A MODEL FOR GRAPHICAL INTERACTION APPLIED TO GESTURAL CONTROL OF SOUND

Jean-Michel Couturier
Blue Yeti
France
jmc@blueyeti.fr

ABSTRACT

This paper is about the use of an interaction model to describe digital musical instruments which have a graphical interface. Interaction models have been developed into the Human-Computer Interaction (HCI) field to provide a framework for guiding designers and developers to design interactive systems. But digital musical instruments are very specific interactive systems: the user is totally in charge of the action and has to control simultaneously multiple continuous parameters.

First, this paper introduces the specificities of digital musical instruments and how graphical interfaces are used in some of these instruments. Then, an interaction model called Instrumental Interaction is presented; this model is then refined to take into account the musical context. Finally, some examples of digital musical instruments are introduced and described with this model.

1. INTRODUCTION

Graphical interaction is present into most of computers and also in computer-based gesture-audio systems. Nevertheless, creating a digital musical instrument that is controlled through a graphical interface is not obvious: graphical interaction adds complexity to the design, especially when this interaction is more sophisticated than the one of classical computer software. To describe graphical interaction, some models have been proposed into the Human-Computer Interaction (HCI) field; the aim of an interaction model is to provide a framework for guiding designers and developers to design interactive systems. Some of these models can be used with digital musical instruments, but they need to be refined because interactive sound systems have very particular characteristics regarding usual computer programs.

This paper introduces the theoretical work that I have carried out during my PhD thesis to propose an interaction model that is suited to digital musical instruments. Rather than creating a new interaction model, I use the *instrumental interaction* model [4] and I have refined it to fit to specificities of digital musical instruments. Then, this model was used to describe some digital musical instrument I have created.

2. GRAPHICAL INTERFACES AND INSTRUMENTAL CONTROL

Graphical interaction is an important area of Human Computer Interaction (HCI) field. Nevertheless, instrumental control, which characterizes musical instruments, is very specific regarding current applications studied in HCI. This section introduces the specificities of instrumental control of sound processes and shows how graphical interfaces are used in this context.

2.1. Specificity of instrumental control

Musical instruments, except voice, are tools (tangible objects) that human have to manipulate. Contrary to other tools like knives, hammer or drill, musical instruments require strong synchronisation of movements and the temporal development of actions is essential. These specificities are present in digital systems dedicated to instrumental control of sound processes, according to M. Wanderley [28]:

> “Gestural control of computer generated sound can be seen as a highly specialized branch of human computer interaction (HCI) involving the simultaneous control of multiple parameters, timing, rhythm, and user training. Therefore, input devices, interfaces and interaction modalities involved in computer music need to take into account these specificities.”

Hunt and Kirk [12] have confronted the musical context to interface paradigms that are usually accepted in HCI, like WIMP (Windows, Icons, Menus, Pointing) or direct manipulation. They assert that control interfaces for musical instruments are in stark contrast to the commonly accepted choice-based nature of many computer interfaces. The authors suggest some attributes of a real-time instrumental control system, like:

- There is no fixed ordering to the human-computer dialogue.
- The human takes control of the situation. The computer is reactive.
- There is no single permitted set of options (e.g. choices from a menu) but rather a series of continuous controls.
- There is an instant response to the user's movements.
- The overall control of the system (under the direction of the human operator) is the main goal, rather than the ordered transfer of information.

- The control mechanism is a physical and multi-parametric device which must be learnt by the user until the actions become automatic.

M. Wanderley and N. Orio [29] specify that an important feature of digital musical instruments is that the goal of the interaction (the performance) is part of the communication between the performer and the instrument. The performance relevance is situated into the action development and does not follow the usual actions → result scheme, like classical computer interfaces, which enable to obtain a result through series of actions realized in a sequential manner.

Concerning input devices of a digital system, the most usual is to have standard devices (e.g. keyboard – mouse) that will be used with every software. These devices generally meet the needs for the manipulation of WIMP interface and standard computer tasks, even if they only exploit a small part of our gesture possibilities; nevertheless, for musical instruments, other input devices are necessary to sense gestures in a more complete and precise manner, then allowing a better expressivity.

According to Dragicevic [9], numerous works show that input devices have to be suited to the task, the user and the environment. In our case, the task is the real time control of sound processes, the user is a musician, novice or expert, the environment can be a stage, an exhibition or a studio. The standard input devices, by their generic nature (used for all), suit here neither to the task, nor the user and nor the environment: more adequate input devices are required in the musical context.

In the HCI domain, other interaction styles have been developed and are more suited to the musical context: the post-WIMP interfaces, like bimanual interaction, mixed reality or virtual reality. Jacob, Deligiannidis and Morrison [16] give their definition of a post-WIMP user interface:

> "The essence of these interfaces is, then, a set of continuous relationships some of which are permanent and some of which are engaged and disengaged from time to time. These relationships accept continuous input from the user and typically produce continuous responses or inputs to the system. The actions that engage or disengage them are typically discrete (pressing a mouse button over a widget, grasping an object)."

This definition can include the digital musical instruments, which need continuous control of numerous parameters.

Another specificity of instrumental control is the necessity of learning an instrument to be able to exploit all its expression potentialities. When one designs a classical software, the rules to follow are generally: a fast learning phase, simple and comprehensive interfaces,… Some tests on musical interfaces have shown that in some cases a complex mapping between sound and gesture and a compulsory learning time is more efficient than a simple mapping and a fast learning phase [13].

