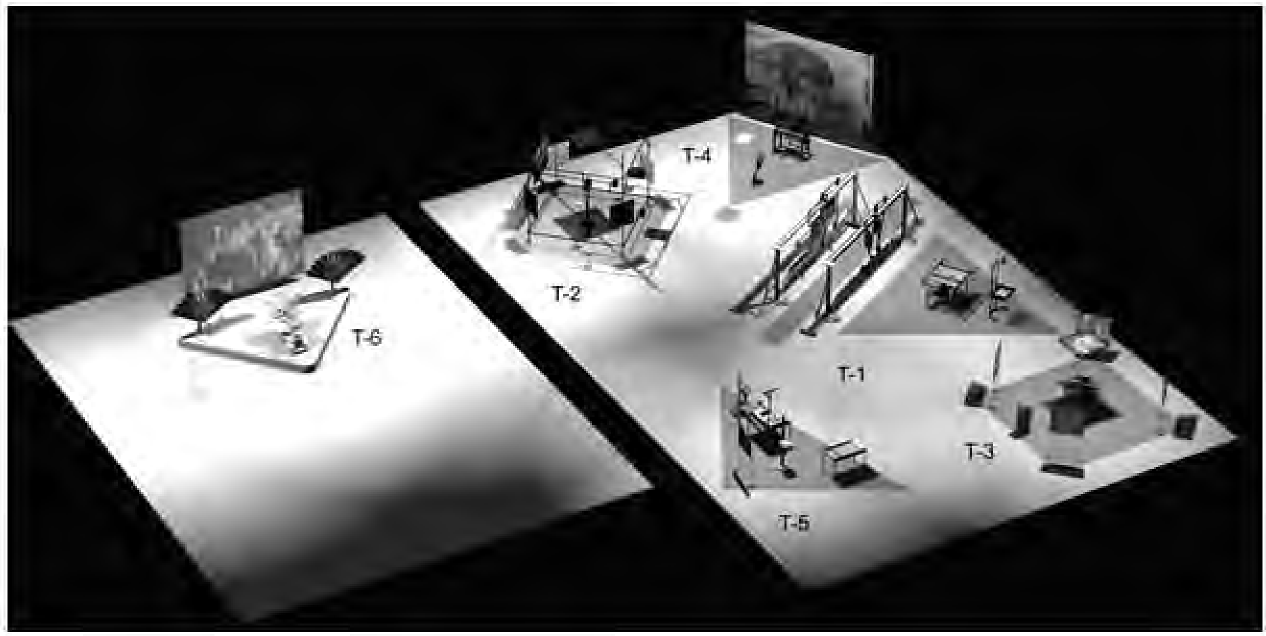
Archiving *The Brotherhood:*

Proposing a Technical Genealogy for Time-Based Works



Dimensional Representation of *The Brotherhood,* *http://vasulka.org/Woody/Brotherhood/Brotherhood.html*

Thesis Project in completion of a Master of Arts Degree in

Moving Image Archiving and Preservation (MIAP),

New York University, 2014

Joey Heinen

Executive Summary

The following case study is a proposal for the creation of a "technical genealogy" for time-based artworks, specifically focusing on works which use a multitude of hardware and software in a highly-customized environment. *The Brotherhood*, a six-part installation series by Woody and Steina Vasulka, is an example of an artwork where a thorough understanding of the employed technologies which mediate the work and the intended user experience is just as if not more of an essential component to the artwork than the experience alone. Many models currently exist for defining the relationship between materials and behaviors in interactive and technological artworks so as to inform more pragmatic approaches towards ongoing care and stewardship for the artwork, which more often than not is guided by an interview with the artists. In this particular case study the artists are donating their collection to the Brakhage Center at the University of Colorado - Boulder University Library whose mission is to study and historicize media art and technology. 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.

This case study is a comprehensive analysis of the artwork - its development, its functional challenges, and its key principles for ongoing care - as well as a guide for reading the archival package. This package includes an inventory of the technical documents in the

collection (notes, schematics, user manuals, etc.), the hardware (exhaustive list from past installations), the software elements (package from the work's final instantiation in 1998), and a grouping of digitized documents from the collection which are referenced throughout the inventory as places for further research and inquiry. All of these elements inform the model and are best read as a group, though my following case study should fully articulate how to effectively document and contextualize a technologically complex work so as to continually educate and inspire future researchers and audiences.

Introduction

The oft-cited expression in archiving states "preservation without access is pointless."[1] The concept of "access" is conventionally applied to an object that is rendered and delivered to the user such that it can be witnessed and experienced. An object that cannot be continually accessed thusly fails to perform its function and may as well not exist. In conventional cases these objects are independent works such as texts or photographs, needing no other context besides perhaps some historical background, provenance of the object within the collection, or in the case of digital work perhaps some stipulation of the preferred software for how to view or experience the work. In the case of moving image or dynamic media, access is usually provided via a derivative or copy, in recent years more increasingly in digital form. In the case of complex media art, specifically custom-designed software and interactive works, the notion of access takes on a unique new set of definitions given media art's complexities and interdependencies. Most visual artists and art exhibitors would likely say that the work is the experience,

---

[1] The Committee for Film Preservation and Public Access, "Preservation Without Access Is Pointless," 1993. http://www.loc.gov/film/pdfs/fcmtefilmprespubaccess.pdf, last accessed 12/13/13

particularly in time-based works in which a whole multitude of effects - moving image, sound, lighting, temporality, interactivity, chance, etc. - inform the “essence” of the work. How is the piece to be continually accessed, then, when a work requires a specific space, when software or hardware obsolesce and must be replaced by more efficient and user-friendly technology, or when works are uniquely situated within their required platforms and specifications (e.g. dedicated RAM, clock speeds, processors, and so on)?

For conservators and archivists, the nature of an original object is a thorny issue - is the object the ephemeral or aesthetic experience of the work, or is it the physical properties and technologies which originally enabled or mediated the work? While the experiential and material elements of a work are co-dependent, defining each component and its function creates a polemic for describing our intent for preserving the work - do we wish to more thoroughly understand each component’s function as records for scholarly analysis or for more informed conservation actions? Furthermore, is the object understood to be the culmination of all the parts or is it not also the process and documentation that describes its functionality and development? The conservator and the archivist have similar approaches towards relating contextual information (technical notes and manuals, documentation of previous iterations, added historical context) back to the central work and paying equal attention to both physical objects and their behavioral qualities for preservation. Of course, the physical and the behavioral properties diverge over time as what were once considered new technologies obsolesce, giving way to more widely adopted and supported technologies with the behaviors encapsulated within new forms and languages. This passage of time in and of itself implies that the historical context within which the work was created is already gone and the work’s technologies irreversibly shift from contemporary to artifactual. While the archivist (and conservator when one exists) may pay the same due attention to physical properties and behaviors, their work will be undertaken using slightly different lenses. The archivist and conservator in the context of a library will strive to preserve the historical evidence of a work, also preserving or stabilizing its physical objects as a means of adding value to that history and providing access to the native environment of that work to the best of their abilities.[2] The conservator in the context of a museum or exhibition space will also note the historical evidence and functionality of the work's original instantiation so as to inform a future installation that is as faithful to the original as possible. This process may involve substitution or recreation of certain obsolete components though the conservator will never alter the work in a way that is irreversible but rather maintain additional instantiations across time.[3]

Naturally many of the decisions around how access is provided - evidentiary or experiential - are determined by the collecting environment. Furthermore, decisions on how to provide historical context and its impact on the authenticity of the reinstallation or authentic representation to the researcher are the responsibility of the curator (both museums and libraries). While the conservator in an art museum will maintain certain contextual documents about an original so as to inform conservation actions, the archivist in a research library might argue that it is just as important to focus on these contextual documents as materials for public access. In the case of an especially complex, interactive work that employs innovative technologies, relating the artwork and its technical specifications can historicize the technologically "deterministic"

---

[2] Society of American Archivists, "SAA Core Values Statement and Code of Ethics." May, 2011. http://www2.archivists.org/statements/saa-core-values-statement-and-code-of-ethics, last accessed May, 2014.

