

‘Isadora: an interview with artist/ programmer Mark Coniglio’

in *International Journal of Performance and Digital Media*. 1:1. 2005, pp. 31-46.

In July 2002, I initiated an email dialogue with close colleague and interactive media and performance artist Mark Coniglio about the unique software tool he had created. I had proposed this as an article for a book Johannes Birringer was planning to edit on interactivity to be published in Germany that September. That plan did not materialise, and I distributed the finished dialogue on-line in November 2002 to the Dance-Tech discussion list.¹

In early 2003, I was invited to join the Editorial Board of the *International Journal of Performance and Digital Media* and proposed this dialogue for the first issue. After peer review and final editing was complete it was published in the first issue of the journal.

The International Journal of Performance Arts and Digital Media is a forum to energise, innovative and inspire creative thinking and practice surrounding the combination of digital technologies with the performance arts (theatre, dance, music, live art).

Disciplines may be domain-specific or in convergence. Source:

<http://www.intellectbooks.co.uk/journals/> (accessed 7 May 2010).

¹ See: <http://www.scottsutherland.com/dancetechnology/archive/2002/0368.html> (accessed 7 May 2010). Also available in this pre-published version on the ‘Software for Dancers’ documentation site: <http://huizen.dds.nl/~sdela/sfd/isadora.html>. (accessed 7 May 2010).

Isadora 'almost out of beta': tracing the development of a new software tool for performing artists

Scott deLahunta *Dartington College of Arts*

Abstract

Mark Coniglio is an artist who has recently programmed Isadora, a real-time software tool to support the creation of interactive live performance work for installation or stage. Programmed primarily to manipulate digital video, the Isadora software interface is designed to be easier to use than other similar real-time software tools. In dialogue with Scott deLahunta, Coniglio discusses his motivation and intention in developing Isadora, explores connections between his artistic and programming practice, talks about the use of the software in his own work with the performance company Troika Ranch and speculates on his role as author of the software in relation to those who use it to create their own work. This is followed by comments from some artists who have been using the Isadora software. The main aim of this dialogue is to provide an insight into the creation of software tools for artists to those who may be relatively new to this practice.

Keywords

dance
interactivity
real-time software
artist-programmer

Introduction

Real-time computer software tools designed to support the creation of interactive live performance work (either for installation or stage) and be useable by non-computer scientists have a history of being written by artists who were also programmers or were writing programmes in the context of arts organizations such as IRCAM (Institut de Recherche et Coordination Acoustique/Musique in Paris or STEIM (Studio for Electro-Instrumental Music) in Amsterdam. A few of these artist-programmers include David Rokeby who began to design his interactive Very Nervous System in the early 1980s, Miller Puckette who programmed Max in the late 1980s, Tom Demeyer who during his period at STEIM in the 1990s wrote the BigEye and Imag/ine software programmes, and the enigmatic Netochka Nezvanova who is credited with having created NATO.o+55 (2000).¹ The emergence of the Internet in the form of the World Wide Web in the early 1990s increased the dissemination of these software tools and now artists are customizing them to some degree, creating tools within tools, and sharing these developments with others. This short history, and in particular the list of artist-programmers, is by no means comprehensive, but is intended as a partial frame for the following dialogue with Mark Coniglio, an artist-programmer

1. Brief descriptions of some of the software referred to in the Introduction:

The Very Nervous System by David Rokeby is an interactive sound environment for which he has written the software SoftVNS which is a real-time video-processing and tracking software for use in the Max software environment.

Max is a screen-based patching language that could imitate the modalities of a patchable analogue synthesizer. Much of the early development of Max was aimed at

supporting music performance, but now is used in the support of real-time video manipulation.

BigEye is a computer program designed to take real-time video information and convert it into MIDI messages. *Imag/ine* is a program that allows a user to manipulate visual source material in a live performance environment in real time.

NATO.o+55 was built around comprised a set of QuickTime externals for the Max environment that could manipulate with any sort of QuickTime media (films, images, sound, QuickTime VR, QuickDraw 3D, Flash movies, etc.)

who has recently created the real-time software tool, *Isadora*. Some of above-mentioned software is referred to in the exchange below as *Isadora* has evolved from or alongside their development. The following exchange with Mark, conducted via e-mail in September 2002, is intended to provide an insight into the creation of real-time software tools to those who may be relatively new to this practice.

Biography/Background

Mark Coniglio is an artist who crosses the disciplines of music, dance, theatre and interactive media. Dubbed an 'interactive performance pioneer' by the *New York Times*, his work has been performed nationally and internationally primarily with Troika Ranch, a New York City-based performance company committed to creating multidisciplinary works of which he is co-director with choreographer Dawn Stoppiello. A native of Nebraska, Mark studied at the California Institute of the Arts (CalArts) with electronic music pioneer Morton Subotnick and received his degree in music composition in 1989. He was on the staff of the Center for Experiments in Art, Information and Technology at CalArts teaching courses in interactive music from 1990 to 1994. Troika Ranch was launched in 1993, while Stoppiello was still performing with Bella Lewitzky, and in 1994 they moved with the company to New York City. Besides the rehearsals, performances, symposia appearances and residencies that make up the bulk of their creative contribution to the field, Troika Ranch regularly conducts their popular Live Interactive (Live-I) workshops primarily in New York City. These give participants the opportunity to explore the use of interactive computer technology in the creation of live performance artworks. Participants also have a chance to learn to use the custom hardware and software Mark has created.

