Towards a General-purpose Infrastructure for Novel Collaborative Multimodal Human-Computer Interfaces

JoAnn Kuchera-Morin
UC Santa Barbara
jkm@create.ucsb.edu

Tobias Hollerer
UC Santa Barbara
holl@cs.ucsb.edu

Bo Bell
UC Santa Barbara
bobell@mat.ucsb.edu

Lance Putnam
UC Santa Barbara
ljputnam@umail.ucsb.edu

Stephen Mangiat
UC Santa Barbara
smangiat@umail.ucsb.edu

Charlie Roberts
UC Santa Barbara
Charlie@charlie-roberts.com

ABSTRACT

We describe our investigations towards creating a multi-user audiovisual painting canvas and its role in developing a more general-purpose infrastructure for multi-user orchestration of complex spatiotemporal patterns. The infrastructure consists of a device server for orchestrating multiple input controllers that send messages to a virtual interaction space.

1. INTRODUCTION

If there is one concern shared by artists and scientists alike it is with the mathematical processes underlying the formation of spatial patterns and how these patterns vary with time. The overarching goal of this research is to allow an ensemble of users to specify and orchestrate the unfolding of complex spatiotemporal patterns using "least"-encumbered methods of gesture sensing. This *synchronized* musical ensemble-style of human-computer interaction can allow more complex modes of real-time control through its highly parallel nature; something that would be impossible to achieve by a single user alone. In this spirit, we have developed two iterations of an audiovisual painting application that allow multiple users to record and play back simple two-dimensional gestures to control "sonic curves" that interact with each other within a shared virtual space.

Our experience while developing and using the painting programs led us to consider a more general infrastructure for future applications. This infrastructure adopts the model-view-controller paradigm commonly used in software engineering. In our case, we call the model-view component the Digital Information Interactor (DII) and the controller the Allo-I Device Server.

These two components provide a basis for the long-term infrastructure of the Allosphere [1, 2, 3], one of the largest artistic and scientific instruments in the world that we are developing for interactively generating and manipulating large-scale multi-dimensional data sets visually and sonically in real-time. In order to develop a general-purpose infrastructure that will be flexible enough to scale to different platform configurations and eventually to a system as large as the Allosphere, a C/C++-based Linux infrastructure is needed. Existing real-time multimedia applications, such as pd and Max/MSP/Jitter, are not easily extendable and robust enough to handle distributed, computationally intensive systems. We are developing a new infrastructure that will be general-purpose enough to manipulate complex scientific and artistic data in real-time with multiple users having multimodal control [4].

2. PREVIOUS WORK

There is a history of multimodal interaction systems constructed for the creation and manipulation of spatiotemporal patterns. One of these, Sketchpad [5], allowed users to create line-based forms using a light-pen, display, and various push buttons, switches, and knobs. The program pioneered the idea of composing figures using an object-oriented approach of classes and instances. Individual objects could also be assigned to a group to create more complex hierarchical patterns. Since Sketchpad stored topological and metrical information between various parts, it allowed animations and certain types of physical simulations to be run. In this way, the forms created were not just illustrations, but were directly influenced their function. A second interaction system, GROOVE [6], was developed at Bell Labs for composing temporal control functions. It could record and store output from a set of dials and a 3D wand and combine these through algebraic manipulation and sequencing to create complex output patterns. The system was not tied to any specific input device as it simply sampled a set of one-dimensional voltages. While both of these systems included multimodal input and integration, their output was generally oriented towards a specific modality, i.e. sound synthesis in the case of GROOVE.

3. COLLABORATIVE PAINTING

A multi-user audiovisual painting program was deemed an appropriate starting point for creating more complex applications involving ensemble-style multimodal interaction.

The basic requirements of the painting program are to 1) record gestures, 2) transform gestures, and 3) sense the spatial proximity of the strokes. The program receives control messages specifying the position of a user cursor and two mode-switching flags – record trajectory and transform trajectory. The record trajectory mode stores the cursor position coordinates sequentially into a buffer. The transform trajectory mode specifies that some type of parametric transformation that is mapped to the cursor's position should be applied to the record buffer. Two specific implementations of a multi-user audiovisual painting program have been developed.

The first, named Collision Painter, allows multiple users to draw, in a free-hand fashion, two-dimensional trajectories of entities on a shared canvas. The canvas recognizes when two entities collide as they traverse their drawn path. When a collision occurs the entities provide feedback by flashing white and emitting a brief sound whose pitch and echo time depend on their position on the canvas.

In the second iteration, called Curve Synth, users draw shapes on the screen to control sound synthesis (figure 1). Features of these shapes such as curvature and bending energy determine their color as well as the audio they produce. When placed into the space the shapes also move and collide with each other, dynamically morphing the sounds and images. Also built into the Curve Synth is gesture based control, allowing users to change audio parameters using eight spatial gestures. This is desirable as traditional keyboard controls are impractical for multi-user collaborative spaces such as the Allosphere.

