Toward participatory design of computer-assisted immersive environments

A position paper for development of new facilities and programs at
the School of the Art Institute of Chicago

Insook Choi

1. Introduction

The discussion *Toward participatory design of the computer assisted immersive environment* involves two problem areas: the participatory design process itself and the technological/physical configuration of the immersive environment. The proposition is that a group of people with diverse expertise and skills participate in designing a computer software environment as well as designing physical facilities.

The concept of participatory design opens a rather unpracticed territory for an art community. Several implications for planning such process hinge around the problem of articulation and communication capability for the processes of art production rather than the art products themselves. For the participants in art production design the question is "what do we design and how do we participate?" Though this question might sound simple and obvious many conditions add complications before shedding light on the question. To mention a few, among the prevailing conditions are:

1. Heroic model of individual artist
2. Lack of historical precedent for the need to articulate and communicate the art production processes
3. Lack of the evolution of aesthetic criteria that counter-faces the evolution of technology.

For an individual artist there has been no historical precedent that demanded such degree of precision for articulating work processes if one is to dive into the participatory design process. It has been a generally assumed description that rolled-up-sleeves studio artists go into the closed room and come out with magic results that impress the public. The process behind the magic results is left unarticulated while it might have been empowering to a few selected individuals as "genius". However, outside of the art community a significant part of technological evolution is achieved through the endeavors to make available shared knowledge bases and data bases, which often lead to successful growth of that field. To do so the community establishes common foundations for languages and formalized systems and structures that convey working models that are functional to the domain applications. Examples can be seen in fields such as medicine and library science. Under this consideration the current reputation and habits in an art community may not necessarily empower its future.

At the same time the philosophy of art production as an activity and creativity as cognitive issues resides in different and alternative motivations in the human mind from and to the other fields mentioned above. The philosophy demands more than establishing common knowledge and database systems. Not only should we establish the common grounds in our working environment, also we can seed the functional descriptions such that what we establish in computational tools and environments does not affirm or confine individual creativity. This criterion suggests that we assume two research areas in an ongoing base: 1. *Computability problems*: how much processes can be formalized in computer software environment and how much is to be left for the human mind, and 2. *Intelligent and adaptable environment design and testing*: how can we design and test an environment that assists and functions along with human intelligence with adaptability.

The proposition on the table is to design a facility with a supporting software environment in which we can conduct both art production and related research, and where the design process is to be participatory.

2. Participatory design

The participatory design practice has an origin in Scandinavian countries where the community of “users” actively participated in influencing the look and feel of final products in industrial settings. Our present situation goes further to the extent a participatory design is a proposition as much as a method for determining consensus. We are not necessarily bound to certain product design. One aspect of the proposition is more about the design of an environment where the description of people in that environment goes beyond that of users. The function of users is replaced by the function of observers in the sense of cybernetics: the observers change and influence the dynamics of objects and environments by ways of making observations.

An initial part of such designing process is to examine our own task environment: what are the tasks and where are they usually carried out. Then the analysis should guide us, not prescribe, for planning the technological implementation.

Our goal is to establish community languages that empower communications for within and outreach.Unless we know how to communicate with each other the body of knowledge does not empower the rest of the society.

Examination areas involve

- Tools and working paradigms for individuals
- Tools and working paradigms for group of people
- Computer assisted collaborative framework
- Social organization and social interaction portraits.

Implementation areas involve

- Multi-modal display and interactive devices
- Division of labors between human and machines
- Division of labors within computational tasks
- Computational architecture to host collaboration and social interaction.

To see why these are new problem areas we revisit HCI definitions and fundamental implications of computer architecture.

2.1 Human computer interaction (HCI)

The goal of research in HCI is to achieve an extended working process by creating synergy between human and machine. We note HCI has been focused on novel interactive devices and psychological studies given the particular devices. What have been missing are comprehensive case studies on the integrated multi-modal signals in inter-operative environments between human and machine, and between human and human in a computer assisted environment. Human computer performance systems put an emphasis on human operability and functional presence by configuring computer performance reciprocal to human performance. The nature of reciprocity, however, goes deeper than just mirroring or extending human action. Computation takes the significant portions of division of labor with the degree of autonomy that often exceeds human capacity to grasp processes undergone. This is good without agreeing with Minsky who states that human brains are badly designed and can be replaced by better designed chips in the future. Before we rush into "Neuromancer" scenarios and to genome gold mines the sober political human minds need to generate a wealth of environments where human computer intelligent interactions can be practiced and rehearsed. Desirably this kind of initiative comes from the art community.

