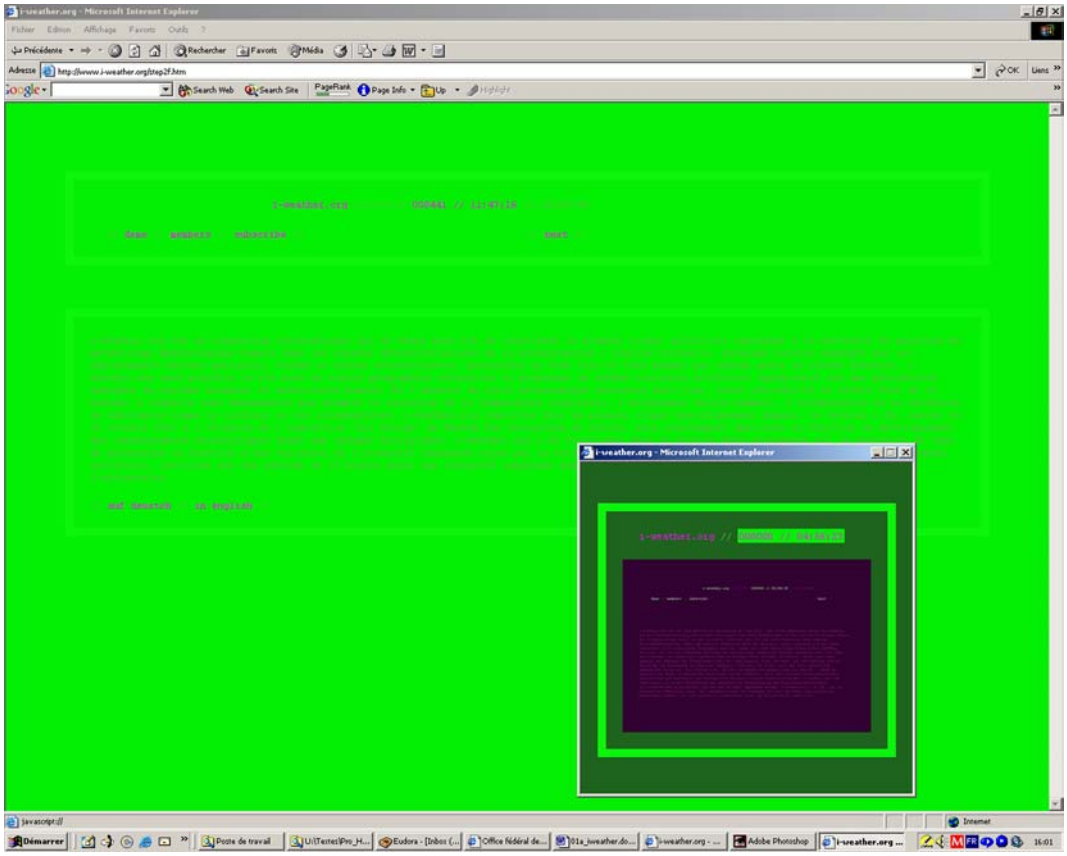
i-weather.org

en collaboration avec Décosterd et Rahm, associés

climat artificiel
climat lumineux distribué [open source]
serveur/clients multiples
architecture climatique

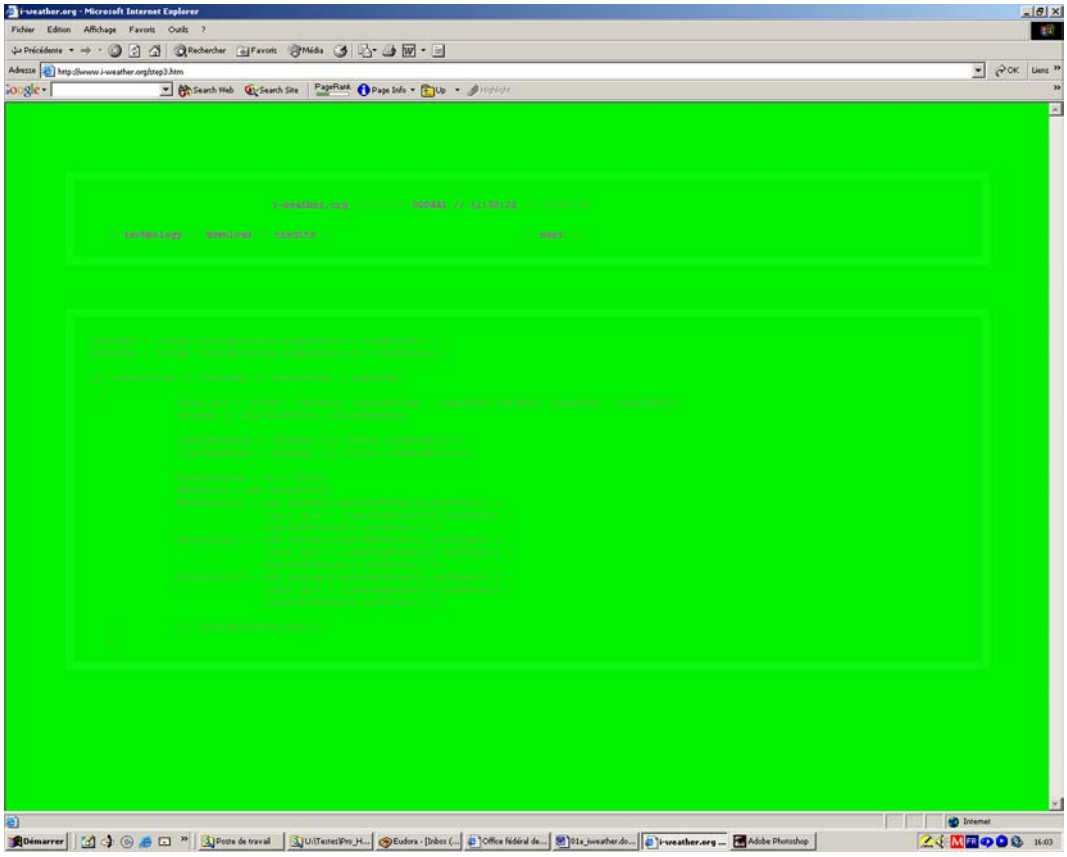


__www.i-weather.org, jour 441, heure 12:19:24 / 25:07:40, client html

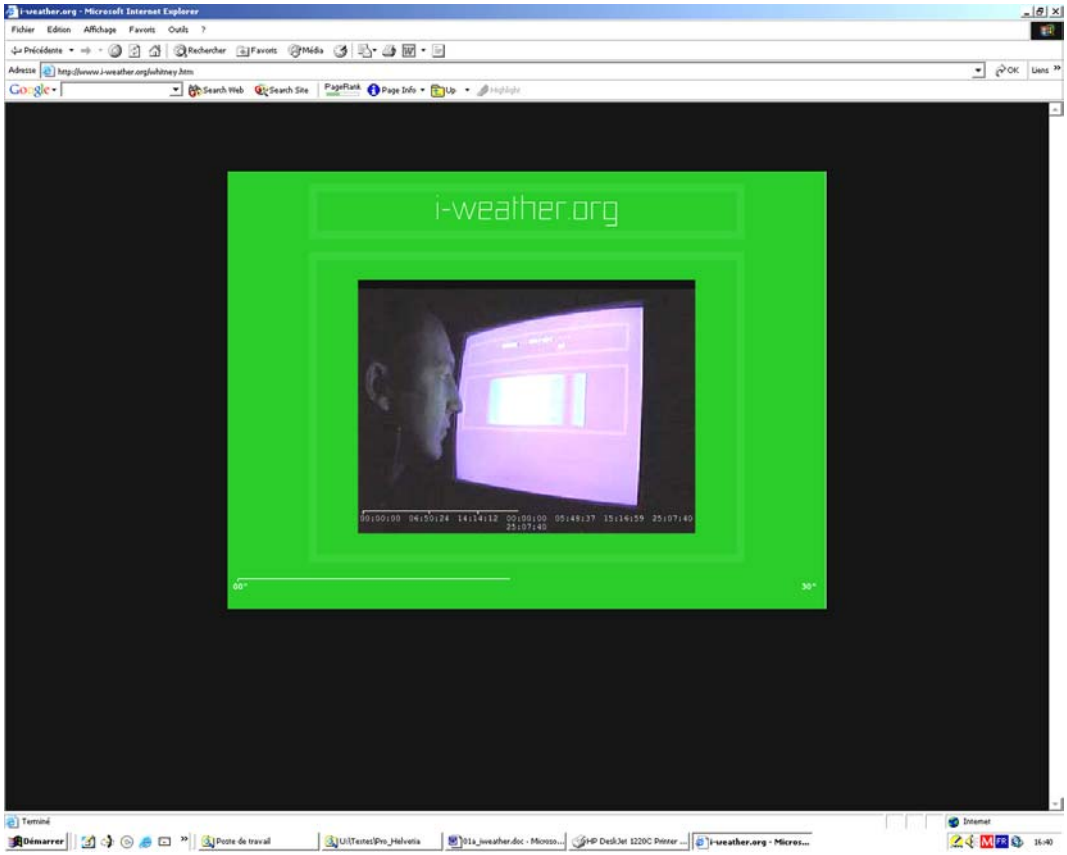


__client html et démo. Le climat oscille entre 2 tranches du spectre lumieux.

// map_IT ---------- //
// projet de campus digital pour l'EPFL // fabric | ch // www.fabric.ch // mai 2003



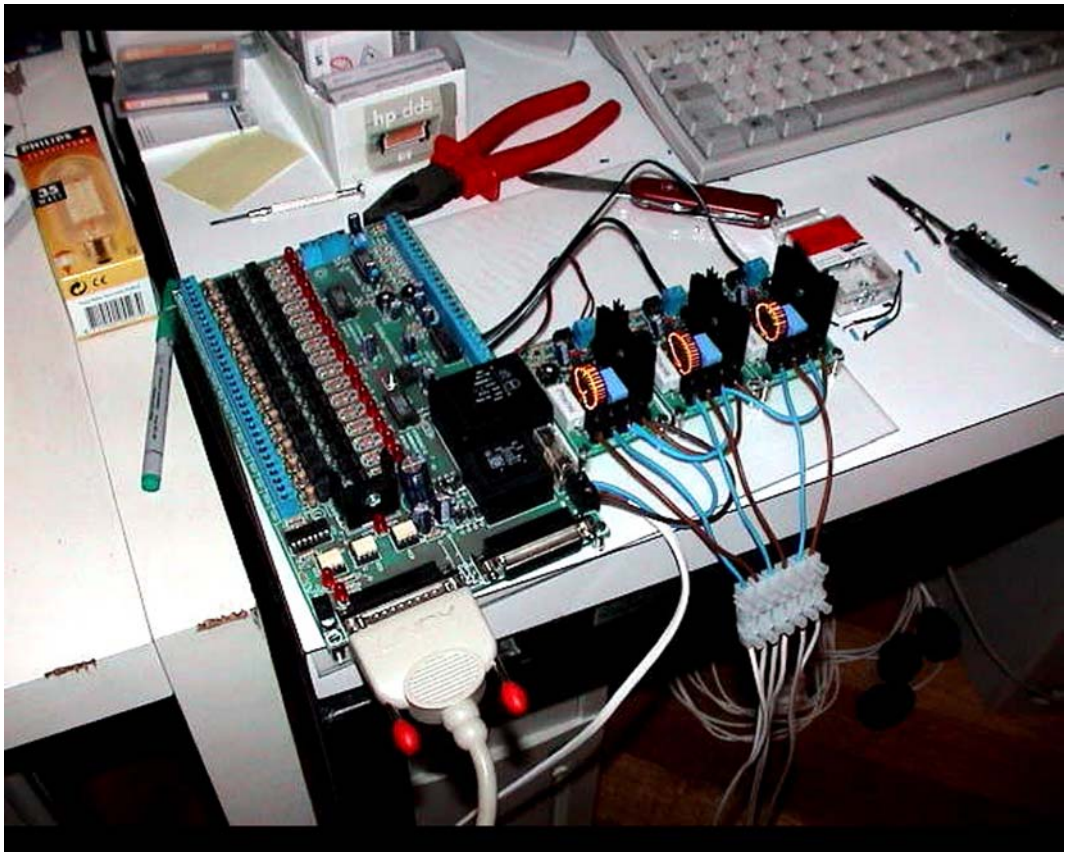
__i-weather.org est un climat électronique piloté par un serveur. Même climat distribué, à tout instant, partout



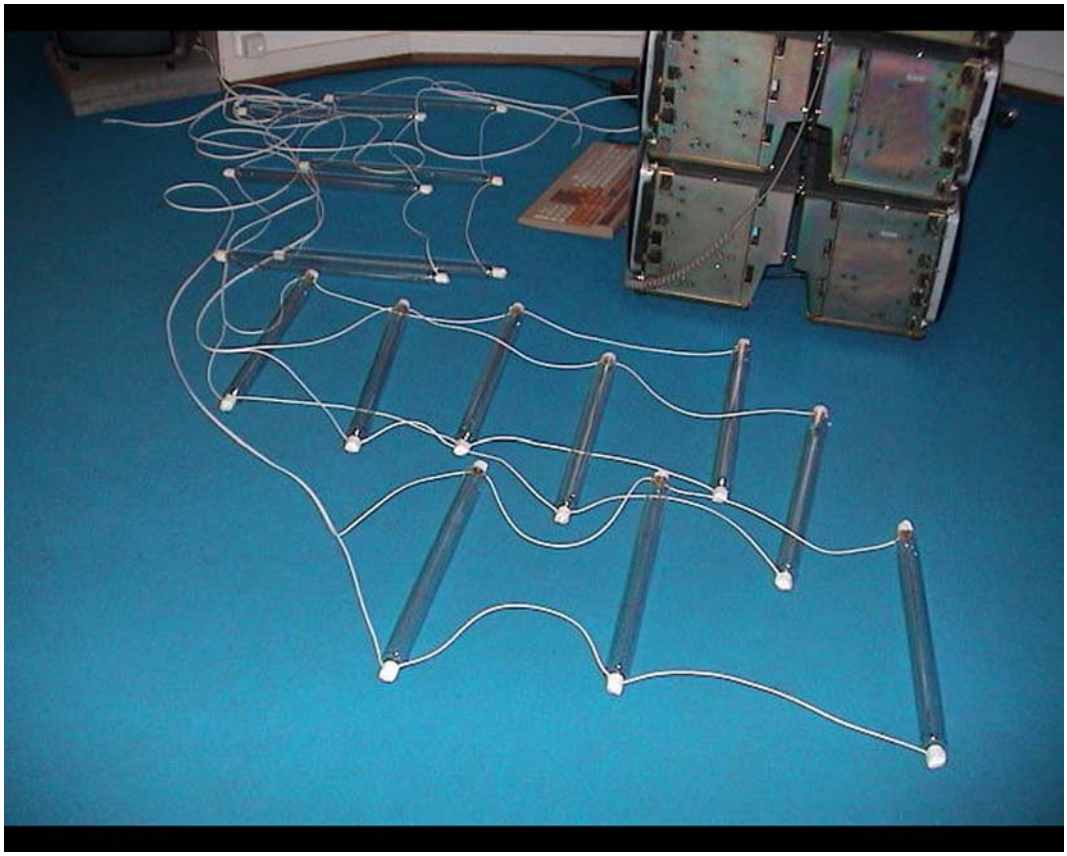
__rythmes circadiens stimulés en réseau pour personnes délocalisées, déterritorialisées.
__démo au www.whithneybiennal.org

2002 +++
i-light tool experiment

visual data flow
information lighting
Kbytes flow to lumen
light emission
light generated real space

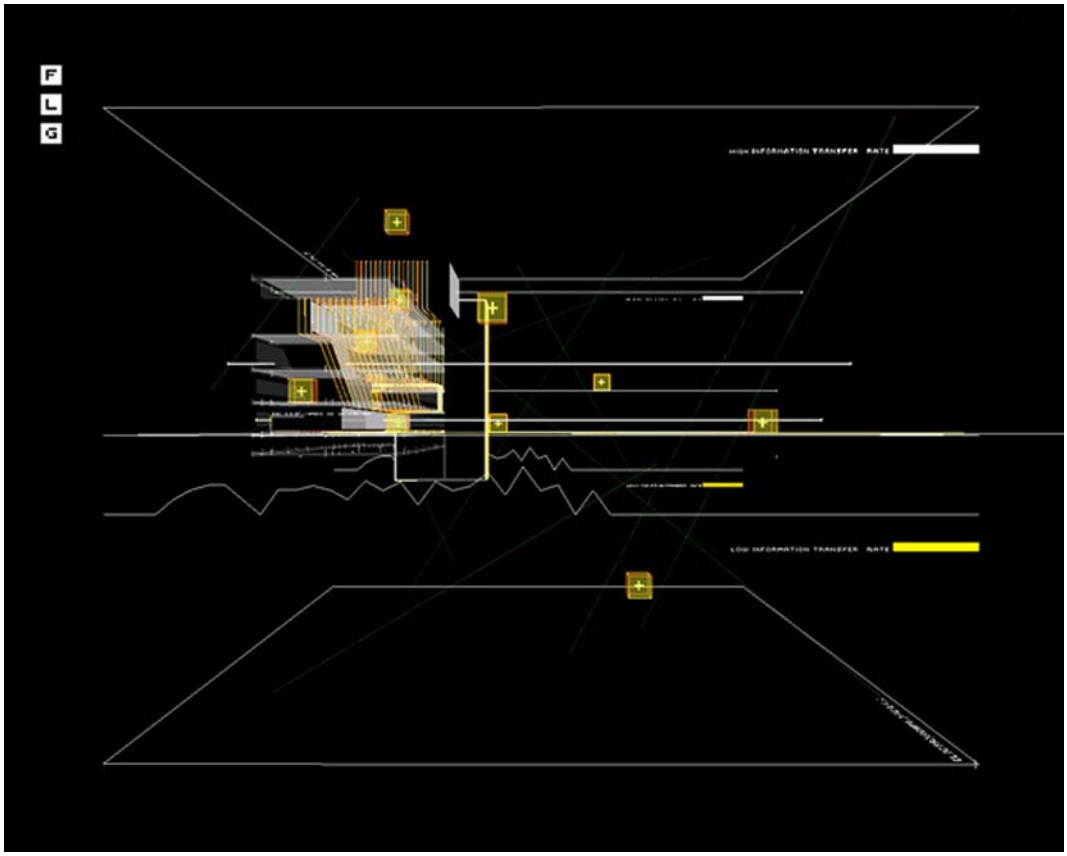


__the tool inbetween the network's data flow, the computer and the lights.

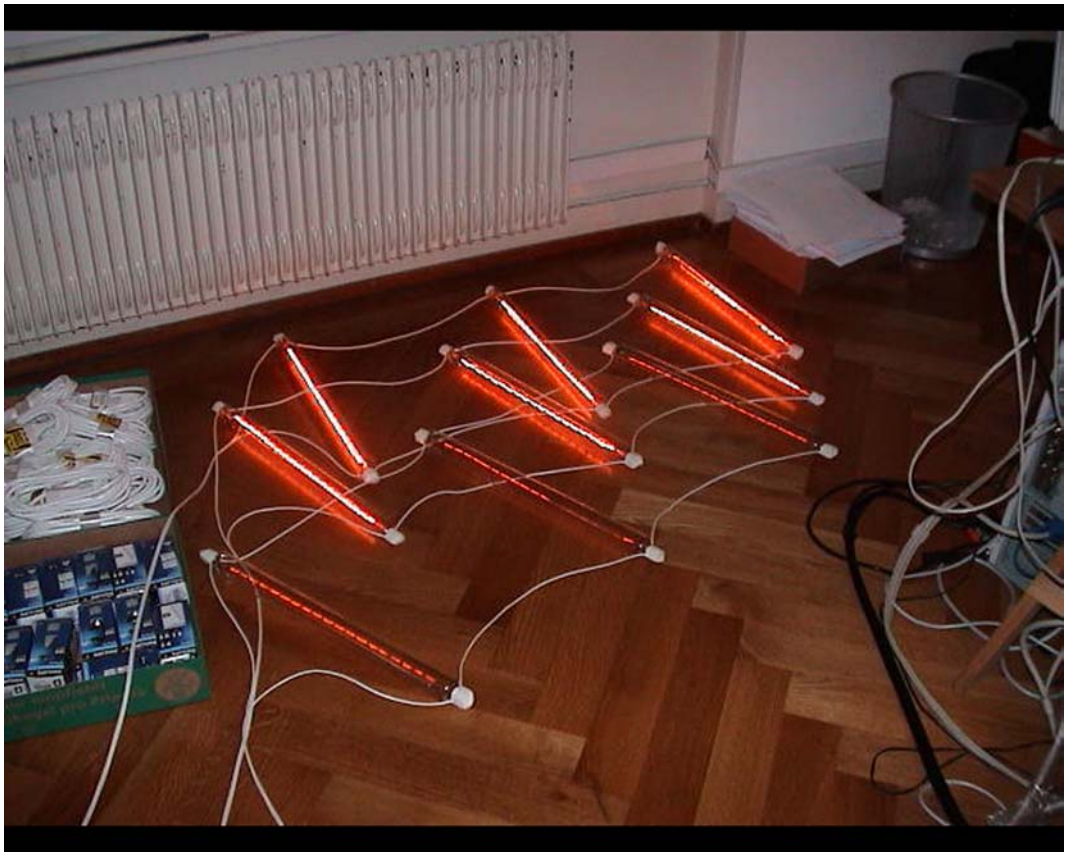


__a network of lights for beta testing.

__graph of the network's data flow [right] and digital lights driven by i-light tool [left].

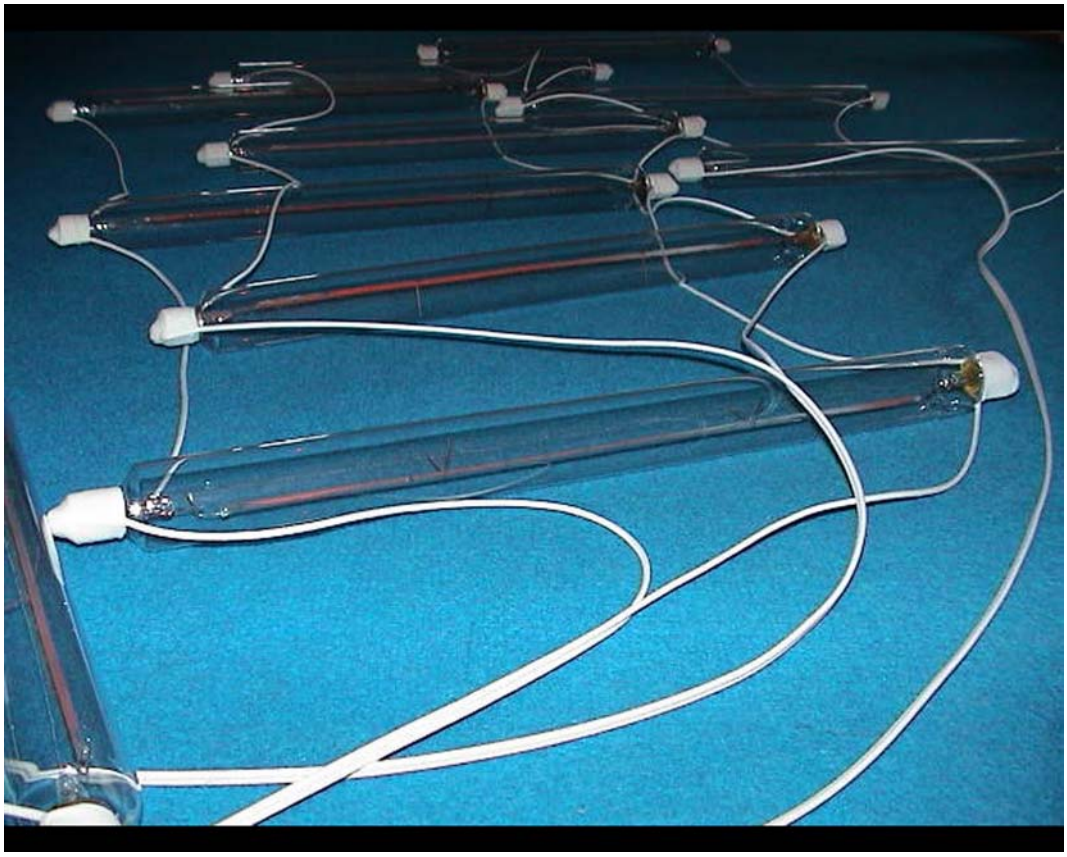


__network's data flow graph within a project and global emmissive lighting of the digital content driven by the i-light tool.
__www.electroscape.org/002



__the physical lights reacting the same way as the digital ones and translating the data flow of our network into lighting.

