Why Toys Shouldn't Work "Like Magic": Children's Technology and the Values of Construction and Control

Mark D Gross
Computational Design Laboratory
School of Architecture
Carnegie Mellon University
mdgross@cmu.edu

Michael Eisenberg
Craft Technology Group
Dept. of Computer Science
University of Colorado, Boulder
duck@cs.colorado.edu

Abstract

The design and engineering of children's artifacts–like engineering in general–exhibits a recurring philosophical tension between what might be called an emphasis on "ease of use" on the one hand, and an emphasis on "user empowerment" on the other. This paper argues for a style of technological toy design that emphasizes construction, mastery, and personal expressiveness for children, and that consequently runs counter to the (arguably ascendant) tradition of toys that work "like magic". We describe a series of working prototypes from our laboratories–examples that illustrate new technologies in the service of children's construction–and we use these examples to ground a wider-ranging discussion of toy design and potential future work.

1. Introduction: Magic Considered Harmful

There is a longstanding philosophical tension, in writing on technology and design, between the values of "ease of use" on the one hand, and "intelligibility" on the other. Roughly speaking–and without striving for nuance–the former philosophy is one of creating technological artifacts whose interior workings (whether software or hardware) are hidden and whose interface, as presented to the user, is intended to be quickly mastered and to suppress errors. The second philosophy permits users to understand the construction's workings, usually with the intent of encouraging the user toward a greater mastery of the artifact and (in many cases) toward a greater familiarity with engineering, science, or design more broadly.

For the most part, the "ease of use" camp in this traditional rivalry has been in the ascendant, to the point where–not to mince words–few of us have any sense of understanding or mastery of our day-to-day world. The issue is well-expressed by physicist and science writer Jeremy Bernstein:

> Most of us, myself included, are increasingly surrounded by objects that we use daily but whose workings are a total mystery to us. This thought struck me forcibly about a year ago. One day, for reasons I can no longer reconstruct, I was looking around my apartment when it suddenly occurred to me that it was full of objects I did not understand. A brief catalogue included my color television set, a battery-operated alarm watch, an electronic chess-playing machine, and a curious fountain pen that tells the time. Here I am, I thought, a scientist surrounded by domestic artifacts whose workings I don't understand. [1] [p. 196]

This passage is worth quoting at some length because it so aptly reflects the stance of virtually every adult–including highly scientifically literate adults–to their technological environment. Moreover, it can be argued that the very design of that environment actively discourages curiosity and mastery: even so simple an artifact as a light switch is designed so that the result of flicking the switch is mysterious–"like magic", in fact.

The design of toys, and in particular the incorporation of novel technologies into toys, raises the very same set of issues as the design of "grown-up" technology. Perhaps these issues are even more poignant when applied to children. After all, the need for learning and mastering technology is felt to be especially acute among the young; and presumably it is at this stage of life that longer-term attitudes toward self-competence and control of one's environment are formed. The world of toys is often referred to as a microcosm, within children's culture, of the larger adult world. Pesce's observation is representative: "[T]oys show us how we teach the ways of this new world to our children. Their toys tell them everything they need to know about where they are going...." [12][p. 12]

This paper might be seen as a discussion, with examples, of a genre of technological toys that emphasizes values of construction, building, mastery, and control. There is a frankly polemical side to this discussion, drawing on a tradition in educational technology represented by writers such as Papert [11] and Resnick [13], among others. At the same time, we wish to argue that there are certain technological developments that make this approach to toy design both more urgent and more potentially attractive than heretofore.

The following section of this paper presents a variety of projects in the genre of "technologically-enhanced construction kits". For the most part, these projects are working prototypes–they are designed to illustrate the emerging landscape of possibilities in the creation of new types of constructive media. In the third and final section of the paper, we use these collective examples as the basis for a discussion of children's construction and toy design more generally.

2. Some Toys We've Made

Many people agree that toys should do more than entertain; toys should somehow teach children something. This leads quickly to the idea that a toy should deliberately embed an object lesson: a musical toy should teach musical scales, or harmony; a building toy should teach structural mechanics. Although we certainly intend that the toys we build will do more than keep children busy, we doubt that it is fruitful to attempt to deliberately incorporate specific lessons into toys. There is more subtlety to this than toys-as-sheer-entertainment versus toys-that-teach-a-lesson. The construction toys that we describe here attempt to identify points in the space between or beyond this simple dichotomy.