2.2 Human Computer Performance

The environment envisioned in a Human Computer Performance System enables the circularity of interactive signals passing through its constituents. Inevitably included are non-computable ranges of possibilities that human performers bring. When we actively interface the computable domain to non-computable ones, new problem areas emerge in the circularity in ways we can revisit the task environment and address the new problem areas with concrete engineering approaches. The issue of competence should be addressed in both ways: am I literate to the language of the computer, and is the computer literate to any language of mine? The compatibility between human and computer has to be evolved in dialectical processes. Why this has to be even explicitly stated has to do with contemporary research interests which tend to put heavy weight on rational thinking strategies that are more apt to be computable. This condition is founded on the historical references represented in Figure 1.

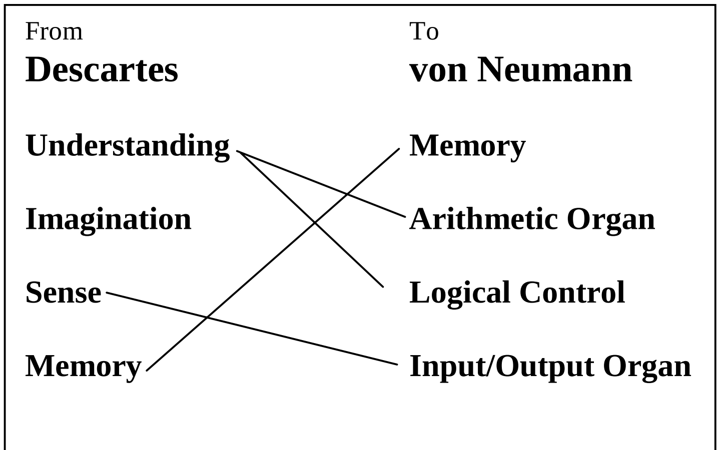

Figure 1: Historical references for computer architecture.

The four faculties Descartes brings are "only useful faculties" for mathematical thinking. They have loose relations to von Neumann's basic specifications for what computers ought to have to (attempt to) resemble the human brain. Descartes' four faculties are understanding, imagination, sense, and memory. Almost in parallel von Neumann specifies the computer capacities: memory, arithmetic organ, logical control, and input-output organs, and these four capacities are not necessarily equal in terms of nesting since most iterative functions are stored in memory (an active organ) and most computations are iterative processes. What is notable when we put two charts together is there is no obvious connection to "imagination." Descartes might have had a benefit of not being confined in the realization of machines and had a luxurious position for abstract thinking. Von Neumann's computer design specification is obviously founded with respect to the physicality of machines. What is important here is to realize neither of the two modalities thus described will be sufficient for us in designing an HCI environment, and they were never meant to be. Not only do we need more memory capacity in the computer, also different ways of organizing memory that facilitate sensory feedback and interactivity. This is a research issue.

3. Designing a computer assisted environment

I am concerned with undesirable political implications between human and machine only because of the mistaken assumptions we live with in this technological era. These include human inferiority to machines, hacker models of techno-wizards, and underestimating what is NOT computable. The last, to state, ***not everything is computable***. And I am confident to say that we may not run out of resources of non-computable aspects of life as long as we respect our own intelligence.

Proposition: a laboratory that deliberately integrates interactive signal paths. These paths are re-configurable for projects with components both inside and outside of the computer. The design of this laboratory actively assumes concern for the non-computable domain of activities and the interface with computable domains as much as one can. This design leads to on-going processes of pushing the boundary of computability.

We identify two key principles for designing these capabilities:

<u>Social Organization as Computational Architecture</u>: the technologies that we take up as tools either reflect, facilitate or predetermine our interactions as a group.

<u>Distributed Cognition</u>: design of division of labor leads to the distribution of computational tasks among components in an immersive environment.

To consider tools in social context, we distinguish tools for an individual purpose vs. tools for a team. Can we design tools to accommodate their use in both social contexts? This accommodation may be described as a *utility bandwidth requirement*.
Design for tools for a team requires more information concerning the use of the tools: how are the tools used individually, and how do people use the tools together?