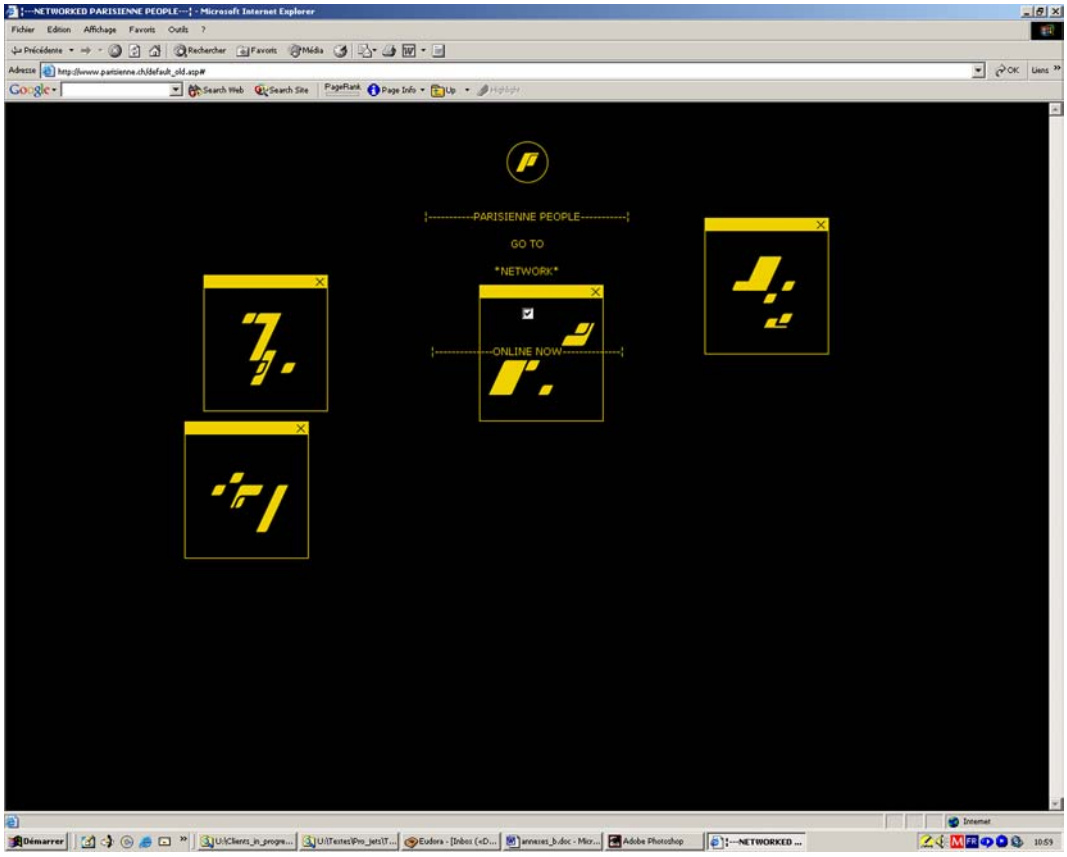
__infra red image of the server's room whith no other light source than the data flow lights. the amount of information getting through our network drives the lighting of the room containing the servers.



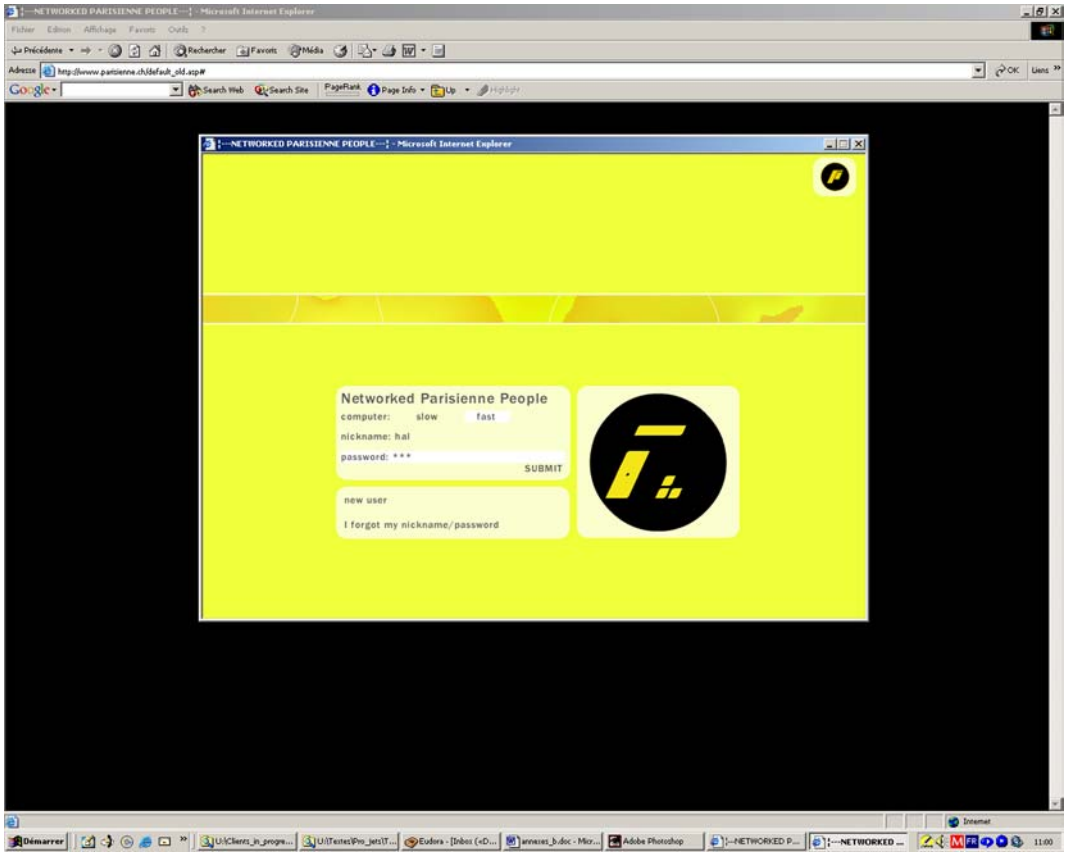
__detail on the light bulbes

2001 +++
parisienne.ch

electronic community
digital presence / parisienne
shared multi-user web territory
communication toolbox
interlaced networks

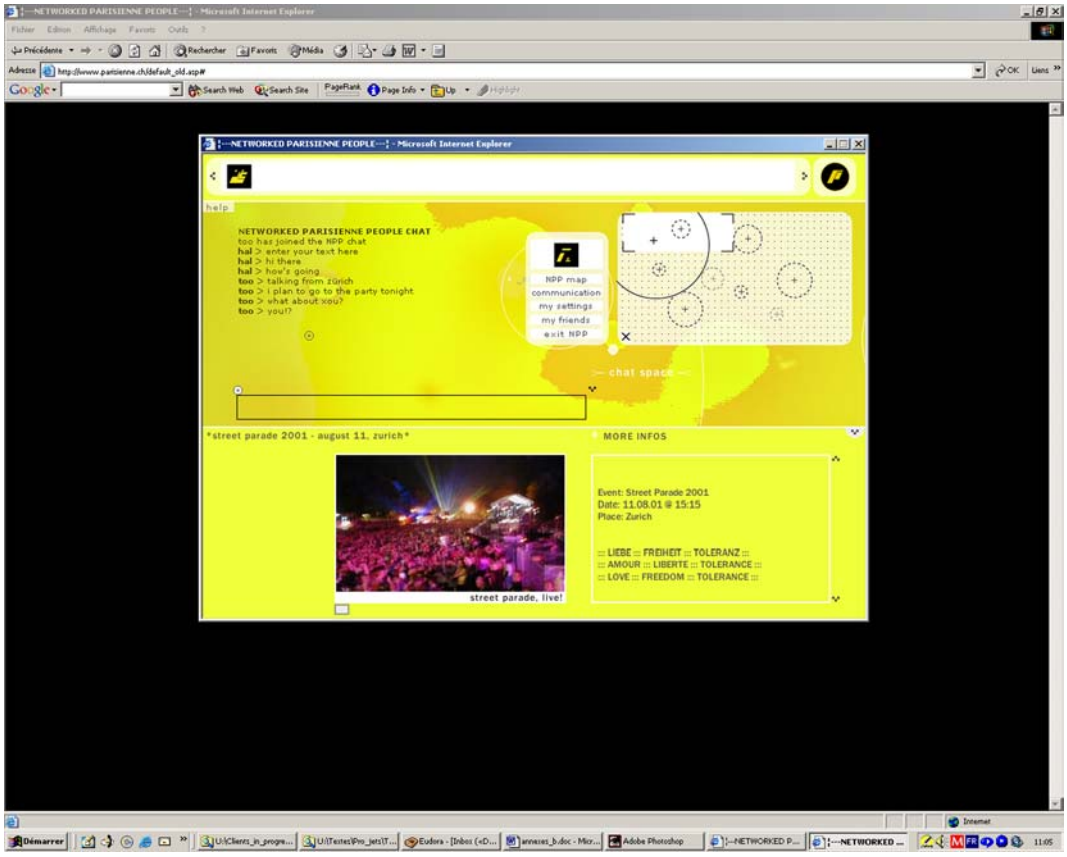


__entrance of the parisienne.ch website.

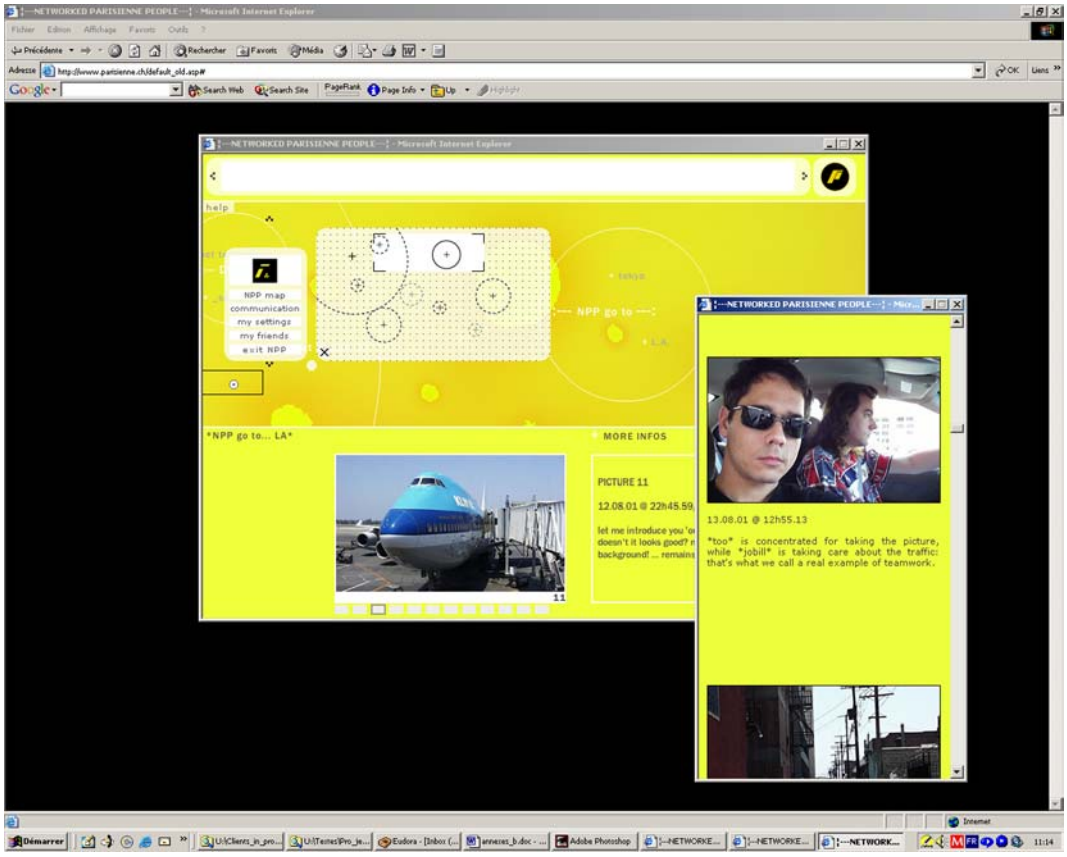


__the login [nickname, password] turns into the creation of a visual logotype. the logotype is the main part of the project as it will follow the user on several different supports.

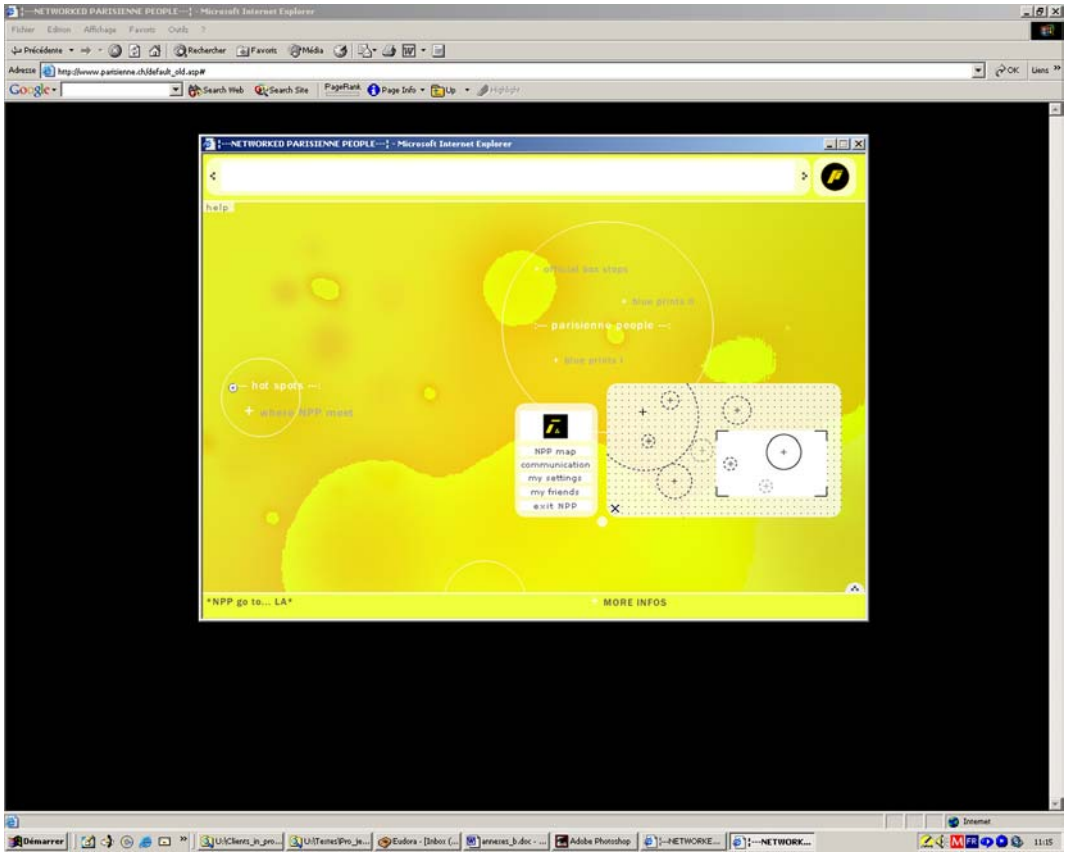
__inside the website. a pesonnal toolbox with the map of the community's territory/content [on the right] and another user conneted in tha same area. bottom part of the display is reseraved to give some preview of some accessible content.



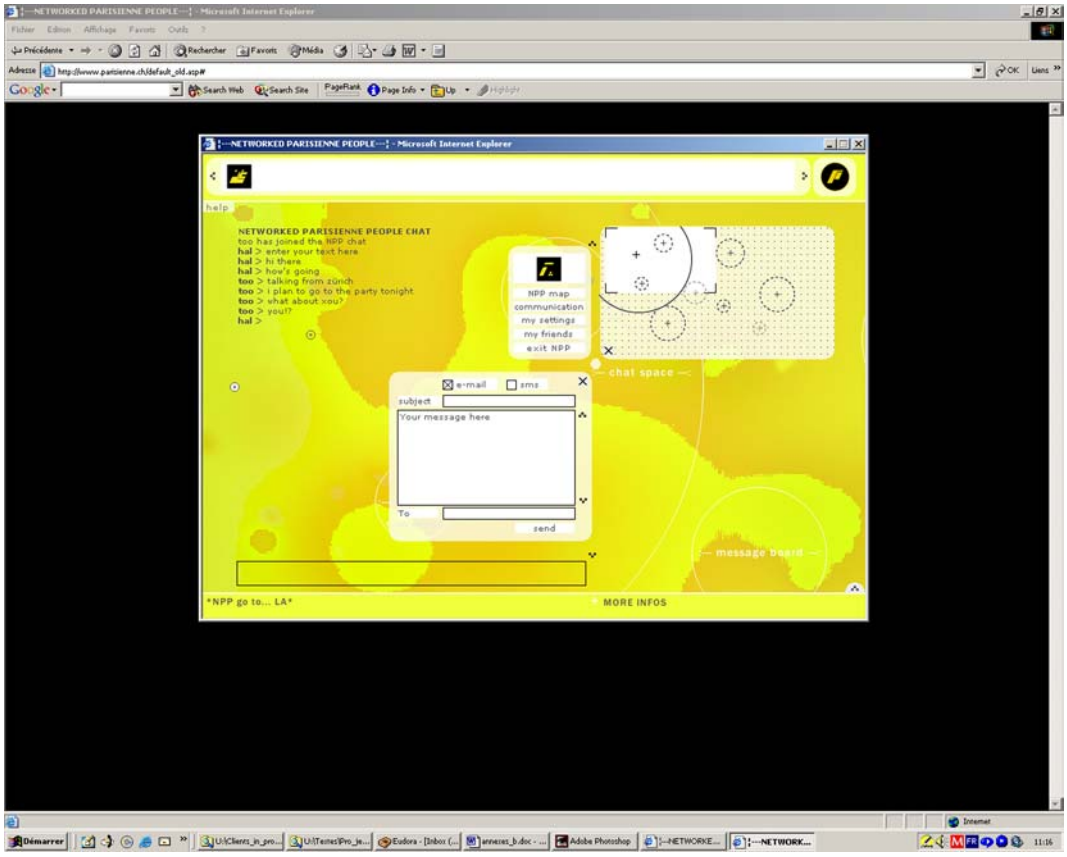
__both users have moved into the *chat space*'s part of the online community. another contextual content is displayed in the bottom part.



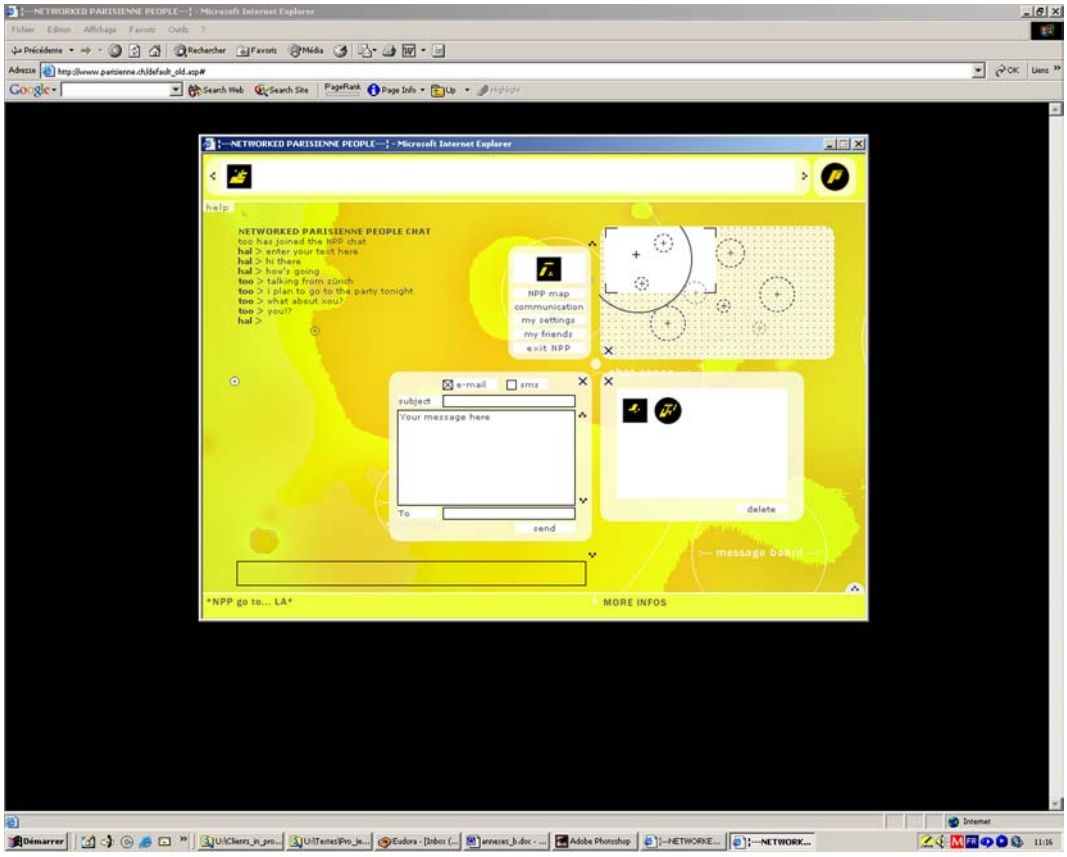
__again the user has moved in another part of the digital animated territory where he's alone. contextual content has been further opened in a specific window.



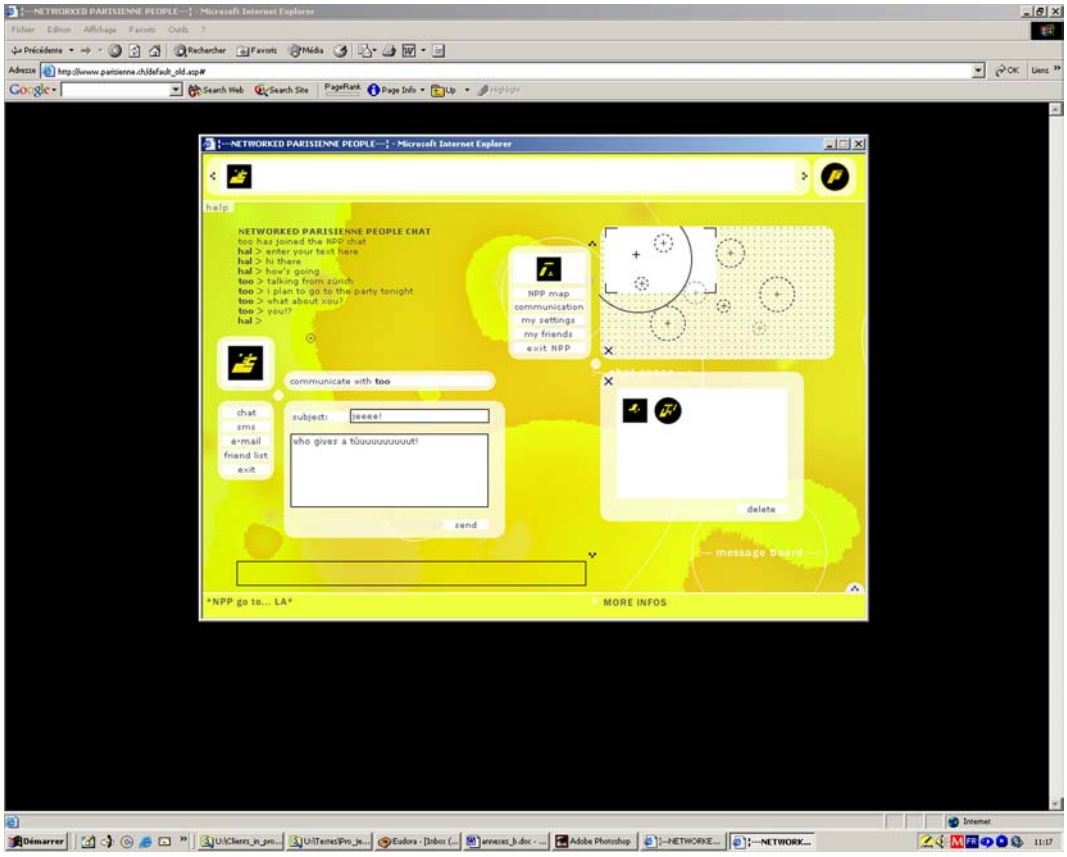
__the interface allows the user to fully open the window on the territory and use it has a visual quietly animated display. the user has alos room for other activities.



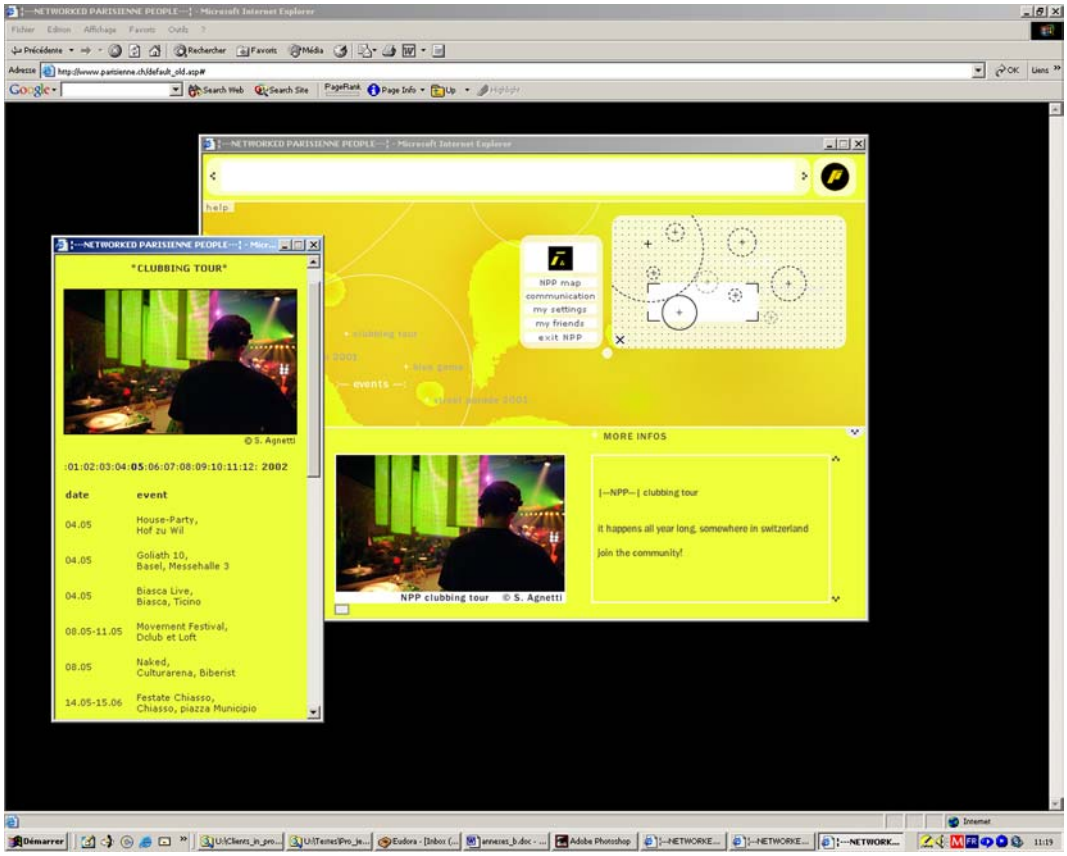
__the user opens its communication toolbox: he can write mails and sms-es and send them from within the interface and the website.