2.1. roBlocks – a kit for building robots

roBlocks [14] is a kit for building robotic constructions. It consists of a set of small plastic colored blocks that snap together. Each encapsulates a function that is useful for building robots: a sensor (light, touch, sound), an actuator (rotating or translating motor, sound, light), or control (logic). Each roBblock has an embedded microcontroller and connectors on each face. The connectors both attach one block to another and conduct power and data from block to block.

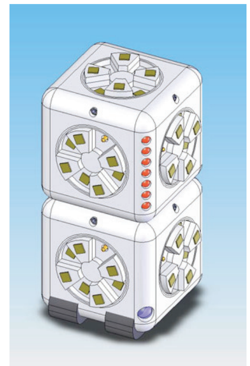

Figure 1. A light sensor atop a twisty block turns toward light.

To build a robot you assemble roBlocks that have the desired sensing, actuation, and control behaviors. For example, to build a phototropic (light following) robot, you attach a light-sensor block to a "twisty" motor block that rotates the assembly. To modify the robot to be photophobic, you add an inverter ("not") block.

2.2. Furniture Factory and Designosaur

The Furniture Factory, and its cousin software, the Designosaur [10] are environments for making model kits that can be produced on a laser cutter and then assembled by hand. The Furniture Factory is tailored to making miniature furniture, and the Designosaur is tailored to making model dinosaurs.

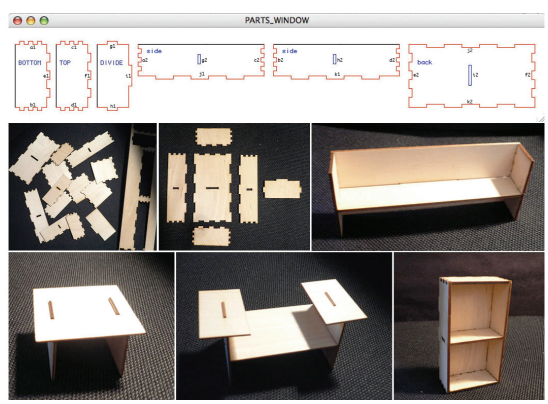

Figure 2. Furniture Factory produces model chairs, tables, and bookshelves out of basswood.

To make a furniture kit in the Furniture Factory you draw an isometric sketch of the object you want to build. The program identifies the planar parts of the furniture object, and suggests a scheme for jointing the parts together, which depends on the local arrangement of the parts. It then lays out the parts in a file to be laser cut. Figure 2 shows the parts laid out on the screen (above) and cut parts as they are assembled into furniture (below), Likewise, in the Designosaur you draw individual parts of a dinosaur model you would like to build. The software coordinates the parts, produces a three-dimensional graphic model, and adds notching details to aid in assembly, then outputs a file for the laser cutter. Figure 3 shows the computer graphic model on screen (above) and an assembled model produced with Designosaur (below).

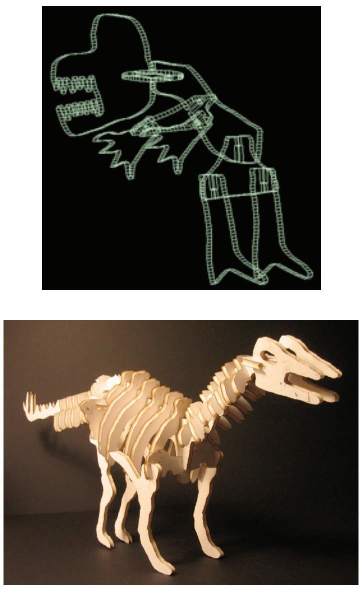

Figure 3. Designosaur produces custom dinosaur kits out of wood or foamboard.

