Electronic Media Review

The postprint publication of AIC's Electronic Media Group

http://resources.conservation-us.org/emg-review/volume-4-2015-2016/heinen/

Archiving The Brotherhood: Proposing a Technical Genealogy for Time-Based Works

Joey Heinen
The Electronic Media Review, Volume Four: 2015-2016

ABSTRACT

The following case study is a proposal for the creation of a “technical genealogy” for time-based artworks, specifically focusing on works that use a multitude of hardware and software in a highly-customized environment. *The Brotherhood,* a six-part installation series by Woody Vasulka (b. 1937), is an example of an artwork where a thorough understanding of the employed technologies mediating the work and the intended user experience are just as much—if not more— an essential component to the artwork as the interactive experience alone. Many models currently exist for defining the relationship between materials and behaviors in interactive and technological artworks, informing more pragmatic approaches towards ongoing care and stewardship for the artwork, which are more often than not guided by an artist interview. In this particular case study, the artists are imagining the destination for their collection as a research library so that their body of work and the works’ employed technologies can be studied. For this reason, the process of describing and documenting the relationship between behaviors and materials takes on a different role than it might in a museum setting, particularly in how to provide access to this work and through what means. Through documentation (both pre-existing and generated through this exercise) and examining the relationships between hardware and software, I will argue for a new means of archiving a complex artwork with a particular focus on the work’s innovative methods, so as to leverage scholarly intrigue.

THE VASULKAS: ARTISTIC PRACTICE IN CONTEXT

For time-based media artworks, public access is usually thought of in terms of installation of the piece. Increasingly more theory and practice have emerged, often focusing on the stabilization of a time-based media work or determining the parameters of the piece that can be modified without altering artistic intent. In some cases, modifying the piece in order to respond to technological changes may motivate the artist to think of the work as a new version or new artwork, particularly if the work itself comments directly on technology. The intent of the artist is of utmost significance, and an understanding of these parameters emerge out of the artist interview. However, in some cases, the artistic intent of a work extends beyond reinstallation of the piece, and may serve as a historical record, offering itself as evidence of technological innovation that holds a significant place in history. The polemics of scholarly and experiential access can be seen quite distinctly when embarking on conservation plans for the work of artists Steina (b. 1940) and Woody Vasulka (b. 1937). The Vasulkas are two pioneers of video art whose work spans from the 1960s to the present and whose practice has included early experiments with music and performance, image processing, computer graphics, lighting, electrical engineering, and integrated software and hardware in interactive installations. The Vasulkas are extremely prolific artists with an impressively diverse artistic product. However, they also consider their process to be just as significant as the finished works, and are committed more broadly to scholarly and theoretical discussions around art-making. In addition to their more scholarly approaches towards art and technology—at the Center for Media Study, State University of New York at Buffalo (1974-1980) and Eigenwelt der Apparate Welt at Ars Electronica, 1992, and Techne & Eros, 1998-1999 (Bonin 2003a)—they began to archive their own life and work early in their artistic practice, retaining outtakes/raw materials, studies, sketches, and research material in addition to their finished projects. During the summer of 2013, I worked with the Vasulkas to prepare an inventory of their collection of video, papers, photographs, and art documentation, and made recommendations for prioritizing objects in their collection based on various risks, including media obsolescence. My work was intended to facilitate a donation of the Vasulkas' archive to a research institution. At the outset of this project, the Vasulkas were in conversations with the University of Colorado Boulder (UC), which would eventually own and provide long-term access to their entire artistic and personal collection.

During the work in 2013, it was clear that the Vasulkas' artworks, particularly the video masters, were of the utmost importance, although there was another aspect to the collection that was of primary interest to the Vasulkas. In our earliest conversations Woody was quite insistent that in the overall processing of their collection at the UC Archives, it would be essential to bear in mind that there is a "curriculum" around their work. By curriculum he meant that the means used to create their works are just as important, if not more important, than the final works themselves. Another vital aspect of the Vasulkas' artistic process is their relationship with their technological tools and the potential for abstraction, interactivity, and extemporaneous manipulation. It is worth mentioning that their respective practices are rooted in music and engineering, with video eventually emerging as a medium that suited these impulses very well.

One theme that can be seen throughout their artistic career is exploration of the signal and an invisible language contained therein. In many instances, the nucleus of these explorations was found in audio signals. The couple met in Prague while Steina was a student at the State Music Conservatory for violin performance and Woody was at the Academy of Performing Arts, Faculty of Film and Television, studying filmmaking. These melded interests of the musical impulse and the moving image informed their work throughout the several decades of art-making to come (fig. 2).

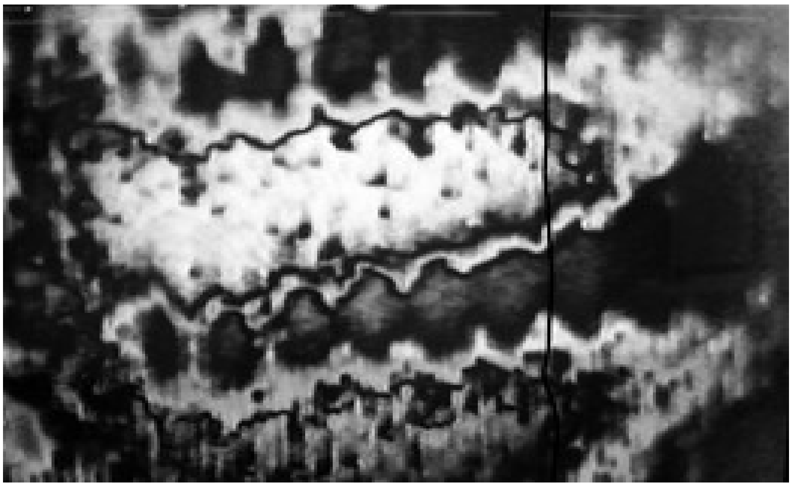


Fig. 2: Steina Vasulka, *Descends* (1972) http://shapesforsound.typepad.com/inspiration/2010/05/steina-and-woody-vasulka-studies-19701971.html c. The Vasulkas

The use of the Putney (a.k.a EMS VCS3) was one of their first experiments in using an existing audio system and wiring it to a television system, abstracting the video image from a live camera feed by using the amplitude and frequency of the audio signal to interrupt and distort the feed (Bronin 2003b). One of Steina's signature works is *Violin Power* (1978) in which she rigged a microphone in coordination with sound processing tools (she later used an electric violin) and through interference of the audio waves, manipulated the horizontal scanning of a live video feed, resulting in an abstracted moving image (fig. 3).

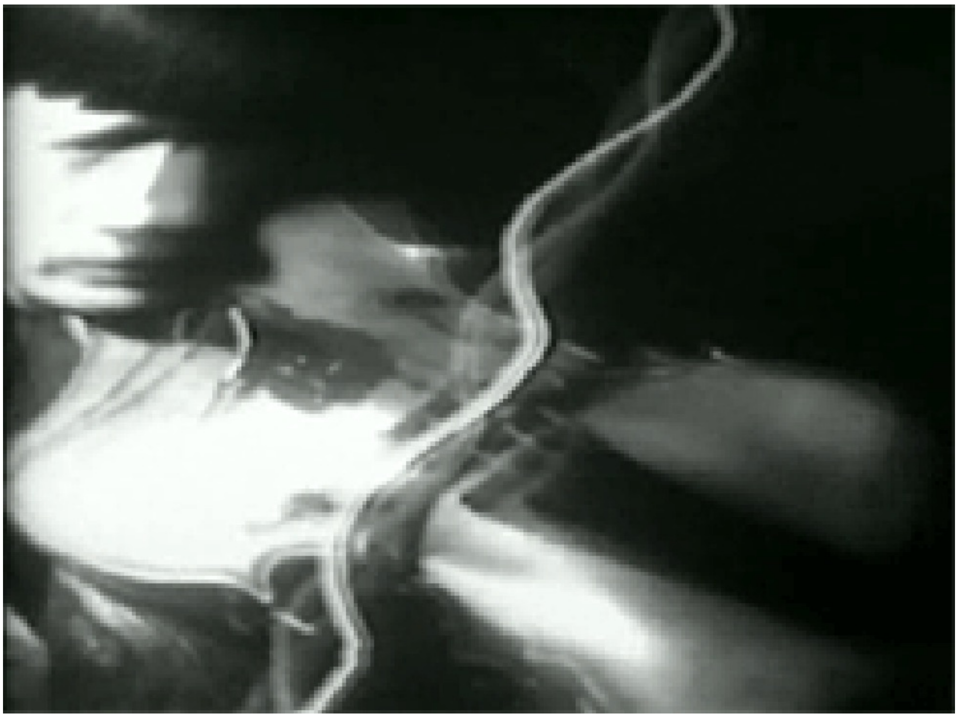


Fig. 3: Steina Vasulka, *Violin Power* (1978)

http://www.abstractmachine.net/thesis/machines.php?id=220 c. The Vasulkas

Their relationship with the liberated potential of audio and video signals through synthesizers and oscilloscopes expressed their intense interest in the anatomy of the signal and its relationship with perceptions of reality. Woody famously expressed these theories on the electronic signal in his *Time-Energy Objects* series in which three primary waveforms—sine, square, and triangle—are fed into a scan processor and result in a multitude of fixed shapes on the monitor. The Vasulkas saw how a seemingly invisible and ineffable signal could actually manifest into tangible forms and objects if only given dimensional representation through the appropriate tools. Though these concepts were expressed quite iconically during their pure analog video art phase of the early 1970s, the same credo was, and is, seen in their work as it has continued into the computer age.