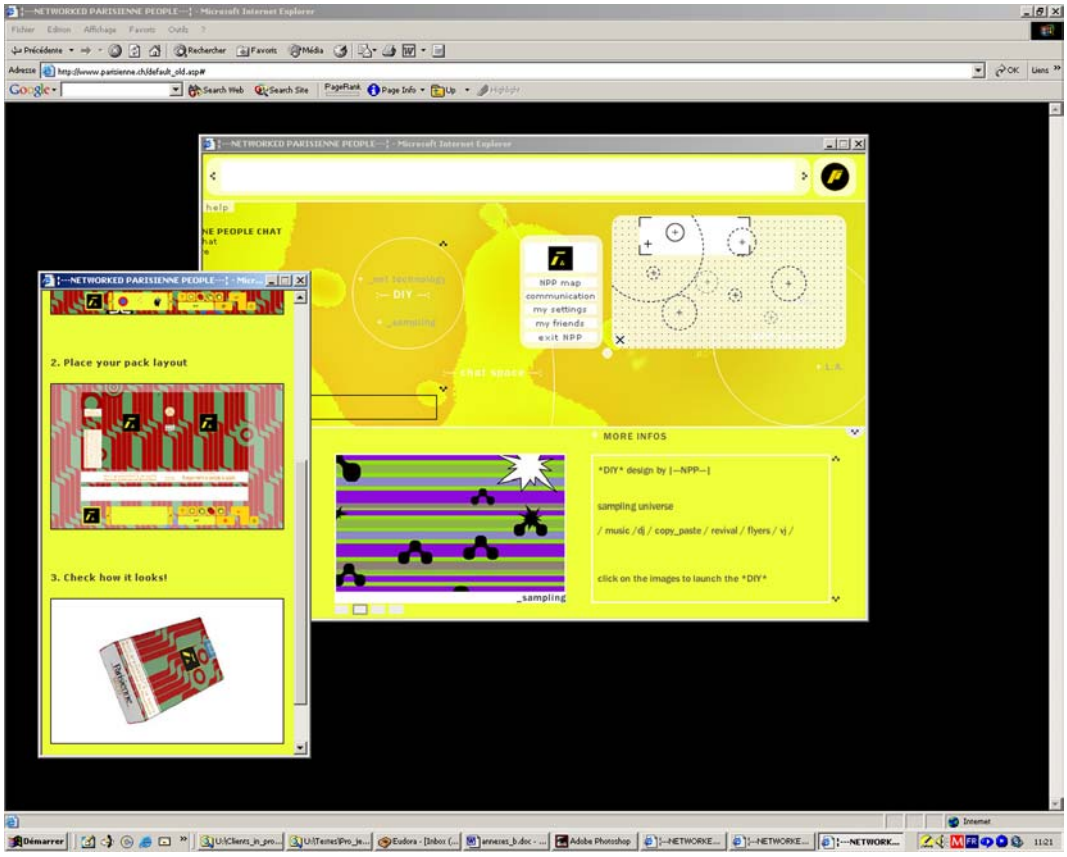
__opening the toolbox a bit further: logotypes of friends



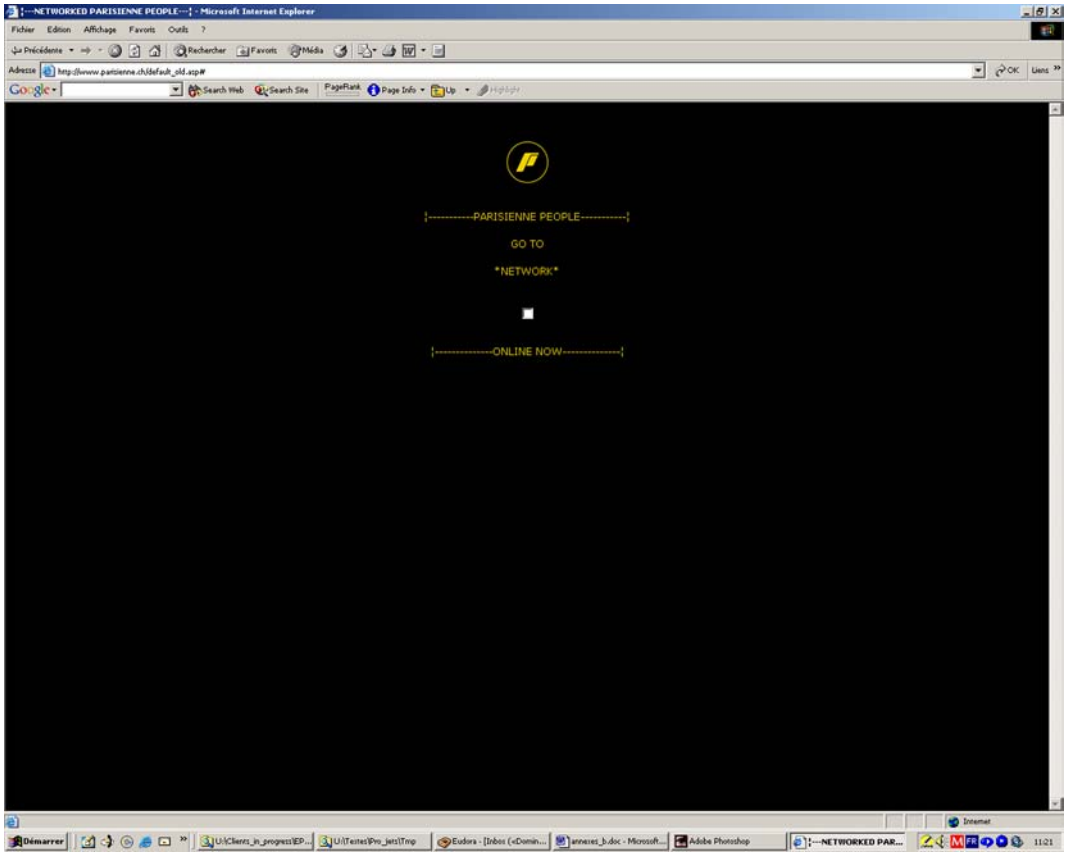
__... friends with which the user can communicate [sms, mail] even when they are not connected, or privately chat if they are.



__back to browsing in the shared and multi-user community territory.



__within the territory: access to another online project. the DIY where each user can customise an exhisting product of the company.

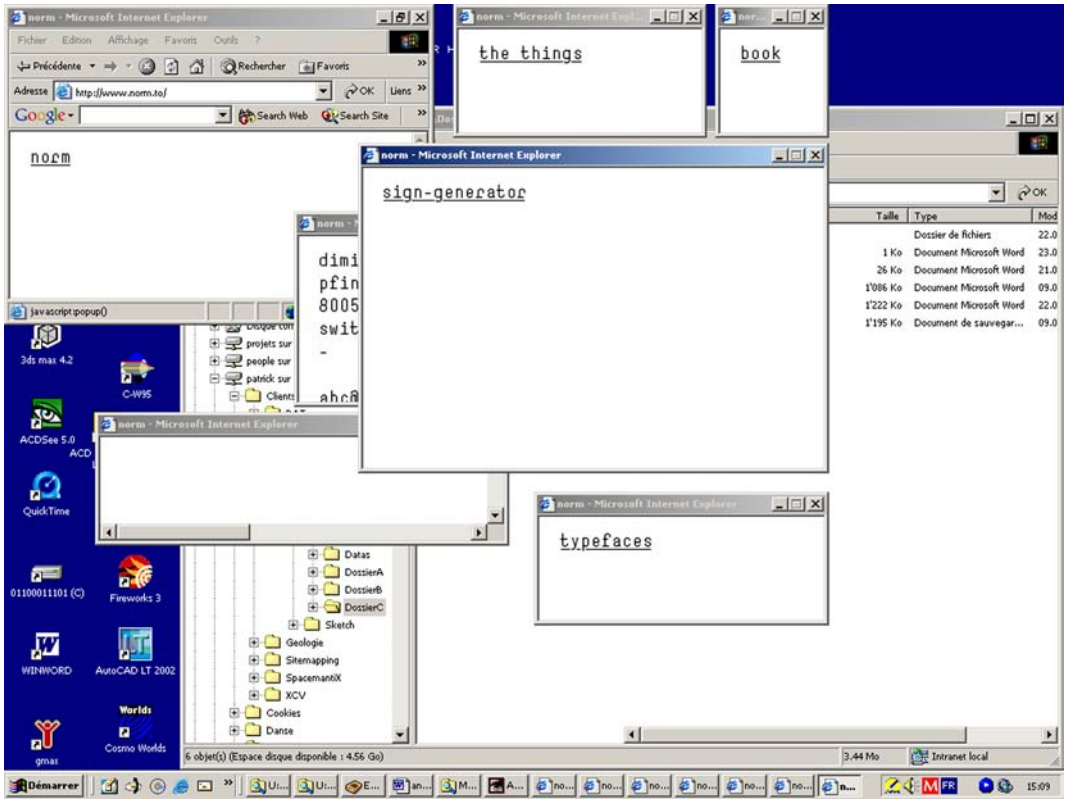


__logout and back to the front page.

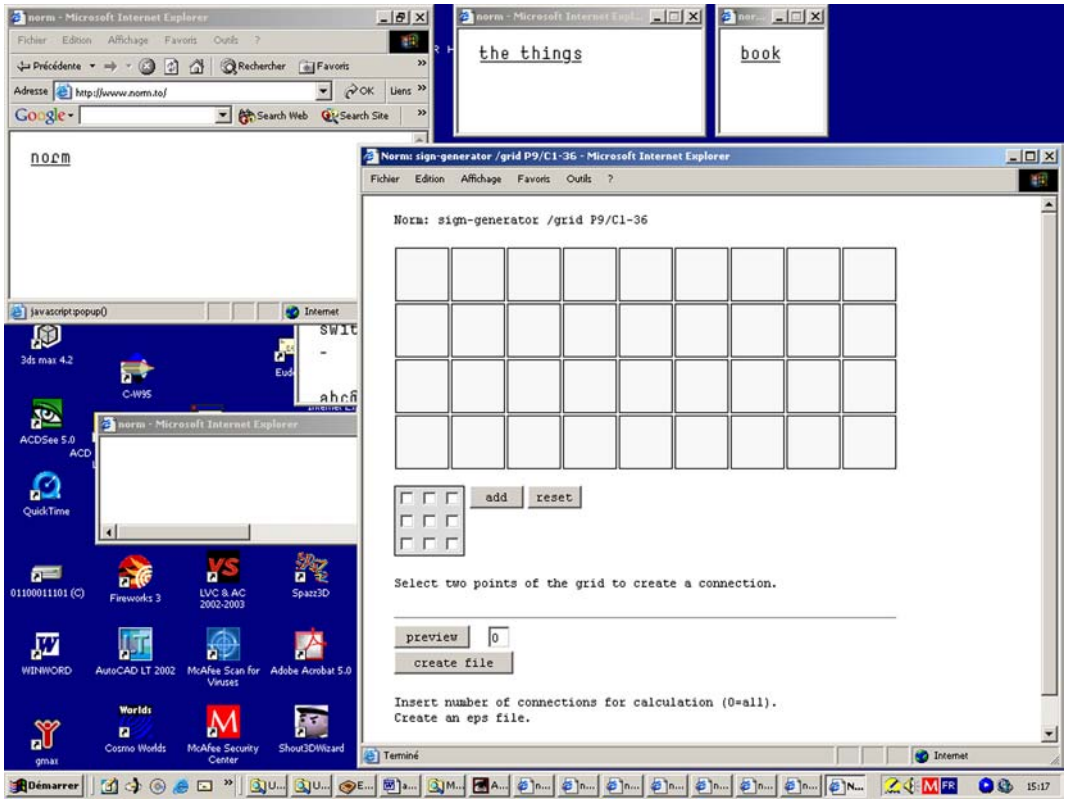
2002 +++

NORM - sign generator

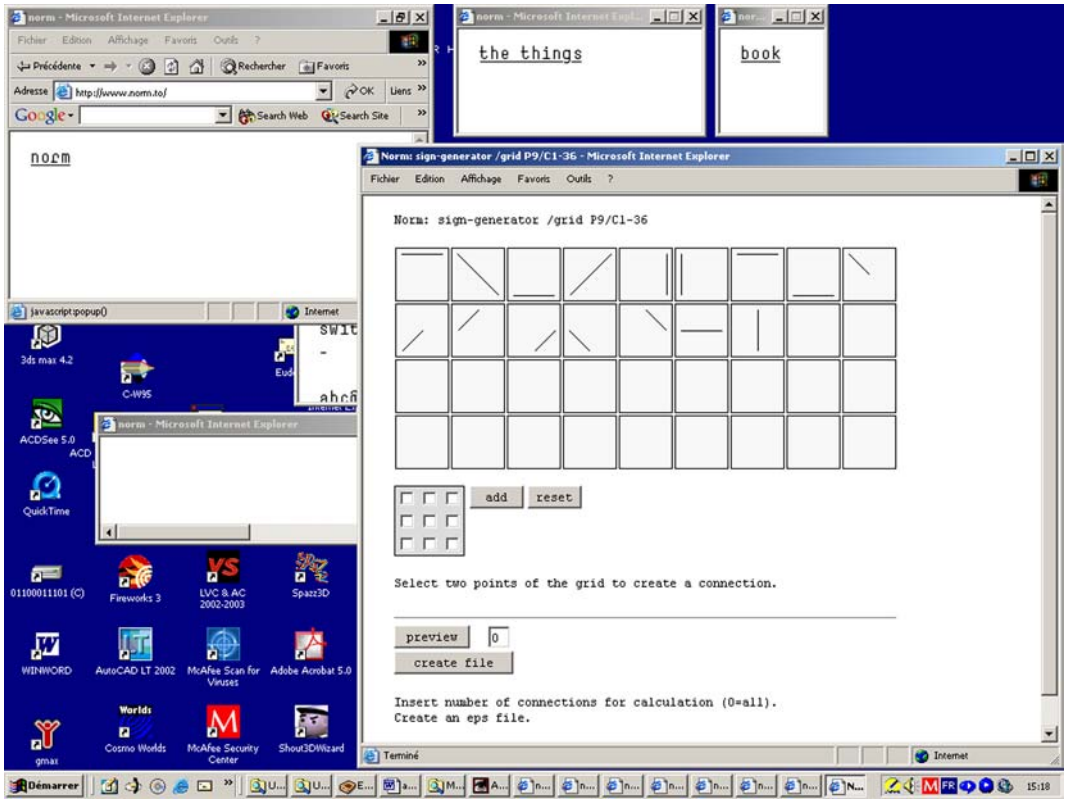
NORM graphic design
basic sign generator for NORM
by default interface
book *the things* by NORM
avatar as logotype by fabric | ch



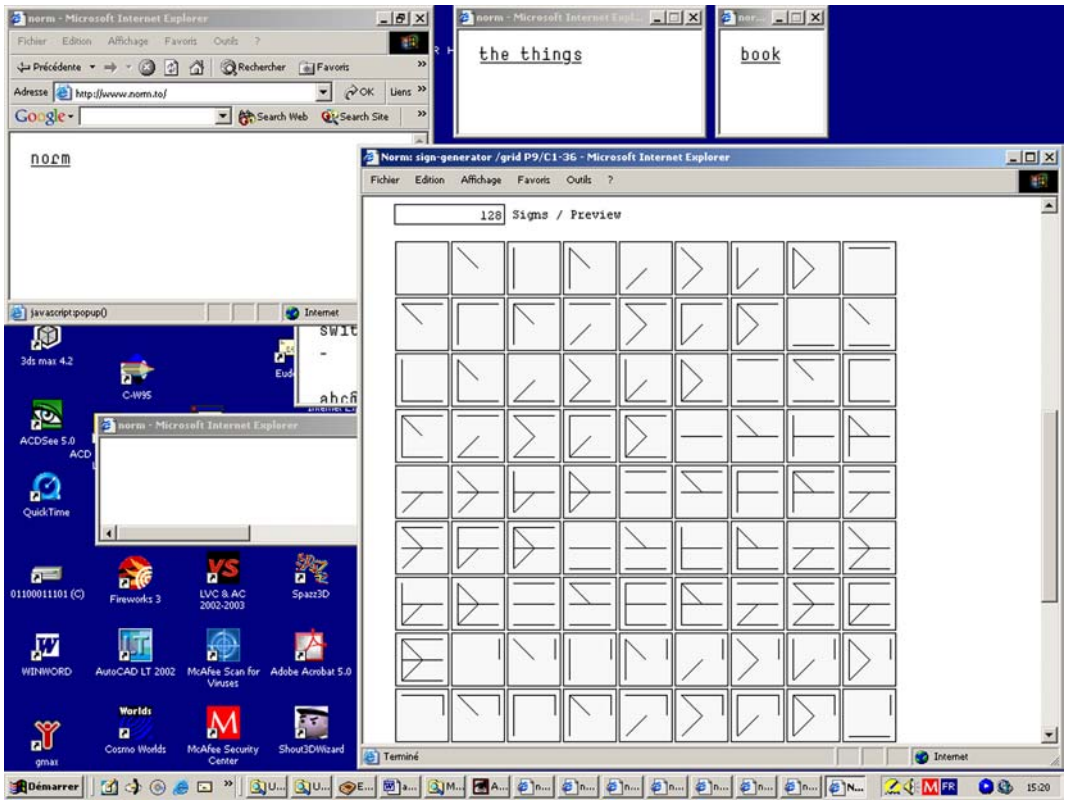
__NORM's website with online access to the sign-generator. the sign generator is in close relationship with NORM's book, *the things*.



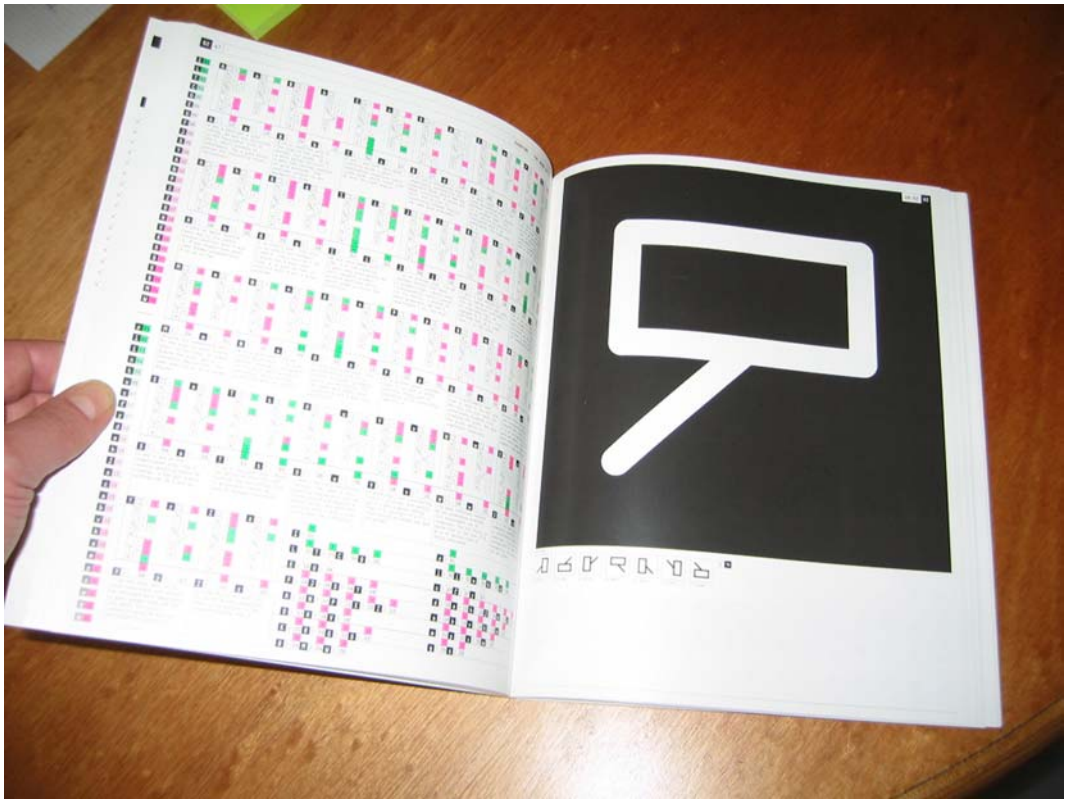
__main *by default* user's interface for the sign generator.



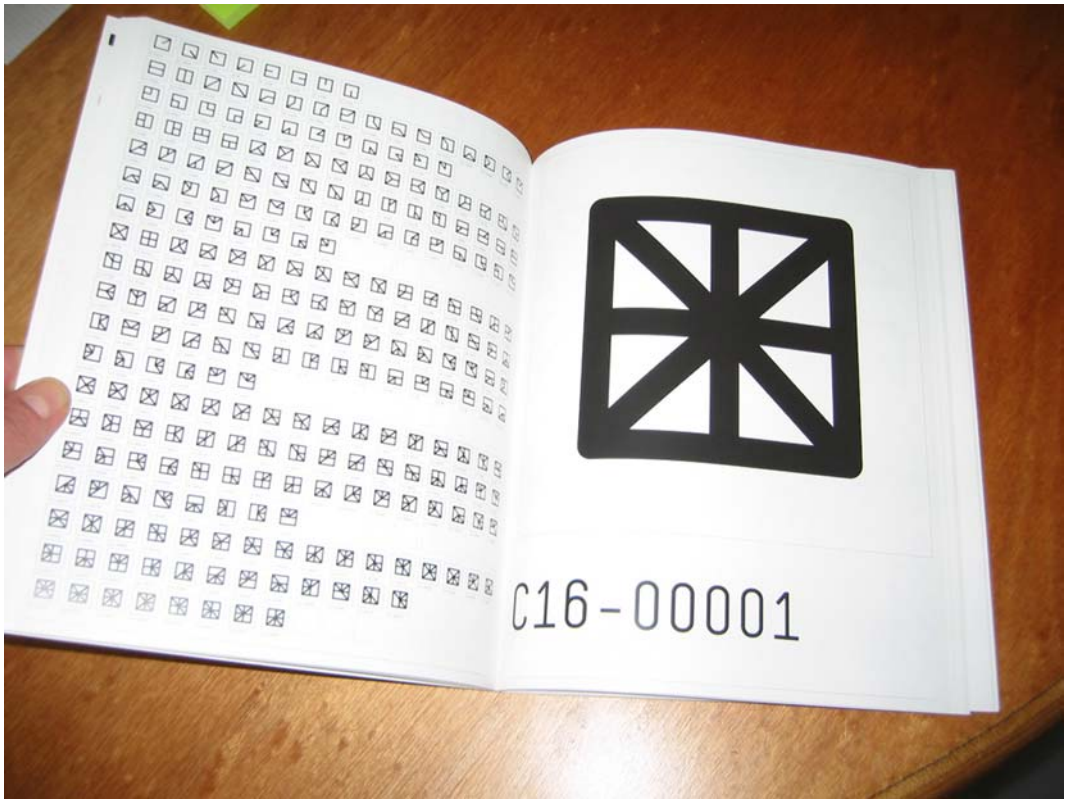
__grid based creation of the main lines that will be used to generate the logotypes.



__some of the logotypes created on the basis of the selected lines. Those logotypes can then be exported to an illustrator format and used for any other purpose [like avatars].