While the building of new interactive instruments or systems has a robust tradition within the electronic music field dating to the early twentieth century (e.g. the Theremin invented circa 1919), Mark is one of a handful of artists who have specialized in creating new interactive instruments or systems to monitor the movements of dancers and use this information to control other media events in real time. The *MidiDancer*, a wearable device Mark first built in 1989, measures the angular change at several joints on the dancer's body. This information is sent over a wireless link to a computer where the data (input) is used to control a variety of media events (output), e.g. sonic, video, lighting and/or robotic. Mark has written two software programs to map this data flow: *Interactor* and *Isadora*. *Interactor* was made available for others to use, however it required time to learn and learning was made easier with some knowledge of software like Max, a graphical programming environment for music and multimedia. Named after the renowned early twentieth-century modern-dance pioneer Isadora Duncan, *Isadora* was designed for use by a non-specialist after only the briefest of introductions, placing the control of the creative software tools in the hands of the dancers themselves. Another key development is that while *Interactor* was focused on handling MIDI (Musical Instrument Digital Interface) data - a communications protocol

established in 1983 that allows electronic musical instruments to interact with each other - Isadora, while it can also work with MIDI, is designed primarily for the manipulation of video.

(Following the dialogue with Mark is a selection of comments from some of the artists who have been working with Isadora and are familiar with the history and evolution of such software tools. This situates Mark's work in direct connection to a community of users of the software he has created. However, the reader will also find that they open up a range of topics and issues that will not be elaborated on here.)

Part 1: In dialogue with Mark Coniglio

Scott deLahunta:

You have been programming actually longer than you have been composing. So do you consider your programming a part of your artistic practice?

Mark Coniglio:

I definitely do, but it does feel a bit secondary to my composition and/or media art-making in that I see it as more of a 'support' activity. The software I program allows the creation of the artwork, whether sonic and/or visual in some combination with live performance, that I envision. I always seem to resort to musical metaphors for things like this. The artwork is like a musical composition; the software tools are like the instruments in the orchestra. If you can develop a more advanced instrument, you can create more advanced music. The French horn is a good example: early versions required removing a piece of tubing from the instrument and replacing it with another piece of a different length to change key. About 1815 the modern version of the instrument with valves was invented. This technological innovation meant that you could now include the horn in passages where there were rapid changes of key. This was immediately taken advantage of by composers, notably by Beethoven in his symphonies. The difference between Beethoven and me is that he was not driving the innovation directly. I am both artist and instrument inventor. My artistic ideas drive the development of software and hardware I need to realize them; while simultaneously the programming I do expands my world of expressive possibilities.

Scott deLahunta:

I understand the analogy with the French horn, but software 'instruments' are comprised of another type of material altogether. Instead of metals being forged into shapes; one could speak of algorithms, formal abstractions and language. As another way to relate programming and art-making, I am curious about the creative process of writing code compared to working with other materials. It seems that most programming is driven by the assumption that the software will have a purpose (more like the notion of the instrument). Once that purpose is understood and defined so there is a goal, only then does the programmer get down to the work of coding. If this is true then programming is a very different creative process than, say, how the choreographer might work with movement material.

Mark Coniglio:

When writing software it is useful to have a clear goal in mind, and this may not be true with other types of art-making, as you suggest. Having a goal

that is too specific can be detrimental to the process of making a dance, for instance, or composing music. I don't think I understood that early on, but over the last three or four years I found myself exploring much more organic, open-ended approaches to art-making. Being involved in dance in particular, which relies on improvisation as a primary source of generating material, has profoundly influenced my way of working. Now, my approach is a much more, 'try everything and follow your nose'. By this I mean, try not to preconceive as much, make lots of stuff and follow through with the material that seems interesting and let the material begin to tell you what it is about. Now, it's pretty difficult to program that way, a kind of 'goalless' coding. The architecture of the microprocessor, from which all programming languages derive, is actually antithetical to such behaviour. However, this 'follow your nose' approach has definitely influenced the way I program. I still have a goal, but I don't often plan out the algorithms. I simply write towards my goal, improvising my approach to solving the problem.

Scott deLahunta:

There is quite a big discussion about 'software as art' these days in Europe and I'm sure it's going on in North America as well. Besides its utility, its usefulness in helping to support your artworks, do you consider Isadora a work of art by itself?

Mark Coniglio:

In a word, I don't - not in and of itself anyway. It's bit like asking if the telephone system is a work of art. Does the creation of the technology to support that system display an incredibly high level of inventive thought and uber-craftsmanship? Definitely! I have the utmost respect for the creative people that designed and implemented such a robust, complicated, and reliable system. But that particular technology is dormant until you pick up the receiver, ring your lover, tell her that you no longer desire to see her, and a heated conversation ensues.

Scott deLahunta:

It seems to me that the telephone system is the collective result, over time, of a multiplicity of individuals and institutions labouring together, and it's difficult to locate individual contributions in that, or at least I don't use the telephone system and sense that I am in contact with one of its creators. But when someone opens up Isadora and begins to build a patch that will map an arm motion to the speed of a video clip, do they not encounter your presence directly, as the primary creator, in the look and feel of the interface?