2.3 Quilt Snaps

Quilt Snaps [2, 3] is a fabric-based construction kit in which the pieces are computationally-enhanced fabric patches. Each piece contains a microcontroller, LED, and a set of "input" and "output" snaps (as well as snaps that can be connected to power and ground). Once a quilt snap piece is powered, its embedded computer program iteratively queries its input port; if a signal should arrive, the program sends a display signal to its LED light and then, after a fixed time, send a signal to its output ports. Thus, by snapping together sequences of fabric pieces, with output ports connected to other pieces' input ports, the user can construct fabric-based spatial programs in which light patterns move from patch to patch. Each Quilt Snap piece may be decorated individually with both craft materials (felt, marker pens, etc.) and electronic materials (sewn in with conductive threads). Thus the kit combines elements of electronic design, programming (in a simple dataflow-like model), and old-fashioned fabric crafts. Figure 4 below shows a collection of four decorated Quilt Snaps forming a loop (with a power strip attached to the piece at bottom right, and a "touch strip" connected to the input port at upper right that allows the user to initiate the counterclockwise looping pattern of light).

Figure 4. A Quilt Snaps construction.

2.4 SmartTiles

SmartTiles are small cubes (about one inch per side), each of which contains a microprocessor, an LED light, and a piezoelectric disk to make the cube touch sensitive. When assembled in an array that supplies power, communications, and a clock signal, the cubes can collectively enact cellular automaton programs (like the well-known "Game of Life" program popularized by Martin Gardner in *Scientific American* [7]). Each cube runs its own internal, and individual program: for the Game of Life, all cubes run the same simple program that can be summarized as "if I have three live (red) neighbors, then I will be live (red) at the next time signal; if I have two live neighbors and am currently live, then I will be live at the next time signal; otherwise, I will be 'dead' (dark) at the next time signal".

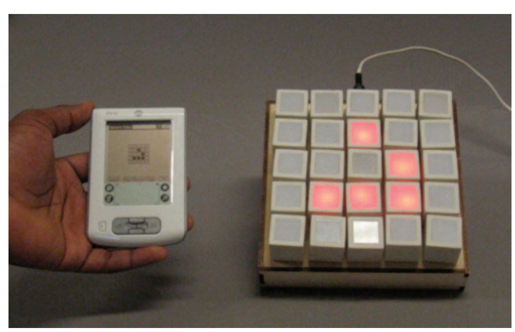

Figure 5. A user communicates wirelessly with a five-by-five array of tiles using a handheld

computer. Here, the user resets an initial configuration of cells (see next figure).

Importantly, there are numerous ways of programming the cubes. An individual cube may be removed from the array and brought to a desktop computer, where it may be given its own particular program and re-inserted into the array. Alternatively, a set of cubes can be programmed through the use of a handheld device. Figure 5 shows a user communicating with a five-by-five array of the tiles using a handheld computer. In Figure 6, several different modes of "reprogramming" are shown. The user can reset an initial configuration of cells (choosing which will be red or dark); she can use a menu-driven interface that permits selection of a wide range of cellular automaton rules; or she can go still further and program the tiles using a textual language interface.

Figure 6. Three types of SmartTile programming: setting initial configurations of "live" and "dead" cells (upper left); textual programming (upper right); and menu-driven interface (bottom).

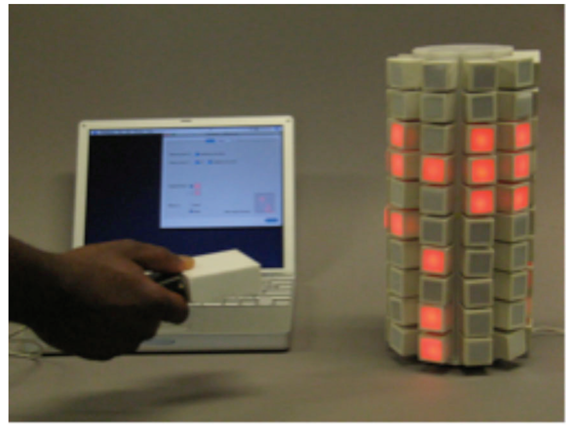

Figure 7. A cylindrical array of tiles being programmed wirelessly.

Much more thorough description of SmartTiles may be found in earlier publications [4, 5]; for the present, the essential point to make about the tiles is that they give the user a high degree of control in dictating their behavior. Our ultimate goal is to think of the tiles as an all-purpose programmable "surface covering" medium–to create large-scale arrays of tiles over surfaces with interesting geometries, and then to allow users to create and combine programmed behaviors over those surfaces. A hint of this idea can be gleaned from Figure 7, in which a set of tiles has been arranged around a cylinder; more generally, we could imagine tiles being placed on wall-sized surfaces (indoors or outdoors), on spheres, and so forth.