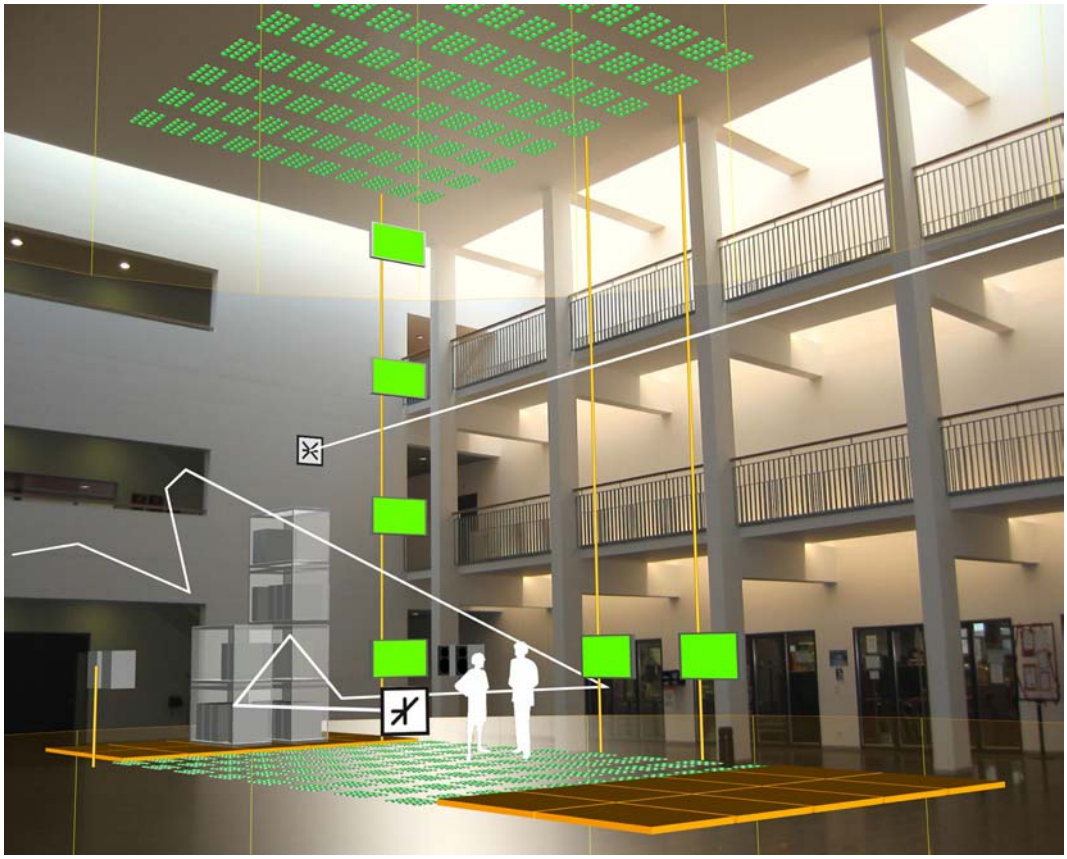


__one use of the logotypes in the graphic design and research book *the things* by NORM.



__another use of the logotypes in the same paper edition.

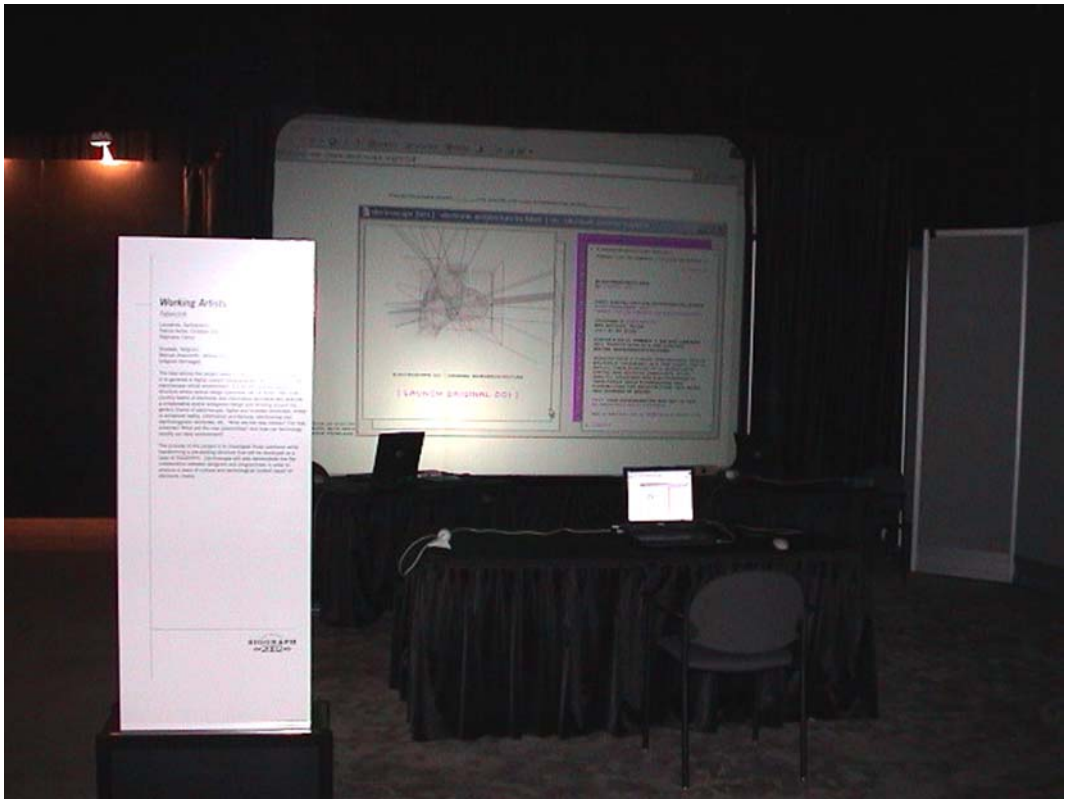
// map_IT ---------- //
// projet de campus digital pour l'EPFL // fabric | ch // www.fabric.ch // mai 2003



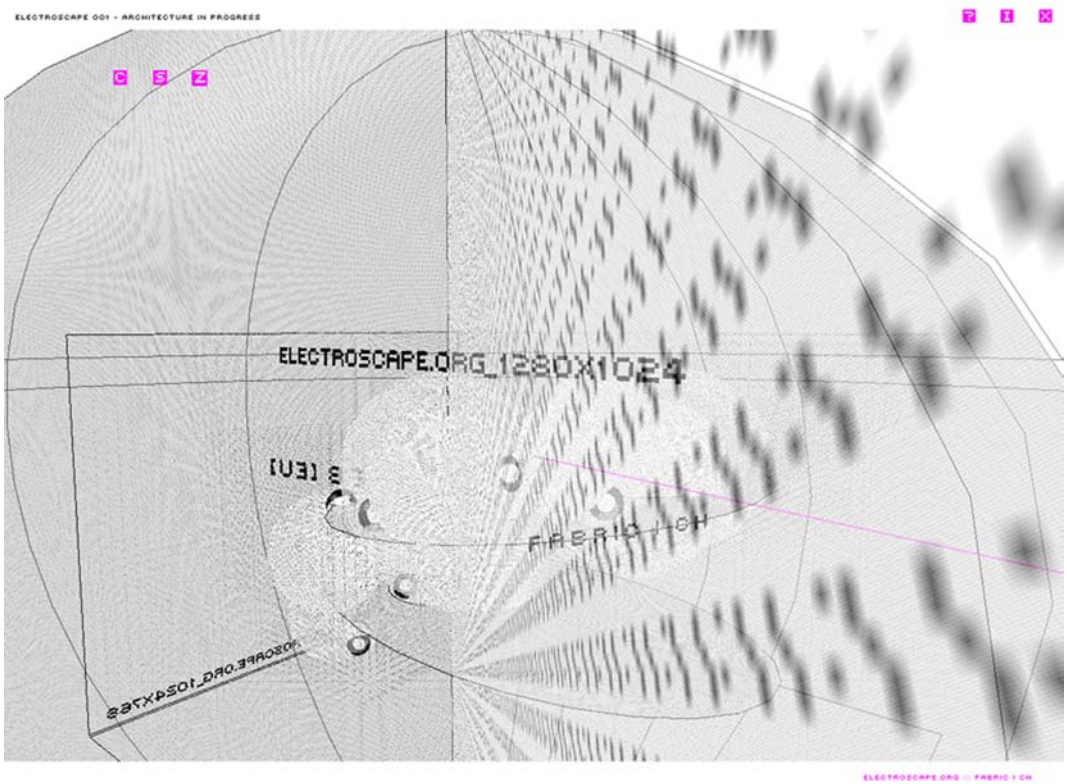
__avatars use of the logotypes by fabric | ch, in the context of a master plan for the digital campus of the Swiss Federal Institute of Technology -Lausanne.

2002 +++
electroscape 001

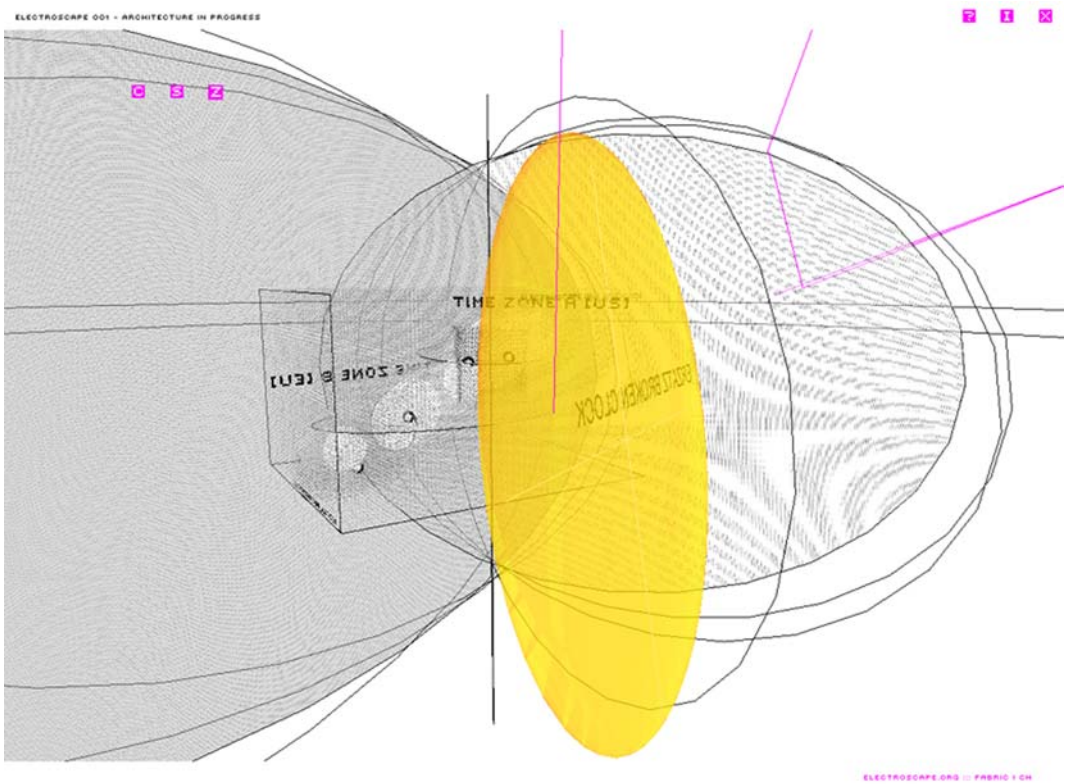
performance @ siggraph 2002
performing architecture
distant collaboration [usa/eu]
datagraph architecture
multi-users environment
netlag, multiplying time
screenscape & soundscape



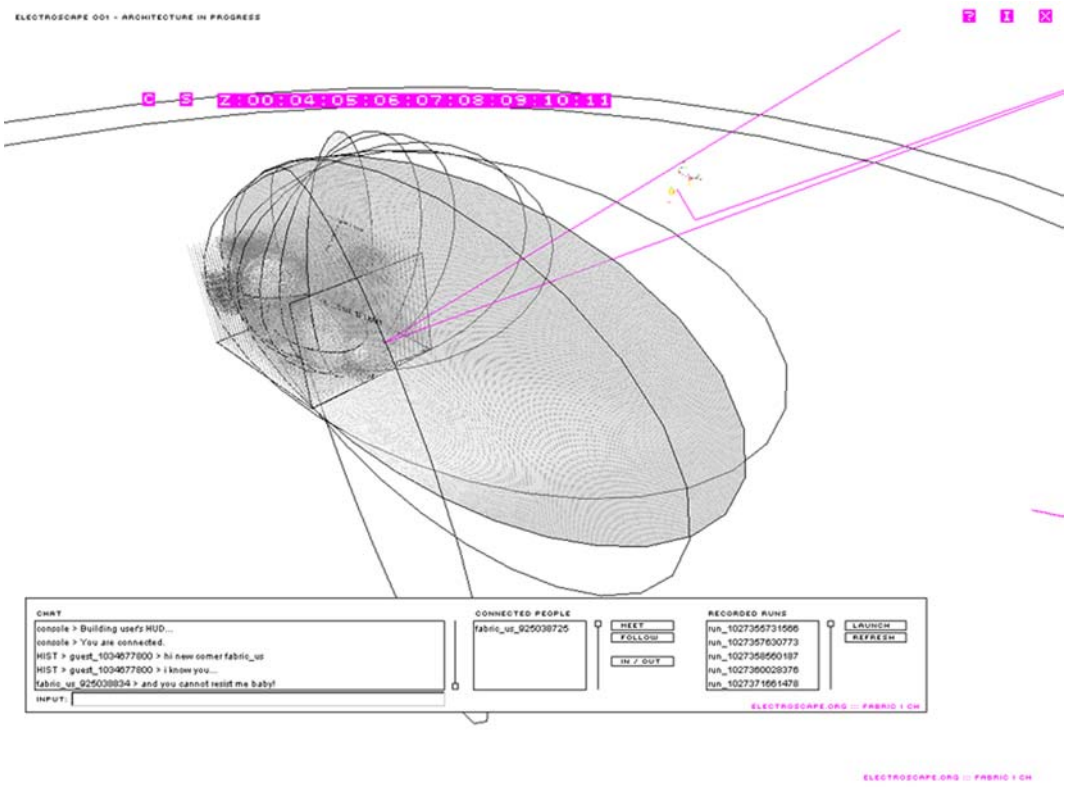
__installation at siggraph'art gallery 2002, screens in the black box…



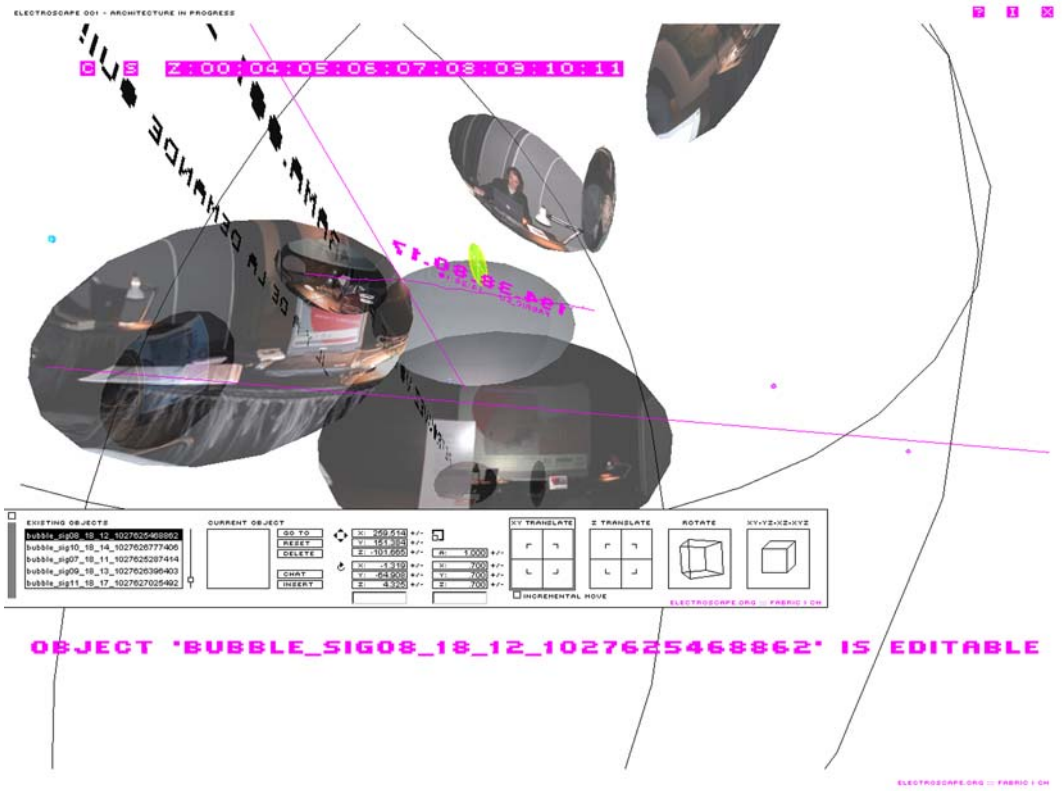
__view in the shared environment [between teams in EU and in USA] of electroscape 001



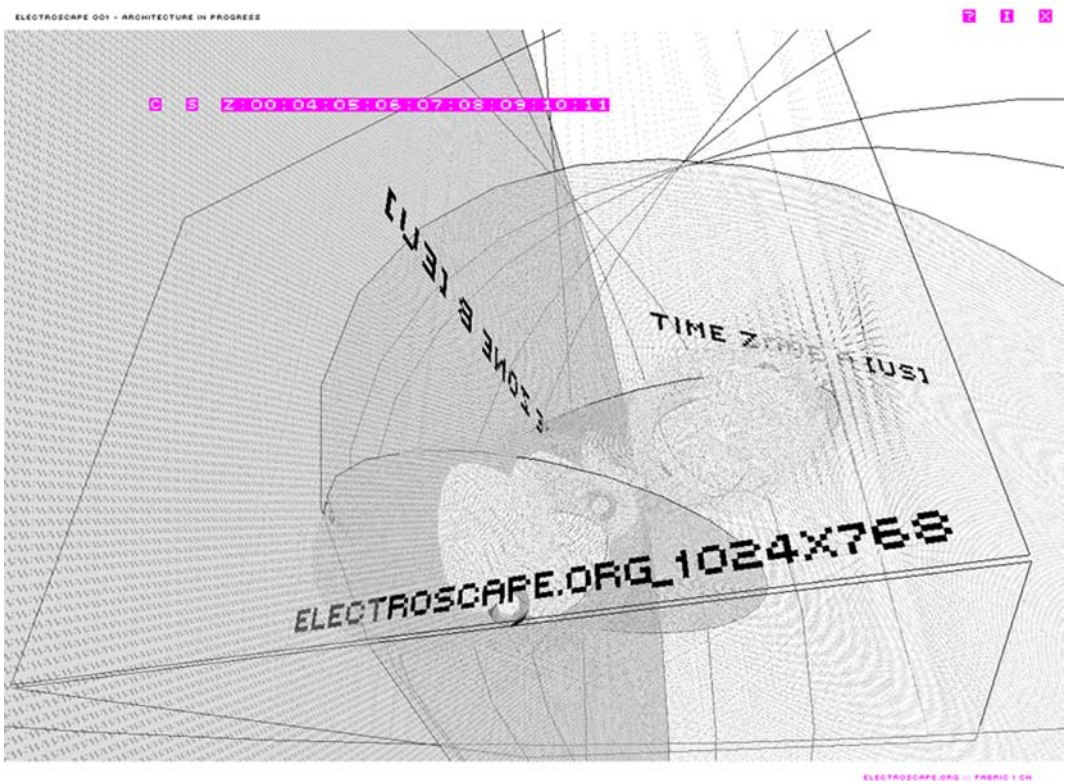
__beginning of the transformation of the space, a sound module [in yellow]



__public side interface and traces of users from europe [in magenta]

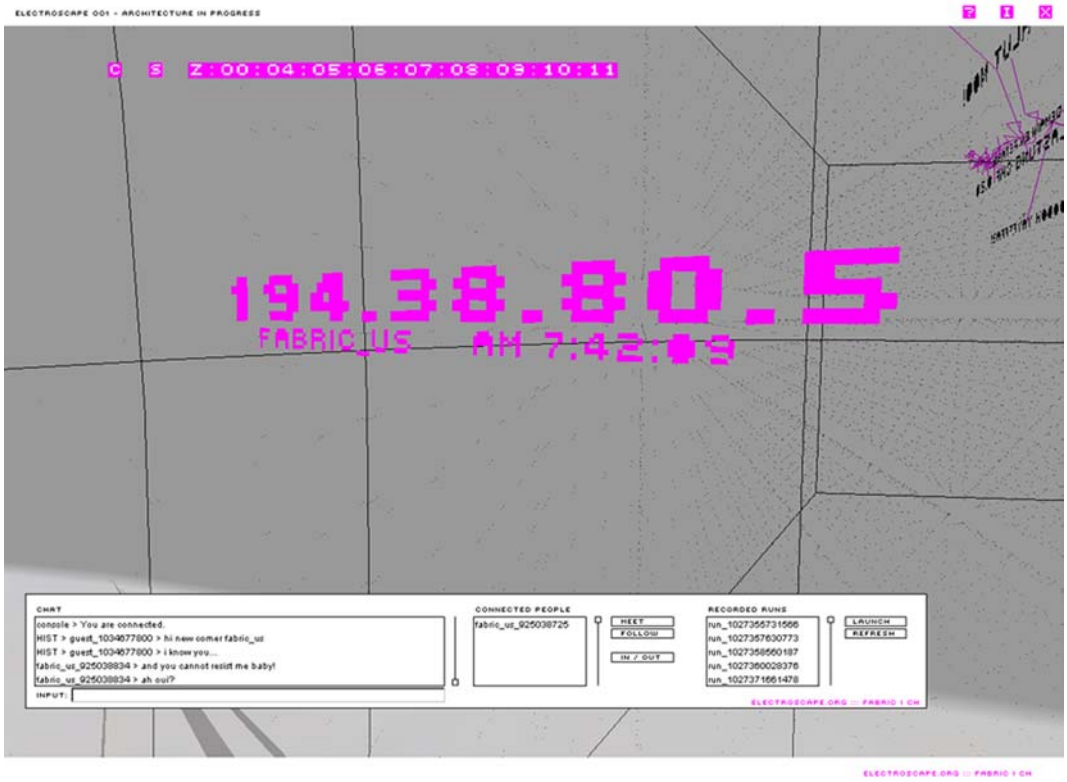


__private side interface with shared editor and editable objects from fabric's team in siggraph

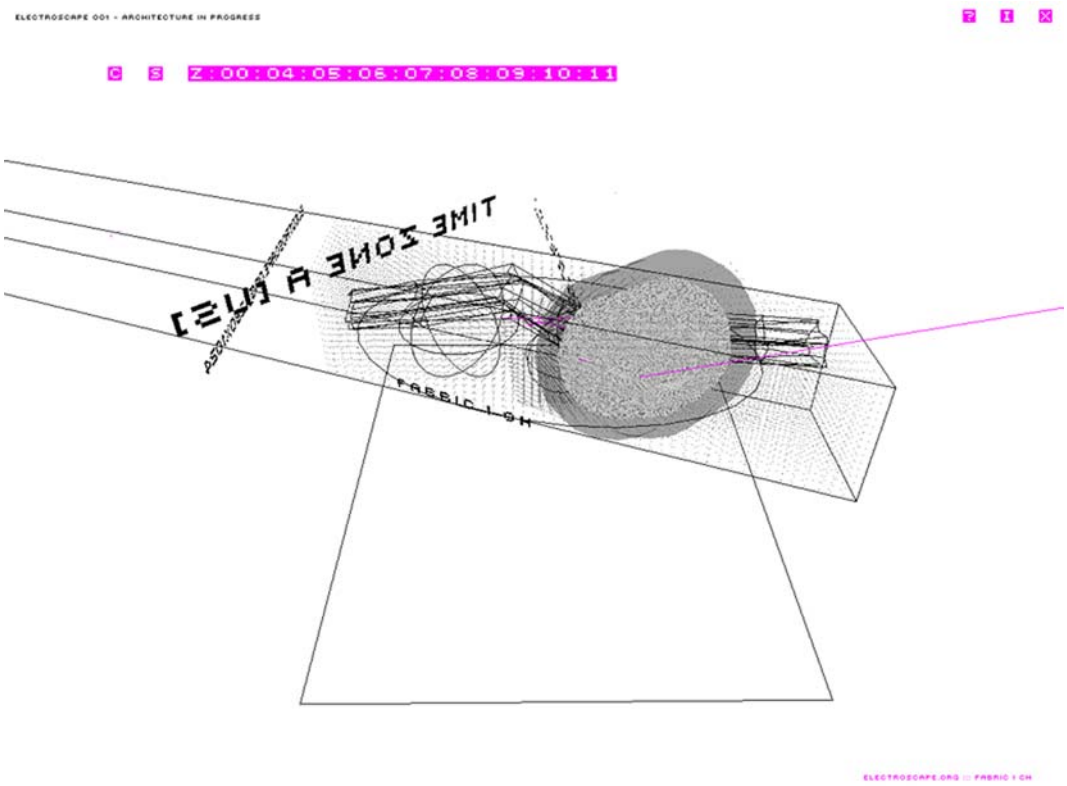


__electroscape 001 is a *screenscape*: 1024x768x1280 units. this is the space to transform

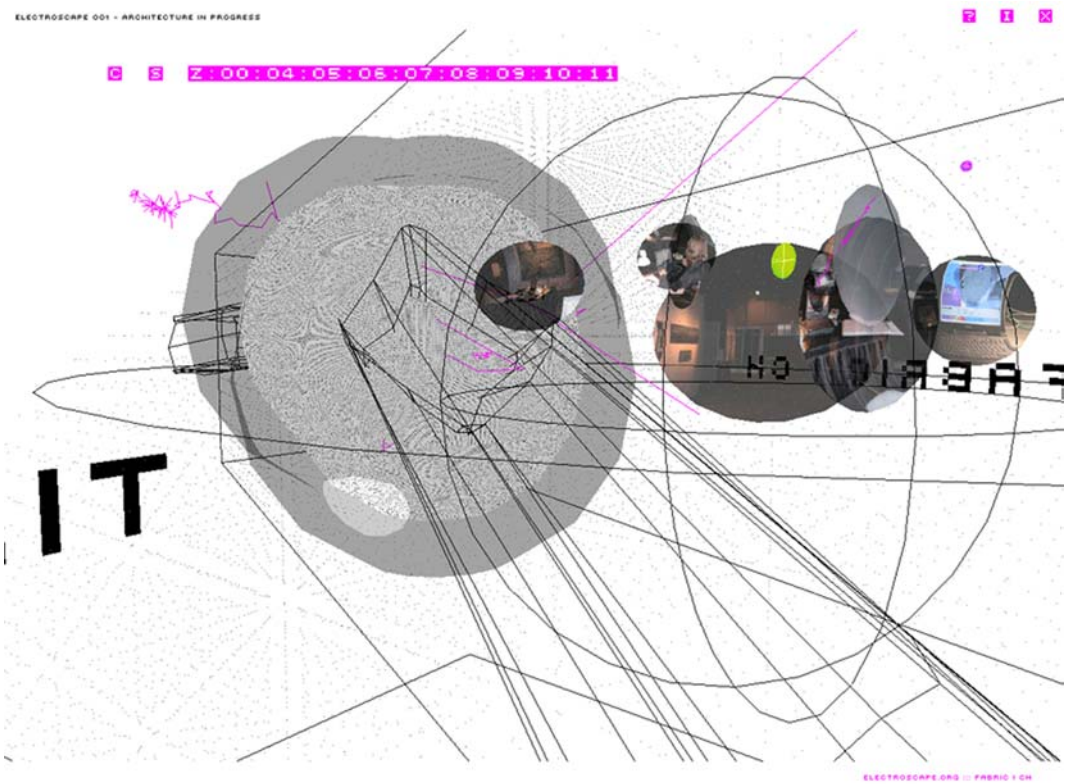
__inside 1: testing of screen effects and specificities of a *screenscape*