Mark Coniglio:

Well, I guess the only way that I *would* consider Isadora to be an artwork is the personal stamp that I have on its design and functionality. To take that a bit further, in a broad sense I could say that I collaborate with each Isadora user as they use the program. Because I can't totally erase myself from the software I create, they have to embrace some of my predilections to make use of the program; which is what happens whenever you choose to collaborate with another artist. It's just a question of how apparent the influence of the software's creator is. That's where software designers and artists who make software may differ, I think. A typical software designer does everything he or she can to filter out their personality and create something that is useful in a general way.

Scott deLahunta:

Maybe we could say that the extent software like Isadora is an artwork is dependent on the degree to which the maker tries to remove him or herself from it? I suppose Isadora then is more of an artwork than say, Photoshop, but is maybe less of an artwork than something like Auto-Illustrator by Adrian Ward (which won the Software Art prize at the Berlin 2001 Transmediale Festival). Auto-Illustrator looks like a vector graphics program, but it doesn't act like one. It misbehaves frequently because it has seemingly autonomous behaviours built into it that take over for you. Adrian creates these behaviours using certain algorithms when he does the coding, so as such his authorship is revealed every time the software does something you did not expect it to, which is frequently.

But let's talk a little bit about the background of Isadora as a graphic programming environment for real-time manipulation of media. You made something similar with Interactor first didn't you?

Mark Coniglio:

Here's a bit of history. In 1986 my soon-to-be mentor and Interactor collaborator Mort Subotnick had just come from a residency at MIT where he was using a program called Hookup created by a student there named David Levitt. Hookup was the first program I know of that used the 'patch-cord' metaphor, i.e. modules that manipulate data are linked by virtual wires, the connection of which is determined by the user. For those in the world of early analogue, patch-cord programmed synthesizers, this was a familiar interface. Mort was using David's program to do tempo-following of MIDI instruments - this allowed him to lock hardware MIDI sequences to the tempo of the live performers. I was a composition student at CalArts at the time, and word had gotten around that I was a good programmer. So Mort contacted me to see if I could hardcode some of the ideas he had implemented in Hookup on a Mac, so that he could use them in his next performance. That program (used in Mort's 1987 multimedia work *Hungers*) would eventually become Interactor. Mort designed the functionality of the early versions, but I became more influential in the design as time went on.

Scott deLahunta:

I guess the hardware and software development of the early to mid 1980s where we saw the advent of the personal computer and more importantly the graphical user interface (marked by Apple's introduction of the Macintosh to the consumer market in 1984) created a context out of which the 'computer' could emerge as a creative instrument or tool. The electronic music field was already well advanced in analysing and exploring the formal and physical properties of music as part of compositional and performing practice, so moving to programming real-time graphic interfaces for this seems like a rather natural progression.

Mark Coniglio:

Yes, that's true and importantly a kind of creative intuition was creeping back in through the development of these new visual interface possibilities for software. Part of the thing I reacted to in Hookup was the way you could easily drop modules into the program and try things; a lot like you could do with the patch-cord synthesizers. I may not have realized it explicitly then, but this ability to program improvisationally allowed for that kind of artful

playfulness that is so important. So I set out to make a similar user interface for Interactor. The creation of Isadora was a natural outgrowth of Interactor. In 1996 Troika Ranch had a two-week residency at STEIM, where I first saw Tom Demeyer's real-time video processing program Image/ine. I first started using Image/ine in concert with Interactor, because Image/ine didn't allow the kind of complicated interactive decision-making that I was used to having in Interactor. So, Interactor would process the MIDI data from my interactive sensors, and then tell Image/ine what to do. By 1998 I was using Image/ine in a major way in my performances with Troika Ranch.

But, while I loved what Image/ine did, I wasn't fond of its table-based interface. And there were problems with crashing during performance, which is unacceptable when there is an audience. Furthermore, we were teaching our Live-I (Live-Interactive) workshops using Interactor and Image/ine, and the students (especially the ones with weak computer backgrounds) found learning both programs, and figuring out how to have them communicate with each other daunting at best. I wanted Isadora to take the qualities that made Interactor and Image/ine great, and put them together in one package that was easy to learn but still offered enough depth to satisfy 'power' users. And, I wanted it to be more affordable to members of my community, which I consider to be choreographers, because of my involvement with Troika Ranch.

Scott deLahunta:

How does Isadora compare to Max, which is probably the most successful and widely used graphic programming environment for controlling and mapping data flow?

Mark Coniglio:

Isadora and Max both inherit the modules linked by the patch-cord metaphor from Hookup. But unlike Max, each Isadora module shows the parameter names and current values for all of its inputs and outputs, and many modules give real-time graphic feedback about their operation. This is important from the perspective of helping new users understand what's going on right away. But perhaps the biggest difference is that Max is a very powerful, open-ended programming language in which you could solve any number of problems. Isadora isn't that. It is a lot like Interactor in that each module is essentially a macro that accomplishes some specific function. This approach helps people who are just beginning to do this kind of work, as it means that useful functionality is already embodied for you and it's very easy to start doing things and getting interesting results quickly (like with Image/ine). Max allows the most flexibility, but may be somewhat more difficult to program because more things have to be built up from scratch. Isadora offers somewhat less flexibility, but is still open-ended enough for the user to imprint his or her aesthetic on the result.

Scott deLahunta:

You have told me that your most important Isadora user is yourself. How do you use Isadora in Troika Ranch's performance work and in particular in connection with the MidiDancer which is the sensor system you built to get data input from a moving body.

