

# Combining Research Strategies for Exploring Expressive Interactions

Olivier Bau

In Situ, LRI/INRIA Futurs  
Batiment 490, Université Paris-Sud  
91405 Orsay Cedex, France  
bau@lri.fr

## ABSTRACT

The goal of my research is, by using multidisciplinary approaches, to define and conceive *expressive interactions*. Common interactive systems provide a conventional language created by designers. I am interested in giving the tools to users for creating their own way to interact with the system, by representing their personal vision of a need or function in a visual, tangible or auditory way.

## Author Keywords

Expressive interaction, open-ended devices, graphical user interfaces, tangible interfaces, communication appliances.

## INTRODUCTION

Several projects like InterLiving Project [5], the work of Gaver et al. [2] on ambiguity in interactive systems suggest that flexibility in the use of interactive systems can lead to different interpretations, which allows for the appropriation of the system by the user. Generally speaking, common interactive systems provide a conventional language created by designers. This “language” statically links functionalities to actions. In opposition to the explicit interaction rules given by the interface to the users, I am interested in what happens if we do not “impose” a way to act on the interface, but if we let the user decide/participate in the creation of his own way to interact with the system. The user should be able to express himself in a more personal way to the system.

## Expressive interaction

The goal of such expressive interactions is not to make the use of a device more productive, but to allow the appropriation of the system by the user. Beaudoin-Lafon et al. [1] introduced guidelines for the design of visual interfaces based on three principles: reification, polymorphism and reuse. Expressive interfaces should allow users to reify i.e. to create representations of their needs, or their vision, of the device's functionalities. Those

representations can be tangible, visual, auditory, gestural,... Expressive interactions give tools for the creation of those representations. For example, in order to launch a music playlist that reminds the user of a particular person, instead of choosing the playlist in the file managing system of his music player, we could give him the ability to make a sketch of this person, or draw something that characterizes him. Interacting with the system in this way allows the user to give information at different abstraction levels as opposed to choosing a label in a list. We are looking at interaction not in terms of different separated functionalities, but in relation to the user's expectations in front of a system. In opposition to conventional interfaces, in which the interaction “language” is imposed, the result will not be precise but will rely on probabilities. For example, the frontier between 2 playlists will not be discrete but continuous and the result will not be “rock” or “classical” for example but a probability of “rock” and a probability of “classical”. This can be linked to a drawing that looks much more like a “guitar” than a “violin”, for example. The behavior of this kind of system will rely on co-adaptation [8] between user and system.

## METHODOLOGY

In conception phases as well as in evaluation phases we are adopting a multidisciplinary approach, inspired by Mackay's framework for triangulation across disciplines[6]. We try to take advantage of computer science research, engineering, social sciences and product design approaches. Our methodology is influenced by Gaver et al.'s [3] work on cultural probes as well as Hutchinson et al.'s [4] work on technology probes. We use this probe approach in particular during the evaluation phases of our projects. Indeed, technology probes allow us to simultaneously test artifacts in situ and introduce new prototypes, collecting ethnographic and user data. We are trying to evaluate some aspects of our work in a more quantitative way with controlled experiments.

## Experience VS precision

PixelCom is a simple, always-connected communication device. The output of each device is an image representing the remote user. The image is displayed in low resolution. Each pixel moves randomly around its static position. The user can set the range of those movements around the static position, and the speed of the pixels. All the pixels have the same speed and amplitude. Those two parameters can be set

independently by the user. We conducted an experiment in which participants had to set a speed and amplitude that caused the image to represent their reaction to given scenarios. Different input device were used, the first being an open ended device. It was made up of a small box with a piezo sensor in it. The more the user hit the box, the more the pixels moved. A second more precise interface was a keyboard. On pressing given keys, the participants could increase or decrease the values of the two variables. This experiment suggests that, although the keyboard was more precise, the open ended device was preferred by the participants. The participants preferred this input device mostly because they could use it in different ways (eg hit it on the table in place or hit it with their fingers). This is supported by the data coming from the log of the devices. Different patterns emerged from the use of the box, in opposition to the keyboard which was used in the same way by most of the users. A questionnaire suggests that the box made the experience more enjoyable, and the users felt they could express themselves to the system, in a more closely related way to the content that the image evoked.

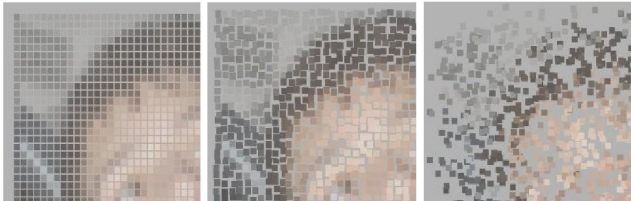


fig. 1 : PixelCom output examples

## CURRENT WORK

Mp3 players have a huge storage capacity. In front of this kind of device, users have a more high level way to think about their needs, ie the music they want to listen to. For example “calm music”, “music that makes me think to my girl/boy friend” or “fuzzy music”. There is a different level of abstraction. Although users can create playlists, neither the visualization of those playlists nor the interaction to create them has a link with those representations. The user has to make a translation between his way of representing and expressing things, and the representation and interaction rules imposed by the player .

Swirl is a on-screen sound-sharing application. Two users are sharing the same 2D space (fig.2). Users can draw 2D shapes in the window which are defining music categories. If the user wants to create a “rock” category, he can draw the word “rock”, or a guitar for example. A user that wants to draw a more abstract representation of a category can draw a square, a swirl or whatever he wants to draw. Those drawings are then shared between the two users. In opposition to common music players with which the user has to write a label for those categories, users can create a graphical representation. Those representations can be a support mechanism for communication between the two users.