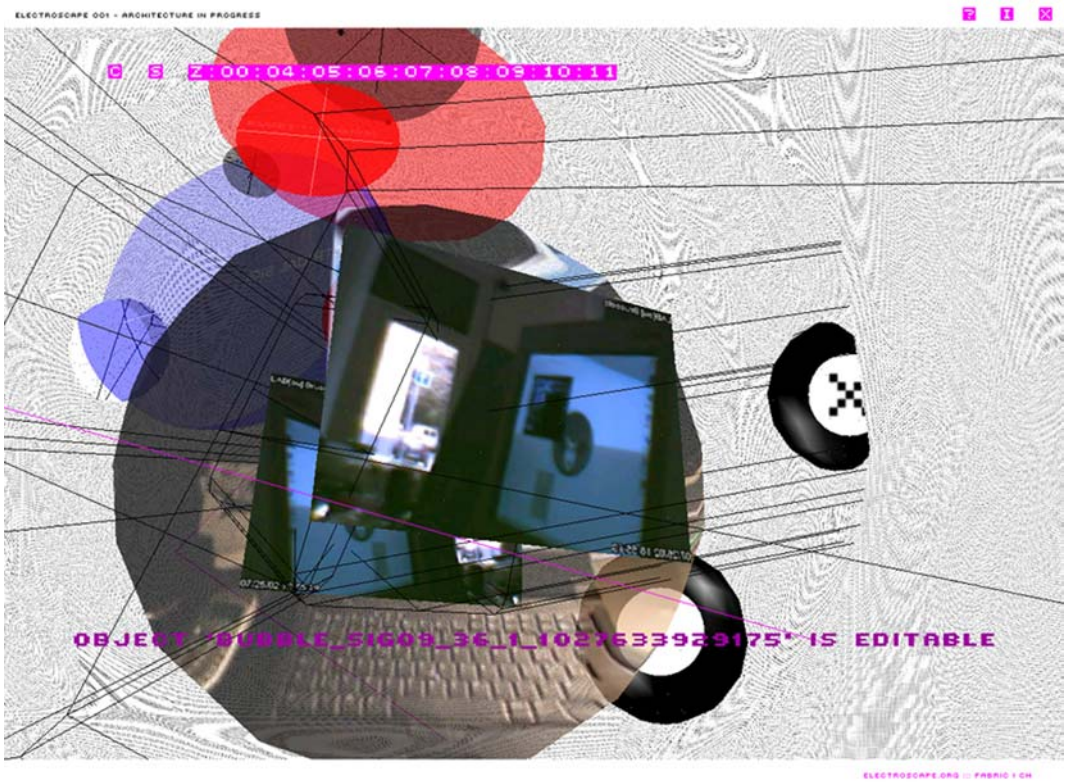
__an avatar [user's representation] of a fabric's user in USA. early in the morning



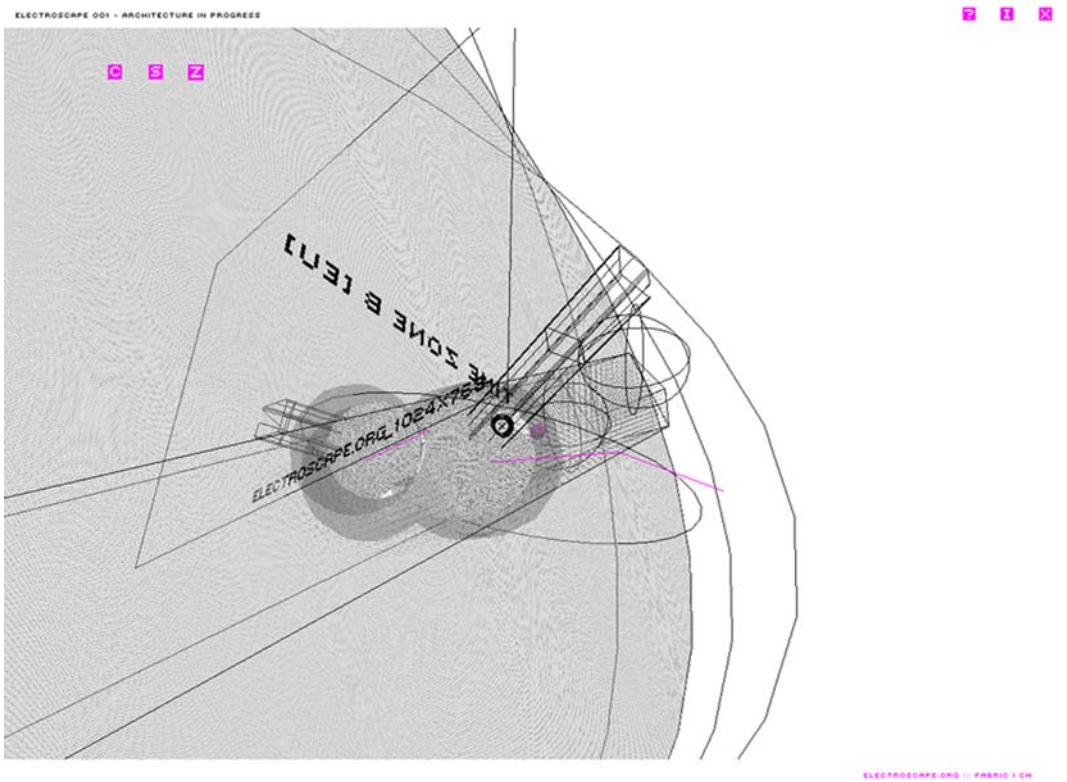
__outside view of a part of the screenscape, the one dedicated to people in EU



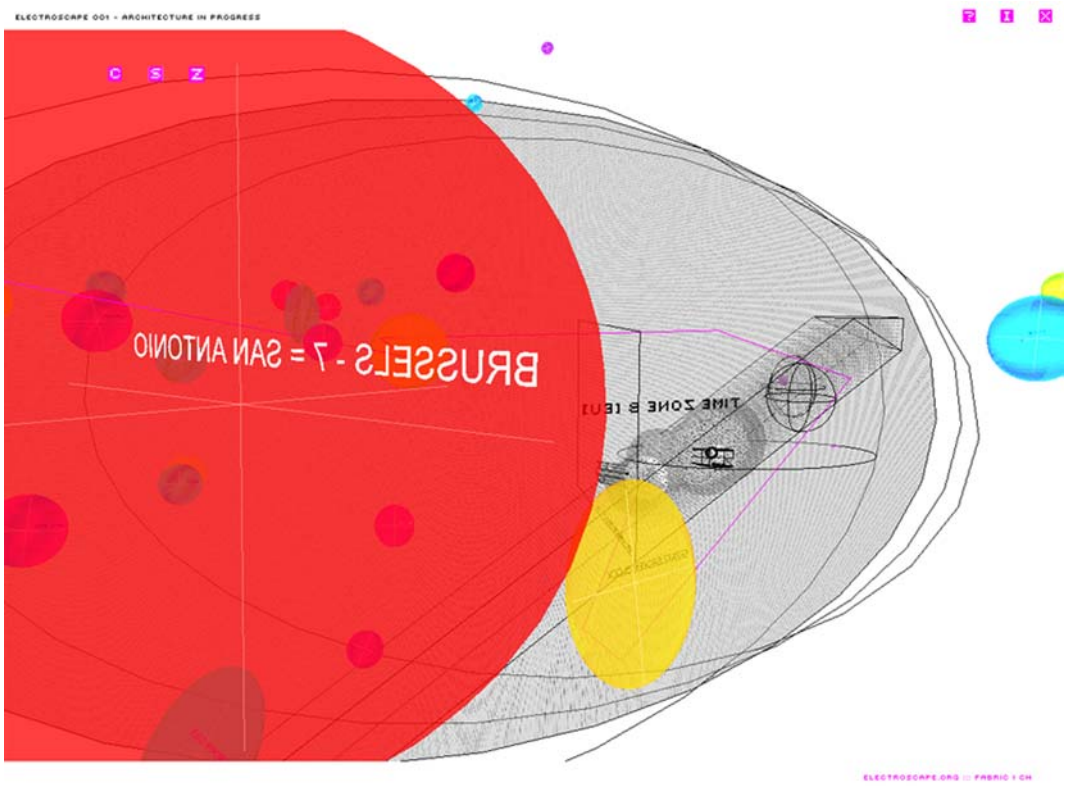
__siggraph's bubbles in the air. recombining our space at siggraph with the screen space



__siggraph's and sound's bubbles in the eletroscape 001 environment. mixing and recombinating…

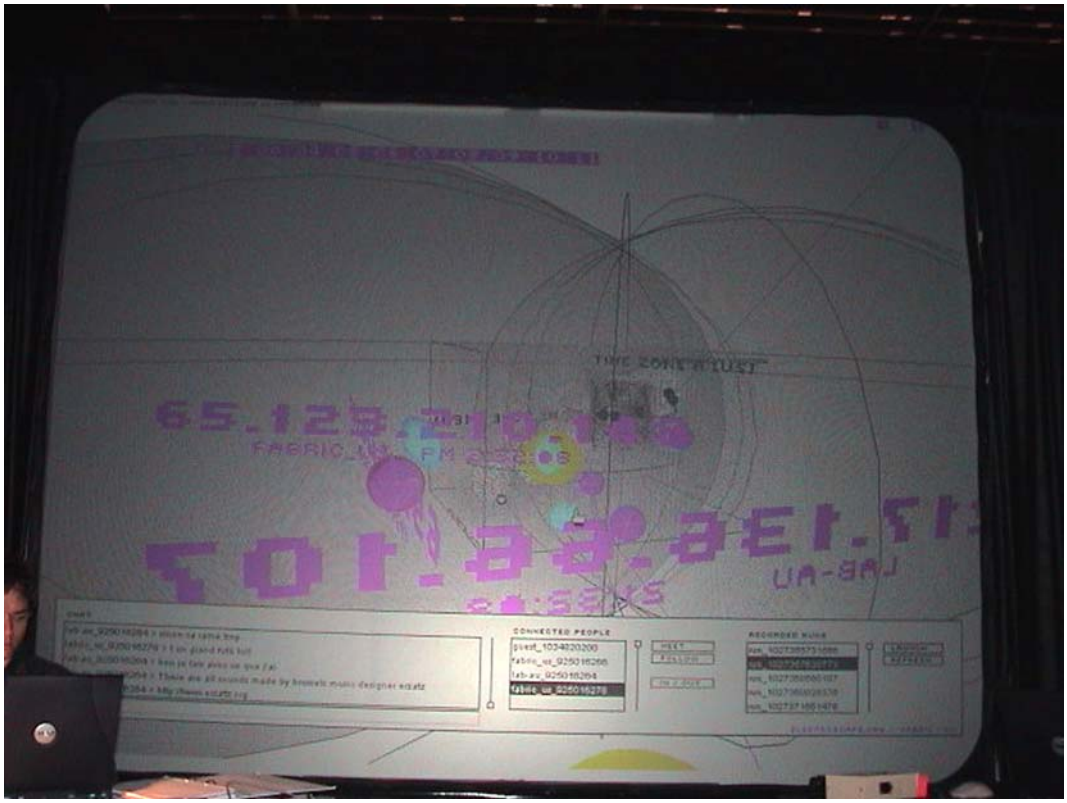


__another outside view with user's traces



__a sound bubble from the team in brussels

__a person from fabric | ch performing in the siggraph art gallery

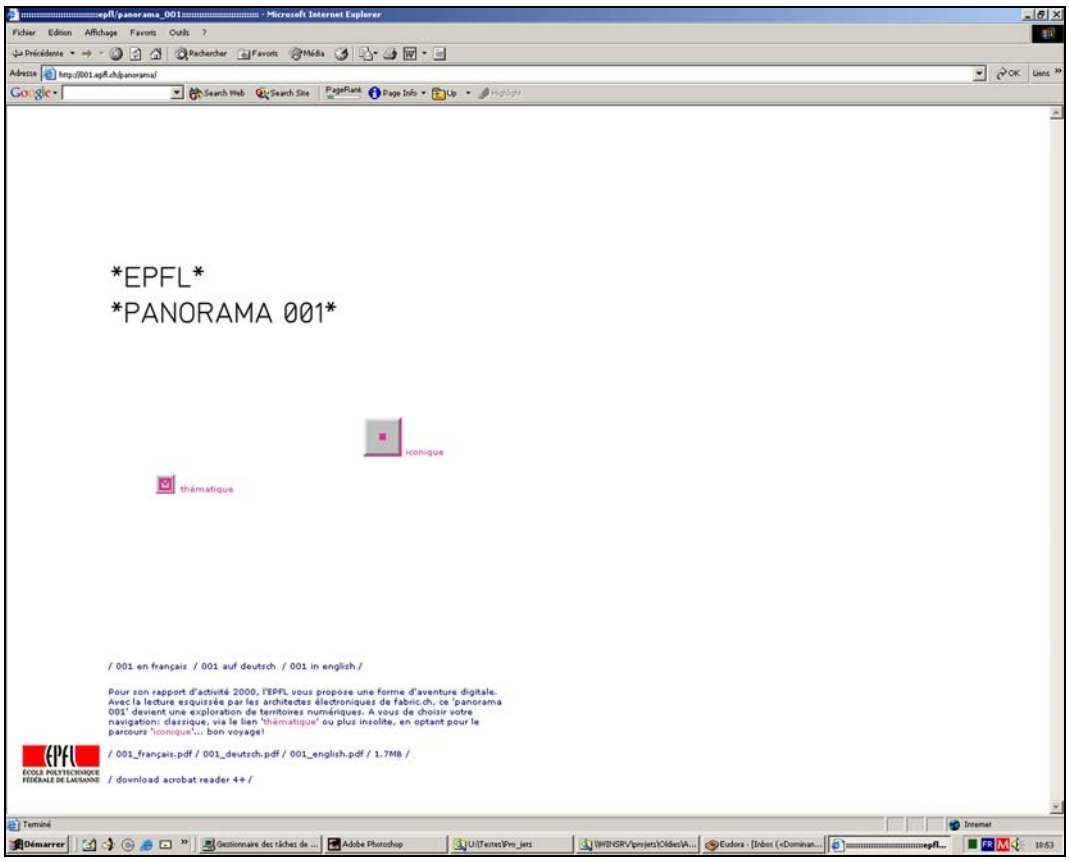


__a screen view of two user's avatars facing and discussing in the electronic world

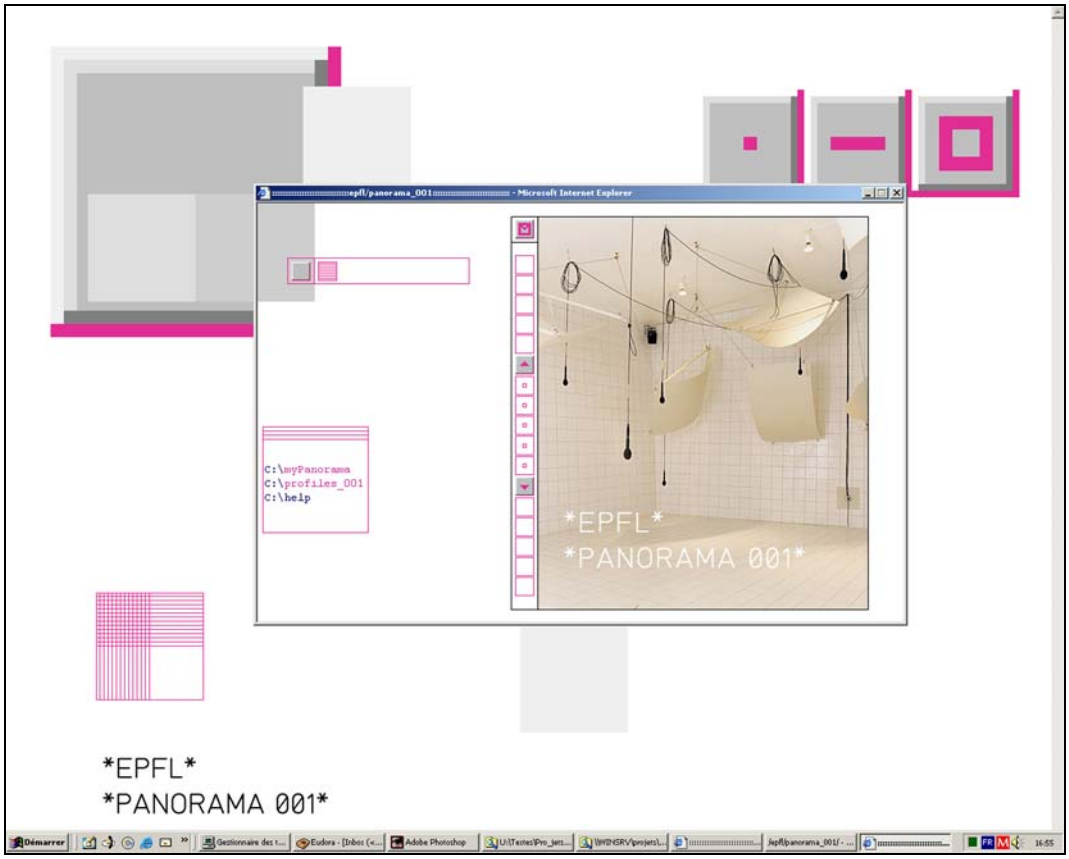
2001 +++

panorama 001

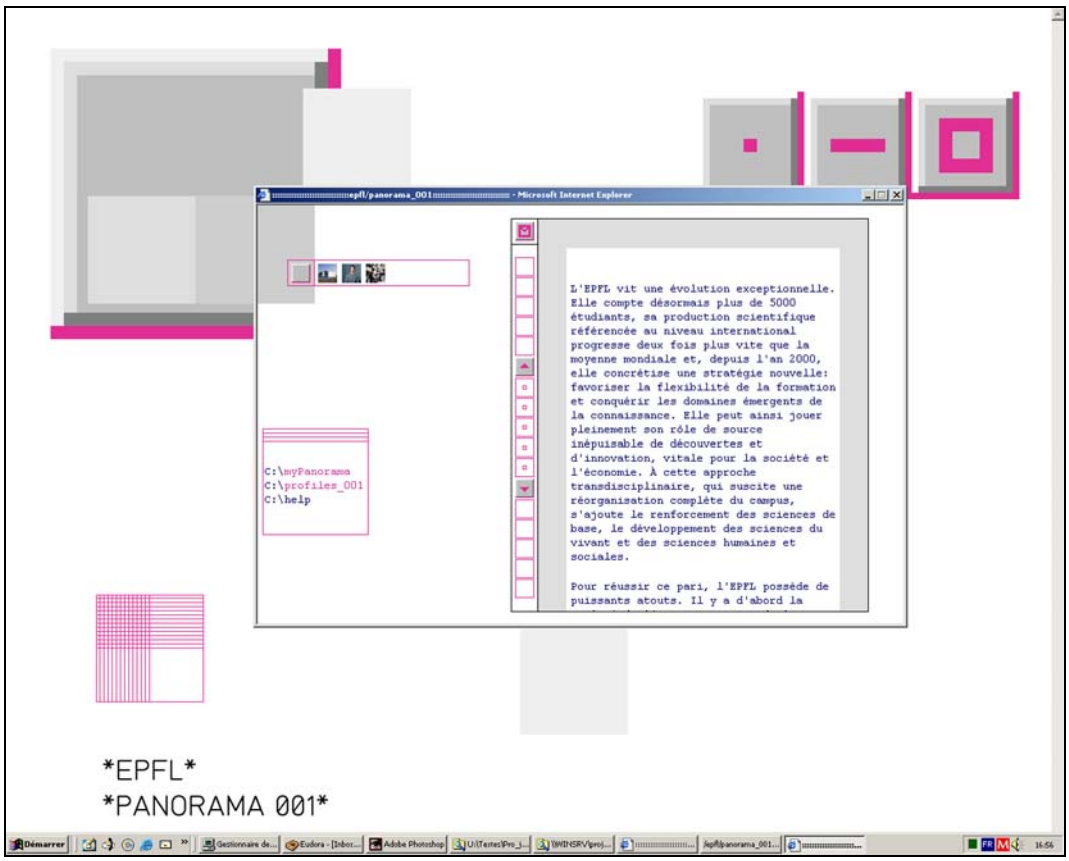
tracked epfl website
open profiling
data traces
shared profiles
mixed & extended networks



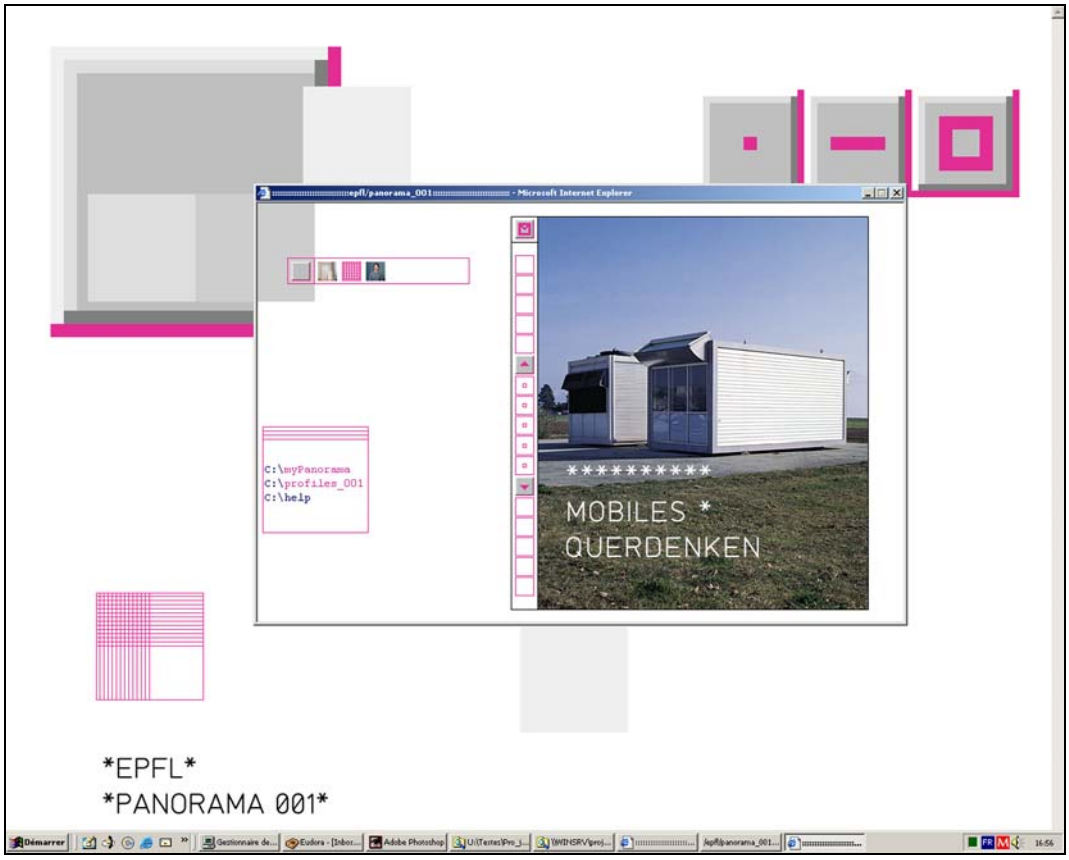
__panorama 001 presents itself in the form of a classical website, with a choice of a *traditional* navigation or a more *intuitive* one



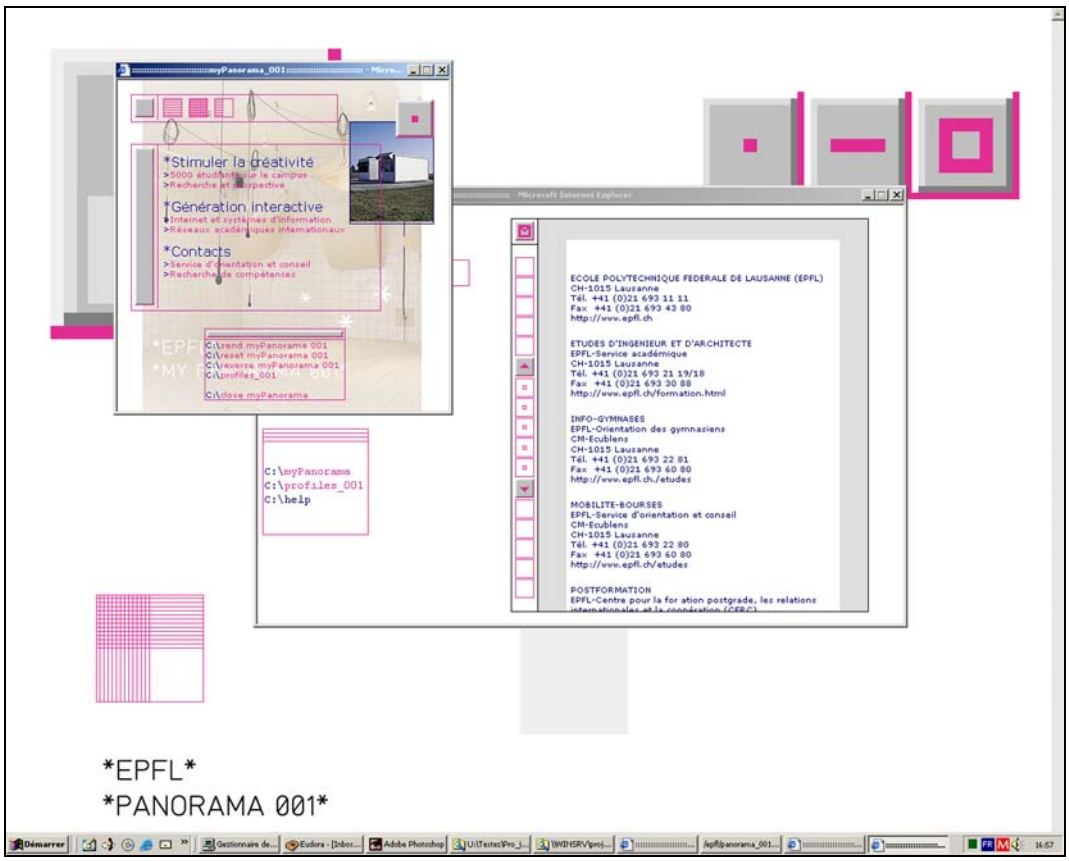
__once inside the website -here the *intuitive* navigation -, no words or precise menu, mostly images and icons