THE BROTHERHOOD

In the 1980s, Woody Vasulka began sifting through castaway objects at the Military Research Centre in Los Alamos, New Mexico. While these objects would have been of little use to most people, Woody's propensity for electrical tools and his talent for engineering hybrid objects made one person's trash another person's treasure. Military research has motivated innovation in technology for decades, ultimately trickling down to the masses once these projects transition into the commercial market. Video tape recorders and cameras, personal computers, and infrared motion detection can all be traced back to military research (Schoenherr 2005). While the sordid past of these machines and their inherent purposes of surveillance and defense is at the heart of Woody's work *The Brotherhood* (1990-1998) (fig. 1), the Vasulkas also imagined a new potential for these machines that they hoped would remind visitors of how machines are often designed to convey and expand upon human behaviors and capabilities.

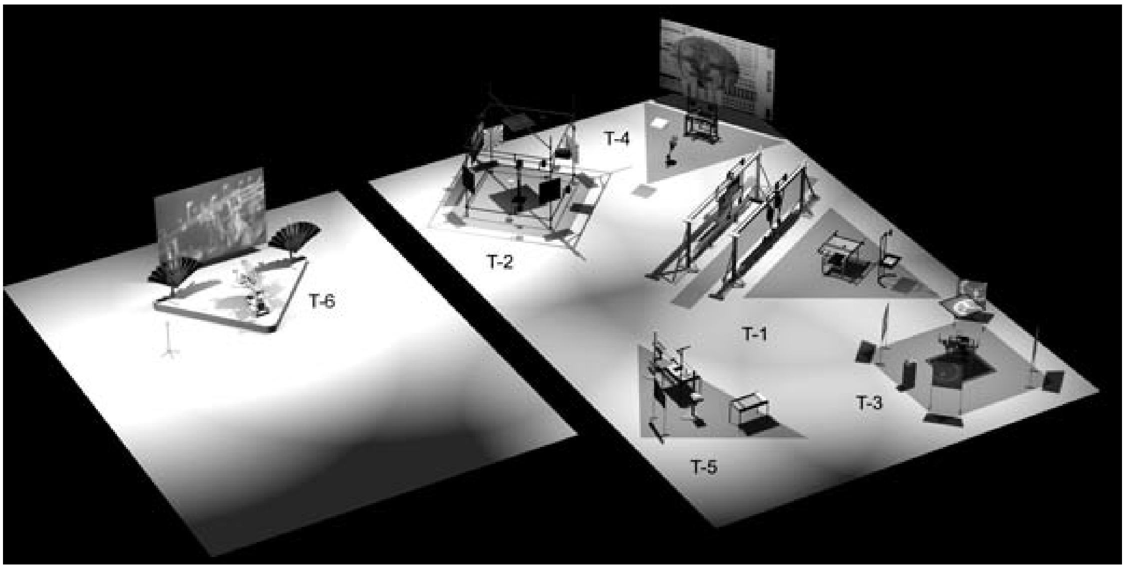


Fig. 1. Installation Diagram for *The Brotherhood* http://vasulka.org/Woody/Brotherhood/Brotherhood.html 1998. Last accessed 12/2015 c. The Vasulkas

The Brotherhood was born out of a singular work-in-progress, which then grew into a work with six movements or "Tables." The Tables were displayed in various ensembles and in some instances as a complete group. Though each of the six movements had its own unique hardware/software, physical configurations, and varying degrees and modes of user-interactivity, each Table followed a relatively similar structure. These machines with seemingly human behaviors acted in dialog with human presence and actions; they were intended to be experienced by the visitors as if they were entering the realm of the machine and were encountering them in their natural habitat. As the title of the piece also suggests, Woody wished to explore themes of masculinity that are intrinsically tied with militarism. This theme of masculinity further extends to notions of power through use of surveillance and the evoking of a sense of dominance of the machine over humans. These themes are critiqued and necessarily deconstructed in the work's use of hacking and in the unpredictability of the machines, intending the viewer to rethink whether machines are merely instruments of power or can in fact demonstrate their own sense of autonomy.

For brevity's sake, two of the six tables from *The Brotherhood* will be described. The first chronological movement of *The Brotherhood* (and the initial impetus for the project as a whole) was *The Theater of Hybrid Automata*, a work that was not purely interactive, but that demonstrates the Vasulkas' interest in the melding of humans and machines. The Vasulkas created *Automata* (as it was later simply called) initially for the 1990 exhibit *Events in the Elsewhere: Performance and Exhibition*, organized by Joan LaBarbara at the Center for Contemporary Arts in Santa Fe (fig. 4).

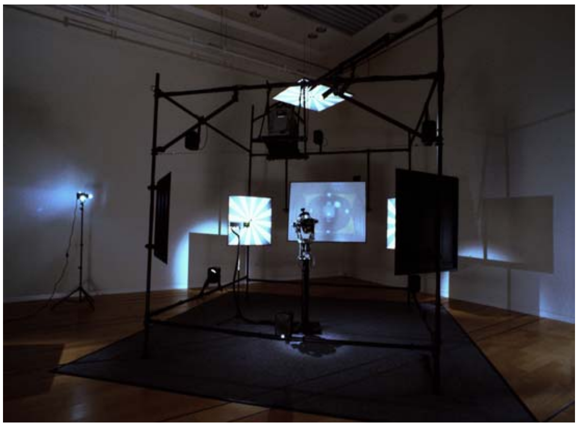


Fig. 4: Installation shot of *Automata*, 1998, ICC Tokyo, http://vasulka.org/Woody/Brotherhood/Automata/PBG_Automata08.html c. The Vasulkas

The centerpiece of *Automata* was the RPT (Rotate-Pan-Tilt) camera, mounted on a gyroscopic slip-ring assembly head that was originally used in locating missile targets in aerial attacks. A computer-generated voice called out different spatial commands based on pre-programmed targets; these commands then prompted the RPT to locate the targets within the cubed demarcations of the space. The camera projected a live feed of the target over-lain with a 3D rendering of the space, creating a liminal space where both the human and computer realm could coexist. The visitor did not affect the space besides being implicated in the live camera feed; the interactive and variable nature of the piece was found in the random sequence of directional commands and use of chance and scripted algorithms.

The sixth and final Table, *The Maiden*, was often considered the most iconic work in the series; it was simple in structure, but extremely complex in its mechanical and computational processes. The piece was comprised of a microphone into which the visitor spoke, a screen that displayed sequences of video activated by the visitor's voice, and the Maiden, a solenoid-valve sculpture whose skeleton was controlled by the stimulation of specific appendages. The sculpture was based off of a chiropractic table that could be mechanically lowered, tilted, and extended for enhanced procedures. The sculpture expanded, contracted, and moved its various appendages based on pre-programmed pitches and velocities from the visitor's voice, which were fed through a voice recognition program. When installed at the ICC (InterCommunication Center) in Tokyo, Steina performed along with the Maiden using a Musical Instrument Digital Interface (MIDI) Violin to control the movements of the sculpture (fig. 5).