[3] American Institute of Conservation, "Code of Ethics and Guidelines for Practice." August, 1994. http://www.nps.gov/training/tel/Guides/HPS1022_AIC_Code_of_Ethics.pdf%20, last accessed May, 2014.

moment of the work's inception. By this I mean that art which pushed the boundaries of what was possible at a particular time can help explain what innovation looked like in that period; for example, a work that involved programming with spatial sensors before there were tools such as Max MSP[4] (designed for interpreting sensor data) reveals the origins and purposes of those very tools and relates them to broader developments in computer science and engineering occurring at the time of creation. Often surprising parallels between the worlds of arts and sciences emerge from these separate histories. The archival record for legacy technology and its manifestations through art objects may unlock some secrets to the past which are otherwise cast aside in light of emerging forms. In order to fully understand the behaviors of these technologies and how an artwork is positioned as especially innovative, documents that describe how the piece was designed and implemented must be made accessible so they can be referenced by scholars.

Perhaps a more important point to consider is the circumstances which dictate how and a when a certain work is to be conserved in relation to the other holdings in a collection. Time-based media collections have existed for several decades though attention to the unique relationship of behaviors and physical properties as pertains to a continuously "authentic" representation of a time-based artwork is comparatively new. These methodologies maintain and contextualize the historical record and the physical carriers largely in service of the experiential rather than evidential record of the original work. Museums have proposed many successful models for informing how a work could be continually installed, though useful contextual elements such as early notes from the beta-testing period or early drafts should also be presented along with the work so as to be used in the scholarly analysis of media history. Furthermore,

---

[4] Cycling 74, Max MSP. http://cycling74.com/products/max/

documentation often is not arranged and abstracted so as to exploit a work's inherent complexities as a boon to the work's inclusion in a broader history of innovative technology, a genealogy of interactive computation. Rather, a highly complex work will likely remain untouched in the museum's stacks due to the heroic efforts required to rehabilitate it, becoming more and more obscure as time passes and technologies obsolesce. If a work is as part of an archival collection along with disorganized papers with obfuscated technical information and custom-built electrical equipment it will typically not be rehabilitated as the cost and labor will likely be prohibitive to the work. On the other hand, librarians and archivists are aware that there are scholarly communities in emerging disciplines within the Digital Humanities, such as Media Archaeology, and that a rigorous approach towards the history of legacy technologies and their usage is growing. Time-based media works of a complex nature rarely find their way to library holdings due to the conventional notions of the archival object (i.e. evidentiary in archives, experiential in museums) and how to deliver it, but the proliferation of digital media means that libraries are increasingly becoming ideal environments for historicizing collections with mixed notions of access. The current shortcomings of each realm – libraries/archives and museums – can each be satisfied by the other by comparing their two methods for historicizing a work and how it is to be experienced by a public. Scholars and researchers in an archival setting can position a work as significant by virtue of its innovative technologies. A museum may duly note the attention the work is receiving and may be able to leverage the time and resources to bring a work back to experiential spaces through the faithful rehabilitation/reinterpretation of the electrical and technical components. As will be demonstrated in the following artistic case study, the quandary of understanding the behaviors and interdependencies of hardware and software in a complex artwork is made even more complex when the employed processes of the work predates what was considered technologically feasible at that time. For works which demand an intense scholarly study and a deep understanding of the materials, how can libraries and museums work together to preserve a variable, technologically-dependent work for historical posterity and perhaps also for reinstallation?

The following case study focuses on new approaches towards documenting and historicizing complex interactive artworks, particularly those which largely exploit the use of innovative technologies within emerging areas of study in computer science. I will first introduce the artists, Steina and Woody Vasulka, and the artistic intent and practice which serves as the nucleus of the work. I will then introduce the artwork, describing its basic configuration and the interactive properties found therein. Following this diagram, I will detail each of the significant technological and computational components of the work, drawing out their relationship to the intended behaviors when appropriate. From there I will identify the key challenges that face this work from the perspective of conventional archival and conservation praxis. Consequently, by turning our attention towards existing methodologies in museums and libraries, I will expose some methodological gaps in processing time-based collections and argue that it is necessary that they be filled for artworks that have both experiential and scholarly value within a library. This complete analysis will then bring us to a new cross-institutional model for organizing and pointing to documentation of hardware and software elements, and relating these elements using a method that best diagrams an interactive artwork's innovative and historical nature.

The Vasulkas - Artistic Practice in Context

The polemics of scholarly and experiential access can be seen quite distinctly when embarking on conservation plans for the work of artists Steina and Woody Vasulka. 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 amount of 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 taking part in artistic residencies and exhibitions, they taught at the Center for Media Study, part of the State University of New York at Buffalo (1974-1980); organized an exhibition on live image processing tools and published one of the first comprehensive histories on this subject (Eigenwelt der Apparate Welt at Ars Electronica, 1992); and organized workshop series on the mechanics of their more complex works (Techne & Eros, 1998-1999).[5] They also published and contributed to a multitude of essays on technology and philosophy throughout their entire artistic careers. Lastly, early in their artistic practice they began to archive their own life and work, retaining outtakes, studies, sketches, and research material in addition to simply 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 – media obsolescence being one of them. My work was intended to facilitate a

[5] Vincent Bonin, "Steina and Woody Vasulka fonds." le fondation Daniel Langlois pour l'art, la science et la technologie, 2003. http://www.fondation-langlois.org/html/e/page.php?NumPage=422 last accessed May, 2014.

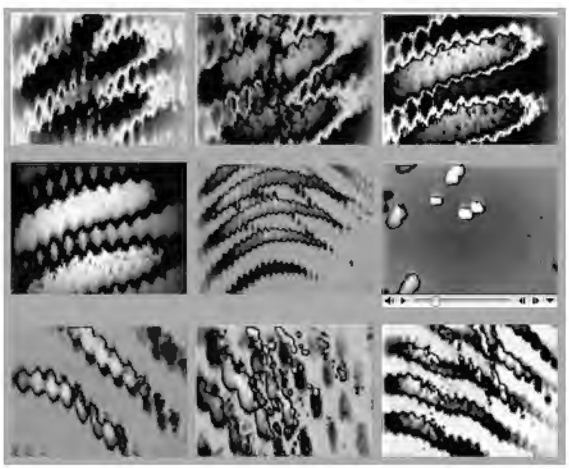
donation of the Vasulkas' archive to the University of Colorado Boulder (UC) that, as noted above, will eventually own and provide long-term access to their entire artistic and personal collection.

In my 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 which 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 work are just as important, if not more important, than the final works themselves. Another vital aspect of the Vasulkas artistic process beyond their finished products is their relationship with their technological tools and the potential that the tools allowed 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.

Steina and Woody 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. In fact, the nucleus for their iconic phase of utilizing the Rutt-Etra Scan Processor,[6] an image-processing device that re-scanned the lines of video on a television monitor and allowed for deflection of the new scan lines to create a topographic video effect, can be traced back to

[6] Woody Vasulka, "Rutt/Etra Scan Processor." *Eigenwelt der Apparate Welt*, Ars Electronica, 1992. http://www.vasulka.org/archive/eigenwelt/pdf/136-139.pdf, accessed May 2014.

their use of analog audio equipment. The use of the Putney (a.k.a EMS VCS3)[7] 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. 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.