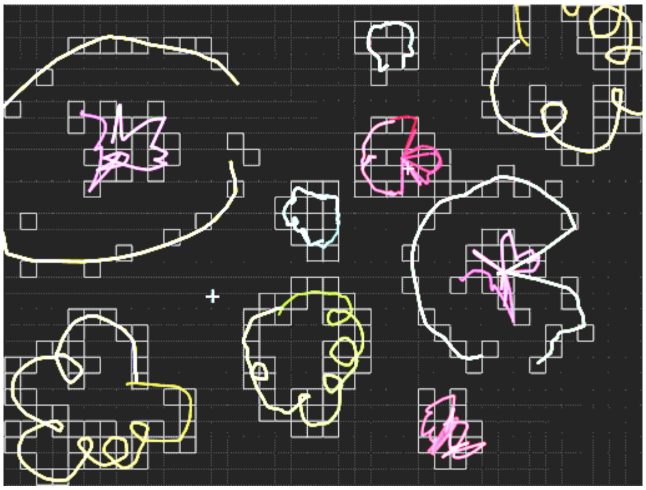

Figure 1. Screenshot of CurveSynth showing sketches from multiple users and the underlying collision grid.

By creating these two iterations of the painting program we began to formulate a more general-purpose infrastructure for creating future applications.

4. MULTI-USER MULTIMODAL INTERACTION SYSTEM INFRASTRUCTURE

As mentioned previously, our infrastructure consists of the Allo-I Device Server and the Digital Information Interactor as components of a model-view-controller paradigm.

4.1 ALLO-I DEVICE SERVER

At the base of the HCI infrastructure is an application known as the Allo-I (Allosphere Interactive) Device Server, a server application that provides connection of HCI devices to applications transparently. In general, all outbound control messages are mapped to user (application) space parameters from natural device-specific controls to help curb mapping complexity [7]. Figure 2 shows a schematic of the device server and how it interacts with client applications. The Allo-I Device Server ensures scalability by providing a single point of reference for all interaction in the Allosphere. A list of devices available to the server is published online for developers to view and each device is described in an XML file. Since the server and device descriptions are network accessible, any computer, PDA, or cellphone with a network connection can run applications that utilize these devices without having to install specialized (and potentially costly) software or drivers. This centralized location also enables the interaction in the Allosphere to be adjusted by simply altering the XML configuration files or the server itself. Individual applications will typically not need to be modified when such changes are made.

The Allo-I Device Server communicates with applications using the Open Sound Control (OSC) protocol. Applications can choose to receive all the messages a device generates via OSC or limit themselves to receiving messages from the individual controls the device possesses. Signal data from these controls can be modified inside the Allo-I Device Server by Javascript expressions. These expressions provide filtering, massaging, and unit conversion on the server side and stop the client application from having to make these calculations every time a signal is received. In applications where many control signals are utilized this capability can offload hundreds of calculations away from the client processor(s) every second.

There have been similar applications written to manage interaction using OSC and XML, such as the Mapping Tools software developed at McGill University [8]. There are notable differences between Mapping Tools and the Allo-I Device Server. The current public release of Mapping Tools relies on Max/MSP while the Allo-I Device Server avoids reliance on third-party software with the goal of ensuring future development flexibility. Mapping Tools was designed with the specific aim of mapping device signals to sonic parameters, while the Allo-I Device Server makes no such assumption about the usage of devices. The Mapping Tools interface allows users to manually drag connections between devices and application properties but provides no simple way to monitor incoming and outgoing signals. In contrast, the Allo-I Device Server was designed under the assumption that developers will have carefully selected the interaction paradigms of their applications and that these paradigms will be relatively stable. In place of a drag-and-drop connection manager aiding experimentation, the Allo-I provides a flexible signal monitor that developers can use to debug interaction difficulties.