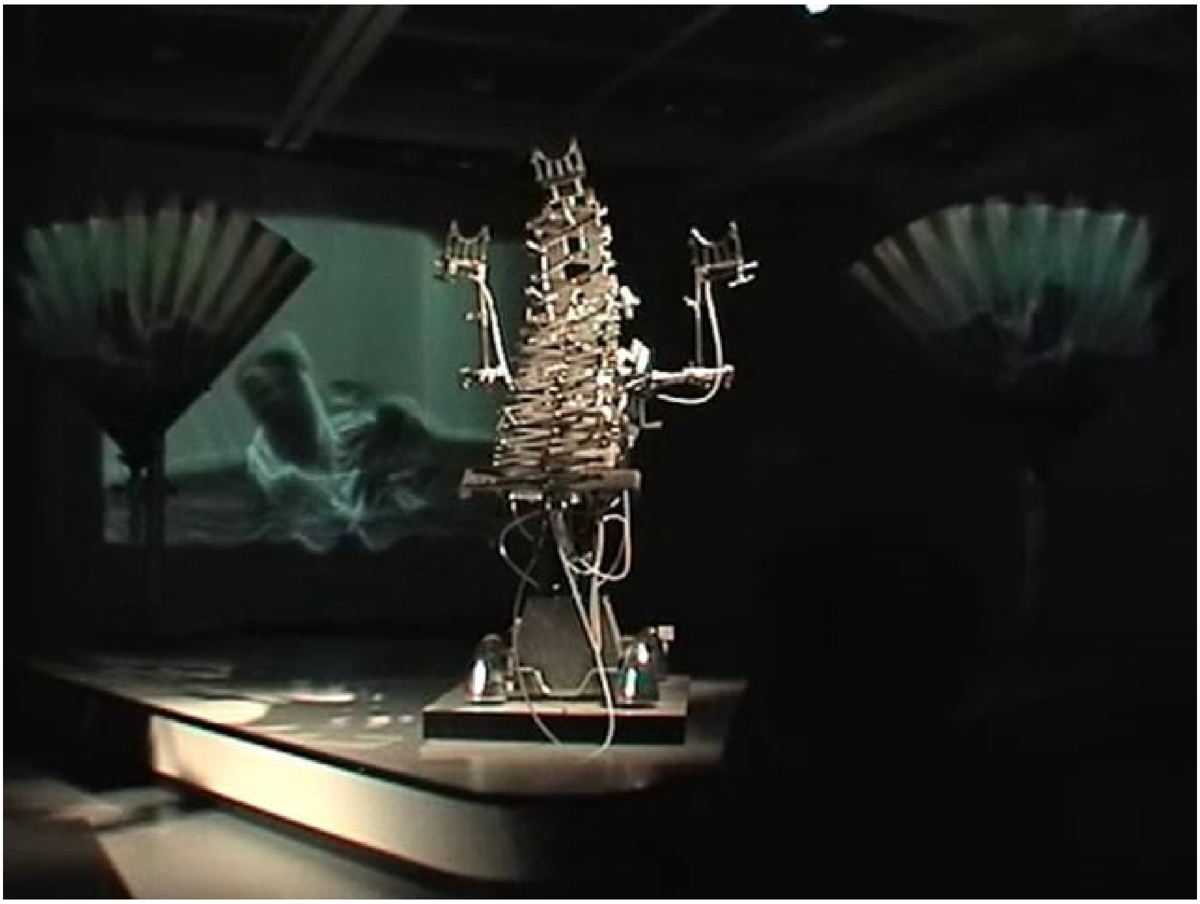


Fig. 5: Installation shot of *The Maiden*, 1998, ICC Tokyo, http://vasulkainterviews.org/page/6 c. The Vasulkas

Each Table of *The Brotherhood*, though highly complex and unique in its own right, can be defined based on a primary structure of stimulus and response. They were also all controlled through the use of a common host terminal that configured each Table through a singular system. However, it is not as simple to define all of the interactive possibilities for each given Table in the work, or to unanimously describe the nuances that can occur therein. The types of responses and the different degrees of responsiveness contained in each work—although in some cases the responses can be directed and in others seemingly random—are not always as they appear. In some pieces there may not be a clear one-to-one relationship between the stimulus and the response. For example, in some cases the visitor did not necessarily understand how mechanical movements or changes in atmosphere (lighting/audio) corresponded to specific user interventions. Other nuances included differences in response time to actions by the visitor and the degree to which the machine was "in dialogue" with the visitor, as opposed to simply exhibiting a one-way response to a visitor's command. Woody has never articulated specific response times for each movement, but rather has insisted that the installations were made to feel organic. He preferred that it appear that the art works reacted rather than responded; that they appeared to be their own autonomous entities, even though they were specifically programmed. In some instances, a digital log jam—in which certain commands may have been lost or overwritten—was desirable to Woody because in that case the machine seemed to be behaving of its own volition.

The behavioral aspects of the piece proved to be some of the most significant and challenging properties for preservation as we delved further into the development and functionality of *The Brotherhood*. To make matters somewhat simpler for the conservator, the installations were video documented and the system design exists in various forms such that behaviors can be more closely studied. However, knowledge of the behaviors will need to emerge from a deeper understanding of the precise integrative design of *The Brotherhood,* which becomes immediately apparent when examining the system concept for the entire work. *The Brotherhood* is unique in that rather than having a precisely controlled configuration system that resulted in a well-defined stimulus and response, randomness was incorporated as an inherent property, allowing for the organic feeling which Woody insisted upon. Thusly, determining the best vehicle to deliver these mechanical responses required an assessment of what available technologies, could, in concert, approach the desired outcomes as closely as possible. This required strong attention to a basic, yet manipulatable, programming language and a means of communicating with external hardware. These aspects of the design of *The Brotherhood* are the most fundamental elements for the archivist/conservator's understanding of the work's contribution to a visionary genealogy of innovative technology in interactive artworks.

MAKING *THE BROTHERHOOD*: CONCEPTS AND COMPUTATION

With the notion of the time-based media artwork as historical record as our guide, we will explore the origins of *The Brotherhood*, which will inform the means through which the piece is cataloged and how the remaining piece of documentation and source code will be contextualized to form this descriptive framework. In relating this work to broader discussion of the history of technological innovation, it is important to note some of the more noteworthy processes—created for *The Brotherhood*—that can be placed within a technological genealogy. A genealogy traces lineages and histories within a logical group or family, and the innovative features of *The Brotherhood* are traced by three distinctive lifelines within the work, which describe its functionality. This is to say that these elements of the work are also what sustain its existence and are thus some of the most essential (and difficult) aspects of the piece to adequately archive and preserve in order to ensure long-term access. The three elements are: the use of MIDI as a main abstraction language, the use of programming languages and interfaces to control a diverse amount of external hardware, and a custom-designed computer environment that can handle the volume of information being processed. In identifying these three lifelines, we not only see a potential diagram for grouping the related materials (hardware, software, documentation) but can see how some conventional approaches in archiving and conservation begin to fall apart for a work this complex.

As was mentioned earlier with regard to the Vasulkas' artistic practice, the relationship of visual forms to signals and the abstracted languages of sound and image were almost always a presence in their work. While the human-machine dynamic at the core of *The Brotherhood* was a relatively new theme in their work in the 1990s, employing user-generated impulses to expose and manipulate the architecture of signals was already central to their art. With the advent of digital music synthesizing in the 1980s, most prominently with the introduction of the Musical Instrument Digital Interface protocol (MIDI), the Vasulkas began to think about how they could manipulate a digital stream in ways they had never before imagined. MIDI is a communication protocol that enables specially designed instruments and external converters to process an analog audio stream as a bit stream; the bytes are packaged based on their purpose (i.e., start/stop, pitch, volume, velocity/punctuation, tremolo/vibrato). It occurred to Steina and Woody that if hardware existed to convert an analog stream into standardized digital binary machine code, and if this bitstream could be easily hacked—such that its intended commands could serve as discrete triggers rather than notes for a musical composition—then there would be an infinite number of ways that a stream could be translated and enacted upon via other outputs and programs.

MIDI is an 8-bit protocol, meaning that each byte is comprised of a cluster of eight bits of binary information (1's and 0's). A MIDI byte can take one of two forms: as a basic MIDI header message allowing the computer to recognize the signal as MIDI, or a means of packaging the information from an analog input into some sort of digital output. These two types are called Status or Data, respectively. Status indicates the type of message being sent; examples being whether the note is on or off (in other words, indicating the start and end of a single command) (Manning 1993, 277). Data includes the content of the message, defining in most cases pitch (numerical assignment of the note within a scale) and velocity (the volume/amplitude) of the note, although Data can also include information related to voice, tremolo, accent, sound dampening, and so on. Additional information, usually contained in the header of the initial MIDI message, determines the designated MIDI channel that is delivering the message (Planet of Tune 2013) (fig. 6). The channel information allows the computer to isolate a bit stream and direct it to a designated output destination, a particularly useful function in the presence of multiple simultaneous input and output signals.

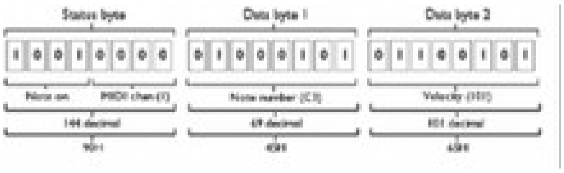


Fig. 6: Architecture of the MIDI bitstream, Planet of Tune, http://www.planetoftunes.com/sequence/se_media/message.gif last accessed 12/2015

While the MIDI bit stream was essential for using the functions of pitch and velocity to contain a large, yet finite, continuum of possible values and mapping these specific inputs to mechanical/audio-visual outputs, it was also the essential means for communicating out to specific hardware points in order to interpret those bit streams as electrical impulses. Equipment such as MIDI samplers/controllers and serial convertor boxes would trigger specific presets (such as lighting conditions) or translate MIDI binary into serial data, respectively. Herein lies one of the most significant aspects of the archival record for a time-based work particularly considered within a broader technological genealogy. While the piece could potentially be reinterpreted using more modern technologies and programming languages/software, the original configuration depends on a specific means for creating a computational language and creates dependencies between hardware and software that are irreplaceable. However, the plot further thickens.