Screen captures from Steina Vasulka *Descends* (1972) (left) and *Violin Power* (1978) (right)

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 the Rutt/Etra Scan Processor and result in a multitude of fixed

[7] Vincent Bonin, "VCS3 (The Putney)" le fondation Daniel Langlois pour l'art, la science et la technologie, 2003. http://www.fondation-langlois.org/html/e/page.php?NumPage=446, accessed May 2014.

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

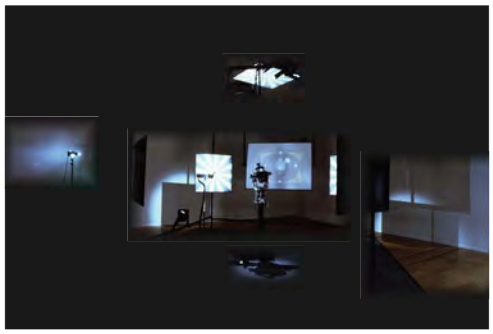
In the 1980s, Woody Vasulka began sifting through castaway objects at the Military Research Centre in Los Alamos, New Mexico. While these objects would have been of little to no use to most people, Woody's propensity for electrical tools and his talent for engineering hybrid objects made one person's 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.[8] While the sordid past of these machines and their inherent purposes of surveillance and targeting is at the heart of Woody's work *The Brotherhood* (1990-1998), 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. Though *The Brotherhood* was born out of a singular work-in-progress (*Automata*) it grew into a work with six movements or "Tables". The Tables were displayed in various ensembles and in some instances

---

[8] Steven Schoenherr, "The History of Magnetic Recording." University of San Diego, 2005. http://www.aes.org/aeshc/docs/recording.technology.history/magnetic4.html, accessed May 2014.

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; these machines 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.

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 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. 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 effect the space by their presence 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.



Installation shot of *Automata*

The second movement (though technically also named as "Table I" of *The Brotherhood*) was *Translocations/Rails* which played off of a similar theme of simulated spatial recognition. The visitor moved their hand over a table made up of a grid of lit squares, 17 across and 17 down. This configuration was reminiscent of a naval warfare game table in which movements and attacks are plotted on a grid and then mirrored in the macro realm. The table sensed the motions of a visitor's hand over points on the grid and on a motorized pulley system adjacent to the table, two x-y cross bars echoed the plotted points. Furthermore, a rail system in the rear of the space also mimicked the x-y motions of the visitor's hands, this time controlling a projector and screen which displayed a live image feed of the cross bar table and its internal mechanics. In this way, Woody felt that this form of mechanical mimicry was similar to the relationship between the choreographer and a dancer. Similar to a choreographer, the light table is an overseer who models movement on a smaller scale and the movement is enacted upon in human scale by an exacting person or instrument. The resemblance of the x-y plotter table to a war table used for "choreographing" combat strategies reminded the visitor of the technologies' more lethal purposes though it ultimately encouraged the visitor to become more deeply involved with the machine rather than merely exploiting its powers.

Woody Vasulka with plotting table from *Rails*

Friendly Fire or "Table III", was the first piece in the group to implement a musical instrument as a controller. The user struck a drum pad which triggered moving images on five monitors affixed to a central table and projected the images onto peripheral screens. Depending on which of the five pads was struck and the intensity/speed of the impulse, the images sped up, played forward or backward, or alternated between scenes. The sequenced images and videos were a mixture of recordings of historical occurrences of "Friendly Fire" (a phrase for a mistaken attack upon one's own troops or allies during warfare) and diagrams of integrated circuits. Similar to *Translocations*, the viewer was confronted with the machine's use for militaristic aims but was also brought into a deeper relationship with the anatomy of the machine by virtue of the mechanical drawings and schematics. The direct relationship of the drum pad to the images also was intended to endow the viewer with a sense of complicity with the machine as their impulses directly correlated to responses on the screens.

Installation shot of *Friendly Fire*

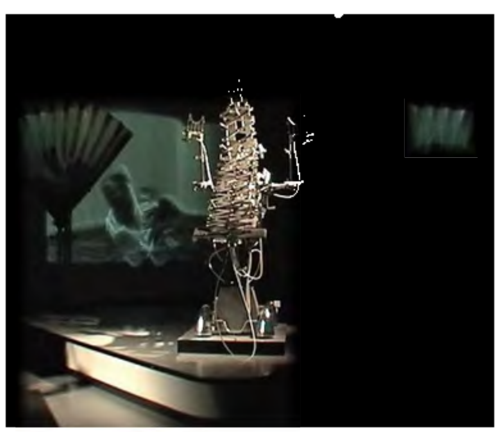
“Table IV” was *Stealth* and was an amalgam of the previous tables. In this installation the visitor sat at sighting station and guided a laser beam at four light-sensitive targets. The beam activated a camera that moved along a X-Y crossbeam in a manner similar to *Translocations*. Depending on which of the four targets was struck by the laser, the camera moved at varying speeds towards pre-programmed positions on the X-Y plane. The camera exposed the visitor to the inner mechanical functions of the table as the pre-programmed images mixed with those from a second camera which scanned the gallery space. Audio scores were also triggered based on the varying brightnesses of the image feed; certain brightness parameters were configured to correspond to MIDI notes (Musical Instrument Digital Instrument, explained further in the following section) which then played through speakers. Thus, the viewer’s attention was at once pulled between micro mechanical functions of the installation and the space as a whole and was confronted by the physicality of the machine space.

Installation shot of *Stealth*

Scribe, “Table 5,” utilized a pneumatically-controlled robotic arm that turned the pages of a book. A camera interpreted the pages of the book using an Optical Character Recognition software program that translated the text into a digital stream. This stream was then sent to a light pen which re-inscribed the work into a new book. The scanned texts were then projected onto a screen combined with intermittent footage of the violent destruction of books. The experience was intended to remind the viewer of format migration and media obsolescence, an inherent element of *The Brotherhood* as a whole as it sought to rehabilitate cast-aside objects and create a new, dynamic hybrid form. A commentary on the violence of technological obsolescence across time was also present in this work, further obscured by the work’s highly complex technologies which were dubiously aware of their own inevitable decay.

Installation shot of *Scribe*

The 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 surgical operating table which 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.



Installation shot of *The Maiden*

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 which configured each Table through a singular system. However, it is not as simple as to define all of the interactive possibilities for each given Table in *The Brotherhood* 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 *Stealth*, the viewer did not necessarily understand what about the four targets was causing the camera to move and to prompt the audio scores. Other nuances include differences in response time to actions by the viewer (such as how quickly the drum strike changed the video sequence in *Friendly Fire*) 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.

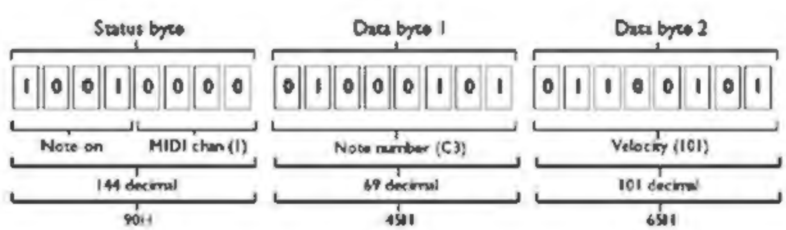
The behavioral aspects of the piece will prove to be some of the most significant and challenging properties for preservation as we delve further into the development and functionality of *The Brotherhood*. To make matters somewhat simpler for the conservator, the installations were video documented 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 technologies were available at this time which, in concert, could approach the desired outcomes as closely as possible. This first and foremost 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 innovation technology in interactive artworks.