2.5 Popup Workshop

Technological tools can also be used in the service of relatively "low-tech" crafts and construction. An example of this is the Popup Workshop program, a design tool that permits children to design, decorate, and print their own popup constructions.

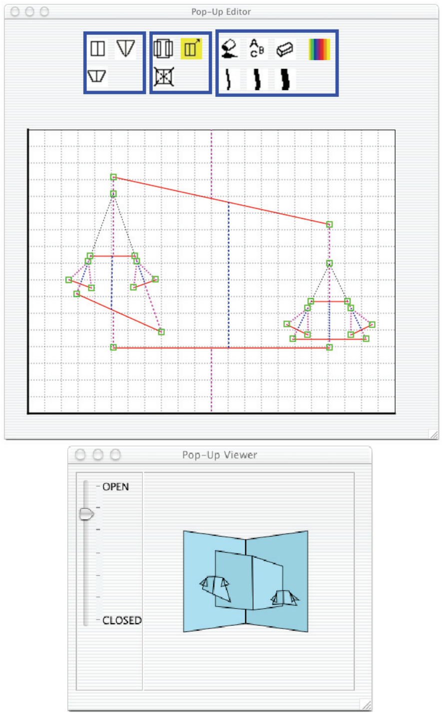

Figure 8. A screenshot of the Popup Workshop application in the midst of a construction. At top, the "virtual paper" has a number of popup elements placed along creases; at bottom, a 3D animated view of the eventual construction.

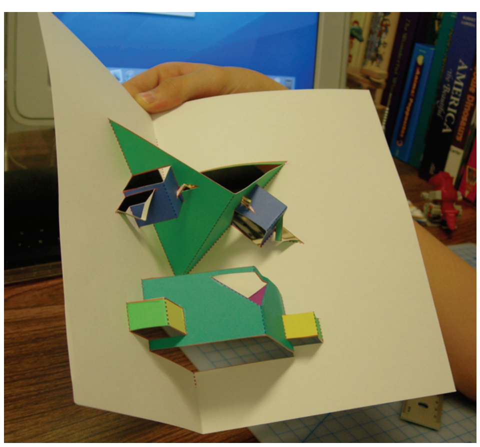

Figure 9. A popup card designed and created by an elementary-school student using the Popup Workshop system.

Although space does not permit a thorough description of the program [see [9]], the basic idea behind the system is that children can place popup elements on a "virtual sheet of paper" on the screen; decorate the resulting sheet using computational tools; and then print out the form on a color printer, cut at the specified locations, and create a tangible popup card. A screenshot of the tool is shown in Figure 8: here, the user has specified a number of popup elements on the virtual paper, and can see an animated 3D rendering of the eventual popup form as well. Figure 9 shows a popup construction done by an elementary-school student using the program.

3. Toys for Makers

The previous section described a number of projects illustrating different types of technologically-enhanced construction–of robots, of furniture and dinosaur models, of programmable quilts and surfaces. We can now step back and use these projects as points of reference to discuss the issues of construction and control more broadly.

Making things is hard. Making good things and complex things requires practice and ultimately mastery. The satisfaction of achieving mastery of making, and the kinds of things that can be made are the rewards for the effort and learning that inevitably goes into it.

There is no virtue in unnecessary labor. Good makers look for ways to streamline the process of making things. We try to design toys and construction activities to minimize needless nonsensical annoyances that distract and discourage. Clearly, there is a fine balance between eliminating needless complexity to make a more elegant design environment, and hiding important detail in the name of ease-of-use.

Our roBlocks construction kit, for example, avoids the hassles of wiring up electronic components and line-by-line programming in a low-level (i.e., C) language. This low level of robot construction demands that the designer think about details such as voltage levels on components and program memory allocation. Instead we provide a kit of parts that are appropriate for the kinds of things robot makers really should think about: processing sensor inputs and planning and controlling motion. We hide uninteresting and unimportant details to allow our robot designers to focus on the main task of robot design: making things that sense, plan, and actuate. Or, to take an additional example: the Popup Workshop program permits the designer to experiment with a variety of potential elements quickly on the computer screen; experimenting with a huge number of paper constructions would likely prove dauntingly tedious for a newcomer to the craft. Because the user is able to explore a much larger variety of design possibilities, she can create more complex, content-rich, challenging, and aesthetically compelling creations (as suggested by the example in Figure 9).