Based on their understanding of MIDI and experiments in digital music, the Vasulkas were convinced that much of this degree of programmability was possible with the MIDI protocol; however, there was still an essential set of knowledge that they needed in order to exploit its power. Early on in the process of drafting a system design for *The Brotherhood*, it became clear to the Vasulkas that they needed outside expertise for the process of hacking and reprogramming MIDI into the other hardware-specific languages that were unique to each machine. They hired Russ Gritzo, the son of a former collaborator Ludwig Gritzo, who was then an electrical engineer at Los Alamos Labs. Russ had familiarity with the electrical concepts present in the Vasulkas' past work and his burgeoning knowledge of scripting languages (particularly C) made him a particularly strong asset for the Vasulkas as they created an actionable system design for *The Brotherhood* (figs. 7-8).

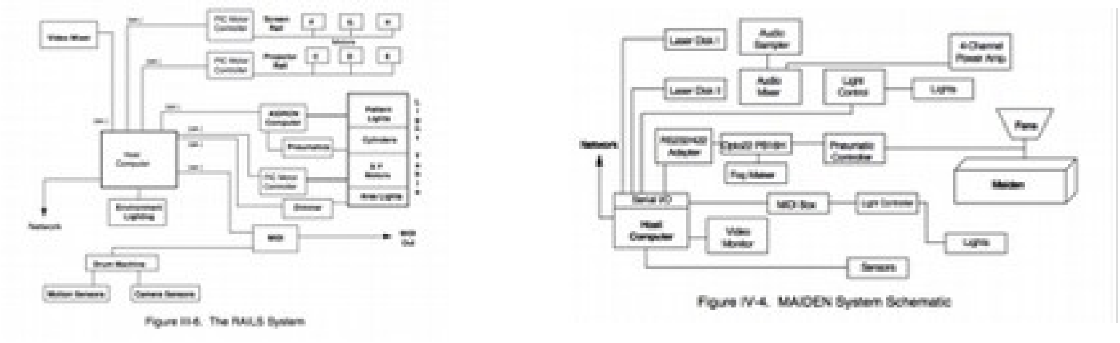


Fig. 7 & 8: System concept block diagrams for *Rails* and *The Maiden*, Russ Gritzo, 1998, c. The Vasulkas

After hearing about the Vasulkas' interest in organic human-machine interactivity and in the alternative use of MIDI, Gritzo knew that a system that utilized Interrupt Service Routine (ISR) would be essential to the configuration and functionality of the work, due to the many languages, stimuli, and moving parts. ISR was developed along with the introduction of the Intel 8086 Microprocessor in 1986 and it essentially aided a computer in two ways: in clearing its queue of user-generated impulses and in the actions of translating those impulses (inputs) into outputs (ECEn 425 2014). Defined another way, it was a means of facilitating communication between user commands and external devices, and allowed for efficient ways of restarting, continuing, or terminating that line of communication. The host computer is truly the brain of the artistic organism. The host computer needs to continuously receive data from sensor points, placing that data into a continuous queue of commands generated by the user and the corresponding input data, and to direct these messages out to many different outputs existing in many different Tables or parts of Tables. User-generated data is constant and fluid within each table, and in order to appear responsive, the system needs to maintain consistent communication with all of the external devices and to refresh input data regularly (BonaFide OS Development 2003). For example, without ISR it would have been difficult to program specific MIDI notes and events to an intended output, such as signaling to the RPT camera to move towards its target or to return to the home position in the absence of a command; without RPT the result is a disoriented system with lost, jumbled, or ignored commands.

In the late 1980s and into the early 90s, ISR could only be found in DOS (Disk Operating System), and could only be compiled through specific models of microprocessors, such as the 8086. The Intel

8086 was the first microprocessor chip designed with ISR capabilities that was implemented on a microcomputer (Edwards 2008) and was the first computer on which methods for *The Brotherhood* were tested. The 8086 was also one of the most widely adopted chips in PCs throughout the 80s and early 90s. Later OS models began to incorporate ISR and it was adopted more broadly. The incorporation of ISR into additional operating systems sparked a shift in *The Brotherhood* in 1992 from a DOS-based design to one that was Linux-based. This shift to Linux was also determined by an increased ability to build programs for communicating with peripheral devices in a real-time fashion, something that DOS increasingly lacked as Linux reached wider adoption amongst programmers (Gritzo 2013).

An essential example of how the programming language was compiled into machines language in *The Brotherhood* can be found in the use of ISR. As mentioned ISR was a means of communicating from user-generated input data out to hardware and facilitates constant flows of data to simultaneous destinations, ensuring that commands are not lost or distorted along the way. Given that DOS was the first operating system to use ISR, it was also one of the first operating systems to allow for direct communication from the computer terminal to the peripheral hardware. In computer science, the Basic Input Output System (BIOS), introduced along with the IBM PC in the early 1980s, was the first standard using firmware. Firmware is defined as a set of software programs that are exclusively designed for facilitating communication with external hardware via designated computer ports (Fisher 2014). BIOS is firmware that is directly programmed into the machine language of the microchips installed in the IBM PC family (in the case of *The Brotherhood*, the Intel 8086 microprocessor). The BIOS inherent to a computer can be accessed through source code. Essentially the BIOS acts as a compiler in its use of pre-programmed software packages, native to the 8086, in order to manipulate the machine language that sends commands out to the peripheral ports. With the 8086 and DOS a programmer could write commands and source code that could alter the structure of how a given peripheral device communicated with the computer, a process that is otherwise only enabled by binary machine language. With BIOS a programmer can build upon what an external device would natively understand, expanding on the kinds of messages it could receive and the ways in which it will respond to those messages. For example, many Tables in *The Brotherhood* incorporate laserdisc sequencers that call up specific excerpts of video in response to a user-generated impulse. BIOS allowed Gritzo to program a script that communicated directly to the laserdisc player, sending out information about which video sequence to call up in the laserdisc switcher’s machine language, which was stimulated through the serial connection port on the player and was expressed as serial data.

In order to more efficiently automate the process of designating commands to the BIOS and to achieve desired corresponding effects in external hardware, a programmer builds software programs called device drivers (“Actors” in the parlance of *The Brotherhood*). Device drivers are written to directly interface and “call out” to the BIOS routines of external hardware, initializing calls out to the hardware and returning a confirmation message when the connection is understood. Device drivers essentially configure a computer system and deliver specific messages to its designated peripherals or outputs. In order for configuration to occur, a system must be able to reference a standard set of instructions for compiling source code to machine language. As mentioned, the programming language C was initially chosen as the programming language for *The Brotherhood,* and this was in large part because of its flexible syntax structure and the ease with which it could communicate with BIOS and external devices. This was later replaced by a low-level programming language created by Russ Gritzo and Bruce Hamilton which largely depended on ASCII to contain MIDI data (expressed in hexadecimals) and delivered to specific ports via device drivers.

The software package for *The Brotherhood* is made up of more than 1000 unique files. At the time in which *The Brotherhood* was created, the use of BIOS and assembly languages (e.g. C) was a more user-friendly way of accessing and modifying a computer's hardware-interface and machine code although this practice was unique to the IBM PC and an era where that system was in wider use. Understanding the transition and development of technologies described above is an extremely vital component of our overall understanding of *The Brotherhood*. Additionally, understanding the system concept of the work helps in describing the sentiments behind its development, the desire to cast an entire system in a language that could contain interactive range (MIDI) and using this language could call out to other devices in ways that otherwise could not have been possible without these alterations to the relationship between the OS and the microprocessor. This "hacking" of the native system marks the work as not just an example of using an innovative means of obfuscating existing functionality but rather as a historical record of experimentations in computer science during a period of swift and epochal growth, aided for the first time by BIOS in the IBM PC. Initially an IBM PC with the 8086 microprocessor was used as the host terminal for *The Brotherhood*, namely for early installations of The Theatre of Hybrid Automata, the Table that, as noted above, was completed as a solo artwork. Later as the work developed to include several Tables, there was a need to efficiently configure all of the pieces together under one system in order to streamline the initializing process when the work was installed in a gallery. A more portable and customizable machine with faster processing time than an IBM PC was required; hence the move to a Toshiba 1200XE laptop with an external I/O (input/output) brainboard and integrated circuits which also used the Intel 8086 chip (Gritzo 1992).

In addition to ISR and BIOS, integrated circuits (ICs) were an essential component within the custom computational environment of *The Brotherhood.* ICs allowed for a boosting of data transmission speeds and multiplexing of signals, facilitating the one-to-many responses that are seen in many of the Tables in the work. ICs allowed for discrete control of signals that were being sent out largely to servo and stepper motor controllers, ensuring that these device drivers would be separately directed and timed so as not to crash the host terminal.

Given that the Vasulkas and Gritzo insisted upon pushing the confines of a singular computer system in order to achieve their goals for *The Brotherhood* demonstrates how they looked beyond the currently possible and designed their own machine for achieving these feats. It further shows how the commercial market had yet to develop and release computer systems that could adequately accommodate the work and the orchestra of machinery that *The Brotherhood* required. The Vasulkas and Gritzo designed their own system and thusly were responsible for innovating a totally unique environment. Of course, this further means that the work now lives in an environment that is perhaps irreplaceable.