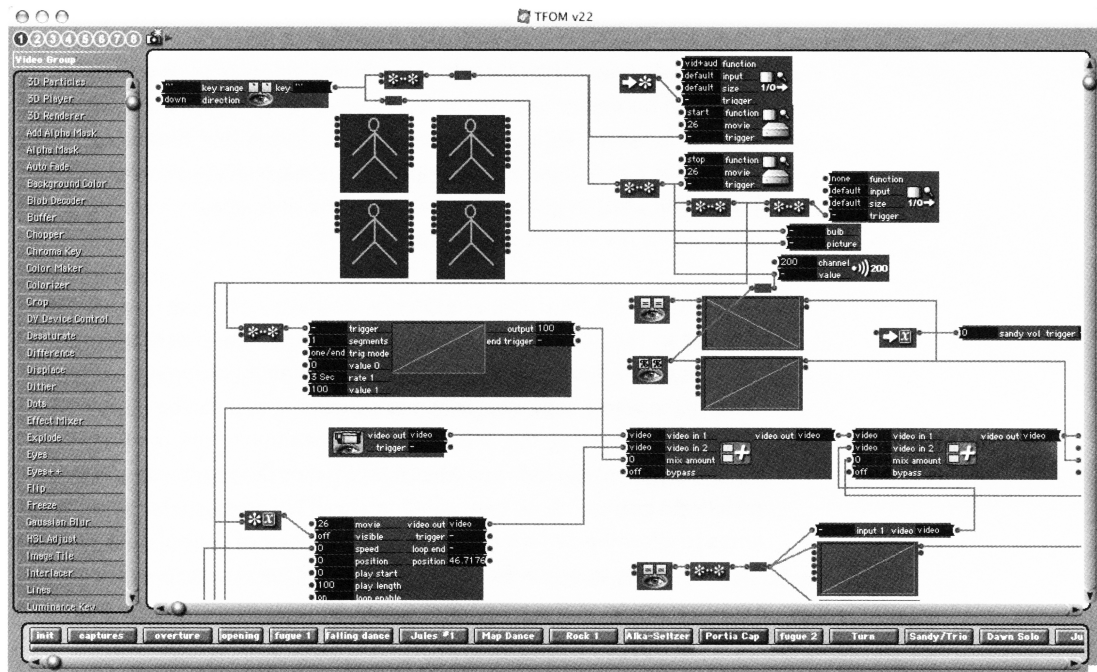


Figure 1. Isadora interface showing how explicit labelling and patching combine to make the software easy to learn.

Mark Coniglio:

I first created MidiDancer in 1989 while I was still a student at CalArts. While it is now much smaller and more reliable, the basic functionality has not changed much since then. Basically, it is a set of up to eight sensors that measure the flexion of joints on a dancer's body. Thirty times a second this information is sent over a wireless, radio link and a receiver up to 150 feet away decodes the information, reporting the angle of each joint in the form of a MIDI message. Any computer with a MIDI interface can accept and process these messages as desired.

The problem with MidiDancer is that, to really play it, and for the audience to see that the dancers are playing, you need to move like a musician. What I mean is that the movement of the dancer needs to be in service of the sound or image that they are generating or controlling. We have worked hard to find ways to make this work choreographically, but it is quite difficult to do. My basic instinct in putting the sensors on the 'gross' joints (elbows, knees, hips, and wrists) was correct, in that these angular changes can be clearly seen by the audience. But I have really been seeing lately that this is not the gestalt that we perceive when we watch a dancer move. We really see energy - that's a bit vague, but it's the best word I can think of to describe it. We're not looking at the individual angles of the joints, but the way that the dancer moves through space and the overall articulation of the movement.

Scott deLahunta:

Well, electronic musicians have been building sophisticated playable interfaces for a long time, but these tend towards either the hyper-instrument (extending

Scott deLahunta: 'Isadora "almost out of beta" ...'

37

an existing musical instrument's capabilities) or a few unique hand-orientated interfaces. But I've always thought that one would need to think quite differently to develop an appropriate system for a trained mover or dancer. I think more research is needed with various sensor input devices and maybe not always towards the aim of live stage performance, but maybe just experimenting much longer with what might emerge from the kind of feedback conditions for the senses interactive systems can generate.

Mark Coniglio:

That's why we'll be using a residency we have next year to make some changes to the MidiDancer. I want to start working with accelerometers in addition to the flexion sensors. The act of turning, or stopping suddenly, or shaking the whole body, now becomes something that can be measured. My instinct is that using this information to manipulate the media will be a more natural linkage between what the audience sees in the dancer and the resulting sonic or visual manipulation. I can then use the position of the limbs to allow the performer to enhance their level of control - but I suspect that being able to sense the impulse of movement may become the primary source of manipulation. I think that, not being a dancer or choreographer myself, I have been slow to let go of the notion of being a musician. In fact, I have often described the MidiDancer as allowing a dancer to be 'both musician and dancer'. I now think that is incorrect. I need a device that allows a dancer to be a dancer, with the media taking its cue from what it sees the dancer doing.