The 2D shapes representing categories behave like air flows. Once a category is drawn, the user can drop some mp3 files on one of those flows. The mp3's, represented by bubbles, follow the drawn shape. The idea here is to have a dynamic representation of the mp3's in order to be able to visualize functionalities like shuffle, for example.

## Sketching interaction

Mackay et al. 's study of air traffic controllers [7] is a good illustration of the application of interaction by sketching. In Swirl, users can draw functionalities. Once the two users have defined categories, playing music becomes selecting an area of the window. The player randomly choose a track passing in the area enclosed by the selection figure. The user can draw a swirl (in green in fig. 2) or whatever represents for him in this -shared- context, the “play” functionality. The user is explicitly creating the link between the shape and function the first time he creates a new shape. We are interested in empowering the user by combining the drawings to be able to combine the associated functionalities. We will use visual feedback to guide the user during the drawing, inspired by the marking menu gesture. Indeed, the successive use of marking menus could lead the drawing of a shape. We will take into account the first part of the shape and propose directions and alternatives for the following movements in real time .

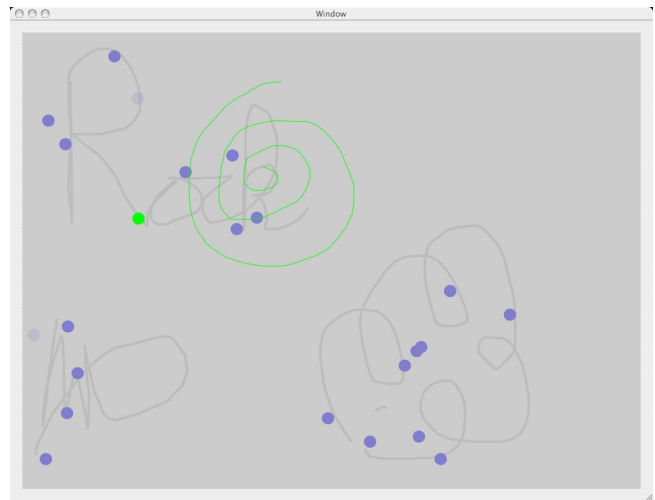


fig. 2 : Swirl

## Sketching interaction and tangible interface

We are working on applying this sketching based interaction to a tangible user interface. We are working upon past work on D20, an icosahedron-shaped screen [9]. For selection tasks, a cursor is displayed on the top of the device. It behaves like a bubble, ie the cursor is always on the top of the device. The user interacts with D20 by tilting it. This shape is constituted of 20 triangular faces. We can observe several face patterns all around the device, like a strip on the equator of the icosahedron, or a pie-like pattern. We use those patterns to display information, and to lead

the user interaction. One idea is to let the user create his own way to interact with the device by selecting by touch the faces involved in a given application. Since selecting a face pattern is equivalent to choosing a gesture, the user is drawing the interaction around the device. As in Swirl, the user is explicitly attaching a path to an application or function, and we are working on visual feedback that can guide the user effectively.

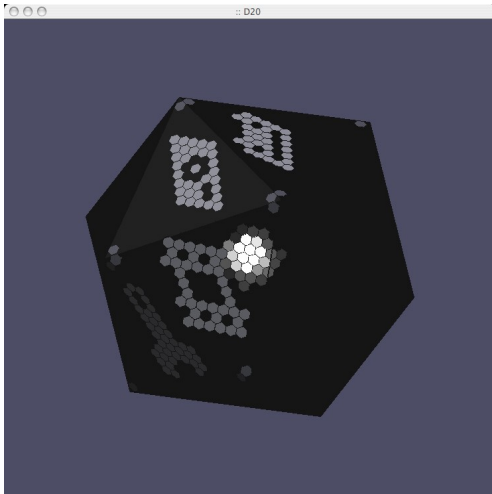


fig. 3 : D20 Simulation

## REFERENCES

1. Michel Beaudouin-Lafon , Wendy E. Mackay, Reification, polymorphism and reuse: three principles for designing visual interfaces, Proceedings of the working conference on Advanced visual interfaces, p.102-109, May 2000, Palermo, Italy
2. William W. Gaver , Jacob Beaver , Steve Benford, Ambiguity as a resource for design, Proceedings of the SIGCHI conference on Human factors in computing systems, April 05-10, 2003, Ft. Lauderdale, Florida, USA
3. Bill Gaver , Tony Dunne , Elena Pacenti, Design: Cultural probes, interactions, v.6 n.1, p.21-29, Jan./Feb. 1999
4. Hilary Hutchinson , Wendy Mackay , Bo Westerlund , Benjamin B. Bederson , Allison Druin , Catherine Plaisant , Michel Beaudouin-Lafon , Stéphane Conversy , Helen Evans , Heiko Hansen , Nicolas Roussel , Björn Eiderbäck, Technology probes: inspiring design for and with families, Proceedings of the SIGCHI conference on Human factors in computing systems, April 05-10, 2003, Ft. Lauderdale, Florida, USA
5. InterLiving Project website <http://interliving.kth.se>
6. Wendy E. Mackay and A-L Fayard. HCI, Natural Science and Design: A Framework for Triangulation Across Disciplines. Proceedings of ACM DIS '97, Designing Interactive Systems.
7. Wendy E. Mackay, Is paper safer? The role of paper flight strips in air traffic control, ACM Transactions on Computer-Human Interaction (TOCHI), v.6 n.4, p.311-340, Dec. 1999
8. Wendy E. Mackay. Users and Customizable Software: A Co-Adaptive Phenomenon. Ph.D. Thesis. 1990 Massachusetts Institute of Technology.
9. I. Poupyrev, H. NewtonDunn, O. Bau. D20: interaction with multifaceted display devices. Conference on Human Factors in Computing Systems. CHI '06 extended abstracts on Human factors in computing systems, 2006