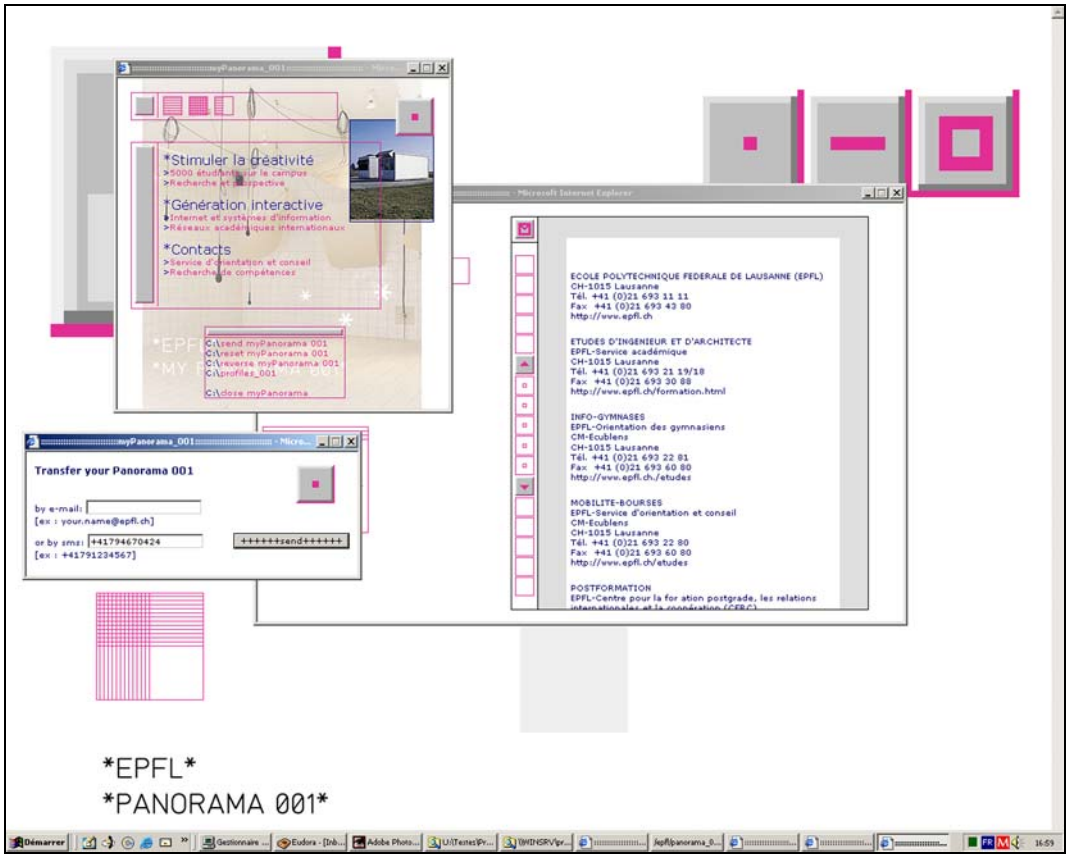
__even if it seems to be a random navigation, it is in fact less than it appears because there are several possible *walkthroughs*, that are like scenarios or fictions. these walkthroughs can then cross themselves



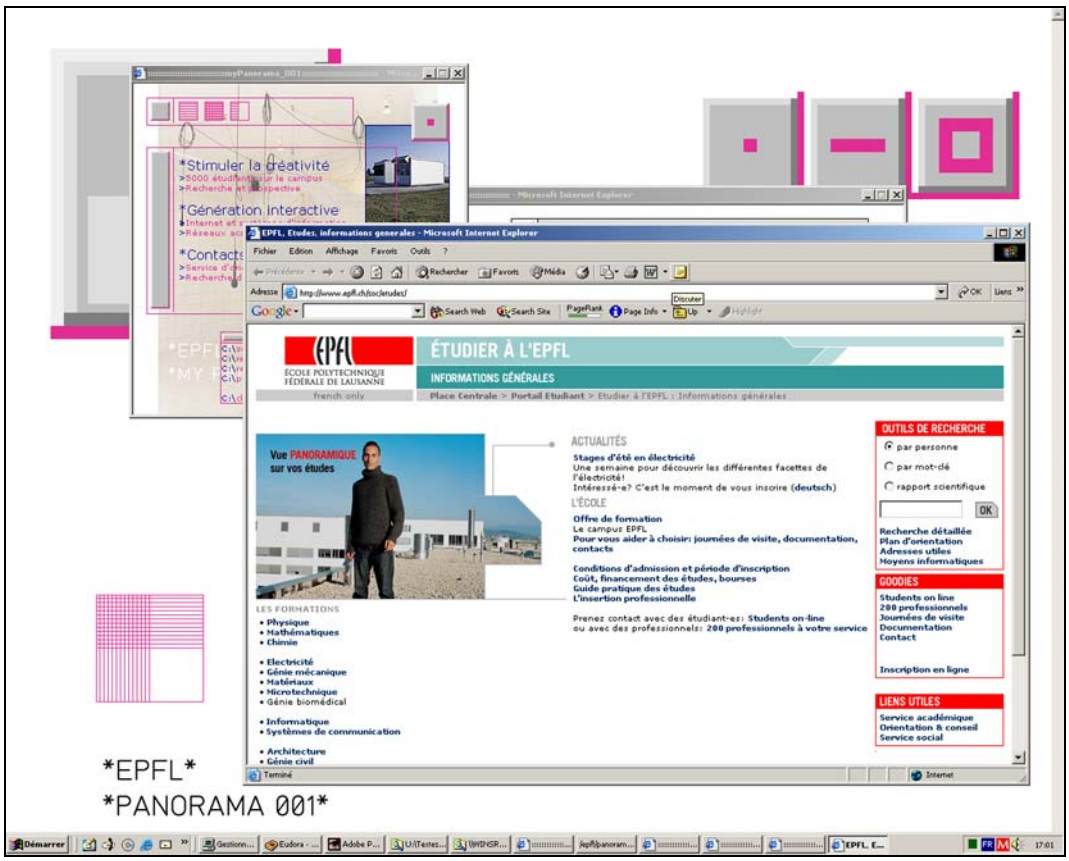
__the navigation in the website continues, but more than what each user is exactly doing, it is the time spent on the different sections that is being tracked and recorded



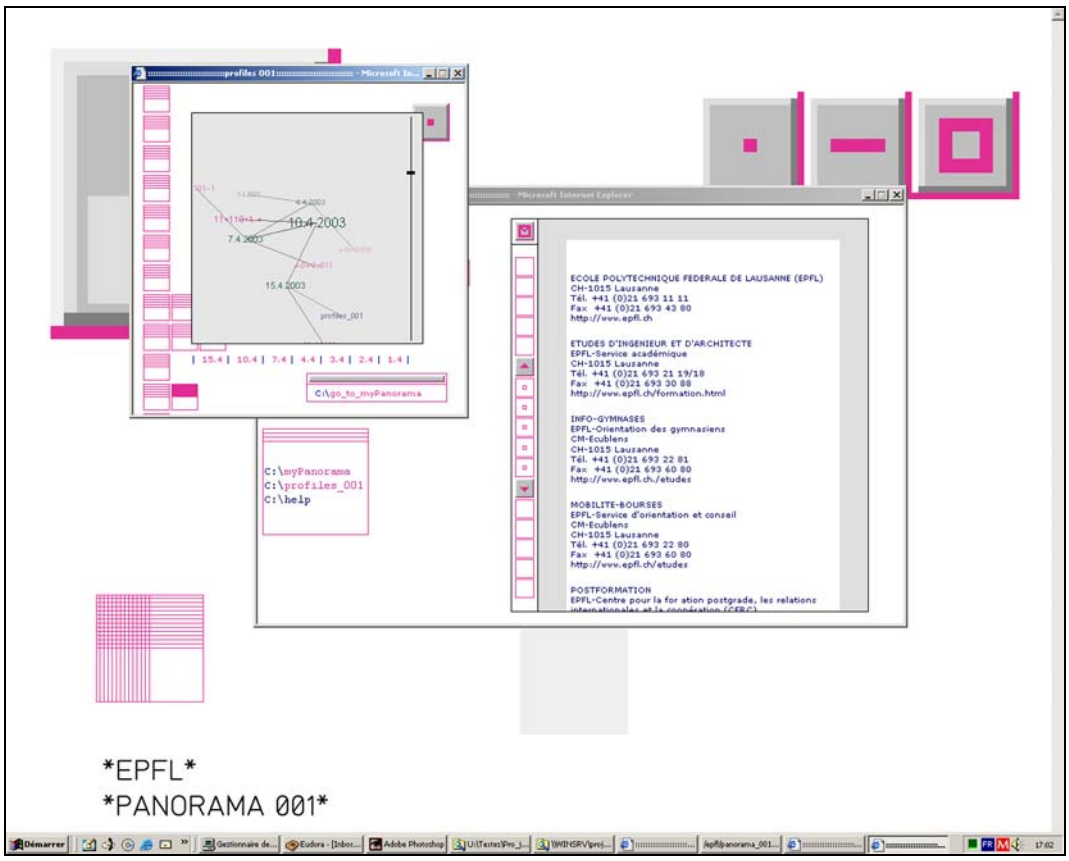
__what the user has done, its interests allow the system to build a profile: a new, mostly visual, small window. it is the actual profile of the user and this window becomes a new navigator, a summary of what has been done, a 2d profile



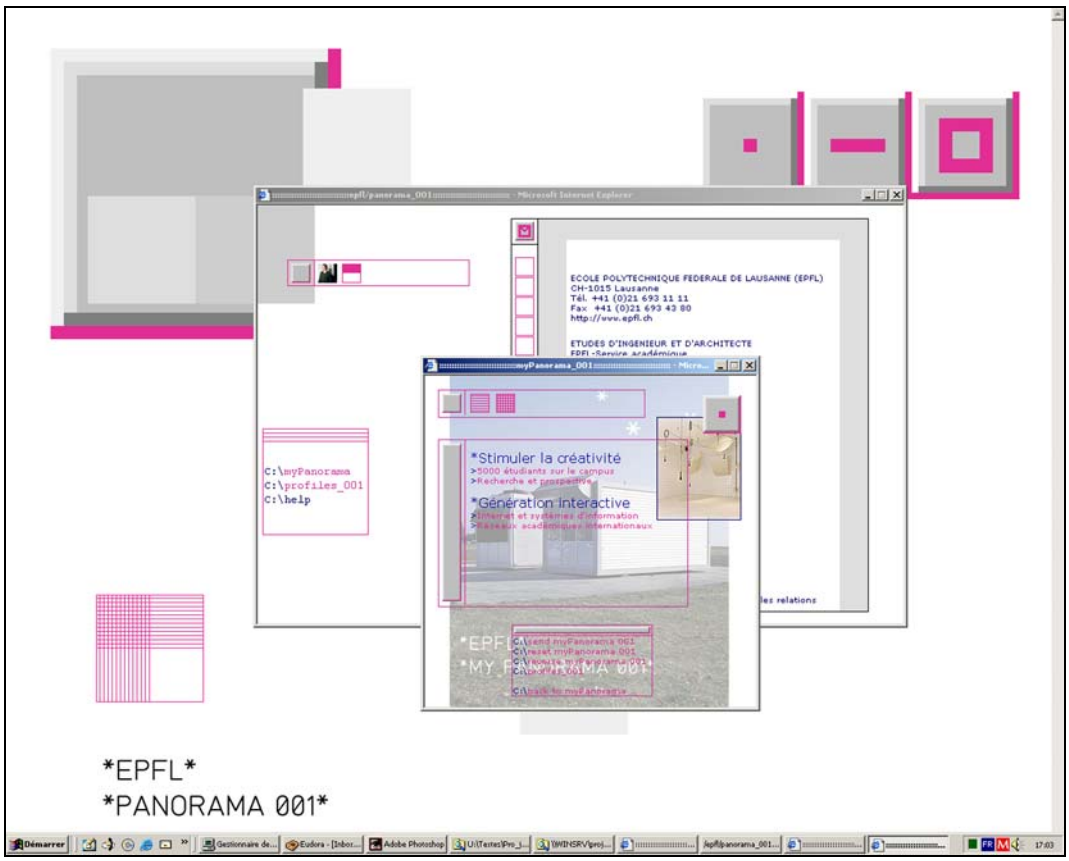
__the 2d profile and its url [stored in a database] can then be shared and sent to new users, to friends or to people that might interested by the same subjects. it can be sent by sms or by email



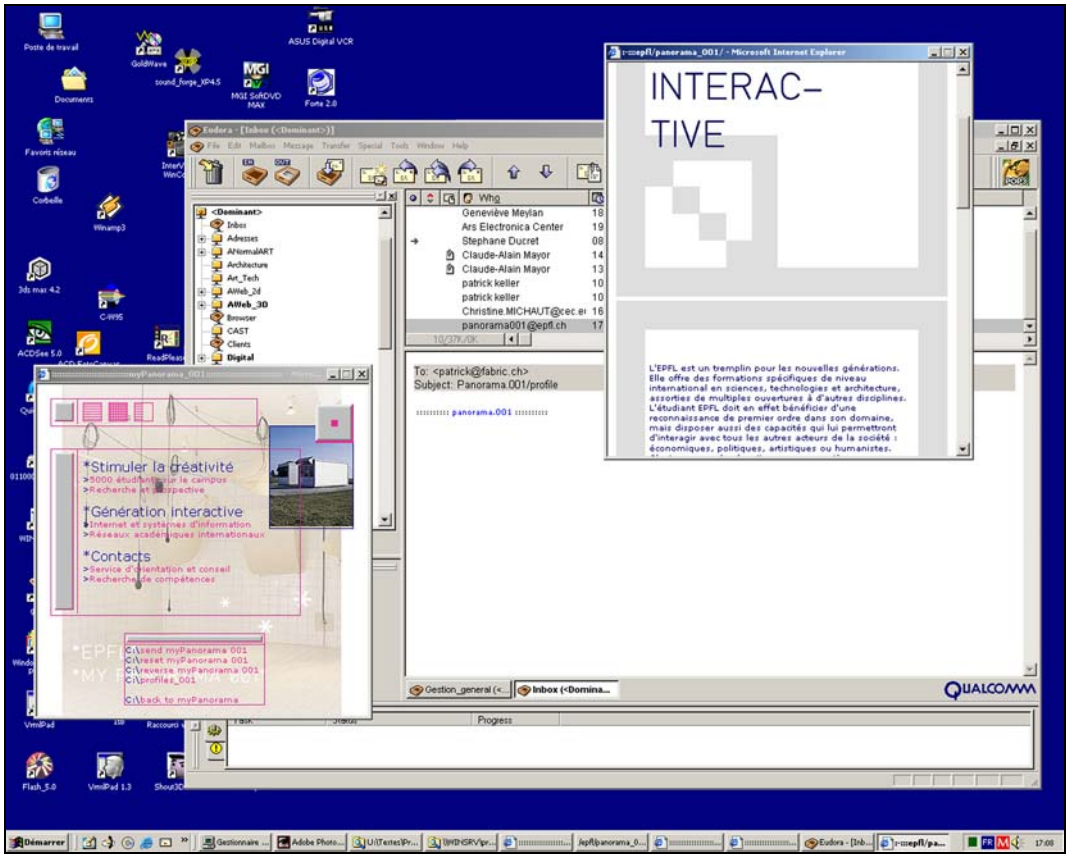
__the 2d profile suggests new links to a bigger site [in this case the site of the polytechnical school of lausanne [epfl] that contains 300'000 web pages]. the suggested links are in close relationships with the areas of interests built in the 2d profile



__in the same small *profile* window, all the profiles created by all users can be seen in the form of a graph, day after day



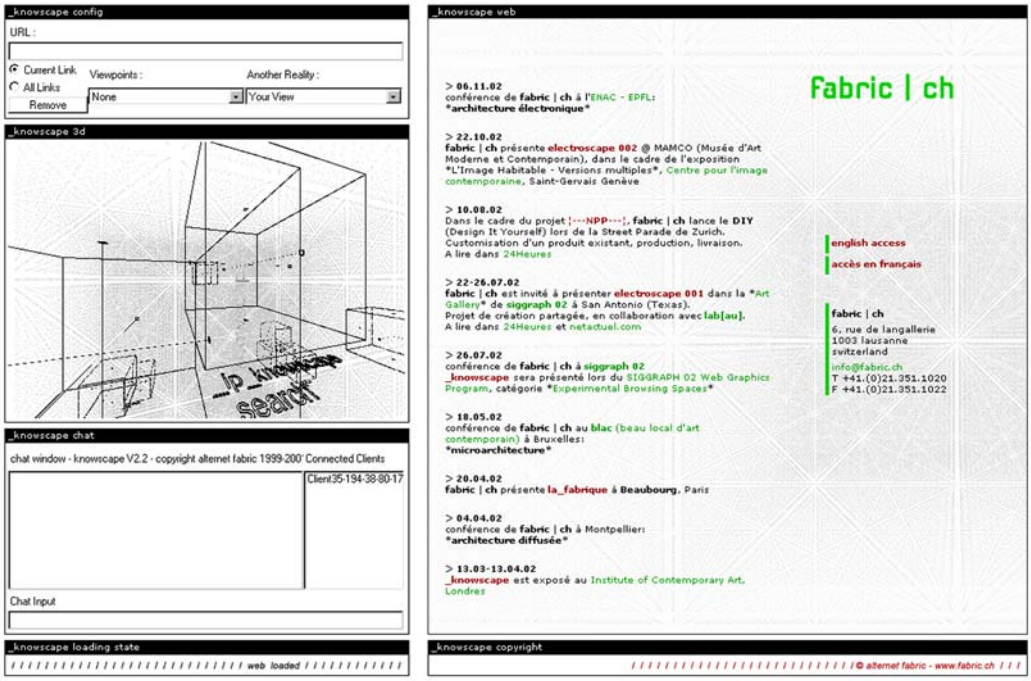
__and somebody else's profile can be open, and then browsed…



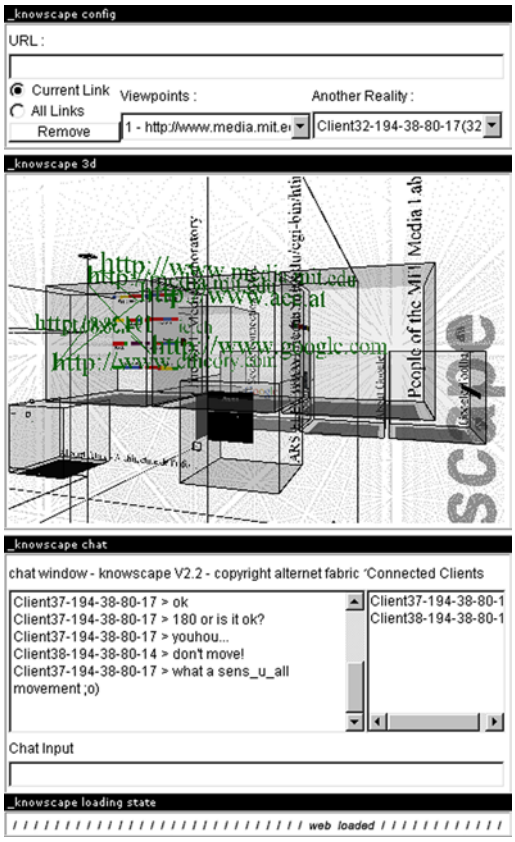
__this profile [like any profile] can again be sent and shared. when received by email for example, the message in the mailer opens the profile's window which can then open new windows either from the panorama 001 online document or from the epfl website. etc…

2001 +++
knowscape

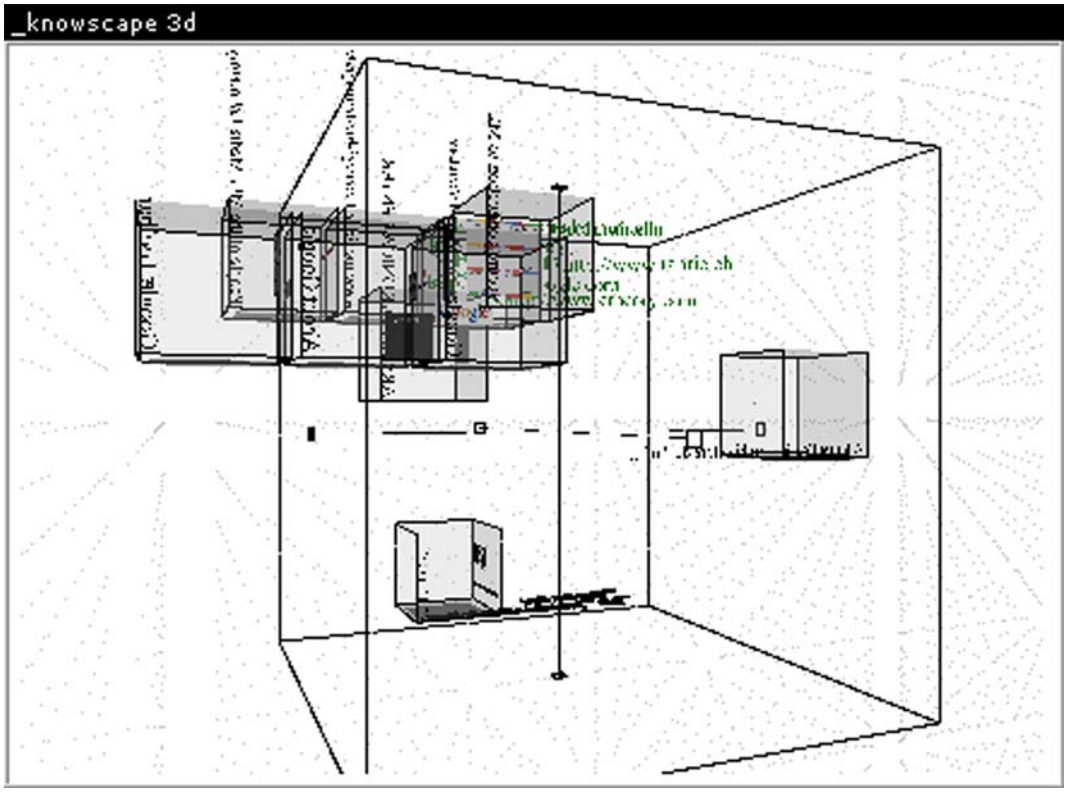
experimental browser
multi-users environment
online personal data identity
online personal data world
users open profiling
shared profiling



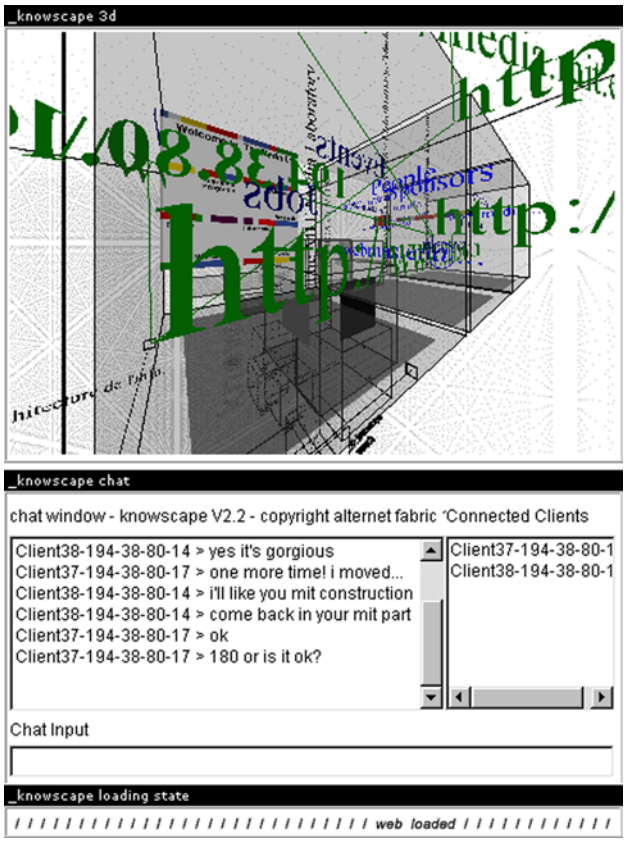
__enter url, 3d world url construction, chat and 2d conventional webpage: the overall interface