Scott deLahunta:

Isadora is just about finished with its public beta-testing phase during which I know you have been working with several artists who have been trying out the software for different projects and giving you feedback and suggestions for additional functions, etc. I have invited some input from some of these individuals [Part 2], but first I just wanted to ask you to say something about how long you have been working on Isadora and your decision to sell the software instead of using an open-source approach (in which software code is released for free into a collaborative development environment).

Mark Coniglio:

As I think I've indicated, Isadora has grown out of a need Troika Ranch had for a reliable relatively easy to use but also sophisticated software for both workshops as well as performance. The end result is that I have taken two plus years to develop Isadora in my spare time so it has grown quite organically. In regard to deciding to sell Isadora, I don't have much interest in starting a real business, so I am feeling out my progress quite slowly. But in the United States, arts funding is very difficult to come by, and discovering ways to supplement what we can get from the government or foundations is essential. I have always had to hold a day job, so I wanted to see if I could make a tool that would be: (1) useful to me; (2) useful to others; and (3) perhaps generate enough income to help me spend more of my time making my artwork. Obviously, an open-source model would not generate any income, and thus wouldn't help to support my artistic pursuits. As I have mentioned, it is important to me that Isadora is affordable to those in

the dance community, so I figure, for what the program does (and will do as it grows) the single licence fee is a reasonable amount. I have yet to make a final determination about site licences for schools etc, but they will definitely get a break if they purchase a five-seat licence for example. I have no idea at this point if this whole plan will work, or if the burden of supporting a growing user base will actually be more work than the day job, but I am hopeful.

Scott deLahunta:

Thanks Mark. Now, what I would like to do is to invite some comments from some of the artists who have been working with Isadora and are familiar with the history and evolution of artist software tools to reflect on some of our dialogue and to talk about how they use or could use Isadora in their work.

Part 2. Comments from Jean-Baptiste Barrière, Jem Finer, Armando Menicacci, Giorgio Olivero, and Steina Vasulka

Scott deLahunta:

The type of artist/toolmaker relationship Mort Subotnick and Mark had is an interesting one to trace back historically. Steina, you have been working as an artist and researcher in electronic media arts for over thirty years. Would you have something to say to this?

Steina Vasulka:

It is a big topic, but it would be interesting to investigate *all* artist/toolmaker relations in history: French horn, Stradivarius, the well-tempered clavier, the invention of photography, film, acrylics, video, etc. We (the Vasulkas) have almost always worked with toolmakers; analogue first, then digital and now software. On our website we have the 'Eigenwelt der Apparatewelt' exhibition catalogue (put together for the Ars Electronica Festival 1992) devoted to early audio and video toolmakers. There is a going model here: sometimes creators themselves, sometimes collaborators with artists, these toolmakers have invariably been gifted visionary individuals far removed from the industries who adapt to as well as inspire and invent within the technology environments of their time. In the early days of video, the buzzword was 'modification'; when one would go into these tools created for consumers and retrofit them for the artist. We also see developments and knowledge passed along from one tool generation to another; as Mark mentioned, the software packages of now are the synthesizers of recent past.

Jean-Baptiste Barrière:

I would just like to add here that there have been many graphical software tools created for music. One that could be mentioned, if only because of its complementary nature to more performance oriented languages, is Patchwork, which became more recently Open Music (both developed at IRCAM, Paris). It is a language based on Lisp, for computer-assisted composition, with a strong emphasis on musical notation representation. Incidentally, unlike most people think, the patch chord metaphor (first used in software by Max Mathews at Bell Labs) is not derived from analogue synthesizers. Quite the contrary: Robert Moog came to know the work of Mathews and then was inspired by it to design modular synthesizers. The

patch chord paradigm was applied to software from the experience of telephone switchboards.

Scott deLahunta:

Just for those who may not know, Max Mathews has been referred to as the 'father of computer music' who, in 1957, was the first to synthesize music on a computer; performing a 17-second piece on an IBM 704. In 1970, when fast digital chips and new algorithms made 'real time' possible Mathews developed Groove, the 'first computer system for live performance' (<http://www.csounds.com/mathews>). So, here we have the early research into systems out of which eventually software and hardware like Mark's Interactor and the MidiDancer would emerge.

Steina Vasulka:

A very interesting part of the dialogue was when Mark talks about the implications of this dancer interface, the MidiDancer; the attempt to have a dancer be a musical instrument player. When I was a kid, it was not presumed that dancers could act, that actors could sing, that singers could dance. Now they have to know it all. Interestingly, in talking about developing Isadora, Mark refers to wanting to make an interface that a choreographer or dancer can learn to use easily. The interface for Image/ine seems impossible for dancers and musicians to learn; whereas it is a piece of cake for video-makers who are perplexed by Max, etc. Another solution might be to think of multiple interfaces to a single program, like Final Cut Pro (otherwise not my favourite software), that makes the same features available in an 'effects' interface for film/video people and 'Photoshop-type' interface for computer artists.

Armando Menicacci:

Getting back to the points that Mark was making regarding the limitations of the MidiDancer for dance; what he says is very true, in my opinion. In fact, for a 'dancer to be a dancer' means not to have to show the control over media. I don't want to re-open the thread about 'Should we see the dancer's control over the media?' I think that Mark's comments are sufficient on this particular subject, to which I can add a brief comment. In my opinion, if what we want to do is 'pedagogy of interactivity' then we have to show the result of the interaction. But if what we want to do is art then we shouldn't care about the visibility of interaction, unless it is vital for a particular aesthetic project.