Making The Brotherhood - Concepts and Computation

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 could 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 8 bits of binary information (1's and 0's). A MIDI byte can take one of two forms: as a basic MIDI header message in order for 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).[9] Data includes the content of the message, defining in most cases pitch and velocity (numerical assignment of the note within a scale and 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.[10] The channel information allows the computer to isolate a bitstream and direct it to a designated output destination, a particularly useful function in the presence of multiple simultaneous input and output signals.



Architecture of the MIDI bitstream, Planet of Tune

Another example of how the Vasulkas employed MIDI protocol in a unique way was the use of the digital stream for both input and output purposes. For example, in the case of *Translocations*, MIDI messages were generated by the motion sensors in the 17 x 17 light grid that were wired to a MIDI sampler. The MIDI message was created based on a preset in the

---

[9] Peter Manning, *Electronic and Computer Music,* p. 277, 1993. Clarendon Press: Oxford, New York.

[10] Planet of Tune, "Index of Sequences," 2013. http://www.planetoftunes.com/sequence/se_media/message.gif, last accessed 12/13/13

sampler and then sent to a MIDI serial convertor box which translated the MIDI binary signal into serial data. This serial data could then be piped into *The Brotherhood*'s host computer. In the example of *The Maiden*, however, MIDI was generated as an output signal from the host computer to a light dimming device. In this case, the host computer generated a serial MIDI message that was interpreted again by a MIDI convertor box, this time by outputting a binary signal that was sent to a light controller. The light controller also contained presets that generated lighting responses based on the MIDI binary stream. The example of *The Maiden* illustrates the usefulness of the MIDI bitstream in using the functions of pitch and velocity to contain a large yet finite continuum of possible values for interpreting stimuli and facilitating interactivity. In addition, it illustrates MIDI's ability to be repackaged through dedicated hardware. This act of conversion is an extremely vital key and concept to the design of the system for *The Brotherhood*, a point that will be analyzed later.

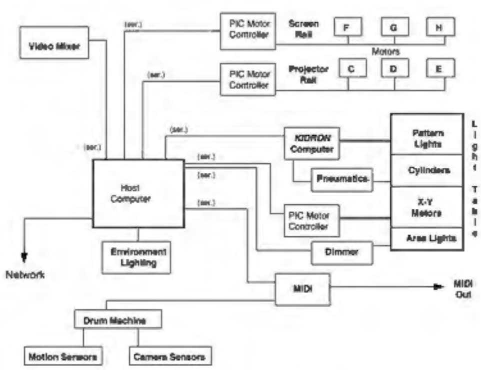


Figure III-6. The RAILS System

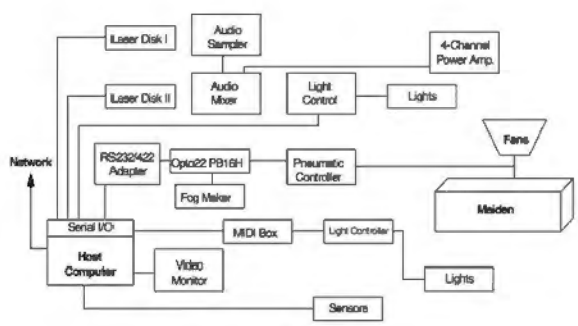


Figure IV-4. MAIDEN System Schematic

System concept block diagrams for *Rails* (left) and *The Maiden* (right), Russ Gritzo

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*.

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 work's many diffuse 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.[11] 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 brains 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

[11] "The 8086 Interrupt Mechanism" http://ece425web.groups.et.byu.net/stable/labs/InterruptMechanism.html, accessed May 2014.

many different Tables or parts of Tables. For example, in Translocations the movement of the visitor's hand needed to correlate with the motions of the x-y crossbeam arms as well as the moving projector and rail system in order to display the live image feed. 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.[12] 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. ISR allows MIDI information to be fed into a computer and to be interpreted as unique notes, each with its own tonal quality and punctuation.

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 that was designed with ISR capabilities that was implemented on a microcomputer[13] 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 thus it was more broadly adopted. 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

---

[12] BonaFide OS Development "Interrupts, Exceptions, and IDTs: Part 1 - Interrupts, ISRs," 2003. http://www.osdever.net/tutorials/interrupts.1.php, last accessed 12/13/13

[13] Benj Edwards, "Birth of a Standard: The Intel 8086 Microprocessor," http://www.pcworld.com/article/146957/article.html, last accessed 12/13/13

peripheral devices in a real-time fashion, something that DOS increasingly lacked as Linux reached wider adoption amongst programmers.[14]

Before further delving into the programming languages and routines that were adopted in the system concept of *The Brotherhood*, we must clarify the relationship of hardware, compilers, operating systems, and processes of translation. Let us first consider two fundamental differences in how code is written and read, namely source code versus object code. Source code is generally that which is written by a programmer in a specified language, which means it is human readable. It uses commands and values which are programmed to perform a specific functions using a controlled language. Object code is a bitstream which was designed to be specifically (and often exclusively) understood by a given piece of computer hardware, encoding the source code commands in a binary language native to the host computer; it is otherwise known as machine language. This language is “the native tongue of the processor in that computer: it consists of much simpler commands than those found in most source code.”[15] Machine code cannot adeptly migrate from processor to processor given that the binary language in each processor may mean something entirely different. Thus, source code is what is used in order for the machine to understand the commands and translate them into the machine language that is enacted within that specific system. The act of taking the human-readable source code and translating it into the language of a native processor is what is referred to as compiling. The process of compiling source code into object code is what causes a program to actually run.

---

[14] Russ Gritzo, personal interview, 08/05/13

[15] Jeff Rothenberg, “Renewing *The Erl King*,” 2006. http://bampfa.berkeley.edu/about/ErlKingReport.pdf, last accessed 12/13/13

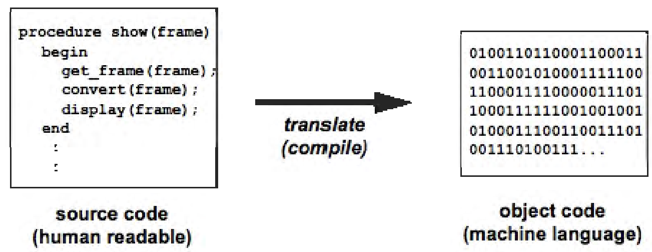


Figure 1: Translating source code into object code

Diagram of source code and compiling into machine language, Jeff Rothenberg

In other words, source code is what allows the programmer to manipulate or modify machine language with commands that are more intuitive and understandable to humans, and the compiler is the software that validates the source code by using instructions for translating those messages into the binary machine language. A useful analogy for the compiler and machine language is the Rosetta Stone and hieroglyphics. The stone references a fixed, proprietary language that is (in contemporary times) not human readable. However, although the stone is based off of a proprietary language, it references simultaneous languages which, in comparison, can unlock the meaning of the symbols and sequences by pointing to more modern, human-readable scripts. The stone acts as a compiler, in this sense, for the Egyptian Hieroglyphics (the "object code").

An analysis of programming languages and machines languages directly correlates to the use of ISR in The Brotherhood. 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.[16] 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 in the ways it will respond to those messages. For example in *Friendly Fire* 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 delivered through the serial connection port on the player.

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. 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

[16] Tim Fisher, "Firmware." PC Support. http://pcsupport.about.com/od/termsf/g/firmware.htm, accessed May 2014.