2.2. How to use graphical interfaces in instrumental control?

In the current musical software dedicated to real time sound control, graphical interfaces tend to reproduce onscreen an interactive environment that looks like a real environment, like the control panel of some electronic instruments. The task of the user consists in controlling sound parameters through the manipulation of graphical objects; the most usual graphical objects represent real objects like piano keyboards, linear or rotary sliders, buttons, … The goal of these interfaces is to give the user the impression he has real objects in front of him.

Figure 1. Some windows of Digital Performer: numerous graphical objects are displayed, often like virtual control panels. These objects have been designed to be controlled trough the mouse or thanks to external controllers.

If graphical interfaces are a good way to control virtual copies of real objects, they can also represent interaction situation that are unrealizable in the real world, like navigation in trees, control of dynamic objects, … In addition to these graphical objects, the interfaces can contain pure information elements, like display of text or icons. The graphical interface is not limited by physical constraints, so that the designer of such interface has more freedom to find the graphical representations that best fit with the sound processes to control.

Nevertheless, the classical graphical interfaces, which are manipulated with a standard pointing device, are not suited to musical performance because all the user's actions get trough the manipulation of a unique pointer. This pointer uses only one hand when most of musical instruments use the two hands, often all the fingers and sometime the feet and the breath. For that reason, one cannot manipulate the graphical elements of the interfaces in the same way than acoustic instruments or the control panels of electronic instruments, in which all the elements can be used simultaneously. To overcome this difficulty, several solutions are possible and can be used in a complementary way.

2.2.1. *Use external input devices*

This solution is the more widespread: it fits with particular uses (like mixing), but the interaction looses in flexibility. One has also to remember which external element is mapped to which graphical element. An example is the use a control surface to control an audio sequencer (figure 2). The user finds back the comfort and the ergonomics of the analog mixing consoles. These input devices are often used by sound engineers in studio, but are also used in live performance.

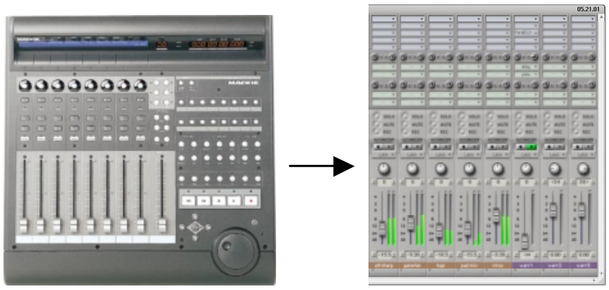

Figure 2. On the left: a control surface; on the right, the software interface which is associated with. The sliders of the external device controls the sliders onscreen and the corresponding sound processes.

2.2.2. *Use advanced interaction styles (post-WIMP)*

As shows the literature, the power of graphical interaction can be improved; interactive sound systems have been developed, using bimanual interfaces, mixing real objects and graphical objects (tangibles interfaces: audiopad [22], ReacTable [18]) or using 3D interfaces (VMI [21], Augmented Groove [23]).

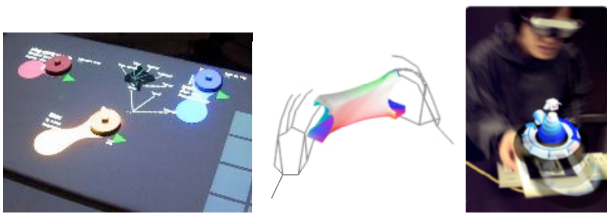

Figure 3. Audiopad, VMI and Augmented Groove systems.

2.2.3. *Use automatic controls or dynamical objects*

The automatic controls define, with a law, the temporal evolution of one of several parameters; this law can be adjusted or modified by the user. The virtual dynamical objects push this principle further: the user controls the sound by interacting with geometrical shapes that evolve according to predefined laws. These shapes can be animated or created by the user and continue to evolve when the user stops to interact with them. Virtual dynamic objects were used in several systems, like AVES (AudioVisual Environment Suite) [20] and FMOL [17] (Figure 4).

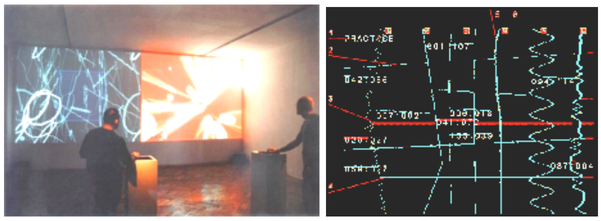

Figure 4. AVES (Golan Levin) and FMOL (Sergi Jordà). In AVES, the performer draws geometrical shapes which start to move movement; in FMOL, the performer interacts with virtual strings displayed onscreen.

The use of virtual dynamic objects is a good example to illustrate the power of graphical interaction into sound control.

3. INTERACTION MODELS

To describe a digital musical instrument, one generally gives its gestural devices, its sound processes and the mapping between gesture and sound; a mapping model can be used, like [1] or [30], to improve the conception of the instrument. When the instrument is controlled through a graphical interface, things become more difficult because graphical interaction adds complexity to the design. Using an interaction model can help the designers to create digital musical instruments with graphical interfaces.

According to M. Beaudouin-Lafon [5], the aim of an interaction model is to provide a framework for guiding designers and developers to design interactive systems. An interaction model can be used from the early stages of the design, unlike ergonomic rules, which are often limited to post-hoc evaluation of a design.