Always there are trade-offs. Potentially, the Popup Workshop user is losing valuable experience in getting a feel for the medium of paper over the course of many failed experiments. Or, to return to the roBlocks example: arguably, a robot designer who works at circuit and C-code level has more power and flexibility than a designer who uses roBlocks. Yet working at the lower level also requires attention to detail that may matter little for the gross design of a working robot. Think of roBlocks as a prototyping tool for robot design. The value of a prototyping tool is the power and richness of the universe of designs that can be made with it. The art of designing a prototyping tool like roBlocks is in choosing what to encapsulate (and thus hide) and what to expose to the maker.

3.1. Discussion

The toys and craft activities that we have described, like the toys and play that we liked best ourselves, provide children the means with which to make things. The particular things that they can make matter far less to us than the sheer ability to make things— and the corollary message that our toys convey: Children can be master makers of a world of their own. Certainly, different spheres of activity (programming, sewing, music, building) appeal to different individuals, but the overall message inherent in the toys is the same.

In our world of consumer products we are constantly encouraged to buy, not to build. Whether dinner from the local fast food establishment or downloaded music from the internet, the idea that people can make things themselves seems more and more remote, at times even quaint. Somehow, moreover, the buy-it mentality is associated with technological advance.

Yet we feel strongly that the ability to make things ought not be seen as a relic of a simpler past, but on the contrary as an essential ingredient of a vibrant future. Our future as a society depends on our ingenuity—our ability to engineer, and our creativity—our ability to make. Creativity and ingenuity must be encouraged and fostered at an early age. That is ultimately why we are working on toys and play activities that will expose the challenges and pleasures of making things.

We believe that creativity and ingenuity can be learned, or at least, it can be fostered. The way to learn creativity and ingenuity is through practice creating and engineering, in short, by making things.

There is a strong case to be made that the current time is right–technologically speaking–for just such a turn toward construction and creation in children's toys and activities. The advent of accessible fabrication tools (like the laser cutter in the Furniture Factory and Designosaur examples) means that children can now work in materials such as wood and plastic with nearly the same fluidity that they have with paper and string. Likewise, the Quilt Snaps example shows that children's fabric crafts can now be vastly augmented through the use of inexpensive conductive threads. Embedded computation allows the design of construction kits with potentially complex and interesting behaviors (as evidenced by roBlocks, Quilt Snaps, and SmartTiles). In short, then, rather than seeing the advent of technology as a reason to make toys more mysterious and opaque, we argue that an emphasis on children's construction dovetails neatly with the potentialities of new fabrication, material, and computational technologies.
When it comes to the educational value of construction, we also argue that much of what children learn in one field of making can apply to others. Beyond the specific skills that inevitably must be acquired to master a particular craft, making things also breeds general skills and ways of thinking that are valuable for any kind of making. For example, although programs and poems would seem far apart on the spectrum of creative artifacts, both result from a deliberate generative process that involves adherence to a system of inner logic (poems and programs both must “work” and adhere to rules that govern their syntax), a knowledge of forms and idiom, and a progressive selection among alternative decisions, and so on.

On the parallel between poetry and programming, Dick Gabriel, a distinguished and visionary programmer, and senior engineer at SUN Microsystems, in discussing his proposed Master of Fine Arts in Software Engineering, says this:

> When I'm writing poetry, it feels like the center of my thinking is in a particular place, and when I'm writing code the center of my thinking feels in the same kind of place…
>
> I'm thinking about things like simplicity -- how easy is it going to be for someone to look at it later? How well is it fulfilling the overall design that I have in mind? How well does it fit into the architecture? If I were writing a very long poem with many parts, I would be thinking, "Okay, how does this piece fit in with the other pieces? How is it part of the bigger picture?" When coding, I'm doing similar things, and if you look at the source

> code of extremely talented programmers, there's beauty in it. There's a lot of attention to compression, using the underlying programming language in a way that's easy to penetrate. Yes, writing code and writing poetry are similar. [6]

And focusing on "makers", here is what, author, programmer, entrepreneur, and painter Paul Graham has to say about "hacking and painting":

> When I finished grad school in computer science I went to art school to study painting…. What hackers and painters have in common is that they're both makers. Along with composers, architects, and writers, what hackers and painters are trying to do is make good things. [8][p.18]

3.3. Teaching magic

We began with "Magic Considered Harmful," arguing for intelligibility over ease-of-use. But in truth we need not sacrifice one for the other: there is no inherent reason why intelligible ("glass box") versions of technologies must be more difficult to use than "black box" versions. In proposing a model for "transparent interfaces" Tanimoto [15] examines the dynamic between encapsulating detail (the black box approach to engineering) and revealing the inner workings of a system (the glass box approach for keeping users informed about what the system is doing).