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 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. C, however, is not necessarily a monolithic controlled vocabulary for programming but is rather what is referred to as a library structure. A C Library outlines how certain syntax structures will be interpreted through the compiler, allowing the program to modify the library in order to include custom commands, or to translate these messages to bit-structures that are natively understood by the external hardware. The software package for *The Brotherhood* is made up of close to 3000 files. The package is mostly comprised of shell scripts and database files, and thus is dependent on the process of referencing C Libraries. This organization ensures a more standardized form of data-packaging and in turn ensures that communication with diverse peripheral devices could be easily automated and was modifiable.

At the time in which *The Brotherhood* was created, the use of BIOS and C Libraries was a more user-friendly way of accessing and modifying a computer's hardware-interface and native language 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 three essential lifelines 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.[17]

Statisticians and historians have analyzed the degree with technological growth has been exponential or constant, plotting developments in terms of RAM size, calculations-per-second, and data storage capacities.[18] Data for analyzing growth of storage and memory capacities in computers from the 1980s to the early 2000s actually demonstrates a consistent growth rather than exponential changes. However, this data correlates more directly with the relationship between amount of data and costs per volume, and does not speak to the sheer number of new products introduced on the market which reached a noticeably fast clip throughout the 1990s.

---

[17] Russ Gritzo, "ExhFour," http://www.vasulka.org/archive/ExhFOUR/Brno/brno.pdf, last accessed 12/13/13

[18] Ray Kurzwell, "Law of Accelerating Returns." Lifeboat Foundation Scientific Advisory Board. https://lifeboat.com/ex/law.of.accelerating.returns, accessed May 2014.

Additionally, data transmission speeds across networks saw an enormous spike throughout this period, going from approximately 1,000 bits per second in 1985 to 1,000,000 in 2000.[19] In particular, developments in integrated circuit technology (ICs) in the 1990s ushered in a new era of product development in computer science which resulted in tools for greater control of logic and timing. Essentially IC technology enabled engineers to fit more transistors (for electrical arithmetic functions) than ever before onto a single chip and also allowed for computers to be constructed in a "building block" fashion, leveraging the tools of computer design to the (albeit technologically savvy) consumer.[20] Integrated circuits were used in T*he Brotherhood* particularly within works that required a boosting of data transmission speeds and multiplexing of signals, facilitating the one-to-many responses that are seen in such works as *Translocations,* incorporating simultaneous cross bar and moving screen reactions. 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*

[19] Figures also related to cost, and specifically pertaining to data transmission through Internet Service Providers, which was nonetheless a noteworthy development as *The Brotherhood* was later controlled via a dial-up modem and IP address.

[20] "Integrated Circuit Classification." Integrated Circuit Help. http://integratedcircuithelp.com/classification.htm, accessed May 2014.

Brotherhood required. The Vasulkas and Gritzo designed their own system and thusly were responsible for innovating an environment that had never before existed. Of course, this further means that the work now lives in an environment that is extremely unique and perhaps irreplaceable.

In addition to ICs, *The Brotherhood* also required the use of the Opto Brainboard, a device for enabling slave/master controls in unison with the host terminal. A master/slave 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. 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.

Another final point for consideration in the genealogy of The Brotherhood is the lifecycle of the various software packages. Three software packages were created throughout the lifecycle of The Brotherhood - INTERCOM, MIDICOM, and UNICOM.[21] Ultimately UNICOM was the final package and was the last to be used in installation (as part of an exhibition in 1998 at NTT

[21] INTERCOM represented some of the first experiments with MIDI as an abstraction language, mostly used in Automata. MIDICOM was a means of casting an entire environment under this singular language and multiplexing messages to be understood through many simultaneous outputs.

InterCommunication Center in Tokyo, Japan). It was also the last to be archived. I found the last iteration of the software from 1998 as a zip file in the Vasulkas voluminous CD-R collection. Early in the process of developing The Brotherhood's system concept, the software was referred to as INTERCOM. It was in this early stage of development that the concept of "actors," a term used in *The Brotherhood* to describe the hardware and device drivers necessary for creating interactivity, was first used. The term referred to a complex sequence of responses or a set of concurrent and parallel responses (i.e., when video, audio, or robotic responses are generated simultaneously) that resulted from scripts which gave the necessary orders. In other words, they were tasks to be enacted by specific pieces of external hardware (enabled through BIOS). Essentially, the primary outcomes of the development of INTERCOM in 1990 were the configuration of a diverse number of hardware items and the incorporation of ISR such that commands could be 'liberated' from a simple one-to-one response relationship to a one-to-many. MIDICOM emerged in 1994 once it was decided that MIDI would be the primary abstraction language for packaging all the diverse digital and voltage-based signals generated by the hardware. MIDI was even used for installations in which musical instruments or audio were not a focal point (i.e., *Automata*). UNICOM, developed in 1996, was ultimately the software that encapsulated all the software, including the means of configuring multiple Tables under one system and streamlining the initializing and boot-up processes.

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. This tension between flawed or obfuscated digital objects and their purpose in establishing the behaviors of an interactive art installation will be explored in full in the following section. Nonetheless, 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 oversimplify 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.

These complexities are merely an abstraction of the piece as a whole as simply identifying and sketching the key technical concepts and processes in *The Brotherhood* does not even begin to describe the complex behaviors that emerge out of these interdependencies within and among software and related hardware. The core of documenting *The Brotherhood* is in the I/O protocols and peripherals and we will necessarily need to discuss a model that builds around this understanding. The section on the documentation model will summarize my work in performing a triage of the technical papers and software, delving deeply into the contents to better understand their relationship. In so doing I have developed a means of inventorying and packaging these materials in a way that is intended to be more intuitive to archivists and conservators as well as more attractive to scholars within the Media Archaeology discipline. But before introducing the documentation model, we must examine what methodologies presently exist for defining and encapsulating interactive works in the museum and library communities and to what extent they are adequate for a piece such as *The Brotherhood*. Also, it will be necessary to examine how provenance and the state of the overall collection enter into my process for untangling a rather sinewy artwork.

The Collection - Documentation and Remembrances

The software for *The Brotherhood*, namely the UNICOM software package, was last archived on one of the Vasulkas' PCs in 1998. 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 which 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 artifacts – some useful, some merely airline receipts or print-outs of news articles – 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, Buchla. 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 accounts of the development of *The Brotherhoo*d were going to be essential to understanding the artwork by better organizing the documents, so we interviewed him in August 2013. While Steina recorded, Woody and I presented a multitude of documents to Russ, some of which sparked his memory and others that did not. What did emerge was a very useful account of Russ' early impressions of the piece and a picture of how, through the remaining documents, a holistic understanding of the piece could be achieved by way of a logical ordering of records. For example, Gritzo recalled his apprehension about the use MIDI for the work, particularly given that it was a finite language (e.g. only having space for 128 unique note messages), and thus perhaps being too limited to accomplish a wide array of interactive outcomes. He recalled that, in the early 1990s, few examples had been tested for an abstracted computer language that would cast so many diverse machine languages into one standard protocol. He also recalled the decision to use DOS as the operating system based on its inclusion of ISR. Despite a degree of uncertainty about the use of certain technologies or approaches, Gritzo began to experiment with MIDI and BIOS programming, ultimately successful in satisfying the Vasulkas' requirements.