As digital musical instruments are very specific interactive systems, a question occurs: do existing interaction models well suit to such systems? This section introduces different interaction models and describes one of them, the instrumental interaction, that will be used in the next section to describe model sound systems with graphical interfaces.

3.1. Different models

Different kind of interaction models have been developed, from high-level design guidelines, such as direct manipulation [25], to detailed rules such as those described in style guides like those of Apple or Microsoft. To describe Post-WIMP interfaces, different models have been proposed. M. Beaudouin-Lafon has introduced the instrumental interaction model, which is well suited to describe interactive systems with graphical interfaces [4].

Tangible interfaces were introduced by Ishii and Ullmer [14][26]. These authors have proposed an interaction model to describe the tangible interfaces, called MCRpd for Model-Control-Representation (physical and digital). Keiichi Sato et al. [24] have proposed 8 interaction models to distinguish the different interaction situations between numerical data and the real world. These models are more a taxonomy of the different interaction situation than operational models. Concerning mixed reality systems (which include tangible interfaces) ASUR and ASUR++ [10] [11] and IRVO (Interacting with Real and Virtual Objects) [6] have been developed. ASUR++ takes mobility into account and IRVO can describe multi-user systems.

In this paper, the objective is to have an interaction model that helps to design digital musical instruments with graphical interfaces. I choose to use the instrumental interaction model because it is well suited to interactive systems with graphical interfaces and simpler than the others mixed reality models.

3.2. Instrumental interaction

This subsection presents instrumental interaction, which has been introduced by M. Beaudouin-Lafon [4]. Instrumental interaction is a model of interaction that enables to describe current interfaces and also to describe a wide range of new interaction techniques (post-WIMP) such as bimanual interaction or augmented reality. This model is an extension and a generalization of direct manipulation model by Ben Schneiderman [25]. It sets a layout for the design of graphical interfaces.

The paradigm of instrumental interaction comes from our interactive experience of the physical world, ruled by the use of tools. Tools such as paintbrush or drill, or even switch, are intermediate objects (instruments) we use to have an effect on other objects (domain objects). In this section, we present the model of instrumental interaction such as described in [4].

The model is based on the notion of interaction instruments playing the role of mediator between the actions of the user and the items-objects of the application. In order to present this model, we first define the objects manipulated by the instruments (domain objects), then the instruments themselves and at last the principle of reification that enables to create new objects and new instruments.

3.2.1. Domain objects

Domain objects are, within the considering application, the domain objects of application. In a text editor domain objects correspond to letters, words, sentences, paragraphs; in a math editor, it corresponds to cells, sheets; in a drawing software, they are the graphical objects composing an image. Each object is composed of a set of attributes, such as, for a text editor, font or the size of letters. Finally, each object can have one or several representations: for example, a text editor offers the choice between outline mode or page mode, a CAD software can offer wire representation or solid representation, etc.

3.2.2. *Instruments*

Every action on a domain object (creation, modification or destruction) is made through an instrument. A simple example of an instrument is a scrollbar manipulated through a mouse: the action of the mouse on the bar controls the position attribute of the object “document” linked to the bar.

An instrument splits up the interaction in two parts (figure 5): on the one hand, the interaction between the user and the instrument, defined by the physical action of the user on the instruments and the reaction of the instrument, and on the other hand the interaction between the instrument and the domain object, defined as the order sent to the object and the response of the object, which can be converted in feedback by the instrument for the user.

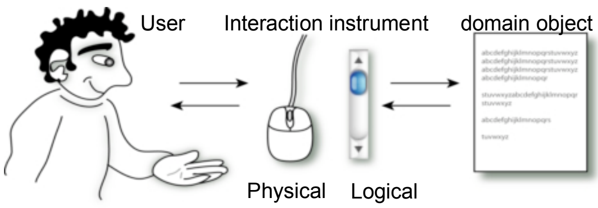

Figure 5. Interaction instrument.

An instrument is composed of a physical part, the input device, and of a logical part, the representation of the instrument in the application and onscreen. At a given time, the graphical interface displays a great number of potential logical instruments, but the user cannot handle them all at the same time, because of the limited number of input devices. In the most frequent case, when the interface is manipulated through a mouse, this unique input device has to be multiplexed between a great number of potential instruments: a unique physical part can be associated to different logical parts.

The association between logical instrument and physical instrument composes the activation of an instrument. This activation may be spatial or temporal: the activation of a scrollbar is spatial, because it requires the displacement of the pointer (and then of the mouse) within the area where is positioned the scrollbar; The activation of the creation of a rectangle is temporal because it is made before the creation of the rectangle (choice in a panel of tools, for example) and remains active until the activation of another instrument.

The activation phase makes the interaction less efficient since it imposes to the user an action of selection of the logical instrument. The use of additional input devices, dedicated to precise tasks, enables to reduce the number of activations of instruments and to use several instruments at the same time. For example, the mouse wheel, associated to the vertical scrollbar of the window, is an instrument permanently activated. An extreme example is an audio mixing console, which has commonly a hundred of sliders and switches, each one linked to a different function. From the unique mouse device to the hundreds of sliders, there are numerous possibilities to use at the very best input devices and to reduce the costs due to activation.