In addition to ICs, *The Brotherhood* also required the use of the Opto Brainboard, a device for enabling follower/leader controls in unison with the host terminal. A leadermaster/follower device allows for unidirectional control across two computational systems. Opto was specifically used in controlling mechanical sculptures such as in The Maiden to activate solenoid (pressurized air) valve chambers that were directly wired to the brainboard. The host terminal would send commands specific to the Opto's programming language in loads, generally packets of commands that were intended to be directed to several ports on the electrical sculpture at once. On this subject, it is important to note that, above and beyond the complexity of the computer in this work, the external hardware is extensive (more than 70 pieces of unique external hardware). For example, elements such as the solenoid valves in The Maiden, laserdisc switchers in Friendly Fire, or the extremely unique RPT camera head in Automata mean that several proprietary languages were being used simultaneously, all of which needed to be sent out to hardware that could understand the machine language and repackage it into serial or analog information.

We have now explored three aspects of the technical configuration for *The Brotherhood*, which can be seen as evidence for how innovative approaches towards use of emerging technologies were greatly at play in its creation. We also see the difficulty in relating behaviors to the work's significant properties, most significantly seen in how the source code often relied on bugs in the code to create the feeling of machine autonomy. These significant properties are clearly seen through the highly customized structure of the work aided by the existing technologies at that time. First, understanding the use of MIDI and unpacking the signal down to logical, manipulatable clusters helps us to define the interactive formula of the piece as expressed through a singular language. Documenting the work without a rigorous understanding of MIDI, only paying attention to the literal causality between stimulus and response, would over-simplify the dialog between user stimulus and system response. It is also unlikely that another language would unite all the diverse pieces of hardware given that MIDI acted as the lingua franca for all the related hardware. Second, the use of data input routines that were present in operating systems at that time represented a cutting-edge approach to modifications to the interfaces of external hardware based on what was currently possible to achieve with programming. Finally, the use of external integrated circuits and brainboards to create a custom computer for centralized control within each Table fixes the piece in a particular era in a period where the speed and simultaneity of data transmission had reached a certain threshold. The corresponding source code which interacted with this custom computer is thus reliant on that technological environment. Serious challenges will be found if the source code was to be migrated to a more modern processor. All these customized features that were added to an off-the-shelf system are not likely to be maintained once the code is moved to a new system that functions in an entirely different way.

THE COLLECTION: DOCUMENTATION AND REMEMBRANCES

The software for *The Brotherhood* was last archived on one of the Vasulkas' PCs in 1998, contained in the form of a zip file which was copied to an optical disc in the early 2000s. This was the only known copy to have survived. Although a select number of system-level manuals were digitized in 2006 and stored on CD-Rs as recently as 2008 along with the executable files—those needed for enacting the piece—the collection has remained in a dormant state ever since 1998. At that time the piece may have seemed relatively self-contained and its functionality and configuration was still fresh in the memory of the Vasulkas and Gritzo. However, its stagnation has resulted in many essential pieces that comprise a full expression of the work falling through the cracks. My study of

The Brotherhood and reassembling its technical genealogy was at first unintentional. At the time, I was trying to locate papers which directly pertained to the Vasulkas' image processing tools as part of a triage to provide context to the videos that were soon to be digitized. I came across a multitude of technical papers that were unlabeled and stored in seemingly arbitrary locations. As I sifted through boxes of these composited, largely unrelated artifacts I began to pull out any technical documentation which seemed that they could serve the purpose of the "curriculum" that Woody and I had so often discussed. As I interviewed the Vasulkas about these documents, asking them to recall their purpose, random bits started to fit together but not yet cogently: RPT, MIDI, Gritzo, etc. In many cases we were not able to identify the documents or to understand their purpose. Nonetheless, as I collected the documents I began to see patterns emerging—also more explicitly labeled documents were found—and I created separate troves for different periods of the Vasulka's work. Eventually, I was able to create separate sets of documents for each unique Table within *The Brotherhood*.

Early on we agreed that Russ Gritzo's and Bruce Hamilton's accounts of the development of *The Brotherhood* were going to be essential to understanding the artwork by better organizing the documents, so we interviewed them in August 2013. Their accounts and the paper trail that they created during the development of *The Brotherhood* has been vital for two reasons. The oral history was essential to better understand not only what was significant to the Vasulkas and the impetus to create the work but also their assessment of these aims which manifested in his programming. Second, what emerged was that the Vasulkas and Russ Gritzo were forging a path and a computational design that was completely their own. Thus, if the work is to continue to function in the experiential realm, this unique computer environment must be continually stabilized in order for the source code to migrate effectively. Additionally, detailed documentation and an intuitive key for these documents and the relationship to one another must aid in the process of excavating what behaviors emerged from the code.

One particularly interesting aspect of Gritzo's account was his need to develop a workflow that was adaptive to Woody's somewhat extemporaneous decisions around what he wanted the system to do. Gritzo needed to respond to specific aesthetic and behavioral requirements through creative programming. Arriving upon what it meant to create an organic response required a great deal of trial and error on Gritzo's part; he scripted programs that gradually approached the results that Woody desired. His pursuit of seeking the desired outcome through an iterative process meant that the code was responding to the affectual requirements through a process of discovery within the confines of a given language (e.g. MIDI).

The question now emerges of how to best arrange all this knowledge around the artwork and its development. Proper arrangement and description will enable new forms of scholarship, and the way works of art such as *The Brotherhood* are documented in an archival setting will help to reveal the relationships between technologies and behaviors. *The Brotherhood* is more than an artwork; it is an example of innovation set in a particular time period. Within an archival setting an understanding of the innovative qualities of the work are better understood if there is a documentation model that facilitates this study. In this case, scholars can help to further draw these distinctions between behaviors and materials and, in relating this to broader concepts and other historical examples in computer science and engineering, can better explicate what emerged out of this affectual programming.

So how are we to imagine what this archival record would look like for a work of this nature? Who might our designated community be and in what we could they be folded into the process of creating further meaning out of this mass of information? For scholars in computer science, robotics, or interactive programming, a more rigorous descriptive framework for describing the relationship of the parts would create a breadcrumb trail, improving the discoverability of the collection based on topics, themes, or keywords that might pertain to their work, focusing on technological processes and not just the specific environment of the artwork. At the same time, the relationship that the individual technology has to the whole still needs to be evident to the researcher and should be reflected in how the collection is organized. The researcher would also benefit from an understanding of how the work's functionality set itself apart from what was technologically feasible at the time, focusing especially on the unique and innovative elements of the work – which are more likely to be the subjects for scholarly inquiry.

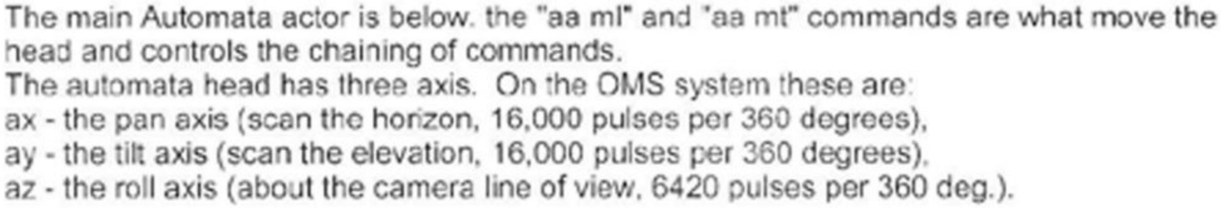
Besides benefiting our scholarly understanding of the *The Brotherhood*, building a curriculum or genealogy can also benefit a broader field of inquiry. *The Brotherhood* is a perfect example of an art work which uses a modified version of a programming language that, though widely adopted in its time, has changed greatly over time through subsequent versions. The highly customized configuration of the code further means that methods exploited through the programming language leave unexplained traces of what functionality (or potentially lack thereof) was allowed through its syntax. Identifying those examples, such as unique characters, commands, or sequences, could help to aggregate knowledge around the quirks of a programming language which often go undocumented. These pieces of evidence can contribute to a deeper understanding of not just the programming language but how it interfaces with the unique system of hardware and how it facilitated interactivity.

PROPOSING A MODEL

Before choosing a course of action for documenting a complex artwork, one must first understand how a work functions by creating an exhaustive inventory of the hardware and software. Essentially, before deciding what behaviors emerge from the hardware and software, one must first and foremost completely understand their interdependencies. While I have argued that an understanding of a work as complex as *The Brotherhood* needs to go much further than simple inventories in order to fully grasp the interoperability within this customized, hybrid environment, we will still need to create a basic structure for commencing with the classification and relational quality of the hardware or software items. In order to do this, one must go further than a simple listing of all the component parts (hardware, software, necessary peripherals), but also build other data fields which further illustrate how the various component parts communicate within a generic system (basic computing platforms, operating systems, etc.), as well as within the specific environment of the artwork (microprocessors, external timing devices/controllers, integrated circuits, etc.). The chain of communication of all these moving parts, in what direction the various messages are to be sent, and in what manner they are to be packaged, are perhaps the most essential characteristics that this inventory should express; in other words, how many diverse data formats and signals were cast through the singular MIDI protocol in *The Brotherhood*. The form of this inventory or abstract should be concise enough to clearly address the specific interoperability of the specific artwork but flexible enough so as to be relevant in an archival environment that could include other artworks in a collection, works which all have their own unique structures and dependencies. Ideally in creating this abstract possible conservation actions may reveal themselves, such as whether an obsolete MIDI serial convertor box could be replaced by more modern technology that also understands and packages data based on the MIDI protocol. However, for the purposes of *The Brotherhood* I am proposing that what is most important is framing this work within a significant historical context so as to better understand its innovative processes. Thus, a collection for the designated audience of media archaeologists should include an abstract that is descriptive of how the piece functioned but also that allows the researcher to connect to its significance of the development of computer science and interactive technology at large.