Gritzo's account in 2013 and the paper trail that he 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 Gritzo's assessment of these aims which manifested in his programming. Second, what emerged was that the Vasulkas and Russ Gritzo were attempting to design a computer-based work using processes that either they were inventing or that were not standardized processes in commercially available computers. In designing a customized computer environment they were also building a behavioral environment that resisted current or even emerging standards for managing and directing this amount of data, and in doing so were forging a path 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. As Russ Gritzo explained in his 1995 manual for the Intercom software package:

> "The recent integration of microprocessor technology with media equipment has resulted in an abundance of systems capable of communication over common computer interfaces. Examples range from laserdisc players with RS-232 interfaces to violins with MIDI outputs. In parallel, computer technology has progressed to the point of bringing such sophisticated devices as speech recognition and synthesizer units into practical usage in a number of applications...What was needed [for *The Brotherhood*] was a flexible software system that could be easily configured to accommodate interfacing any of a number of systems to each other, without developing new software for each configuration or setup."[22]

This statement exemplifies the inherently unpredictable nature of the artwork and the need for Grtizo 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. Since Woody did not have a background in scripting he articulated his desires for the piece based on whether the desired look and feel was satisfied by the programmed responses. I describe this process of designing the technologies of the work around absolutely defined aesthetic outcomes "affectual" in that the machines need to respond in ways that felt natural and organic to the artists, in a sense operating beyond words alone. Arriving upon what this organic response meant from a computational perspective required a great deal of trial and error on Gritzo's part; he scripted programs that

---

[22] Gritzo, Russ. "User Manual for *The Brotherhood*" 1995.

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. C and MIDI). Fortunately the piece has been well documented with video footage which has means that some of the relations between stimulus and response are continuously observable. However, given the dynamic, multi-directional nature of the signals and messages that yield reactions within the work, it is very difficult to pinpoint exactly what the precise relationships are between stimulus and response from Table to Table. This varying degree of stimulus-response clarity is what gave the piece the autonomous feel that so attracted Woody.

Following from this issue, timing is perhaps the most difficult aspect of the piece to document and thusly contain through documentation. If the direct relationships of stimulus and response are difficult to understand, than the speed with which these behaviors occur is even more difficult to determine. In a similar case study of an interactive artwork presented at the Preservation of Complex Objects Symposia[23], Vickey Isley and Paul Smith explore issues of timing with regard to their artwork "*whirligig*". The work involves a computer environment that sustains life for germ-like organisms which continue to grow, evolve, and develop autonomously across time. One particularly useful excerpt on timing states:

> "To achieve fluidity in this artwork, we had to optomise, cut corners and find compromises; pushing the boundaries of what was possible on the chosen platform and hardware. In our attempt to squeeze every last cycle from the processor, we set everything to run as fast as possible. This was a race we could not win. Within a couple

[23] Joint Information Services Committee: The Preservation of Complex Objects Vol 2: Software Art. November 2012.

> of years technology thundered passed. Appearing from the dust cloud left in its wake were our creatures, whizzing about maniacally at supersonic speeds far beyond our intention. This simple blunder, easily corrected with a single line of code frame rate "(30)"; reminds us we are not the only authors of change. We have to follow good programming practice to ensure the future of our work, if not for future generations at least for future presentations."[24]

As this statement reminds us, timing is one of the most crucial elements of the responsiveness and spatio-temporal quality of a computer-based work, perhaps why "time-based media" has continued to be used as a descriptor for technology-dependent works that have a temporal element along with terms such as "variable media." With *The Brotherhood* timing is not just significant in how evidently responsive the work is to the viewer but how all the hardware and software work in succession, or in some cases in parallel with each other. Understanding that improvements in technology also means an increase in processing speeds, many works that are being restored require addenda to the source code to specify timing which stagnate the piece, given that certain processes will run much faster on current computers than in a work's native environment.[25] However, to understand speed in the source code of *The Brotherhood* is not as simple as the singular speed of the microprocessor but includes all the other ICs, the brainboards, and the cross-directional trafficking of data streams.

---

[24] Vickey Isley & Paul Smith, "In Homage of Change." *The Preservation of Complex Objects Symposia (POCOS) Vol 2: Software Art*. JISC, November 2012. p. 56

[25] This approach insinuates migration, though even in the instance of a virtual or emulated environment, processing speeds will be determined by the hardware that is running the VM regardless of what processing speeds were involved at the time of the original piece.

Now that we have thoroughly complicated whether it is possible to determine the behaviors of *The Brotherhood* irrespective of the hardware and software, we can turn our attention towards some existing models for describing the technological relationships that define behaviors in variable artworks to gauge their adequacy for this piece. However, a central question will be whether existing models can be used for access purposes in both libraries and museums (i.e. the record *of* the work for scholarly study as well as for purposes of reinstallation) and how access to these materials can be facilitated. As I've already argued, scholars (in particular media archaeologists) can help to position a work as innovative by virtue of what other possibilities existed at the time the work was created. Artists who attempted to do the impossible by gerry-rigging existing tools in unconventional ways or engineering their own computational systems created work that is difficult, if not impossible, to stabilize because there are so few outside resources - replacement parts, updated software versions, emulated environments - to sustain it. A conservator will diligently document the work from a material, historically contextual, and behavioral standpoint but their work can only go so far, particularly if there isn't an adequate amount of time and resources to allocate to the work in order to faithfully restore it. As I have argued, restoration is not likely to happen unless the work has a solid place in a larger historical framework, namely that of innovation in technology. Developing a structure for documenting and historicizing works of this nature will not only create a dialogue between these technological scholars and conservation staff, libraries and museums, but will also provide a more rigorous understanding of how these works function by virtue of their legacy equipment.

Existing Methodologies for Time-Based Works - Square Pegs, Round Holes

How best to ensure long-term access to a time-based artwork by means of documentation of its dependencies and behaviors is not a new discussion in the world of media art conservation. In looking at existing methodologies I am inspired by methods that strike a balance between behaviors and technologies and explore how to develop a descriptive framework for this relationship. Most of these methodologies exist in order to contextualize the work so as to facilitate reinstallation, although for *The Brotherhood* my goal is rather to make the work's contextual documents a means of providing scholarly value rather than as a means for the experiential (i.e. reinstallation in a gallery). The models are helpful for this research-based initiative, to enable documents to be organized and component parts to be identified. Ultimately, however, the new model, appropriated from existing methods, will necessarily diverge from them in order to suit different user needs.

The Variable Media Initiative is an often –cited example of cross-institutional networking that had the goal of establishing a model for approaching complex, iterative, aggregative, and behavioral works. The consortium sought to develop a plan for long-term stewardship of these works by developing evolving approaches that outline the relationships between installation elements and their expressed behaviors. Spearheaded by Jon Ippolito at the Solomon R. Guggenheim Museum of Art in 2000, a Variable Media Questionnaire (VMQ) outlined how to describe a work in its "ideal state", portraying a best-case scenario for what materials embody the work, knowing that some necessary deviations may have to occur over time.[26] The questionnaire developed as a descriptive tool for exposing qualities of the work that Ippolito

[26] Solomon R. Guggenheim Museum of Art, "Variable Media Initiative." http://www.guggenheim.org/new-york/collections/conservation/conservation-projects/variable-media, accessed May 2014.

proposed should aid a conservator and/or curator in making decisions about what elements (material or behavioral) constitute the essence of the work, and in so doing, should enable the caretaker to find areas for acceptable loss within the work. In other words, it was a means of identifying the extent to which a work's original modes or technologies are necessary for realizing the work's behavioral and aesthetic properties. Knowing, for example, that laserdisc players are no longer the standard in museum exhibitions for video switching, these components of the work could certainly be substituted within *The Brotherhood* for more modern switching devices, but the VMQ and an artist interview would hopefully expose this fact rather than the conservator jumping to certain conclusions about the essential nature of the laserdisc. In the VMQ, the parameters of the work's ideal state are defined based on eight distinct criteria including, to name a few examples, whether the piece is networked (continually pulling data from another source), interactive (responsive and to what extent), or encoded (has a dependency on certain computational platforms).[27] From there, the conservator can determine what behaviors emerge from these taxonomies and whether or not their physical manifestations must (or can) be stabilized and preserved along with the work. The questionnaire attempts to expose the degree to which technologies are essential to the intent of the work or are secondary, merely serving as a form for mediating the concept that constitutes the work. The questionnaire also serves as an enduring point of reference for the original state of a work, understanding that successive installations are inherently different from the original. This reference gives the conservator some context for why the work was made using certain processes; for example, out of consideration