3.2.3. Meta-instruments

When an instrument acts on another instrument and not on a domain object, it is called meta-instrument. In the real world, it is usual to focalize attention temporally on the instrument itself: for example, a pen can be sharpened by a pencil-sharpener, which itself can be screwed back again by a screwdriver. In an interactive application, menus and toolbars are examples of meta-instruments which can be used to activate other instruments. Meta-instruments can also be useful to organize instruments in a workspace or to configure instruments for specific tasks.

3.2.4. Reification

Reification is the process which consists in changing a concept into an object. For example, the action of suppressing a file is reified in the trash of the operating system. The use of meta-instruments is also an example of the reification of instruments: in this case, instruments are the objects of interest of the meta-instrument. The reification helps to multiply the number and the nature of objects which can be handled in an application. Reification leads the way for new possibilities of interaction, while the interface keeps a uniform and coherent shape, based on the instrumental manipulation.

3.2.5. Instruments properties

The degree of indirection, the degree of integration and the degree of compatibility are three of the essential properties of instruments. These properties enable to compare several instruments which perform similar tasks.

The degree of indirection corresponds to the spatial and temporal interval between an instrument and the object it acts on. Some instruments, like selection handles, are situated onto the domain object. Other instruments, like dialog box, can be situated far from the object they act on. The validation system of the dialog boxes also adds spatial delay and temporal delay.

The degree of integration is the ratio between the number of physical dimensions and the number of logical dimensions that are manipulated during the use of the instrument. Concerning a scrollbar (1D) manipulated by a mouse (2D), the degree is ½. This ratio may be more than 1: for a 3D object manipulated by a mouse, the ratio is 3/2. The term degree of integration comes from the notion of integral tasks [15].

The degree of compatibility corresponds to the similarity between the manipulations performed on the instrument and their effects on the domain object. For example, the degree of compatibility of a scrollbar is lower than drag and drop degree of compatibility, since the direction of the document scrolling is reversed compared to the manipulation.

4. INSTRUMENTAL INTERACTION APPLIED TO DIGITAL MUSICAL INSTRUMENTS

In this section, I refine the instrumental interaction model to make it fits with the musical context. Use this model complementarily with a mapping model helps to design sophisticated digital musical instruments with graphical interface.

4.1. Domain objects

The aim of a digital musical instrument is to generate sound in real time, so domain objects will be linked to the produced sound. At a very high level of abstraction, one can consider that the sound produced is the "primary" domain object of the system; the interaction instrument is then the digital musical instrument itself. Nevertheless, the sound is produced by sound processes and the user acts on the sound by the manipulation of the control parameters of these processes. At a lower level of abstraction, control parameters can be considered as domain objects; this paper stays at this level of abstraction, where domain objects are the control parameters of the sound processes.

Contrary to most classical applications, in digital musical instruments the result of the actions on domain objects is mainly into the sound modality. These objects are not "graphical" and they generally do not have a visual representation in the interface, contrary to many objects of other applications.

4.2. Instrumental interaction: controllers and tools

Interaction instruments are mediators between the actions of the user and the domain objects. The interaction instruments that are specific to sound processes control can be classified in two types:

Controller type: the controller type defines interaction instruments that directly act on control parameters of sound. The term gestural controller is commonly used in Computer Music to define a gestural device dedicated to the control of sound parameters. In many music design software, the control of parameters is often done through graphical elements looking like gestural controllers: sliders, switch, etc.

Tool type: the tool type defines interaction instruments that directly act on the domain objects that are displayed in the interface and those which enable to create objects. Even if this type of interaction instrument is common in the interfaces of classical software, it is less common in systems dedicated to the control of sound processes.

Controllers and tools correspond to different actions. The tool suits to the description of usual HCI tasks, where different tools are used to act on a domain object (like a vectorial drawing edited onscreen) in order to reach a final result (drawing finished and ready for printing). The term "controller" rather distinguishes control of processes and implies notion of regulation; it enables to take into account specificities of the control of a musical instruments, which are: human operator is totally in charge of the action; result (aim of interaction) is situated in action and is not obtained at the end of a succession of tasks.

Compared to the types of domain objects seen before, controllers will rather act on control parameters of the sound, and tools will rather act on controllers (in this case, tools are meta-instruments); moreover, tools could be used to create controllers. Tools can also be parts of a controller; the user can choose a tool among a range of tools having their own specific action on sound, while all tools will act in the same area of the graphical interface. To avoid confusion between musical instrument and interaction instrument, the terms controllers and tools will be used instead of interaction instruments in the rest of the paper.

The interaction instruments are composed of a physical part and of a logical part. In this paper, the physical part of a controller will be called physical controller and the logical part will be called logical controller. Physical controllers correspond to input devices; logical controller is the representation of the controller within the software and onscreen. Logical controllers are represented by graphical elements onscreen, when the system has a graphical interface. In many interactive music systems, controllers do not have a graphical representation: physical controllers are directly linked to domain objects. Furthermore, tools have a physical part and a logical part. The logical part of a tool will be called logical tool. Yet, the physical part of a tool is generally a gestural controller, like the physical part of a controller; for that reason, the term "physical tool" will not be used.

Tools and controllers only represent interaction instruments that act more or less directly on sound. Like in other software, other interaction instruments can be used for other tasks, which are more common, such as tabs, scrollbars, menus or forms not related to sound; we will not deal with them in this paper.