As mentioned earlier, the initial process of creating the technical genealogy for *The Brotherhood* was somewhat haphazard. Whilst digging through miscellaneous papers that might have assisted in facilitating the video triage (i.e. inventories of master tapes, technical papers for the image processing tools), many other documents emerged with unknown purposes or origins. Over the course of many interviews with the Vasulkas and eventually Russ Gritzo and Bruce Hamilton, we were able to separate certain papers based on their association with a given artwork. In the case of *The Brotherhood*, simply grouping papers in this way was not enough to truly help us understand how each component part worked within the overall configuration. After assigning unique identifiers to each user manual, schematic, or block diagram and scanning these documents, I began to pore over all the material, attempting to find patterns in the various commands being invoked, what data formats were being outputted, or what ports or hardware specifications were mentioned. Over time I was able to map out the relationships of the hardware based on these internal clues, and was able to capture both narrative and contextual information about each work in terms of its intended function as well as how it was adapted into the holistic system concept of *The Brotherhood*.

Before going any further in regards to the archaeology of the work, I must stress how essential the unique identifier is in terms of adding validity to the inventory as well as in creating my methodology and structure for archival arrangement and description. All of my conclusions from describing and contextualizing the mechanical guts of the work can be traced back to a document or a series of documents. Every item in the inventory points to a document for further research, and ideally the inventory serves as a guide for a media archaeologist in finding relevant interrelated documents. An item in the inventory, be it a script, database file, or piece of hardware, contains a listing of related items, information on its intended function in the piece, its role in the Input/Output scheme, as well as tags pertaining to the functionality, risks, or necessary commands to engage with the technology. Given that documents in the Vasulka collection are always used as evidence for drawing these conclusions, the unique identifiers can be used such that researchers can find source material to further research whatever piece of hardware, software, or programming method most pertains to their work. In locating a particular document and further researching the technical characteristics, a researcher will be able to cite *The Brotherhood* in their own work but will also be able to build upon our understanding of the nuances within the technology which in turn enabled the autonomous organism that is *The Brotherhood*. For example, perhaps a scholar is researching use of stepper motor controllers in the early 90s and discovered the Vasulka collection based on the explicit mention of this technology within the inventory. They would be able to research the specific devices and computational environment of *The Brotherhood* and add further annotations as to how the controller works which may either build upon the existing abstract or perhaps even correct or modify the information. The relationship to the document is the most essential component to all of this because, after all, we are trying to provide access to a designated community within Media Archaeology in keeping with Woody's desire for a "technical curriculum" around their work (fig. 9).



The main Automata actor is below. the "aa ml" and "aa mt" commands are what move the head and controls the chaining of commands.
The automata head has three axis. On the OMS system these are:
ax - the pan axis (scan the horizon, 16,000 pulses per 360 degrees),
ay - the tilt axis (scan the elevation, 16,000 pulses per 360 degrees),
az - the roll axis (about the camera line of view, 6420 pulses per 360 deg.).

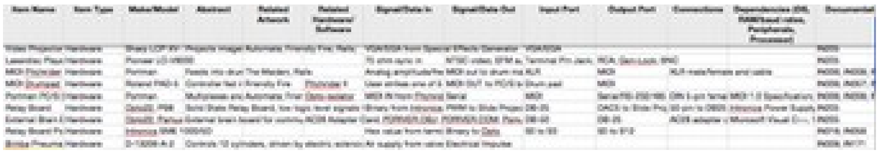
Fig. 9: Example of source code from a user manual on RPT (UID IN040), Russ Gritzo, 1998, c. The Vasulkas

After identifying the relationships between documents and pieces of hardware, it was time to explore the software more deeply. I first scoured all the documents and picked out any sections which noted specific data clusters or command prompts, providing evidentiary support as we examined the source code. After aggregating enough information indicating, for example, that the command "aa ml" is used to control elements of the RPT in Automata (as seen in the excerpt above), I could safely say that any scripts using the "aa ml" command is operating the mechanical positioning of the RPT head. In consolidating such examples of clusters of code into a singular document or "key," I was able refer to the originating documents by their unique identifiers and place this identifier in the "related document" field in the software inventory. A majority of the files in the package are shell scripts and database files which are interrelated. The database files store the essential bits of data for repackaging bitstreams and the shell scripts enact this process, detecting when data is coming from a designated stimulus channel and determining what device driver will deliver that information to the hardware for generating a response. Within these two file types that make up most of the software package, there was more than enough information to answer some of our burning questions: What are the inputs/outputs? At what point do they occur? How is the hardware configured? How can the source code be linked to specific hardware by virtue of the hardware's technical specifications? What is the native/ideal environment (OS, dedicated RAM, transmission speeds, etc.)? Finally, we could also begin to concretize what makes this technology unique or significant and represent this in the documentation in a way that is searchable through keywords.

STRUCTURING THE INVENTORY

In creating the abstraction of all necessary hardware and software, I also outlined the direction of the signals and points at which the messages become repackaged, translated, or scripted out to designated hardware through BIOS control. After completing the inventory and analyzing the component parts through their related documentation, I arrived upon a controlled vocabulary for describing what is happening to the data streams or signals at certain points so that stimuli are processed and sent out to their designated outcomes. This was a useful exercise in not just identifying some elemental function of the scripts but also to create a clearer visual for how the sensor data travels across the system in each Table. My hope is that this inventory will not only be a guide to accessing the materials for scholarly research in Media Archaeology and to understanding the interrelated nature of the elements within *The Brotherhood*, but will also be open-ended enough to solicit further participation and data gathering from the designated community. To this end, data related to specific commands in the scripts are gathered despite a lack of authoritative knowledge on their significance. The inventory is both a map for understanding the elements and a skeleton upon which researchers can aggregate their research so as to further crystallize our collective understanding of this complex work.

Under the Hardware table of the inventory/abstraction, there are a number of fields that are relatively straightforward in their approach (fig. 10).



Item Name	Item Type	Make/Model	Abstract	Related Artwork	Related Hardware/ Software	Signal/Data In	Signal/Data Out	Input Port	Output Port	Connections	Dependencies (OS, RAM/baud rates, Peripherals, Processor)	Documentat
Video Projector	Hardware	Sharp LCP XV-	Projects images	Automatic Friendly Fire; Rails		VGA/SGA from Special	Effects Generator	VGA/SGA				IN005
Laserdisc Playe	Hardware	Pioneer LD-V8000				75 ohm sync in	NTSC video, EFM a	Terminal Pin Jack	RCA, Gen-Lock, BNC			IN005
MIDI Pitchrider	Hardware	Portman	Feeds into drum	The Maiden; Rails		Analog amplitude/fre	MIDI out to drum ma	XLR	MIDI	XLR male/female and cable		IN006; IN008;
MIDI Drumpad	Hardware	Roland PAD-8	Controller fed i	Friendly Fire	Pitchrider I	User strikes one of 8	MIDI OUT to PC/S b	Drum pad	MIDI			IN006; IN007;
Portman PC/S	Hardware	Portman	Multiplexes and	Automatic Fire!	Opto isolator	MIDI IN from Pitchrid	Serial	MIDI	Serial/RS-232/485	DIN 5-pin femal	MIDI 1.0 Specification	IN006; IN008;
Relay Board	Hardware	Opto22, PB8	Solid State Relay Board, low logic level signals			Binary from Intronics	PWM to Slide Projec	DB-25	DACS to Slide Proj	50 pin to DB25	Intronics Power Supply	IN005
External Brain E	Hardware	Opto22, Parlux	External brain board for controlling			AC28 Adapter Card, PORWER.DBJ	PORWER.DDM, Parlu	DB-25	DB-25	AC28 adapter c	Microsoft Visual C++,	IN005
Relay Board Po	Hardware	Intronics SMB 1000/5D				Hex value from termi	Binary to Opto	00 to 03	00 to 813			IN016; IN008
Bimba Pneuma	Hardware	D-13206-A-2	Controls 10 cylinders, driven by electric solenoid			Air supply from valve	Electrical Impulse					IN006; IN171

Fig. 10: Example of Hardware Inventory for *The Brotherhood*

Abstract contains information about the narrative of the piece of equipment, namely its intended purpose and its use within the context of *The Brotherhood*, and Related Document contains the unique identifier for the document that points the researcher to additional information. Dependencies contains any information about the technical specifications related to necessary microprocessors, external power, built-in software, etc. However, this column serves a different purpose as Related Hardware/Software which is intended to contain data that connects to other pieces within the inventory that are specific to *The Brotherhood* system concept (rather than potentially unforeseen external dependencies). The Component Visible column contains data that is perhaps most useful to a conservator is a museum, making it clear whether a certain component is part is visible in installation; that is, the component is part of the aesthetic experience of the installed work and also provides further justification for the item's necessary inclusion in future iterations of the work.