As regards my own use of Isadora: I am not sure, but maybe, if we don't consider Mark and Dawn's own work, I've been the first person to use Isadora in the context of a full-length dance performance on stage. We bought a copy of Isadora in the beginning of May; in my experience even the beta was very stable and we managed to use it in a way that was artistically successful. I was working with the choreographer Rachid Ouramdane in Rheims where we have a residency. We'll use it now in Dijon for the same work, titled '+ ou - la', and from 30 October to 4 November we show the same piece in the Grande Salle of the Centre Pompidou in the Festival d'Automne à Paris. What was interesting for Rachid, was to find particular inspiration for the choreography in the way in which Isadora transforms

digital video media. This is important to note ways in which emerging forms of composition/transformations in the digital medium can be translated to and used in mediums of the body; specifically dance. It broadens the idea of these softwares beyond just their specific functionality.

Scott deLahunta:

Giorgio and Jem, neither of you, as far as I am aware, come out of the contemporary dance field. Can you say something about your backgrounds and how you arrived at this point of working with Isadora?

Giorgio Olivero:

We (myself and Andrea Clemente) come from a background as web and graphic designers. In 2000, we started doing visuals for the club scene when we were students of the Scienze della Comunicazione faculty in Torino. We were very excited by the possibility offered by this evolution of clubbing, and we began experimenting with aesthetics and technology. We were ignorant of what was going on in other places like London, Berlin or Amsterdam, so we built up our knowledge from scratch, finding only later that similar stuff was happening elsewhere. At the beginning, we used mostly found video footage with short inserts of wild motion graphics. Now we produce (shoot, edit, and post-produce) nearly all the basic material that we use.

I had been looking at Max and NATO (a graphic programming environment similar to Max designed to handle Quicktime media) a lot in the last few months but the learning curve was so steep. I discovered Isadora by chance by following a link to the Troika Ranch website and downloaded the beta version. I learnt everything quickly (note that my previous experience with programming was self-taught action script - at a basic-medium level) and in three hours I had built a very complex patch that performed many different tasks. It's so different from the other application that we've been using. First of all you get a candy factory and not a candy packet. There are all the basic functions one could perform on digital video. And you have them in real time, which allows for much more improvisation. I can build and modify a patch as we play ... it's the Lego for video junkies, definitely. Our VJ'ing practice will shift with this tool, because now we do much less post-production and can use video that is more 'raw'. For example, in Isadora you can easily bring in live video input of clubbers dancing into the overall mix.

And we're producing a live audio-visual show that is the most interesting stuff we've been working on so far, and a shift from the work in the club environment. It's called *Città Invisibili*, and it's a multi-projection event that will be premiered at the beginning of October based on city shots taken in Tokyo, LA, London, Detroit and Barcelona. The concept is of the contemporary *flâneur* fused with the poetry of the city jungle. We're preparing many Isadora patches to play visual and audio in a very tight relation, and it will be the first time that we'll use the application without the backup of other programs. Perhaps if one is a very skilled programmer fluent with the Quicktime API (Application Program Interface) Isadora would not be so useful, but for the rest of us it is. I was a big player with Lego years ago, and I can't see the difference. I feel like I am playing while trying out things. And

after sessions of wild patch-cord dragging, I have run into consequences of a chained 'effect' that were not planned but are very interesting.

Jem Finer:

Ever since I first got hold of a computer, probably the Sinclair Spectrum in the early 1980s, I have been interested in programming them to make music. This interest continued and gained momentum to the point of my composing a 1000-year-long piece of music, *Longplayer*, in 1995. I spent a few years exploring numerous possibilities for composing such a piece: artificial life, AI, chaos theory, neural nets, interaction with the environment, etc. At first the problem was that I had no language to program in, but eventually I discovered SuperCollider, which I've used pretty much exclusively ever since (SuperCollider is an environment and programming language for real-time audio synthesis in which you can write programs to generate or process sound).

For the last two or three years I have wanted to create a visual counterpart to the music I make. My live performances are based on a symbiotic relationship between my computer, running a library of SuperCollider patches I've written, and me. The patches do various things to grabbed or streamed input and can run in parallel (up to the limit of the CPU). I want to be able to do the kind of things I do to audio to visuals, or at least to experiment with that approach. It may well be not so interesting as new things I discover. The problem has been finding an environment in which to do this. I tried Image/ine, but had difficulty with the interface, it wasn't intuitive enough for me (something Steina mentioned earlier). I tried Arkaos too (a VJ authoring tool); never tried NATO because I couldn't get a demo, and it was too expensive to buy on spec.

So when Isadora turned up it seemed an answer to a prayer. Intuitive, flexible, not too expensive and it produced good-quality images. Up to now I've had little time to explore it, but enough to know that it's worth an investment of time to get to grips with. I've just been writing simple patches to explore the objects and interaction between them. There are a few things I would like to be able to do that seem out of its range at the minute; one is to have a far greater interaction with sound. I want to be able to link things to precise frequencies, for example, which involves FFT (fast fourier transforms) stuff. The only way at the moment is to write this in SuperCollider and convert it to MIDI info to send to Isadora, which may be messy. I prefer to run just one thing if I can, but I have a feeling that I might need to use Jitter for some stuff. In that scenario I imagine using Isadora for any tasks not dependent on anything more complex than listening to sound in. It certainly requires less programming and the interface is simple and informative, really great. I love the simplicity of the stage set-up and the rendering possibilities.