4.3. Relationship between physical part and logical part

At a given moment, an interface offers a potentially high number of logical controllers and logical tools, which generally cannot be controlled simultaneously. To manipulate a controller or a tool, there is often an activation phase of connection between a physical part and a logical part: physical parts and logical parts are not necessarily connected and a physical controller can be associated to various controllers or tools.

The link between physical and logical parts can be either simple or complex, according to the choices made and the interaction styles. Next subsections describe the different relationships I have defined between physical and logical parts, which implies different programming strategies of the interfaces.

4.3.1. Permanent activation

The link physical/logical is the simplest when both parts are permanently coupled. It is often the case with digital music instruments: in order to avoid waste of time due to activation phase, additional physical controllers are used and linked directly to logical controllers or tools, like in the *Figure 2* of the *2.2.1* section, where the physical sliders of the control surface are permanently linked to the virtual sliders onscreen.

4.3.2. Pointing activation

This type of explicit activation is the most frequent within graphical interfaces; considering control of sound processes, the most typical case is the control of a virtual command panel with a mouse. Generally, the physical controller is represented within the graphical interface by a pointer which moves along two dimensions (x,y). Interaction consists in manipulating the physical controller to put the pointer onto one of the logical controllers or tools of the interface; activation between logical part and physical part is made trough a selection gesture (like button click). Both parts are deactivated by a selection gesture (button released, for example). Using pointer displayed onscreen is not necessary if the physical controller directly acts on the display area of the interface (touch screen, augmented reality).

Using several pointers in these interfaces can extend this type of interaction, which increases simultaneous handle possibilities and enables a collaborative use of the different pointers. Furthermore, pointing activation can be used with every types of physical controllers that send two coordinates (x,y) and button value. Sometime, physical controllers have more than two degrees of freedom: For example, some graphic tablets send the pressure of the tip and the tilt of the pen, in addition to the position of the pen. This data can be considered by computer to enable the creation of more evolved controllers or tools; this data offers a better integration degree between physical and logical parts.

4.3.3. Complex activation

In the real world, controller or tool activation can be considered equivalent to grasping a tool that is going to be used; when a real tool is grasped, the gesture performed is generally more complex than the pointing/selection task of an element within a standard graphical interface. Between computer pointing and real objects grasping, a range of more or less complex possibilities exists, such as gripping an object by two points or a total redesign of a grasping gesture in a virtual reality system, in which hand gestures are digitalized and used to grasp virtual objects.

This type of physical/logical activation is more complex and generally does not use pointers, but a set of data corresponding to the description of gestures (data from virtual reality glove, etc.) or data coming from a complex interface (camera, pressure field sensors, etc.). A complex analysis of this data, suited to the selected type of interaction, is generally necessary to enable the activation of the controllers. These "complex activations" enable to extend interaction possibilities, taking into account gestures in a more accurate way.

4.3.4. Separate activation

Sometimes the activation and the use of a controller are not executed with the same physical input device: for example, if one physical slider is used to control several logical sliders, the choice of the logical slider can be done by a keyboard shortcut.

4.3.5. Activations of several types in a same interface

It is possible to build hybrid interfaces that group together activations of several types. The most widely used example in HCI is the use of a mouse wheel: the mouse enables to manipulate elements of the interface with a pointer while the mouse wheel is linked to the scrollbar of the active window.

4.4. Reification of actions on sound

Reification is the changing of concepts into objects. For example, the action of increasing or decreasing loudness can be reified into a volume controller.

Generally speaking, a controller or a tool is the reification of one or several commands; in interactive music systems, these commands act on sound control parameters. The main difficulty is that these parameters do not have, a priori, graphical representations; visual metaphors, suited to the parameters that have to be controlled, have to be found. The reification of control parameters is an essential stage of graphical interface design; it makes the interaction efficient and comprehensive. Moreover, to drive a given set of parameters, many different kinds of graphical interfaces can be designed.

4.5. Controllers and tools properties

The three main controllers and tools properties are indirection degree, integration degree and compatibility degree. These properties can be used to estimate the efficiency of a system dedicated to the control of sound processes.

4.5.1. Indirection degree

Indirection degree corresponds to the spatial delay and the temporal delay between a controller or a tool and the object it acts on. Temporal delay must generally be avoided because the control of parameters has to be performed in real time. Spatial delay only has a meaning if the domain object is visible on screen; in that case, a tool directly acting on an object will provide a better indirection degree than a controller acting on this object and situated at another location onscreen.

4.5.2. Integration degree

Integration degree is the ratio between the number of physical dimensions and the number of logical dimensions that are manipulated during the use of the controller or the tool. It is not always necessary to try to reach the highest integration degree, which can make interaction less efficient: in the case of a controller, it all depends on the number of parameters that have to be manipulated at the same time, on the integral or separable quality [Jacob 1999] of the parameters associated to the controller and on the user ability to manipulate all degrees of freedom of the physical controller.

4.5.3. Compatibility degree

Compatibility degree corresponds to the similarity between manipulations carried out on the controller or tool and their effects onto the domain object. Using logical controllers that suit to the task will increase the value of the degree, as well as following some established rules: for example, when a vertical slider is associated to loudness control, the user is used to increase loudness by moving slider upwards and to decrease loudness by moving slider downwards.