Science fiction author Arthur C. Clarke is credited with saying that, "Any sufficiently advanced technology is indistinguishable from magic." Technology seems like magic when we do not understand how it works.

Our technology-enhanced toys reveal, rather than conceal, their fundamental principles. By exposing the magic within we aim to intrigue and inspire a new generation of ingenious, creative, makers: in short, to educate a new generation of magicians.

Acknowledgements

Ellen Yi-Luen Do co-advised the Furniture Factory and Designosaur projects and has contributed to many of our discussions of toy-making and computational construction kits. Needless to say, we also appreciate the efforts of the students whose work we mention here: Buechley, Dodson, Elumeze, Hendrix, Johnson, Oh, and Schweikardt. as well as others who have worked on related projects in our laboratories. This research was supported in part by the National Science Foundation under Grant ITR-0326054 and REC-0125363, and the Pennsylvania Infrastructure Technology Alliance. The views and findings contained in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

[1] Bernstein, J. *Cranks, Quarks, and the Cosmos*. Basic Books, New York, 1993.

[2] Buechley, L., Elumeze, N., Dodson, C. and Eisenberg, M., Quilt Snaps: A Fabric Based Computational Construction Kit. in *IEEE International Workshop on Wireless and Mobile Technologies in Education (WMTE 2005)*, (Tokushima, Japan, 2005).

[3] Buechley, L., Elumeze, N., Dodson, C. and Eisenberg, M., Quilt Snaps: A Fabric Based Computational Construction Kit (Poster presentation). in *Wireless and Mobile Technologies in Education (WMTE 2005)*, (Tokushima, Japan), 219-221.

[4] Elumeze, N. and Eisenberg, M. SmartTiles: Designing Interactive 'Room-Sized' Artifacts for Educational Computing *Children, Youth and Environments*, 2005, 54-66.

[5] Elumeze, N. and Eisenberg, M., SmartTiles: Mobility and Wireless Programmability in Children's Construction and Crafts. in *Wireless and Mobile Technologies in Education (WMTE 2005)*, (Tokushima, Japan, 2005), 230-237.

[6] Gabriel, R. The Poetry of Programming, SUN Microsystems, 2002.

[7] Gardner, M. *Wheels, Life, and other Mathematical Amusements*. W. H. Freeman, New York, 1985.

[8] Graham, P. *Hackers and Painters*. O'Reilly, Sebastopol, CA, 2004.

[9] Hendrix, S. and Eisenberg, M. Computer-Assisted Pop-Up Design for Children: Computationally-Enriched Paper Engineering. *International Journal on Advanced Technology for Learning, , 3* (2). 119-127.

[10] Oh, Y., Johnson, G., Gross, M.D. and Do, E.Y.-L. The Designosaur and the Furniture Factory simple software for fast fabrication. in Gero, J. ed. *Proceedings Second International Conference on Design Computing and Cognition*, 2006, in review.

[11] Papert, S. Situating Constructionism. in Harel, I. and Papert, S. eds. *Constructionism*, Ablex Publishing Company, Norwood, NJ, 1991, 1-11.

[12] Pesce, M. *The Playful World*. Ballantine Books, New York, 2000.

[13] Resnick, M. and Silverman, B. Some Reflections on Designing Construction Kits for Kids. in *Proceedings of the ACM Conference on Interaction Design and Children*, Boulder, CO, 2005, 117-122.

[14] Schweikardt, E. and Gross, M.D., roBlocks: A Robotic Construction Kit for Mathematics and Science Education. in *International Conference on Multimodal Interaction*, (Banff, Alberta, 2006), IEEE.

[15] Tanimoto, S., Transparent Interfaces: Model and Methods. in *Workshop on Invisible and Transparent Interfaces* (Gallipoli, Italy, 2004).