Scott deLahunta:

One of my interests with writing up this discussion is to provide some insight for those who may be relatively new to software like Isadora; so I think what is important is to establish the range of different types of environments and contexts the software might be used in; from VJ'ing to installation and stage performances. However, I also don't want to misrepresent this field of possibilities by

implying that Isadora is the only or even the best programme available for everything; in fact, there are several cross-media synthesis programs that take advantage of the possibility to manipulate digital video material in real time. So, it's important that both Giorgio and Jem have mentioned other programs including the newly available Jitter, which has been created for the Max graphical programming environment. Another is Keystroke, which is a media-mixing multi-user environment and should be mentioned as it is another development of Image/ine and Tom Demeyer has been working with that team. There are links to these programs provided at the end of this article. It is beyond the scope of this conversation to delve too deeply into each one or to make any real comparisons. However, I think Jean-Baptiste Barrière may be able to offer some interesting thoughts on Isadora in relation to some of these other softwares.

Jean-Baptiste Barrière:

Well, I pretty much have them all and at least tried each extensively at some point. I consider them for their respective and different qualities and do not want to be restricted to using only one, whatever its capabilities. Until I discovered Isadora, I used Image/ine controlled by Max, starting from September 1997, for a still ongoing series of installations and performances called Reality Checks. I first used Isadora for an installation last year called *Chasing Wind: the Well of Vanities*, presented in the Abbaye of Maubuisson, an ancient convent 30 kilometres north-west of Paris. In this piece, people enter through one side of a large empty room. At the centre of this room is a sort of well where people can see their own image floating, mixed with other images. When they turn around they modify these images, as well as the sound in the room. Another image, much larger, is projected at the other end of the room on the wall. By moving about the space, people are triggering specific sequences of images and sounds, elements of texts (mostly extracted from the Ecclesiast in the Bible). Generally speaking, this is a very meditative installation piece, calling on the viewer to reflect on identity and death.

Technically speaking, first one computer is running Max with Cyclops, which is software that can detect people moving around by analysing the video image of the space. Then this information is transmitted by MIDI to another computer running Max to do the mapping between what is detected and what I call the 'interactive scenario': what is to happen for every 'situation' and succession of situations which I call 'trajectories'. This second computer controls with MIDI two others running Isadora; each one receiving the video input of a different camera: one for the well, the other for the wall. It also produces/synthesizes all the sound, and Isadora is used similarly to play previously prepared video materials (with Final Cut and After Effects), and process it together with live video capture, mostly doing keying and displace.

What I appreciate with Isadora is its clarity and straightforwardness. It is very easy to use, and all sorts of interactive ideas can be implemented with it, both very quickly and efficiently. It has been growing rapidly and intelligently from the original modularization of the Image/ine model, to become an original and unique tool. It obviously grows out of Mark's extensive dual

2. Aiming to expand critical awareness of software, a small number of cultural theorists have started to write about its development. Examples include: Ellen Ullman's *Close to the Machine: Technophilia and its Discontents*, San Francisco: City Lights Books, 1997; and Matthew Fuller's *Behind the Blip: Essays on the Culture of Software*. New York: Autonomedia, 2003 (see Fuller for references to other material).

complementary experience with music and dance composition needs. It has reached a kind of ideal balance between an application and a graphical language that is a very clear and easy one to master. Which makes a clear difference with Max for instance, and should provide it a distinct audience.

It has been a pleasure to watch it evolve and see Mark's responsiveness to users; at the moment he quite literally is collaborating with each user. I hope he will continue to be able to develop it in the same way that he has been doing until now; without losing track of his artistic projects since clearly one activity nourishes the other reciprocally. That may be difficult, while he stays the only developer or source of novelty. This is why I want to encourage him to open up the possibility for users to make their own modules and/or link external codes such as Photoshop's plug-ins or Director's xtras. Why not also have, for instance, Jitter's (Jitter is a customized set of Max objects that manipulate video media in real time) plug-ins inside of Isadora? This would be making the most of the two paradigms, would be equally profitable for Isadora and Jitter. This is the way in which I would like to see Isadora evolve: to continue to offer new ideas related to transformations, control and interface, and at the same time allow other artist programmers to insert/import specific code. This would be the most satisfying for users, and therefore help to provide a sustained development for Isadora.

Scott deLahunta:

I think this last comment will be challenging for our readers who are not so familiar with computer programming. However, it is essential to help develop the understanding amongst a broader public how a unique software like Isadora, the manifestation of Mark's creative vision and hard work, contributes to an open arena for artists to exchange ideas and materials that relate both to the tools they use as well as the artworks they will make with them. Thank you everyone, thank you, Mark for taking part in this conversation.