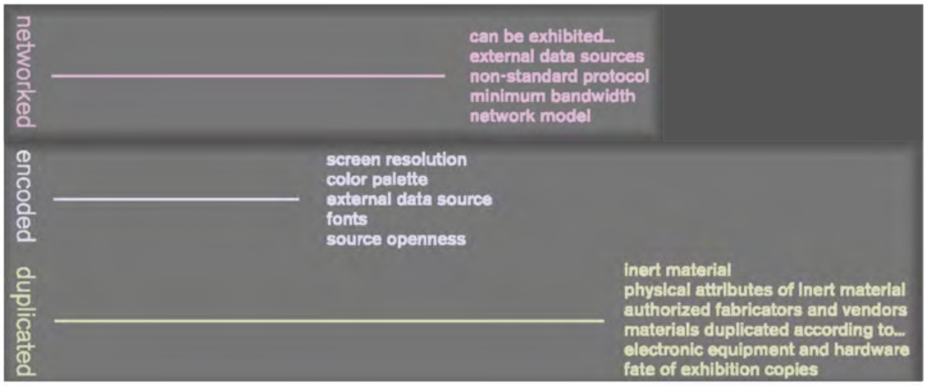
[27] Variable Media Questionnaire, http://www.variablemedia.net/e/index.html, accessed May 2014.

for existing technologies, social conditions of the time, or the impact that technology had on those social conditions.

The VMQ was a revolutionary approach toward new media works and sparked cross-institutional conversations in the context of object conservation in art museums. *Permanence Through Change: The Variable Media Approach* was a publication that addressed issues which many arts institutions were facing despite having diverse administrative capabilities (e.g. a developed conservation staff) and differing collections policies as to the types of materials often found therein. In exploring these issues through specific case studies, they were able to draw out narratives that were in fact quite similar despite these institutional differences. Contributions came from staff members and consultants from the Berkeley Art Museum/Pacific Film Archives, Franklin Furnace, Rhizome.org, the Walker Art Center, and the Daniel Langlois Foundation for Art, Science, and Technology (to name a few). Published in 2003, this group of case studies reified the VMQ's malleable descriptive structure for defining a work of art in which there are complex and proprietary relationships. In analyzing a group of diverse works that shared common problems, they acknowledged a lack of controlled vocabularies for these problems and developed a taxonomy for describing a work's most essential dependencies and behaviors. For example, if we were to apply taxonomy to *The Brotherhood,* the work could be described as one that is networked, encoded, interactive, installed, and contained. Within these headings there are opportunities to further extrapolate on the technologies which embody these behaviors, such as within "networked" where "non-standard protocol" could then describe the purpose of the MIDI abstraction language.[28]

---

[28] Variable Media Questionnaire, http://www.variablemedia.net/pdf/Ippolito.pdf, accessed May 2014.



Excerpt from VMQ Table

While the VMQ begins to address the need for controlled vocabularies for describing the dependencies within a work, one member from within the *Permanence Through Change* collaboration went further in his exploration of a possible standard. In his article "A System of Formal Notation for Scoring Works of Digital and Variable Media Art," Richard Rinehart (then serving as Digital Media Director at the Berkeley Art Museum/Pacific Film Archives) proposed a model for describing complex computer-based works by virtue of technologies that enable variable or aleatoric artworks. For this reason, he began his analysis with FRBR, the Functional Requirements for Bibliographic Records, which aims to describe the multiple iterations that a singular work can take throughout its variable forms. FRBR is the predominate schema in libraries (with gradual adoption in museums) for describing multiple iterations and stagings of a singular work or a logical series of items which together constitute a work. This taxonomy is broken down into four distinct definitions which successively get closer to the precise iteration; these classifications are defined as "Work" (the distinct intellectual work, e.g. the story of *Hamlet*); "Expression" (the realization of work, perhaps in multiple stages, e.g. Bloom's edition of *Hamlet*); "Manifestation" (physical embodiment of the expression, e.g. paperback copy); and

"Item" (an exemplar of the manifestation, e.g. individual copy on a shelf).[29] Rinehart argued that this could be a useful model in terms of describing the relationships among the technologies that exist for the work in a collection of time-based media (e.g. the "items") grouped under the essential behaviors which the items are meant to express (e.g. the "expression"). The "item" could be seen as the most precise way of staging the work, though the "expression" is higher up in the descriptive hierarchy; thus the objects for mediating the work must first be explained through the lens of its intended function.

Though FRBR is a useful model for variable works, Rinehart does not see FRBR as completely scalable in order to clearly articulate the needs of variable artworks which incorporate interdependent hardware and software. Or, put another way, he believes that such frameworks for iterative works are inherently flawed in that it becomes difficult to place technologies in subordination to their intended behaviors without removing the interdependencies *across* technologies in the process. The behaviors of a work should dictate the way that components are arranged within the work's descriptive structure but the individual physical or digital components should also be placed in a framework that describes its relationship to a technological environment or system concept. Thus, he proposes a Media Art Notation System (MANS), borrowing from the notion that music scores are documents that sample from a given language of a notation system, imbuing the behaviors, emotions, and performative elements that constitute a work through its expression. As a response to this lack of useful flexibility in the FRBR structure, Rinehart describes how MANS mimics the adaptability of a music performance through its notation - the score. He states:

[29] Barbara Tillett, "What is FRBR? A Conceptual Model for the Bibliographic Universe." Library of Congress Cataloging Distribution Service, http://www.loc.gov/cds/downloads/FRBR.PDF, accessed May 2014.

> "The formal notation system could be considered an expression of that model and a score considered a specific instance of notation. It is important to note that the conceptual model and expression format are distinct entities. For instance, this conceptual model could be expressed using various formats such as Extensible Markup Language (XML)(4) or a database file. In this way the conceptual model itself defines the integrity of the score while allowing for variability in its expression. The conceptual model could be considered a kind of meta-score."[30]

Following from this proposal, Rinehart also cites DIDL, the Digital Item Declaration Language and adapts it for MANS. DIDL incorporates a controlled vocabulary for declaring the properties of digital items, allowing for more granular description of the discrete parts within a cohesive digital item such as a package. Rinehart proposes that this controlled vocabulary can be adapted in a manner similar to FRBR by incorporating a similarly dependent tree structure. For example, the main "Descriptor" element would contain information about the piece at a holistic level (or in terms of digital objects, the parent directory). Within that heading there would be a sub-element of "Container" which would describe an individual digital element (e.g. a file within a software package) which would have its own descriptor, describing its role within the directory structure. One particularly demonstrative sub-heading is found in "Part," a logical sub-component which can contain information about the variables that effect or enable chance-operated elements, such as an algorithm. This information is contained under the heading of "choice" from within

[30] Richard Rinehard, "A System of Formal Notation for Scoring Works of Digital and Variable Media Art." University of California, Berkeley. http://www.bampfa.berkeley.edu/about/formalnotation.pdf, accessed May 2014.

"Part."[31] This structure, he proposes, can serve the conservator by storing and describing elements of a variable work in a way that also retains information about its behavioral properties.