A design of collaborative tools will encounter these concerns:

- Quantitative vs. qualitative service -- It has been said: Increased bandwidth may provide more data flow quantitatively, but not qualitatively. Can we avoid this either-or situation?
- Confirmation Bias — affirming a prior interpretation and discounting or reinterpreting new evidence that runs counter to the already affirmed interpretation. Why does this continue in society? Can we avoid confirmation bias in a group? “When all you have is a hammer everything looks like a nail” – do systems where we interact tend to uphold confirmation bias or help us to avoid it? Can a new tool be made that solves more than one problem?
- Cognitive dissonance -- The group performing the task may have cognitive properties that differ from the cognitive properties of any individual. Cognitive properties "between" groups may depend on the differences in social organization rather than cognitive properties of individuals in the groups. How do we map individual behavior to provide insight/ information to the distributed team to efficiently plan for coordinating activities?

3.1 Software as an Environment

We are concerned with more than the creation of another computer application, dedicated gadget or tool. A software infrastructure enables software as an environment. Taking a comprehensive view, we envision a space where multiple tools can be designed and integrated.

In rigorous definition, an environment is a living system. It is an adaptive system that changes over time, performs context dependent responsibility, and supports design while being subjected to redesign. For software to constitute an environment, it should be accountable for the four layers of contextual expansion:

1. Performs *functions or tasks* as well as supports the design of new functions and tasks;
2. Provides *utility* of the functions as well as supports the invention of new utilities;
3. Enables the *contexts* in which the utility is meaningful as well as supports creating new contexts or translating the utility into alternative contexts;
4. Provides *infrastructure* for collaboration

To evaluate and to bring about a software environment capability, the Immersive Technologies research project covers both technology and faculty committee development. Both components -- faculty committee and technology development – recognize a need and undertake action items to reach an initial objective to address the need.

The Immersive Technologies project recognizes the need for faculty to collaborate in the design of an environment to support immersive technologies and related interactive design research. To address this need the committee is organizing series of meetings to identify a participatory design process and define a framework to be practiced among the committee members. The outcome will articulate a range of relationships between emerging technology environments and The School, including its pedagogical and aesthetic missions.

The Technology Development project conducted in parallel recognizes the need to envision and provide for alternative to off-the-shelf technologies. To address this need we are adopting a working study system to conduct a series of technology experiments. The outcome will provide working demonstrations of component-based immersive technology and a method for extending the system to include new components supporting new projects.

A working experimental system is vital to constructing a transition from discussions to a realizable research environment. Figure 2 describes the differences between a commercial software application and a software development environment. The differences are embodied in the intent of the interface. In commercial systems the interface provides a set of pre-defined tools, designed to meet the common needs of a group of potential customers identified by market research. The Research Environment provides for multiple interfaces to meet diverse needs a commercial package cannot deliver due to constraints of profit-oriented production. The economics of the component-based Research Environment are sustainable based on our mutual work to identify

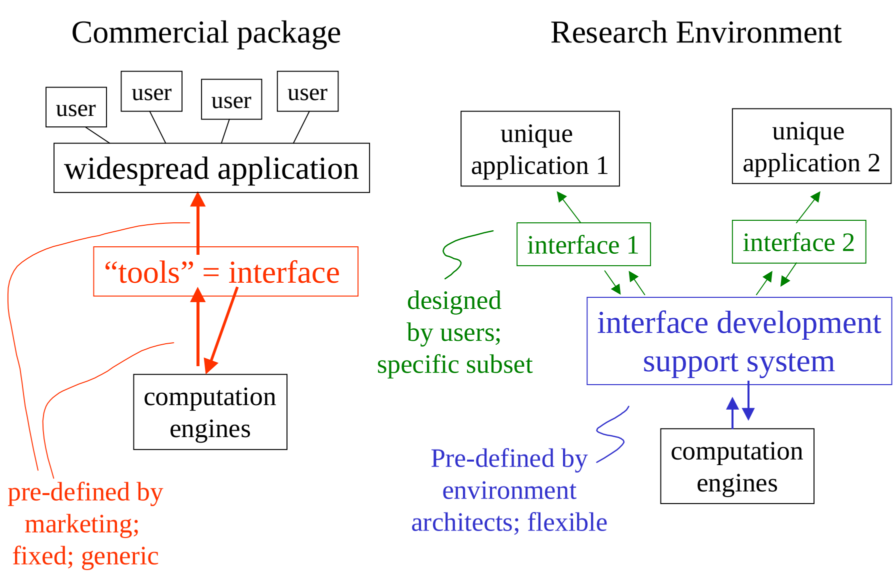

Figure 2: Comparison of off-the-shelf technologies and a technology development environment.