__a close look on the left part of the sceen: enter url, 3d datascape [shared among users], chat between users



__knowscape 3d: the shared world made out of of each user's browsing experience, each profile



__detail on a part of a user's world [he went to the mit medialab's page] and discussion between 2 online users

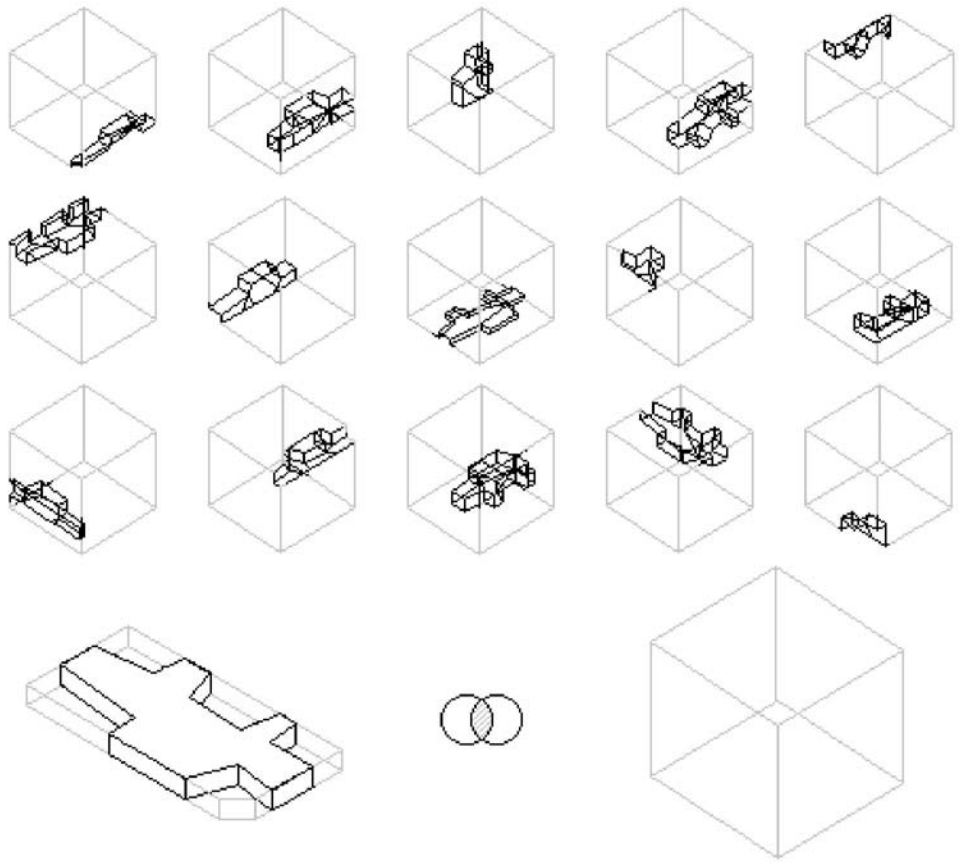
__an avatar [a user's representation] in the 3d datascape

__an avatar alone in the scape: in the center, his IP number, surrounded by the urls he browsed [clickable by any other user]

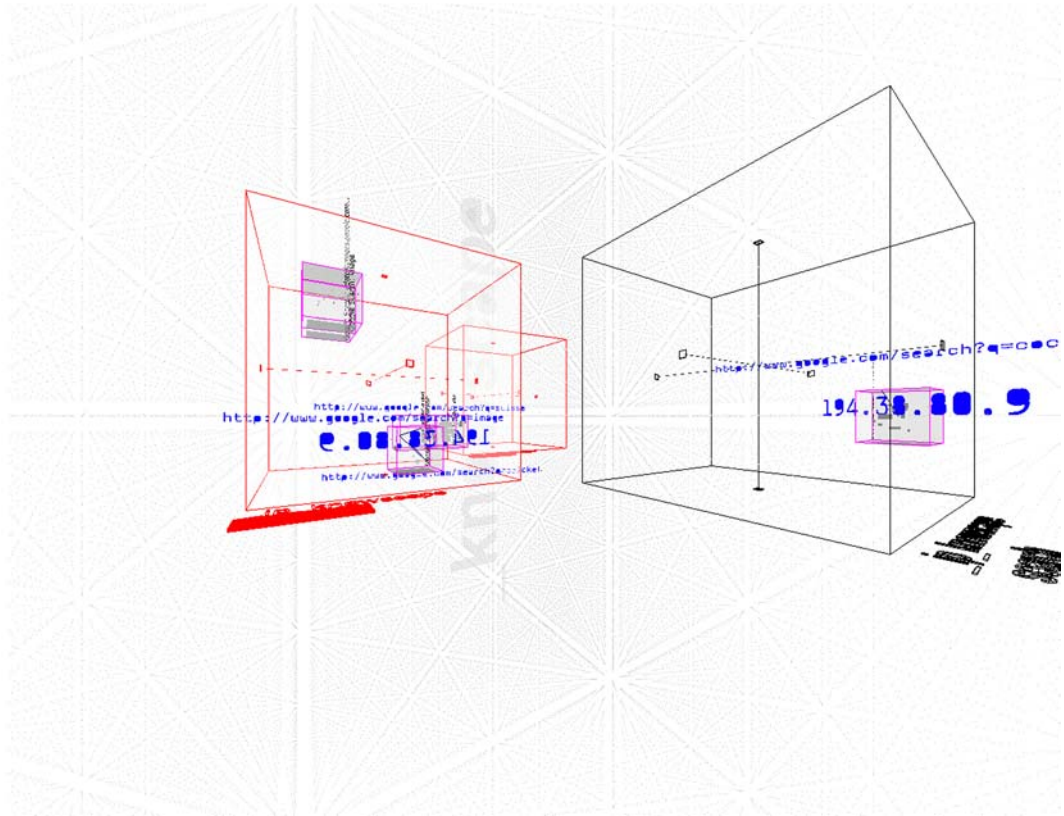
2003 +++

_knowscape mobile

_knowscape [2001] goes mobile
mobile & downloadable space
information space
relational - digital architecture
micro architecture
alternate 3d web browser
installation

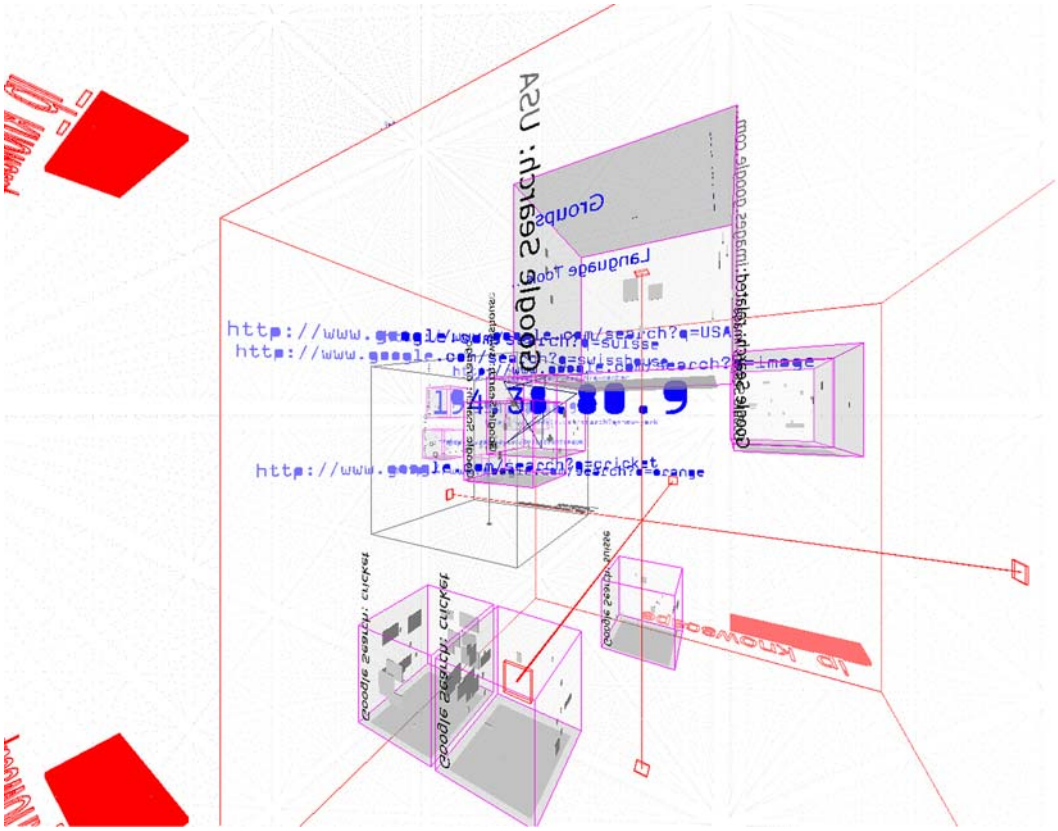


___knowscape mobile is a boolean and hybrid space. principles of associations between physical space and digital one are based on boolean operations. _knowscape mobile will follow fabric | ch and will be geo-re-localised, from place to place, from time to time, from installation to installation.

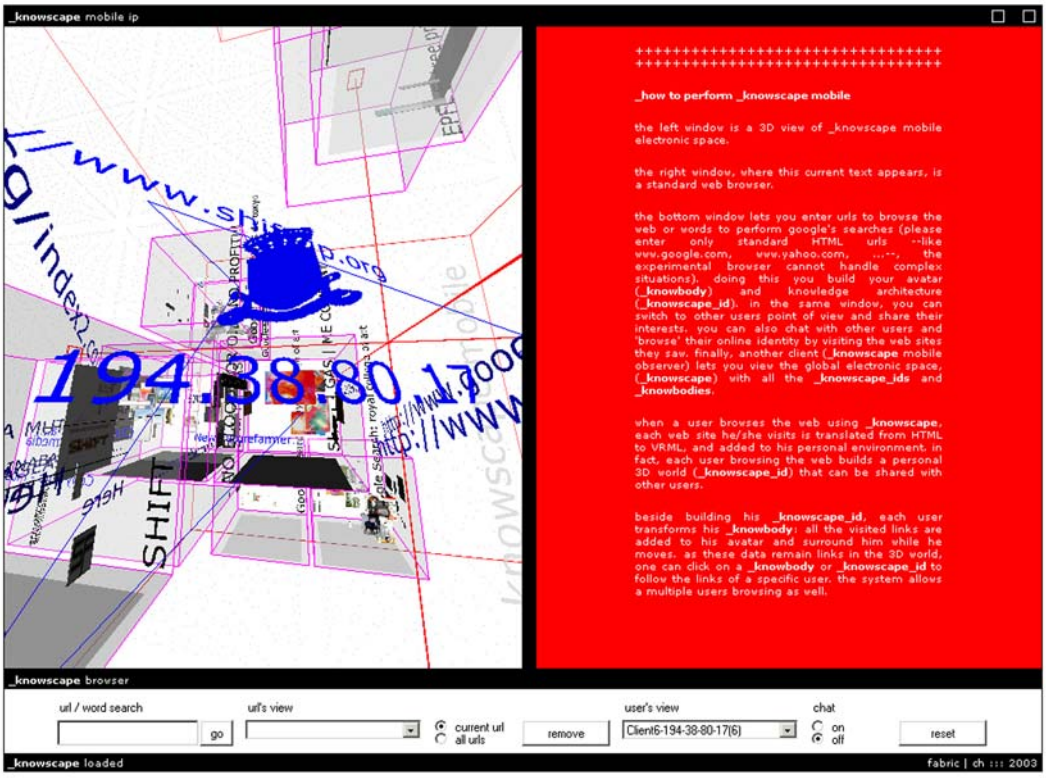


__a view into _knowscape mobile multi-user digital environment. a space with temporary scale, temporary and un-clear location and with both local and distant users.

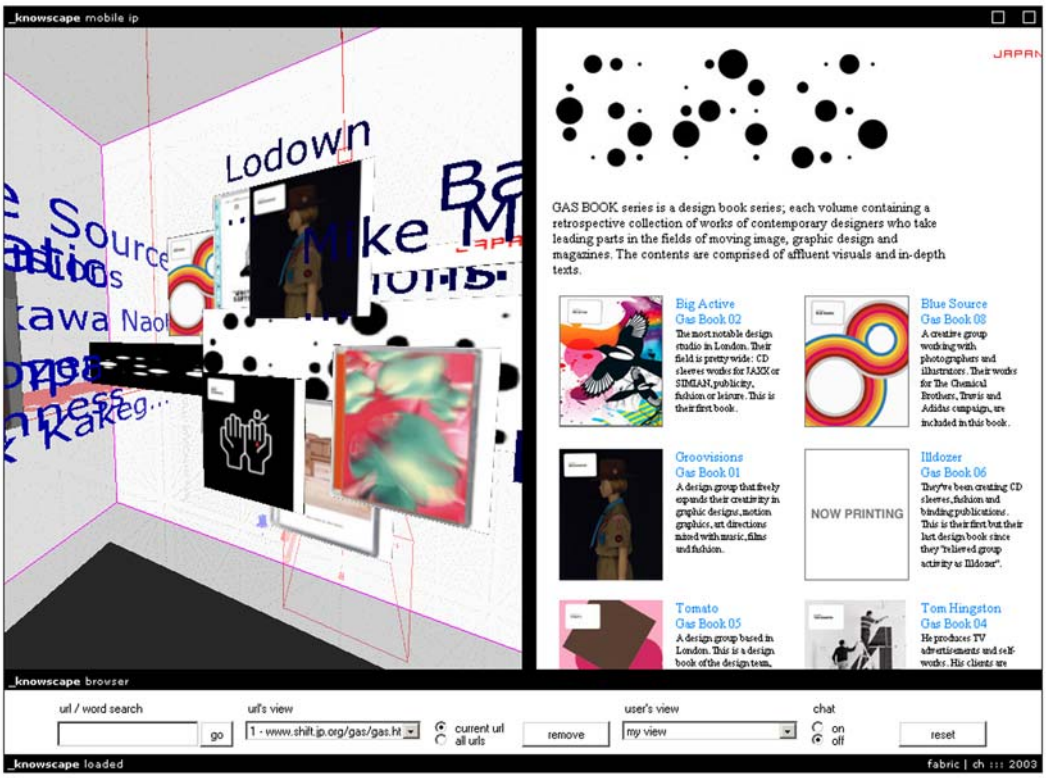
// map_IT ---------- //
// projet de campus digital pour l'EPFL // fabric | ch // www.fabric.ch // mai 2003



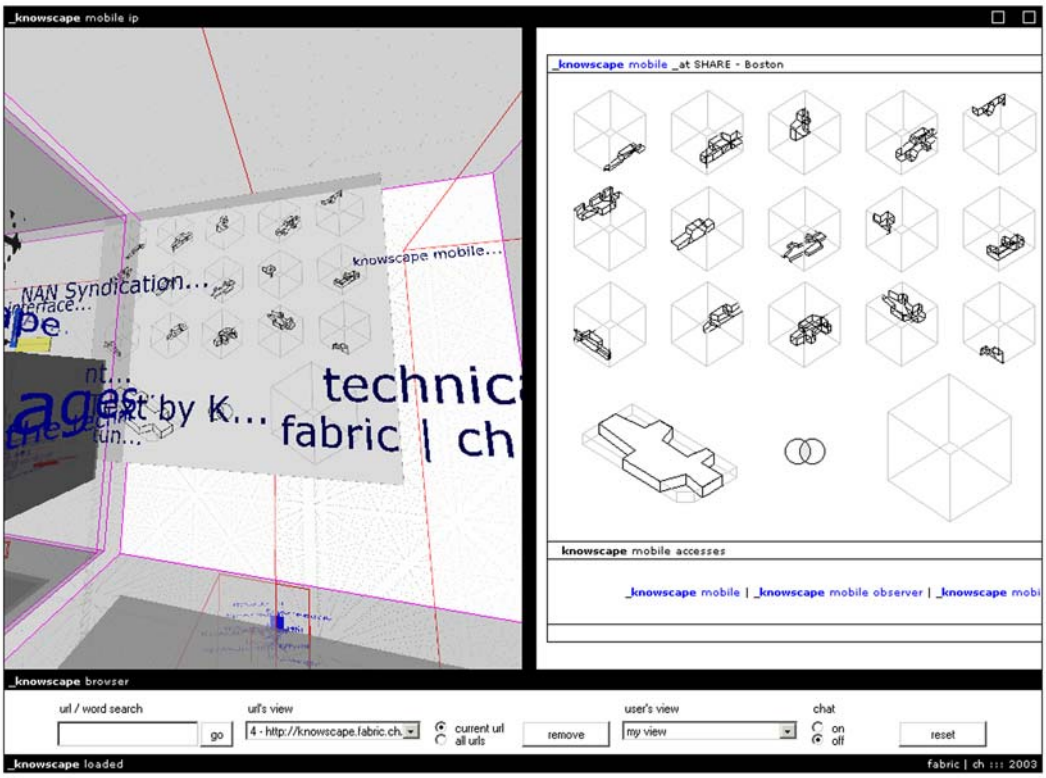
__a user and its avatar persence [in blue] with associated links and information construction made out of visited links.



__another user from SHARE [first _knowscape mobile location in Boston] in front of its construction. the display is the alternate browser with 3d display and dynamic construction on the left and the traditional 2d web browser on the right.

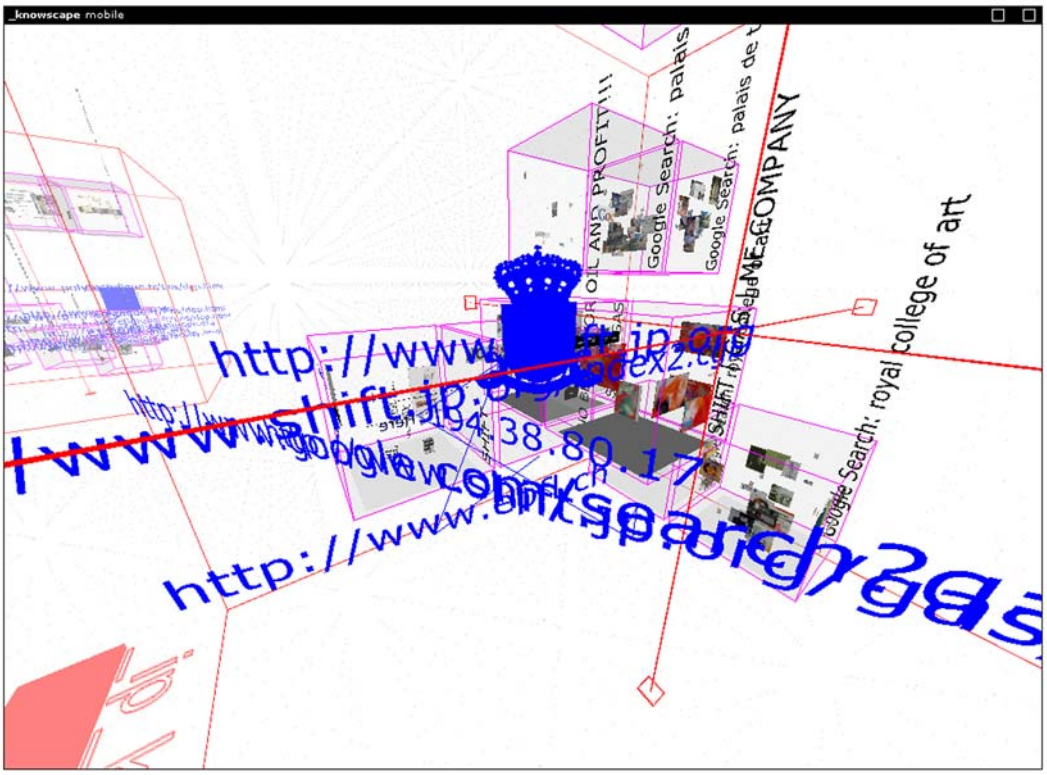


__a 3d multi-user display into the _knowscape mobile space vs a 2d traditionnal and single user display on the right. the content is the web page of SHIFT magazine, japan.

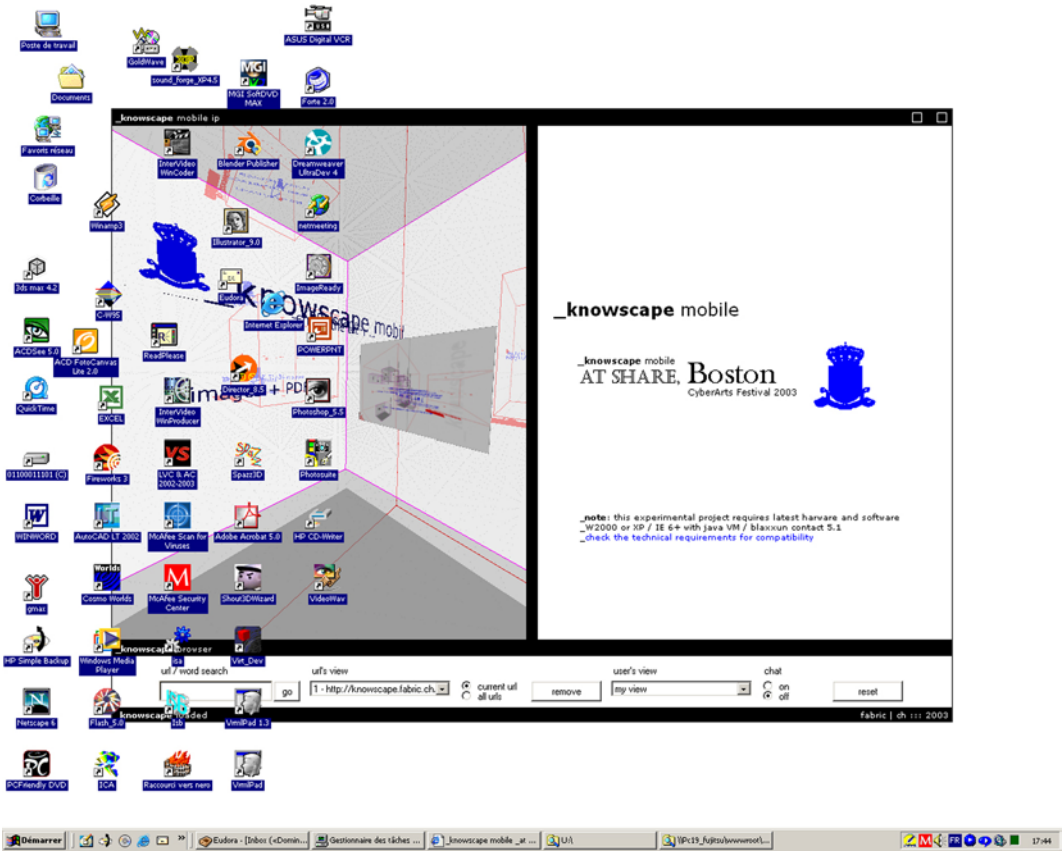


__another situation with the same image [of _knowscape mobile's website] in their relative space.

// map_IT ---------- //
// projet de campus digital pour l'EPFL // fabric | ch // www.fabric.ch // mai 2003



__a close view onto an avatar. it contains all the links that it has browsed. those links remains clickable so to experience the user's interests and share its browsing.



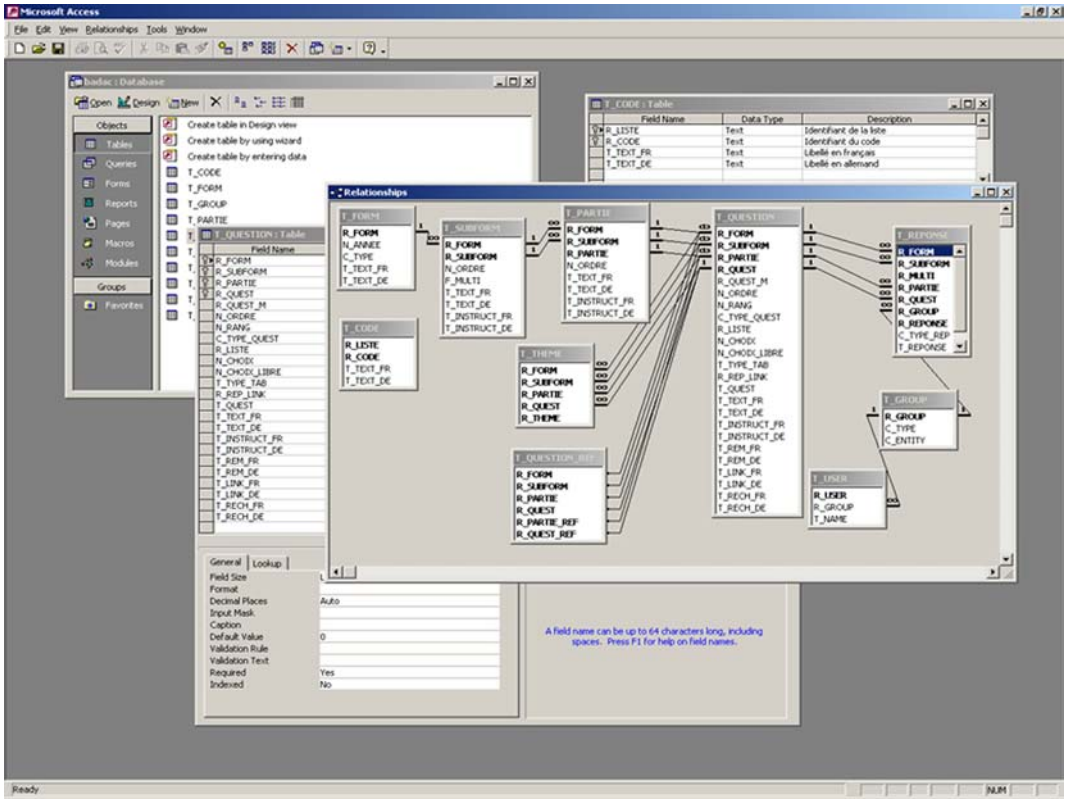
__at SHARE Boston and as an active desktop between fabric | ch's members. a shared information space between us and between our computers.