This act of diagramming the various inputs and outputs marks the first departure of this methodology from the above-discussed ways of documenting variable media works, since data is separated out to describe both input and outputs (whereas most models include this data in one column). I find it essential within a singular piece of hardware to diagram the data that is coming in versus the data that is being sent out so as to better demonstrate what is being done to the signals and at what point, and how every unique piece communicates with another. Thus, the four separate columns are Input Signal/Data, Output Signal/Data, Input Port, and Output Port. Within these columns I note not just the type of data or signal but from where it has been sent, further diagramming this "baton-passing" that occurs ubiquitously throughout the artwork. For example, under "Pneutronics Card," the piece of hardware that is designed for controlling the air valves for the sculpture in The Maiden, Input Signals are marked as "On/Off binary data from Opto Board." Under Related Hardware/Software we also see Opto Board. Thus, we understand not just what role the Pneutronics Card serves, but where it lies in the great chain of command for the system configuration. A final necessary data column is Connections/Cables, which predominately shows how the hardware ports connect to external devices via a port and cable.

Input/Output Signal (Analog)
IR (Infrared Range)
Volts, Direct Current (DC)
RPM
Pulse-Width Modulation
NTSC Video
RF
Pressurized Gas
Audio
NPN/PNP transistors
VCC (Integrated Circuit Power Supply)
Light-Emitting Diodes (LED)

Input/Output Signal (Digital)
MIDI
Serial Data (Hex) / Bauds / FIFO (data buffer algorithm)
Latch Signal Pulse/Flip-Flop
Machine Binary
ASCII
PCM/WAV

Input/Output Ports
Infrared Detectors
LS10R/LS10E (Phototransistors)
MIDI
Switches, SW1-5
LED pins, TP1-TP4
Terminal Pin Jack
Servo to Stepper: SM/PM, ST/PT
5-pin adapter
VGA/EGA
DB-25/50
9-pin RS-232C
XLR (3 and 5 pin)
20/26 pin converter
60 pin daughter board
34 Conductor Ribbon
1/4” Audio Jack
BNC
RCA
COM1-COM3 (Pigtails)
PFC

14-pin connector
CTC DIN connector
JB5/JB9 to CT (CPU extension)
Dual SCR
SCSI
Integrated Circuits
DM15M (gyroscope)
RX/TX pin
37-pin connector (DB37)
J2/J3 memory port
Serial S0-S12
40-pin interconnector
OAC Opto ports

For Software, we also have a number of straightforward data columns, which I like to refer to as "monolithic" (figs. 11-12).

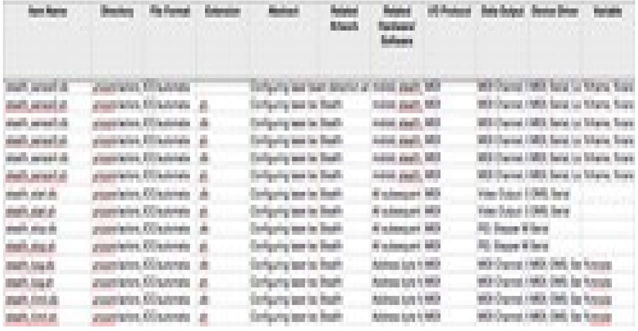


Fig. 11 & Fig. 12: Example of Software Inventory for *The Brotherhood*

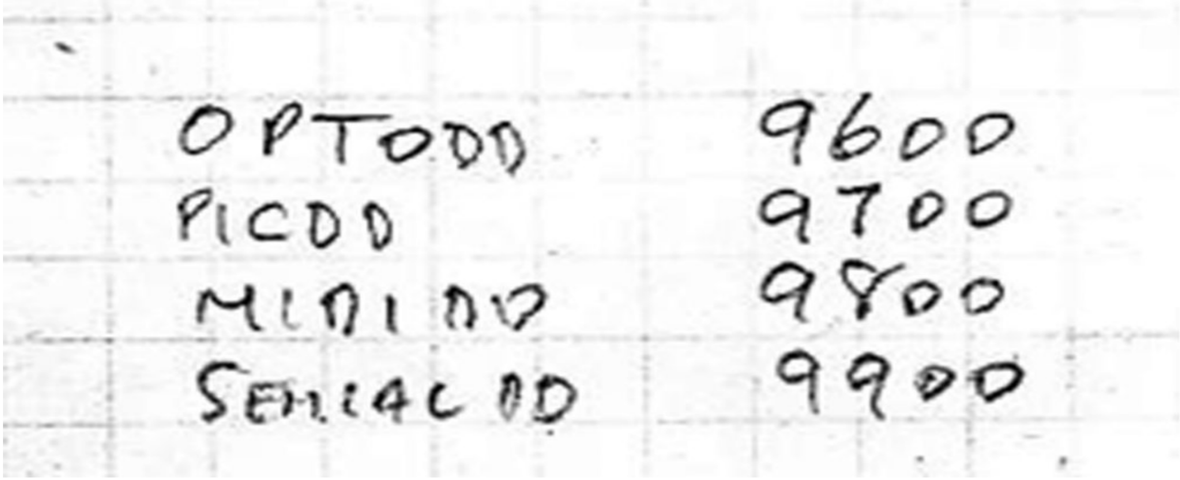
By this, I mean that this data pertains to the software package at large rather than discrete parts therein, mostly out of respect for the fact that other artworks and data packages will enter the collection which merit their own unique system-level forms of description. First, we have the I/O Protocol which describes the singular language under which all hardware and software are cast, in this case MIDI. The Programming Language indicates the scripting language which communicates with the OS which is further compiled into machine language, in this case a custom language (with roots based in C). Similarly, this system relies heavily on the use of source code to access the machine's native means of controlling external hardware, so information is collected about the Firmware, in this case PC BIOS. Finally, we will contain data about the Operating System, GNU/Linux. As was found in Hardware, we will also collect data related to the narrative or Abstract, as well as the Related Documentation for further research and analysis.

Here is where data collecting starts to get really interesting in terms of how to document *The Brotherhood* so as to pertain to wider conversations and points of research within the Digital Humanities and Media Archaeology. For the purposes of *The Brotherhood*, the most noteworthy features include forms of data repackaging and control of external hardware to create an interactive environment. After poring over the source code and related documentation, I was able to create an exhaustive list which speaks to every type of data that is being interpreted as stimulus data or repackaged so as to be useful to external hardware as a response. We can describe *The Brotherhood* in broad strokes based on these twelve functions:

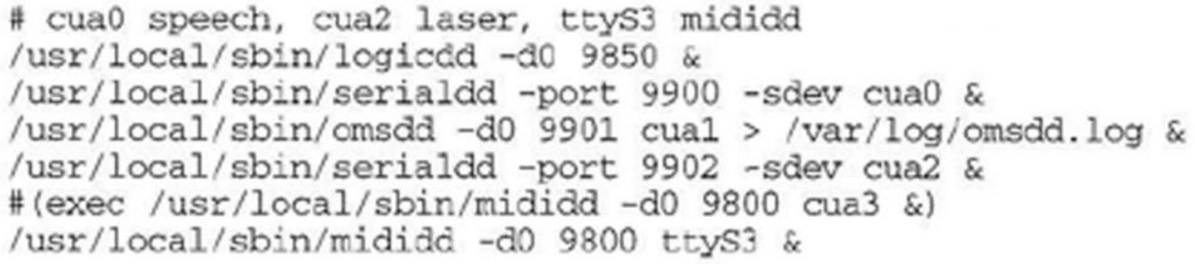
1. PID (Process Identifier)
2. Solenoid Valve
3. Text
4. Hex: MIDI Velocity/Pitch
5. Variable (setting acceptable interactive range)
6. Stepper Motor
7. PIC-Servo Motor
8. MIDI Light Control
9. Laserdisc Sequence
10. MIDI note (audible)
11. Live Camera Feed
12. MIDI Channel

By outlining these twelve functions, we will provide the researcher with a skeleton for the project at large such that specific hardware and software elements can more intuitively be placed in context with one another in terms of function. This will also help us in charting a a system of “classes,” particularly in the case of the software for which there are close to 3,000 files. In grouping this vast directory into data types, the researcher will be able to navigate the elements based on function, which ideally will suit their specific research needs, even if they had never heard of the Vasulkas, but rather are studying the historical use of the relevant technologies.

Of course, all of these bitstream reformatting processes are in service of specific outcomes via external hardware, all processes that are enabled by the BIOS and through device drivers that have been written for communicating with the specific hardware. This led me to also create an exhaustive list of Device Drivers that were created for the piece (also referred to as Actors), informing not just a means of classifying clusters of source code based on the purpose, but also revealing very direct dependencies amongst the shell scripts and database files which simply would not run in absence of the device drivers. One absolutely essential document that I found in my triage was OM037, which provided evidence as seen in figs. 13-14.