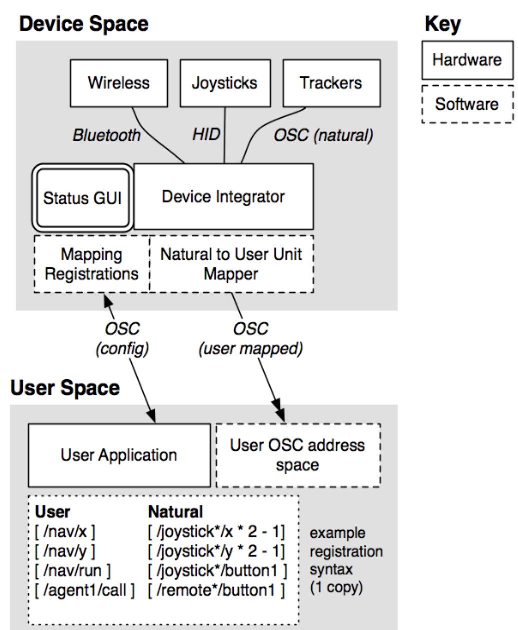

Figure 2. Allo-I Device Server Schematic

4.2 DIGITAL INFORMATION INTERACTOR

The Digital Information Interactor (DII, pronounced "dee") is a system that transcodes digital information consisting of data and/or algorithms for interaction within a unified mediating space [9]. The DII is based on a sparse multi-agent system interacting with a dense regular sample grid. In extreme cases, the “true” model of the system can be either the agents or a particular signal processing algorithm executing on the grid. In the first model, the agents are autonomous systems communicating through the sample grid. In the second model, the sample grid is running a simulation while the agents signify salient features. Realistically, the model will lie somewhere between these extremities.

The samples in the grid can be any type of object or data structure. This generic approach allows developers of a multi-entity interacting system to use custom entities in the space and program custom logic for interactions. For instance, the simplest space would contain a boolean value at each location signifying whether or not an object is currently occupying the respective cell. A more complex space might have a list of objects in each cell so that more sophisticated relationships can be programmed between entities in the system. We have developed a C++ class called *Spacetime* that can represent up to a 4-dimensional grid. The Spacetime class contains functions for checking if points are within the space and for converting between Cartesian coordinates and cell indices.

The processing algorithms we employ are contained within the custom-built Gamma generic processing library. Gamma builds upon the CREATE Signal Processing Library (CSL) [10] to allow use of generic types and single-sample evaluation. Gamma also allows processors to be assigned to different processing domains. For instance, we can use the same low-pass filtering object for smoothing sensor data, processing sound, or moving a camera in 3D virtual space.

An interesting outcome of the DII is that it naturally encourages human collaboration in a common computational space. When used on a large shared screen many users can modify the same data, encouraging the back-and-forth that is common when a group of scientists or artists discuss their work [6]. Furthermore, it also allows for real-time composition of music and art. The benefit of the DII-many-user paradigm is that users can immediately react to what others have drawn to produce a unique piece. In this sense, the DII facilitates "inter-human interaction" as well as HCI.

5. CONCLUSION

Curve Synth and Collision Painter are two applications our research group has developed that capitalize on our investigations into multi-user interaction spaces. There are common requirements inherent to such spaces that the DII and the Allo-I Device Server address by easing device connectivity, shared data access and signal processing in application development. The use of these systems in conjunction with wireless interaction devices will support the future creation of immersive multi-user environments inside the Allosphere.

6.REFERENCES

[1] Amatriain, X., Castellanos, J., Hollerer, T., Kuchera-Morin, J., Pope, S., Wakefield, G., Wolcott, W. (2007). Experiencing audio and music in a fully immersive environment. Lecture Notes in Computer Science, Springer-Verlag.

[2] Amatriain, X., Hollerer, T., Kuchera-Morin, J., and Pope, S. Immersive audio and music in the Allosphere. (2007). International Computer Music Conference. Copenhagen.

[3] Hollerer, T., Kuchera-Morin, J., and Amatriain, X. (2007). The Allosphere: A large-scale immersive surround-view instrument. In EDT ’07: Proceedings of the 2007 workshop on Emerging Displays Technologies, page 3, New York, NY, USA, 2007. ACM Press.

[4] Wakefield, G., Kuchera-Morin, J., Novak, M., Overholt, D., Putnam, L., Thompson, J., Smith, W. (2008). The AlloBrain: An interactive stereographic, 3D audio immersive environment. In CHI Conference Workshop on Sonic Interaction Design.

[5] Sutherland, I. (1963). Sketchpad: A man-machine graphical communication system. In Proceedings of the AFIPS Conference, v. 23.

[6] Mathews, M. (1970). GROOVE- a program to compose, store, and edit functions of time. Communications of the ACM, 13(12).

[7] Wright, M., Freed, A., Lee, A., Madden, T., Momeni, A. (2001). Managing complexity with explicit mapping of gestures to sound control with OSC. In Proceedings of the 2001 International Computer Music Conference, pp. 314-7.

[8] Malloch, J., Sinclair, S., Wanderley, M. M. (2007). A network-based framework for collaborative development and performance of digital musical instruments. In Proceedings of the 2007 Computer Music Modeling and Retrieval Conference, pp. 401–425.

[9] Turk, M., Hollerer, T., Muller-Arisona, S., Kuchera-Morin, J., Coffin, C., Hoetzlein, R., Putnam, L., Overholt, D. (2008). Creative collaborative exploration in multiple environments. In Association for the Advancement of Artificial Intelligence.

[10] Pope, S., Amatriain, X., Putnam, L., Castellanos, J.. Avery, R. (2006). Metamodels and design patterns in CSL4. In Proceedings of the 2006 International Computer Music Conference.