It is interesting to evaluate another compatibility degree, particularly in the case of permanent coupling between logical and physical part: the compatibility degree between a logical controller and a physical controller. If logical controller has a behaviour really different from the gestures that are performed on the physical controller, the user will not have the impression of manipulating the graphical interface: the logical controller will seem to the user a simple visual feedback of his actions. A good compatibility degree will strengthen the impression of really acting through the graphical interface.

5. EXEMPLES

This section introduces examples of digital musical instruments we have developed at the Laboratoire de Mécanique et d'Acoustique (CNRS-LMA); these examples are described using the instrumental interaction model.

5.1. Scanned synthesis

The Scanned Synthesis was developed by Verplank, Shaw and Mathews [27] and uses the shape of slow-moving dynamical systems to create wavetables. We have used this technique with a string (spring-masses model) as dynamical system and designed a graphical interface with different logical controllers that can be activated by pointing.

One of these logical controllers is the shape of the string. The interaction with the string is the following: to activate the logical controller, a finger touches the screen up or down the string; according to its initial position, the finger can push the string up or down. Another logical controller is dedicated to pitch control and is localized on bottom right; it uses the angular frequency control developed by Kessous [19]. The last logical controllers are two sliders that modify the string damping and stiffness and 4 buttons that enable to stop the sound and to choose to play a single note or chords. The pointing action is realised by a custom-made physical controller, call Pointing Fingers [7], which works like a multi-point touch screen. The user has to touch the screen at the location of one logical controller to activate it, and to release the finger to stop the activation.

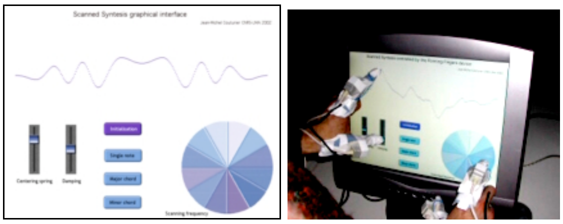

Figure 6. Graphical interface for the control of Scanned Synthesis and manipulation of this interface with the Pointing Fingers. This device enables multiple pointing: one can simultaneously interact with the string and modify the others parameters.

Using such a device makes the interaction transparent and increases the compatibility degree of the whole system, because there are no spatial separation between the fingers and the pointers, like it is with mouse interaction. The multi-pointing feature responds to the specificities of an interactive sound system and allows simultaneous continuous controls.

5.2. Filtering string

In the Filtering String instrument, gestures act on a virtual slow moving string (like scanned synthesis), which is used to filter a noise [2]. The string shape drives the gains of 32 filters and is displayed on a screen; the string model acts both on sound and graphics.

The user acts on the sound fluctuation indirectly, by interacting with the dynamic string using a graphical tablet and a multi-touch surface. Position and pressure of the stylet onto the graphical tablet modify the sound evolution by changing the intrinsic properties of the string (tension, stiffness, damping). Fingers' pressure on the multi-touch surface allows to apply forces onto the virtual string; this changes the string equilibrium position and then modifies the frequency spectrum of the sound.

The graphical interface of the filtering string instrument is very simple and consists of displaying the shape of the string and a logical controller that represents the value of pressure measured at the tip of the pen. The tip of the pen permanently activates this logical controller, which is linked to the pentip pressure; this logical controller is used as visual indicator to help the musician to quantify the pentip pressure.

Even if there are no pointer that manipulate the string, like in the Scanned Synthesis example, the string representation can be considered as a logical controller. Compatibility degree between string shape and gestures is high: first, vertical dimension of graphic tablet corresponds to vertical dimension of the string displayed onscreen; moreover, pressing the right half side of the touch surface will cause a shift of the string rightwards, and vice versa. This high compatibility degree between touch surface and image of the string strengthens the impression that it is the shape of the string displayed on screen that is manipulated: even if gestures on the touch surface have indirect effects on the image of the string, this image can be considered as a logical controller.

Nevertheless, compatibility degree between the shape of the string and gestures performed on the graphic tablet is very low: there is no clear spatial conformity between moves of the pen and their effects on the image of the string. Moreover, the use of the graphic tablet appears the most difficult point to people using the instrument for the first time: it is better to know and to understand how the string reacts to the variations of parameters on the tablet in order to play the instrument fluently. A more sophisticated mapping between graphic tablet and the own parameters of the string could improve manipulation of the instrument, by strengthening the similarity between gesture metaphor and its consequences on the instrument.

The problem seems to be partly due to the integral and separable characteristics of these parameters: the intrinsic parameters of the string, which are stiffness, tension and damping of the string, are not integral; this explains why these parameters are manipulated by separate controllers in instruments using Scanned Synthesis. Concerning the Filtering String, for practical reasons, these parameters are manipulated by three integral degrees of freedom (vertical and horizontal position of the tablet and pressure): this enables to an expert user to manipulate them simultaneously. Yet, concerning the touch surface, the vertical move task and the pressure task are integral, as well as the different elements of the force profile.

5.3. Wave terrain synthesis

This instrument is an implementation of the wave-terrain synthesis, where a pointer follows a trajectory on a terrain and the variation of the elevation of this pointer gives the waveform of the sound [3]. In this instrument, synthesis parameters are driven with a joystick and an interactive pen display.

The graphical interface is displayed on the interactive pen display and is divided in two parts. The largest part of the interface is used to represent the wave terrain in two dimensions. The other part is composed of logical controllers that enable to choose the terrain and the type of trajectory. There is also a visual indicator, represented as a digital value, displaying the speed of the reading point on the trajectory (this speed is linked to the pitch of the sound).