OPTODD	9600
PICDD	9700
MIDI DD	9800
SERIAL DD	9900



```
# cua0 speech, cua2 laser, ttyS3 mididd
/usr/local/sbin/logicdd -d0 9850 &
/usr/local/sbin/serialdd -port 9900 -sdev cua0 &
/usr/local/sbin/omsdd -d0 9901 cua1 > /var/log/omsdd.log &
/usr/local/sbin/serialdd -port 9902 -sdev cua2 &
#(exec /usr/local/sbin/mididd -d0 9800 cua3 &)
/usr/local/sbin/mididd -d0 9800 ttyS3 &
```

Fig. 13 and 14: Examples of Device Driver baud channels (UID OM037), Russ Gritzo, c. The Vasulkas

One other document indicated that 9600 is the native baud rate for the MIDI protocol, bauds being defined as a transmission speed for clusters of data or discrete logical packets rather than individual bits. In other words, the baud rate is much like bps or bits per second but rather indicates the number of packets which contain logical data (e.g. a “note on” message in MIDI) that can be sent at a given rate on a given channel, specifically across a modem (Brainbell 2014). After reviewing all the different baud rates that were in use throughout the software in *The Brotherhood* and referring to documents that made mention of specific device drivers that were developed, I came upon a definite list: Opto, PIC-Servo, MIDI, Serial, Logic, and OMS (Stepper). Interestingly enough, each device driver was also designated at its own unique baud rate, developed as a means of allowing for the drivers to run at slightly different rates (albeit indistinguishable by human perception) on different channels so as to avoid data jamming. In comparing the designated baud rates for the device drivers (evidenced in the shell scripts exclusively by the designated baud rate e.g. 9800 for MIDI) you could also see a direct relationship with the hardware-specific command being invoked. Again, let’s use the example from “automata_midi3.sh”:

```
#10.5.1.12 9901 :aa vl350,350,350;
#10.5.1.12 9902 : 60 sp
if %mvel -eq 6
     {
     10.5.1.12 9901 :aa ml12000,0,0; gd id
     10.5.1.12 9902 : 15070 se  13986 mr\r
     10.5.1.12 9901 :mn 90 7f 07
```

Here we see use of a channel with baud rate 9901 (OMS Stepper) which designates a velocity for the stepper motor of the RPT head. We also see a channel with baud rate 9902 (Serial) which corresponds to a Laserdisc sequence. Laserdisc players are connected to the host terminal by means of a serial port, so this confirms the correspondence of this baud channel with a serial-based device.

For the purposes of the inventory, Device Drivers are given their own data column given how central they are to the Input/Output configuration and interactivity of the artwork. Any specific mention of device drivers within the software by virtue of the baud channels is tagged at the file-object level. The Device Drivers not only indicate how the file functions within the software, but it also broadly expresses what types of technology are being enacted through the software. For example, a researcher of servo motors could be directly connected with the shell scripts that were designed for communicating with these electrical components and would also be able to read more about how it functions in the overall piece by virtue of the Related Document field. Not only that, but it also makes note of the internal dependencies of the overall file directory, noting which programs rely on one another which will be extremely vital to retain as the work receives continuous digital preservation actions within a library's digital repository.

So far, I have diagrammed an inventory for hardware and software elements in an artistic work which aid in the process of historicizing the work from a technological standpoint for use within a research library context. This may also aid in understanding the behaviors as we delve more deeply into the unique configuration of the hardware and software which could further be used in educational environments. Discoveries from these educational pursuits could further aid conservators in both stabilizing the work and perhaps faithfully reinterpreting the work in a reinstallation. However, I do not claim to have exhaustively documented the work nor do I believe that we have all the answers in order to achieve our hazy dream of one day re-witnessing this monumental work. Indeed, much of this process involves pointing to further documents which themselves require much more digesting in order to fully crystallize our understanding of *The Brotherhood*. Scholars are the missing piece and this structure will hopefully tantalize them such that they will continue to lend their services.

PUTTING IT ALL TOGETHER

Throughout their immense careers as artists and pioneers, the Vasulkas have created a body of work that is an invaluable record of artistic and technological innovation. They pushed the realm of the possible and delved deeply within their tools to imagine new possibilities for communicating with and visualizing the language of the machine. *The Brotherhood* was a scintillating example of how a multitude of hardware could be cast together using a singular language, bending their intended purposes to reach a common goal. It is somewhat ironic, then, that highly innovative works which merit a permanent place in the history of art and technology are somewhat resistant to the archival record by virtue of their complexity and precise interdependencies which lie outside of existing knowledge and available historical reference points. We are immensely fortunate that the Vasulkas are so generous with their intellectual property and willing to allow experimental methods for processing and organizing their material so as to better suit the impulses of researchers. We are also fortunate to have Russ Gritzo's and Bruce Hamilton's tireless work in developing the piece and documenting the process, allowing for the possibility of organizing a collection for media archaeologists in the first place.

In examining different methods of access, we see how conservators and archivists in both museums and libraries devote rigorous attention to the relationship of behaviors and materials in time-based media works, understanding that historical context provides an invaluable perspective on the intent of the creator and the social conditions of the time upon which the work comments. Though a library is an unusual home for interactive installation art work, the Vasulkas' desire for an enduring record of their technological innovations offers an opportunity to imagine how existing methods for describing and stabilizing a work in a museum setting can be applied to methods for providing access to scholarly records. By focusing on the innovative nature of the work and how it relates to other developments in computer science and technology at that time, a technological genealogy becomes important in mapping the work's visionary methods: translation of a diverse number of machine languages into one protocol (MIDI), its adept means of controlling an orchestra of external hardware, and the ways in which it pushed beyond the conventional limits of computational power and control. By relating this system concept to a web of documents and outside resources, we can help facilitate study of the materials and leave room for scholarly input within the record itself, further extrapolating on its complexities and building upon our collective knowledge.

In the future I hope that *The Brotherhood* will pique the interests of scholars in media archaeology. I also hope that my genealogy will help researchers and students to navigate the design of the piece such that the source code is not only analyzed but also manipulated through use of legacy equipment and external devices, both for the purposes of experimentation in order to understand the relationship of hardware and software as well as potential reinstallation in a gallery. Indeed, the potential for the use of the Vasulka collection in continuing education rather than it existing as stoic artifacts is the reason the Vasulkas choose a University Library as the home for their collection in the first place. Placing this work in an archival context and creating a map that outlines its functionality makes it much more likely that the heart of *the Brotherhood*—its revolutionary nature and the artists' interests in complex electrical/computation languages—remains visible, allowing users to pay rigorous attention to machine behaviors. In proposing new interrelated data fields and mapping documents to specific pieces of hardware and software, I have created a skeleton upon which further research and knowledge will crystallize. I hope that I have fully demonstrated how highly complex artworks of a technologically-interactive nature merit more thorough attention in the archival processing stage to provide historical context. In connecting these objects to users and audiences of media archaeologists and scholars we may also envision ways to use the objects in both experiential and educational environments.

REFERENCES

Bonin, V. 2003a. "Steina and Woody Vasulka fonds." La fondation Daniel Langlois pour l'art, la science et la technologie. www.fondation-langlois.org/html/e/page.php?NumPage=422 (accessed: 05/01/14).

Bonin, V. 2003b. VCS3 (The Putney). La fondation Daniel Langlois pour l'art, la science et la technologie. www.fondation-langlois.org/html/e/page.php?NumPage=446 (accessed: 05/01/14).

Schoenherr, S. 2005. The History of Magnetic Recording. University of San Diego. www.aes.org/aeshc/docs/recording.technology.history/magnetic4.html (accessed: 05/01/14).

Manning, P. 1993. *Electronic and Computer Music.* Clarendon Press: Oxford, New York.

Planet of Tune. 2013. Index of Sequences. http://www.planetoftunes.com/sequence/se_media/message.gif (accessed: 12/13/13; no longer accessible).

ECEn 425 Online Resources. Updated August 29, 2018. The 8086 Interrupt Mechanism. http://ece425web.groups.et.byu.net/stable/labs/InterruptMechanism.html (accessed: 05/01/14).

BonaFide OS Development "Interrupts, Exceptions, and IDTs: Part 1 – Interrupts, ISRs," 2003. http://www.osdever.net/tutorials/interrupts.1.php (accessed: 12/13/13).

Edwards, B. 2008. Birth of a Standard: The Intel 8086 Microprocessor. http://www.pcworld.com/article/146957/article.html (accessed 12/13/13).

Gritzo, R. 2013. Personal communication.

Fisher T. 2014. Firmware. PC Support. http://pcsupport.about.com/od/termsf/g/firmware.htm (accessed: 05/01/14).

Gritzo, R. 1992. ExhFour. http://www.vasulka.org/archive/ExhFOUR/Brno/brno.pdf (accessed 12/13/13).

Brainbell. 2014. Modem Speeds. http://www.brainbell.com/tutors/A+/Hardware/Modem_Speeds.htm (accessed: 05/01/14)

Joey Heinen
Digital Preservation Manager
Los Angeles County Museum of Art (LACMA)F