__simple presence of the micro architecture in Boston. the volume ok _knowscape mobile [basically a cube with no border] has been re-located over the swiss ambassy. screens are windows into _knowscape mobile.

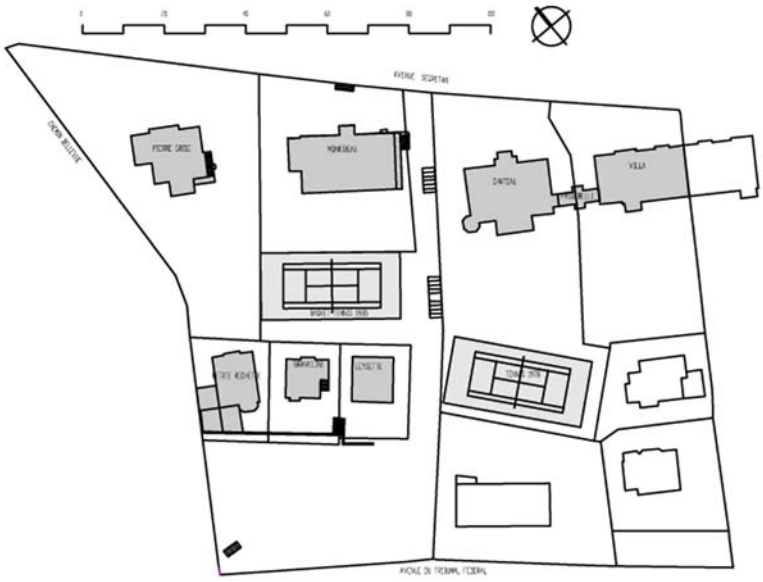
__other screens at SHARE. on the left is _knowscape mobile browser and on the right is another located view into the digital architecture.

2001-2003 +++
brillantmont digital

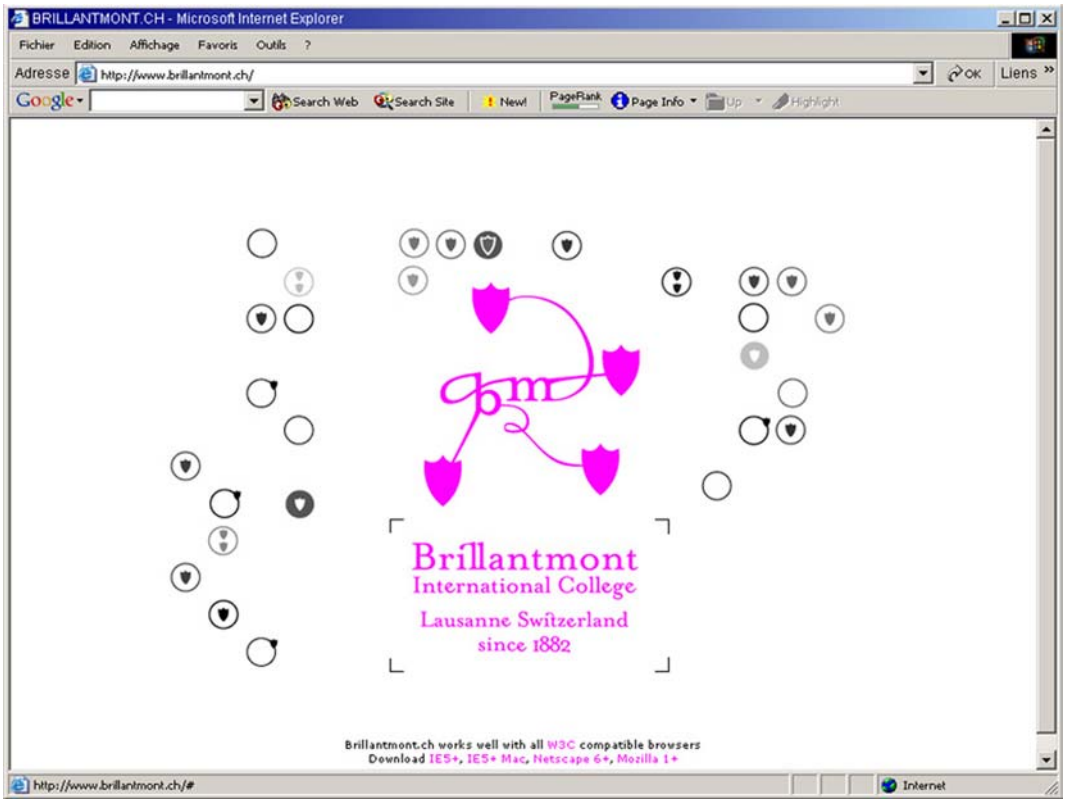
digital renovation of a school
transformation by information
databased environment
data mapping on real world
mixed environment [real/virtual]



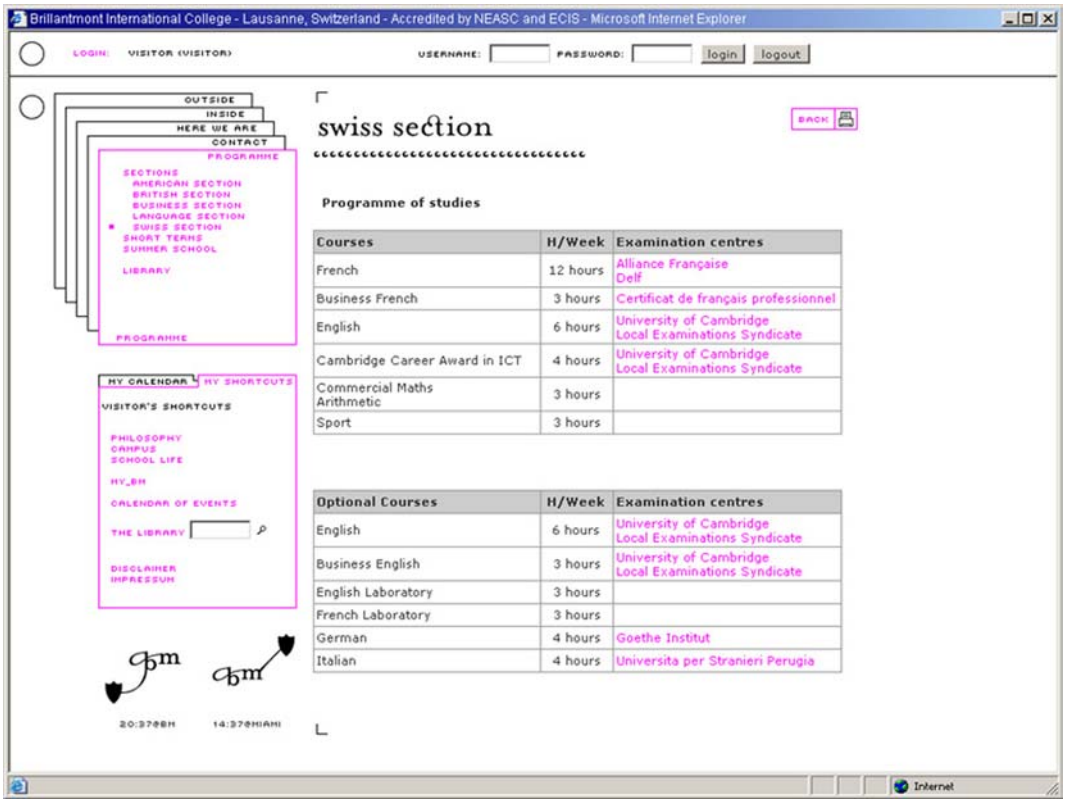
__the main tool of the digital transformation: the database that drives all the application of the school



__the main place of intervention: the school [a plan of it here with 5 main buildings]

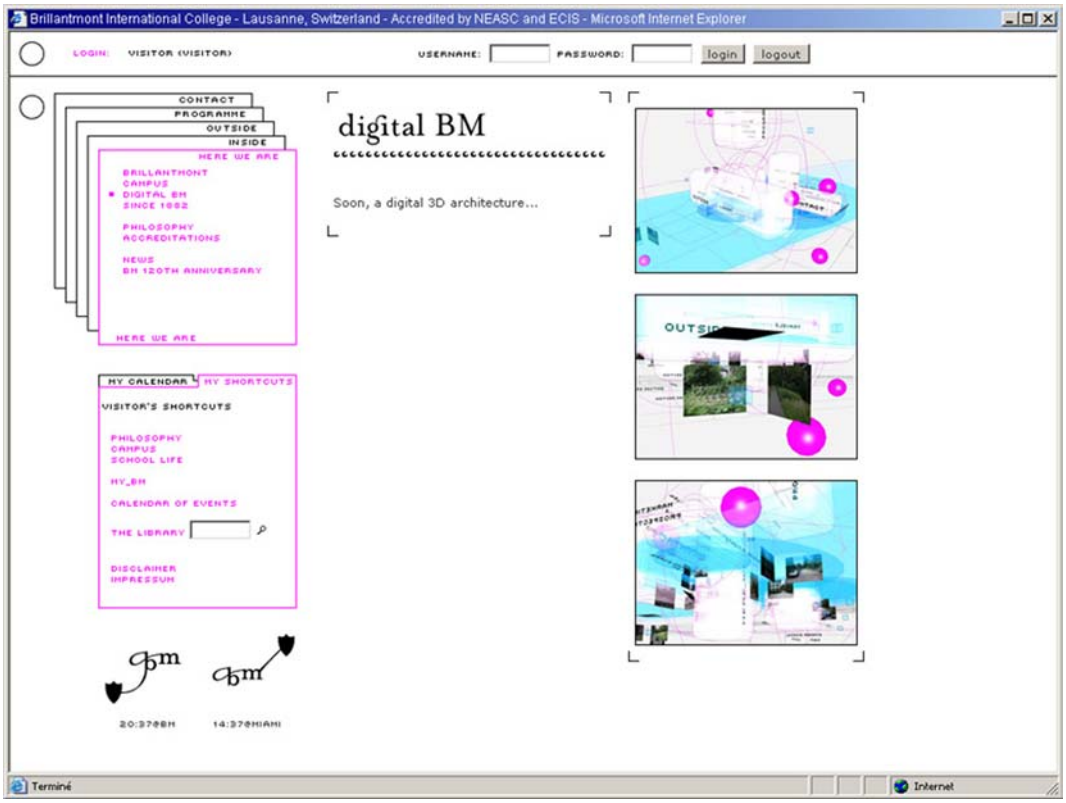


__2d databased driven environment: the web site [actual online users are the black circles]

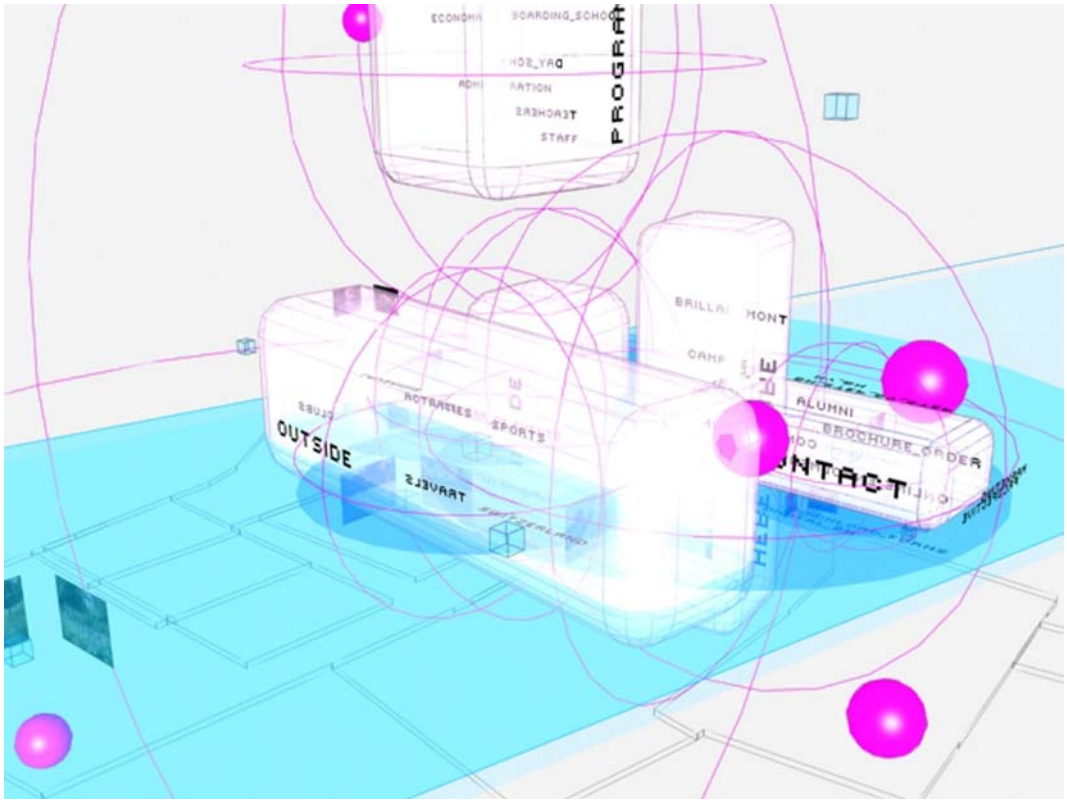


__another 2d databased driven environment: students personal program

// map_IT ---------- //
// projet de campus digital pour l'EPFL // fabric | ch // www.fabric.ch // mai 2003

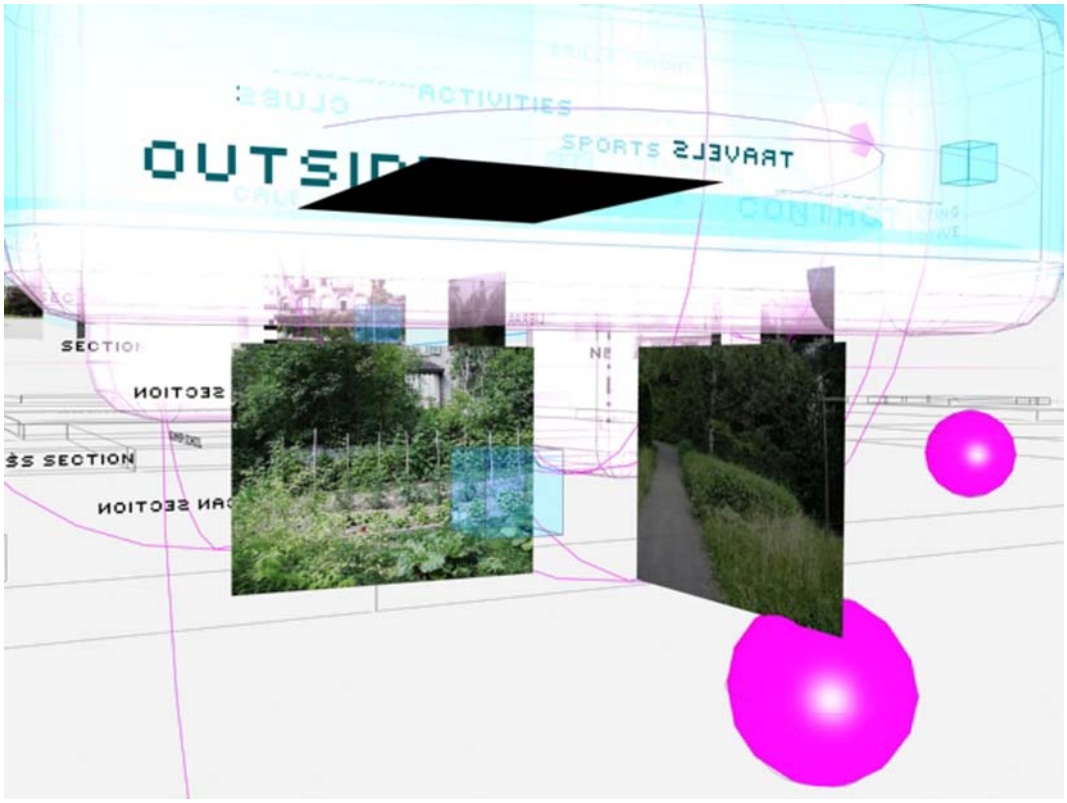


__5 main sections in this website and the annoucement of a coming 3d world

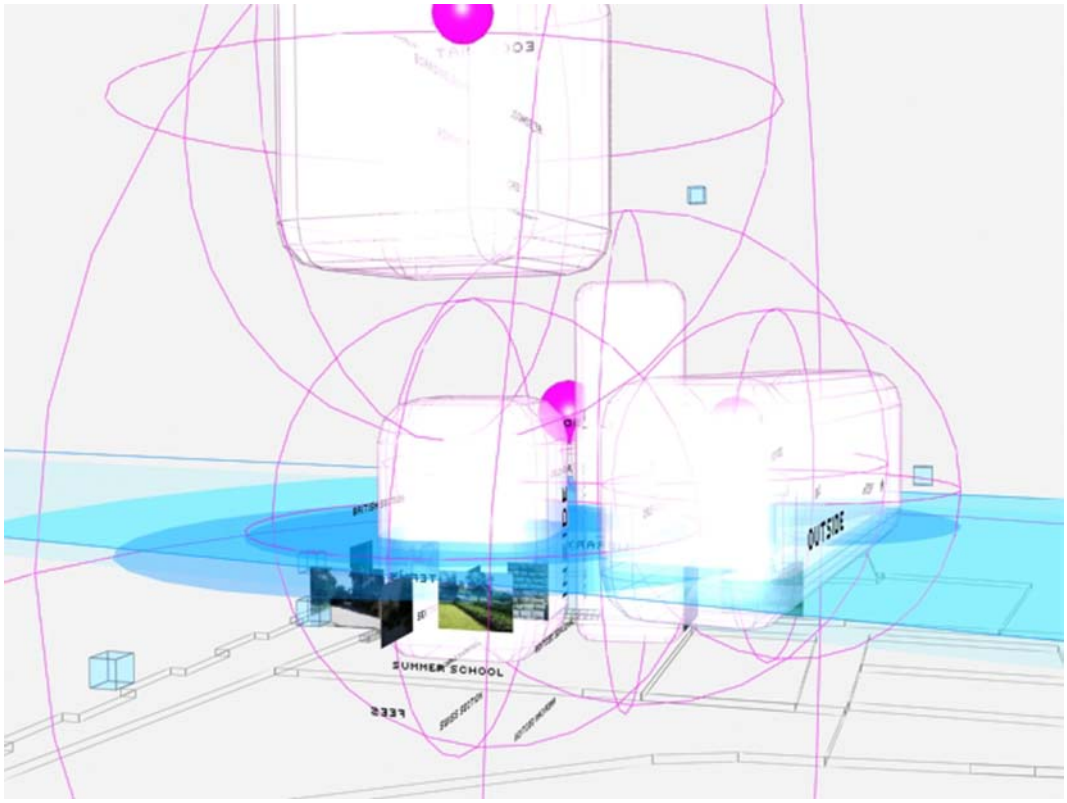


__the data world/graph produced by the database with the same sections found on the website

// map_IT ---------- //
// projet de campus digital pour l'EPFL // fabric | ch // www.fabric.ch // mai 2003

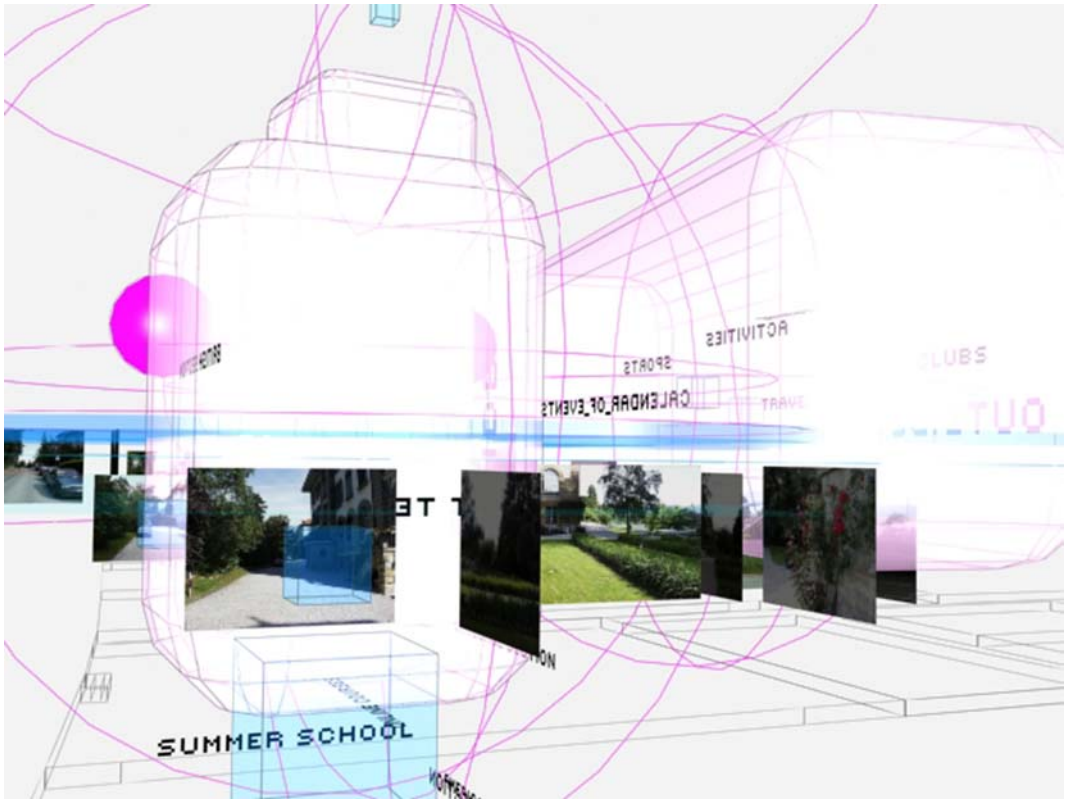


__this data world/graph is beeing organised regarding the functions/positions of the 5 main buildings of the school



__the dynamic 3d graph including metainformation [activity, number of online users, amount of data, ...]

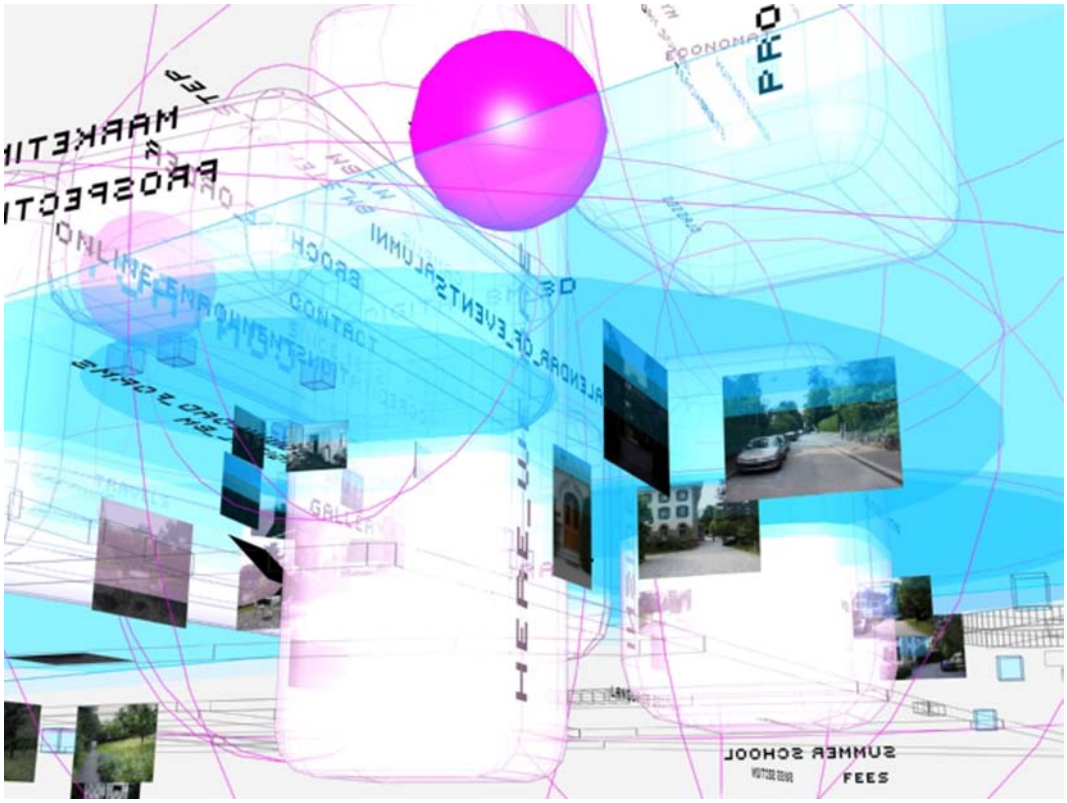
__inside view of tha dataworld



__images of the real site/school are beeing re-inserted in the 3d graph at their exact locations

// map_IT ---------- //
// projet de campus digital pour l'EPFL // fabric | ch // www.fabric.ch // mai 2003

__a view into the data and real world: an information world



__a large view of the dynamic, multi-users graph. including 3d metainformation aout its activity

__inserting the digital landscape into the real one of the school [temporary installation]

__inserting the digital landscape into the real one of the school 2 [temporary installation]

// map_IT -- //
// projet de campus digital pour l'EPFL // fabric | ch // www.fabric.ch // mai 2003

// map_IT -- //
// projet de campus digital pour l'EPFL // fabric | ch // www.fabric.ch // mai 2003