Part 3. Update and summary

The public beta of Isadora for the Apple Macintosh was first made available in March 2002 and the above e-mail exchanges took place in September of that same year. The software was officially released as 1.0 in December 2003, and a PC version is being developed. Over 300 copies have been distributed over the Internet, and Mark is currently the sole official provider of user support; backed up by an online e-mail list of over 200 members where problems and solutions are shared. Versions, releases and users are part of the normal cycle of software development; therefore, these statements about the progress of Isadora are almost universally familiar. However, most software production is notoriously untraceable in terms of its creators. The dialogue with Mark and the other artists indicates that when software, such as Max and Isadora, is created in artistic contexts its inventors tend to be acknowledged. This does not always take place without controversy, of course, but this historical trace of software authors seems to be less evident in other contexts, i.e. commercial, industrial, etc.²

By acknowledging the lineage of creative practice (of making software tools) he is a part of, Mark places himself in a context of evolving ideas and

concepts. At the same time, he is also the author of something original, most visibly the Isadora interface. His individuality, embedded in the software, makes him an implicit collaborator, in a broad sense, with those who are using it in their art-making practice. Jean-Baptiste Barrière also describes his connection to other Isadora users as one of collaboration in terms of direct support. While this interplay between evolution, authorship and collaboration underpins the development of Isadora, its functionality is the crucial component that changes, depending upon the needs of the artist/programmer. As a composer, a performance-maker and a teacher of workshops, Mark creates the software for his own use as well as developing it for other practitioners. In summary, while Mark understandably refers to his software programming as a support activity in relation to his art-making and does not consider Isadora a work of art itself, it seems that there is no single point at which his artistic work stops and his software programming begins.

Related Links

Further information regarding the software discussed throughout the article can be found at:

Arkaos:

<http://www.arkaos.net>

Auto-Illustrator (Adrian Ward):

<http://www.auto-illustrator.com>

Image/ine (Tom Demeyer):

<http://www.image-ine.org>

IRCAM, Paris:

<http://www.ircam.fr>

Isadora (TroikaTronix):

<http://www.troikatronix.com>

Keystroke (a real-time cross-media synthesis multi-user environment):

<http://www.keyworx.org>

Max and Jitter (Cycling 74):

<http://www.cycling74.com>

'Putting Max in Perspective' (published in *Computer Music Journal*, 1993):

<http://mitpress2.mit.edu/e-journals/Computer-Music-Journal/cmjlib/editors-notes/17-2.html>

'A Discussion of NATO.o+55+3d modular' by Jeremy Bernstein (first published August, 2000):

<http://www.bootsquad.com/nato>

SoftVNS (David Rokeby):

<http://www.interlog.com/~drokeby/softVNS.html>

STEIM, Amsterdam:

<http://www.steim.org>

Supercollider:

<http://www.audiosynth.com>

Troika Ranch

<http://www.troikaranch.org>

Interviewee details

Jean-Baptiste Barrière has made studies in the fields of music, philosophy and mathematical logic. From 1981 to 1998, he worked at IRCAM/Centre Georges Pompidou in Paris, France. Besides making his own compositions and media installation works, he has worked with other artists such as Maurice Benayoun (for whom he composed the music of several virtual-reality pieces including *Worldskin* which won the Prix Ars Electronica 1998 for Interactive Art) and Peter Greenaway. <http://www.barriere.org>

Jem Finer began playing music in the 1970s, having left university with a degree in Computer Science. In 1981, he became a founder member of the Pogues, writing, recording and touring for the following fifteen years. Recent work includes *Longplayer*, *TILT*, music for Copenhagen Town Square, various recordings for installations, films and television and *Autodestruct I & II*, a live music performance. <http://www.longplayer.org> (*Longplayer*)

Armando Menicacci has a background in music and dance studies. He is the Director of Mediadanse, a research laboratory of Paris 8 University Dance Department, a member of ANOMOS and of the Rachid Ouramdane's dance company, Fin Novembre, and serves as a consultant for various art institutions and artists. <http://www.anomos.org>

Giorgio Olivero and Andrea Clemente (softly.kicking) have a background in web and graphic-design work. In 2000, they began doing real-time video performances in nightclubs and eventually won the VJ competition of the Arezzo Wave festival (the biggest free rock festival in Italy) in 2001. Their subsequent engagements included a residency with a large group of well-known Italian disc-jockeys; and playing at Fiesta des Sud in Marseille in October 2002. <http://www.softlykicking.com>

Steina Vasulka is a key figure in the field of 'video art' since its beginnings. With her husband Woody Vasulka, she has won numerous awards, and their collaborative works have been widely exhibited internationally. In 1971, they co-founded The Kitchen in New York City, the celebrated media arts theatre. Exhibitions of her individual work have been seen at festivals and institutions including the Centre Georges Pompidou, Paris; The Kitchen, New York; the Museum of Art, Carnegie Institute, Pittsburgh; and the Whitney Museum of American Art Biennial, among many others. Steina lives in Santa Fe, New Mexico. <http://www.artscilab.org>

Suggested citation

deLahunta, S. (2005), 'Isadora "almost out of beta": tracing the development of a new software tool for performing artists', *International Journal of Performance Arts and Digital Media* 1: 1, pp. 31–46. doi: 10.1386/padm.1.1.31/1

Contributor details

Scott deLahunta works from his base in Amsterdam as a researcher, writer, consultant and co-organizer on a wide range of international projects bringing performance arts into conjunction with other disciplines and practices. He is an Associate Research Fellow at Dartington College of Arts and an affiliated researcher with Crucible, an interdisciplinary research network within the University of Cambridge. He lectures on a new postgraduate study in Choreography/New Media at the Amsterdam School for the Arts and serves on the editorial boards of the *International Journal of Performance and Digital Media* and *Performance Research*. Contact: Dartington College of Arts, Totnes, Devon TQ9 6EJ. E-mail: sdela@ahk.nl