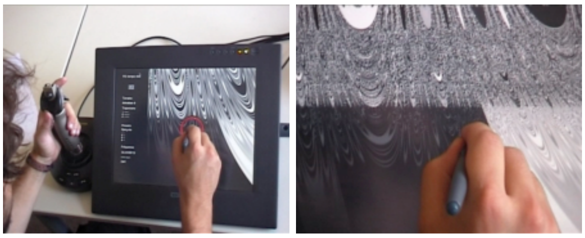

Figure 7. Graphical interface and gestural controllers used to control the wave-terrain synthesis. The user moves a trajectory or the reading point directly onscreen thanks to an interactive pen display. The other hand uses a joystick to modify the pitch, the parameters of the trajectory, the type of trajectory and the terrain. The pen tip pressure controls the loudness.

The part displaying the terrain can be considered as a logical controller that uses logical tools. Each tool corresponds to a type of trajectory: circular trajectories are represented by a red circle, epicycles by a green circle and scrub mode by a dot. Tools activation is done thanks to joystick switches, which corresponds to a separate activation: tool activation is done by another gestural controller than the one it is manipulated by. This solution enables to shift the tool without having to leave the terrain with the pen, which would have been the case with standard tools activated by a pointing task.

5.4. Granular instrument

The granular instrument consists of a granular synthesis controlled with a graphical tablet and a joystick. The synthesis technique uses a sound sample that can be loaded from the computer or recorded live; the algorithm extracts some "grains" of this sample. The musician can zoom and scroll into the sound sample; this enables to use a large sound sample and to choose precisely a grain in this sample. The graphical interface of the instrument has two logical controllers (*figure 8*) which display respectively the waveform of the entire sound sample (on top) and the zoomed part of this sample (on bottom). The first controller is used to choose a selection of the sound sample; it will be called the "zoom controller". The second controller is used to select the grain into the zoomed part of the sound; it will be called the "grain controller".

The physical part of the grain controller is the graphical tablet and the physical part of the zoom controller is the directional buttons of the joystick. The zoom controller is a metacontroller: it modifies the properties of the grain controller. The domain objects are the grain position and length for the grain controller and, for the zoom controller, the size of the position and the length of the zoomed part of the sound sample.

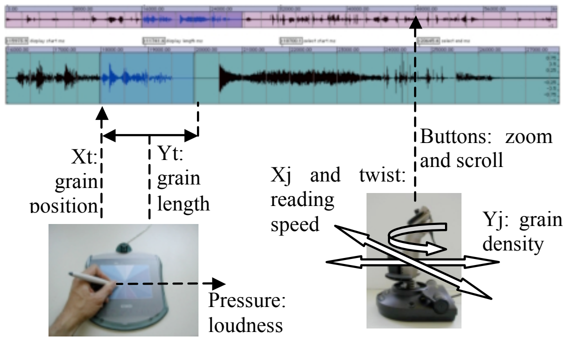

Figure 8. Graphical interface (top) and mapping of the instrument. The twist axis controls the reading frequency and Xj axis enables to add some random to this frequency.

The properties of the controllers of the instrument are represented into *Table 1*. The indirection degree of the grain controller is high because its physical part and logical part are permanently activated; there are no pointing tasks required. This degree is less high for the zoom controller because the physical controller is not continuous and successive multiple button presses are often necessary. The compatibility degree is high because the movement of the physical controllers correspond to the one of the logical controllers onscreen.

	Indirection	Integration	Compatibility
Grain controller	++	++	++
Zoom controller	+/-	+	+

Table 1. Properties of the controllers of the wave terrain instrument.

6. CONCLUSION

Using graphical interface in digital musical instruments can provide very powerful systems, but those systems cannot be designed using the classical rules used to create WIMP interfaces. With the instrumental interaction model refined to the musical context, one has a model that can really help to understand how the interaction works, evaluate this interaction (as shown into the examples *section 5*) and design new systems.

Since five years, several new interaction models have been proposed (see section 3.1), among others to extend the instrumental interaction model and take into account multiples users, multiples systems, mobility, … It will be interesting to look how these models could help to design all types of interactive sound systems, and not only digital musical instruments.

7. ACKNOWLEDGEMENTS

I would thanks the Computer Music team of the Laboratoire de Mécanique et d'Acoustique (CNRS-LMA) where I have done these works during my phD thesis [8].

8. REFERENCES

[1] D. Arfib, J-M. Couturier, L. Kessous, V. Verfaille, “Strategies of Mapping between gesture data and synthesis parameters using perceptual spaces”, Organised Sound, Vol 7/2, pp. 127-144, 2002.

[2] D. Arfib, J-M. Couturier, L. Kessous, “Expressiveness and digital musical instrument design”, in Journal of New Music Research, special issue on Expressive Gesture in Performing Arts and New Media,Vol. 34, No. 1, pp. 125 – 136, 2005.

[3] D. Arfib, J-M. Couturier, J-J. Filatriau, “Some Experiments in the Gestural Control of Synthesized Sonic Textures”, *Lecture Notes in Computer Science*, Volume 3881, Feb 2006, Pages 296 – 299.

[4] M. Beaudouin-Lafon, “Instrumental Interaction: an Interaction Model for Designing Post-WIMP User Interfaces”, *Proceedings ACM Human Factors in Computing Systems, CHI 2000*, La Haye (Pays-Bas), CHI Letters 2(1), pp 446-453, ACM Press, 2000.