common needs across a range of diverse individual needs. Because the system is component-based, multiple needs can be met by reconfiguring a pool of atomic components shared by all projects, utilized in unique configurations. This practice mirrors the industrial economics of software, on a more flexible research scale. In terms of “risk”, the research environment can pursue projects which industries cannot afford because of the cost of commercial production. Effectively the research committee eliminates the role of marketing and the associated high overhead of market research and advertising, focusing the research team resources efficiently to meet agreed-upon needs. The Committee influences the design of components for the “interface development support system” in Figure 2, such that individual project interfaces can be built from the common pool of software elements in the support system.

4. Example Results from an Immersive Environment Research System

The study system adopted for research has been previously developed and applied to create immersive environment performances and installations. This software, known as *ScoreGraph* is described in detail in the previous Synopsis on immersive technologies (Choi: October 26, 2000). *ScoreGraph* supports many of the functions needed to test the principles of Social Organization as Computational Architecture and Distributed Cognition.

One may think of this support capability as an extended network of connected objects, each performing individual tasks while communicating with other objects to accomplish a larger goal. Some of the objects exist in software, others exist as mechanical objects connected to a computer. These networks bridge the computable domain of numerical models and the non-computable domain of human action and experience. Figures 3-5 show an example of a completed configuration, an installation presented at IRCAM (*Institut de Recherche et Coordination Acoustique/Musique*), Centre Georges Pompidou. In this installation a software environment is coupled to a real-world gallery space arranged with interactive devices and displays. The presence and actions of multiple observers are supported by the configuration of a specific network of hardware and software components. Some components render graphics, others generate sounds, and others determine the shape, color and movement of virtual objects in the scene. The visible hardware devices such as MIDI drum pads and video projector are represented in the same software environment as virtual components. By extension the system anticipates the participation of human observers according to the communications among physical and virtual devices and processes. By networking new devices and constructing new virtual objects other immersive projects can be configured.

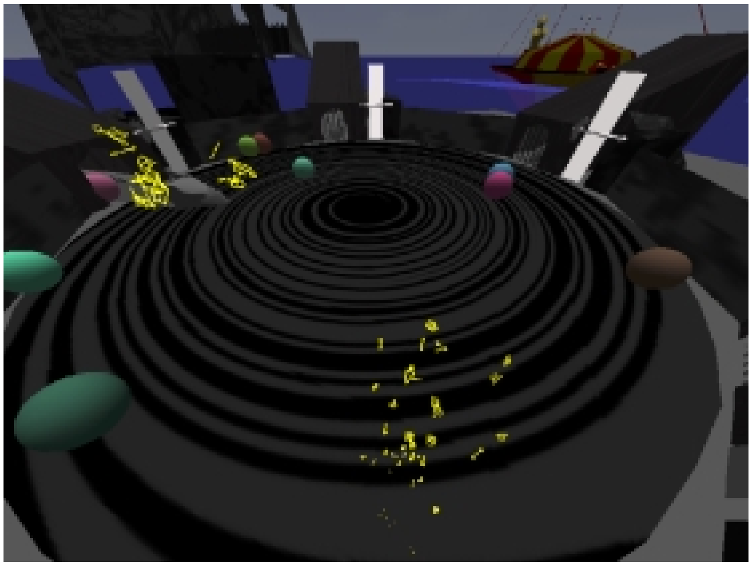

Figure 3: A virtual scene, fully numerical and computable. Spheres move on a spinning surface and vertical paddles rotate, creating yellow particle bursts and sounds when they collide.

Figure 4: Installation participants beyond the computation space. They participate actively by hitting MIDI drum pads to impart forces, which influence the movements of virtual objects.

Figure 5: Installation visitors become involved in the interaction process of the immersive system. While up to ten persons can use drum pads in parallel, everyone can observe the relation between the physical devices and the virtual objects. The immersive environment encompasses both computable and non-computable spaces in a coordinated network of communicating objects.

4.1 Conclusion

Currently there are few if any institutional facilities where art production and research are conducted together in equal standing. Under the umbrella of the two research areas suggested above – *Computability problems* and *Intelligent and adaptable environment design and testing* – there are many subcategories of research topics and art production that can be brought together, not to mention the potential funding resources available to these interdisciplinary approaches.

NB: This unpublished paper served as the basis for design planning for several new programs at the School of the Art Institute of Chicago.