[5] Beaudoin-Lafon M., *Designing Interaction, not Interfaces,* AVI’04, May 25-28, 2004, Gallipoli (LE), Italy, p.15-22.

[6] Chalon R., David B. T. IRVO: an Architectural Model for Collaborative Interaction in Mixed Reality Environments. In *Proc. of the Workshop MIXER'04,* Funchal, Madeira, January 13, 2004.

[7] J-M. Couturier, D. Arfib, “Pointing Fingers: Using Multiple Direct Interactions with Visual Objects to Perform Music”, *Proceedings of the NIME-03 conference*, Montreal, pp. 184-187, 2003.

[8] J-M. Couturier, “Utilisation avancée d’interfaces graphiques dans le contrôle gestuel de processus sonores”, phD thesis, Université de la Méditerranée, Marseille, 2004.

[9] P. Dragicevic, “Un modèle de configurations d'entrée pour des systèmes interactifs multi-dispositifs hautement configurables”, *phD thesis, Université de Nantes*, 2004.

[10] Dubois E. *Chirurgie Augmentée, un Cas de Réalité Augmentée; Conception et Réalisation Centrées sur l'Utilisateur*. phD thesis, Université Joseph Fourier - Grenoble 1, 2001.

[11] Dubois E., Gray P., Nigay L. Asur++: A Design Notation for Mobile Mixed Systems. In Paternò F. (Ed.) *Proceedings of Mobile HCI 2002*, LNCS 2411, 2002, Springer-Verlag, pp. 123-139.

[12] A. Hunt, R. Kirk, “Mapping Strategies for Musical Performance”, *Trends in Gestural Control of Music*, CD-rom, ed. Wanderley and Battier, Ircam, 2000.

[13] A. Hunt, M. Wanderley, M. Paradis, “The importance of parameter mapping in electronic instrument design”, *Journal of New Music Research*, SWETS, eds. J. Paradiso & S.O'Modhrain, Vol. 32 No. 4, pp 429-440, 2003.

[14] Ishii H., Ullmer B. Tangible Bits: Towards Seamless Interfaces between People, Bits and Atoms. In *Proceedings of CHI'97*, Atlanta, 1997, p. 234-241.

[15] R. Jacob, L. Sibert, D. McFarlane, M. P. Mullen. “Integrality and Separability of Input Devices”, *ACM Transactions in HCI*, 1(1), pp 3–26, 1994.

[16] R. Jacob, L. Deligiannidis, S. Morrison, “A Software Model and Specification Language for NonWIMP User Interfaces”, *ACM Transactions on Computer-Human Interaction*, Vol. 6(1) pp 1-46, March 1999.

[17] S. Jordà, “A graphical and net oriented approach to interactive sonic composition and real-time synthesis for low cost computer systems”, *Digital Audio Effects Workshop Proceedings*, 1998.

[18] M. Kaltenbrunner, G. Geiger, S. Jordà, “Dynamic Patches for Live Musical Performance”, *Proceedings of International Conference on New Interfaces for Musical Expression*, Japan, 2004.

[19] L. Kessous, “A two-handed controller with angular fundamental frequency control and sound color navigation”, *Proceedings of the NIME-02 Conference (NIME-02)*, Dublin, Ireland, 2002.

[20] G. Levin, “Painterly Interfaces for Audiovisual Performance”, *Master Thesis*, Massachusets Institute of Technology, 2000.

[21] A. Mulder, S. Fels, K. Mase, “Design of Virtual 3D Instruments for Musical Interaction”, *Proceedings of Graphics Interface '99*, pp. 76-83, Toronto, 1999.

[22] J. Patten, B. Retch, H. Ishii, “Audiopad: A Tag-based Interface for Musical Performance”, *Proceedings of the NIME-02 Conference*, Dublin, Ireland, 2002.

[23] I. Poupyrev, “Augmented Groove: Collaborative Jamming in Augmented Reality”, *ACM SIGGRAPH Conference Abstracts and Applications*, p 77, 2000.

[24] Sato K., Lim Y.-K. Physical Interaction and Multi-Aspect Representation for Information Intensive Environments. In *Proceedings of IEEE International Workshop on Robot and Human Interactive Communication*, 2000, Osaka, Japan, pp. 436-443.

[25] B. Shneiderman, “Direct Manipulation: a Step Beyond Programming Languages”, *IEEE Computer*, 16(8), pp 57-69, août 1983.

[26] Ullmer B., Ishii H. Emerging Frameworks for tangible user interfaces. In *IBM Systems Journal*, vol. 39, n° 3&4, 2000, pp. 915-931.

[27] B. Verplank, M. Mathews, R. Shaw, “Scanned Synthesis”, *Proceedings of the 2000 International Computer Music Conference*, pp. 368-371, 2000.

[28] M. Wanderley, “Performer-Instrument Interaction: Applications to Gestural Control of Music”, *PhD Thesis*, Paris, 2001.

[29] M. Wanderley, N. Orio. “Evaluation of Input Devices for Musical Expression: Borrowing Tools from HCI”. *Computer Music Journal*, 26(3), pp. 62-76, 2002.

[30] Marcelo M. Wanderley, guest editor. 2002. Mapping Strategies in Real-time Computer Music. Organised Sound vol. 7, number 2, 2002.