Score: *didl xml metadata document itself*
<DIDL>
Descriptor: *descriptive data about score document*
<DESCRIPTOR>
    Work: *logical media art work or project*
    <CONTAINER>
    Descriptor: *descriptive data about work or project*
    <DESCRIPTOR>
        Version: *occurrence/state/ account of work*
        <ITEM>
            Part(optional): *logical sub-component*
            <ITEM>
            Choice(optional): *variables affecting configuration*
            <CHOICE>
                Resource: *physical or digital components*
                <RESOURCE>

Hierarchy Structure for Media Art Notation System (MANS)

One tension found in both VMQ and the MANS is that the former offers documentation around an artwork without the use of an adaptable, controlled vocabulary, whereas the latter proposes a controlled vocabulary but without a clear sense of how documentation aids these descriptive decisions that themselves exist as objects within the structure. One project that leveraged the object-value of documents was Documentation et Conservation des Arts Médiatiques or Documentation and Conservation of Media Arts and Heritage (DOCAM) in 2005. This project was organized around research and case studies that focused on the theory and practice of conserving complex time-based media works, much in the same manner as the VMQ, and was more of an evolving consortium of projects with similar goals rather than a singular

[31] Richard Rinehard, "A System of Formal Notation for Scoring Works of Digital and Variable Media Art." University of California, Berkeley. http://www.bampfa.berkeley.edu/about/formalnotation.pdf, accessed May 2014.

organization. However, this particular project incorporated a much wider team of professionals from information sciences, computer sciences, and humanities than its predecessors, proposing best practices for preservation of technology-based artworks with a particular focus on more precise ways of describing a work's lifecycle. One case study from DOCAM is Alexandre Castonguay's *Génerique*, an artwork that captures the silhouette of a visitor and processes it into an abstract animation that mimics foliage, water, and fire. The documentation present through the case study includes information pertaining to materials, spatial dimensions, number of items, a history of modifications/adaptations, lists and images of component parts, and notes on each component. In describing some of the interdependencies and anticipating future risks in the DOCAM report for *Génerique*, conservator Anne-Marie Zeppetelli states:

> "The computer, the operating system, the programming environment and the program are, from the artist's perspective, the four basic elements required for the continued existence of the work. Although the computer can be replaced by a more recent model and the current projector by a luminescent surface, the Museum must reintegrate the same versions of the operating system and software, such as Ubuntu 7.10, GridFlow and Pure Data."[32]

Zeppetelli makes further mention in her report how a chain of custody and an explanation of past conversation efforts (aided by either conservator or artist recollections) or technological updates were often necessary to fully understand the state of the work and whether or not any parts needed to be replaced or emulated in order for the work to survive. She also stressed the need for setup documentation and README files, information that is not always anticipated at the time

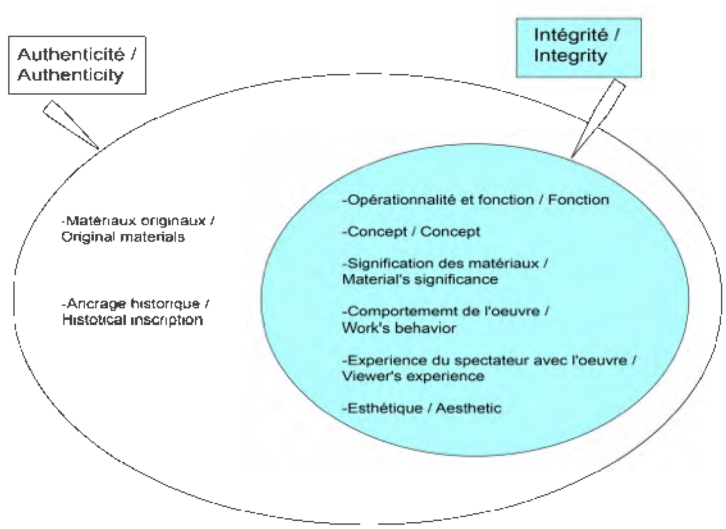
[32] Transcripted interview with Anne-Marie Zeppetelli, October 2009. http://www.docam.ca/en/bringing-together-information/three-approaches/168-entretien-avec-anne-marie-zeppetelli-archiviste-des-collections.html, accessed May 2014.

of creation since computational practices are almost thought to be self-evident into the future; however, she noted that these documents will prove to be essential as the work evolves beyond the original technologies which enabled it. Of course, as we saw with the example of timing in Isley and Smith's *whirligig*, an understanding of the native environment will need to persist along with the artwork.

Zeppetelli's case study was one of many that led to the development of standards and controlled vocabularies for technology-based art works that continued to resonate within DOCAM. In particular, their guide entitled "Descriptive Table - Essential Data-Entry Fields for Documenting New Media Works and Recommendations" outlines ways of organizing hardware and software elements and their points of intersection, attempting to solve similar issues as those that are found in *The Brotherhood*.[33] This information is broken down into two sections - "Components" and "Description. "Components" contains information about the individual piece of hardware and software whereas "Description" goes into detail about the behavioral aspects of the pieces of technology, encapsulated as "Iconographic" (e.g. behavioral and aesthetic qualities or observable details), "Technical" (e.g. configuration requirements), and "Installation" (specific to the artwork, e.g. relationships among components). This information can be very useful for a conservator as he or she determines the relationship between the materials and the artistic experience, a dynamic that is further illustrated by the work of the Conservation and Preservation Committee (a subgroup of the DOCAM project) through their examination of issues in authenticity and integrity for media conservation.

---

[33] DOCAM "Descriptive Table - Essential data-entry fields for documenting new media works and recommendations." http://www.docam.ca/en/cataloguing-collections/descriptive-table-essential-data-entry-fields-.html, accessed May 2014.



Nesting chart for Authentic Technology-Based Works, Conservation and Preservation Committee, DOCAM

As we see see in the above diagram[34], the members of the Conservation and Preservation Committee believe that "integrity" is defined by experiential and physical properties, viewed in context with one another such that the significance of the materials is noted along with the concept, behaviors, and viewer experience. These are all significant properties that need to be recorded for posterity and will inform future installations of the work. Authenticity encapsulates all the essential components found in the integrity of the piece but more generally contains information about the original materials and their historical inscription. By virtue of this model, it would appear that without the historical inscription of the technologies – irrespective of their use within an artwork – the intent of the work and its continued authenticity is lost along with the original obsolescent materials. In regards to whether the artwork must necessarily carry the

[34] Members of the Committee include Emile Boudrias, Alexandre Mingarelli, Olfa Driss, and Richard Gagnier

material significance throughout the entire lifecycle, the Conservation and Preservation Committee has this much to say:

> "We see ourselves as standing at the intersection of the analog and digital worlds, where both technologies continue to be actively used. For this reason it still appears possible, at least in some cases, to offer solutions capable of maintaining a work in its analog mode; indeed for certain works there is simply no alternative. Based on this paradigm, these works would seem destined for a limited lifespan, and we believe such should be the case, as they would otherwise lose too much of their integrity to be viewed and experienced as originally intended. It is therefore vital to determine exactly what constitutes the work and under what circumstances it can continue to be considered as the same work. The case studies put forward focus on doing just that. These works can of course continue to exist in an alternative format; they can adopt the status of a documentary manifestation as witness to a specific production history, for example. But that, clearly, is another case altogether."[35]

As demonstrated here, DOCAM has a rather firm stance on whether art works can be reinterpreted through new technologies, especially when the artist may not have articulated which elements are essential to the work's authentic rendering upon depositing the work in a collecting institution. Information regarding the significance of original technologies is of the utmost significance if the work is to be understood within the context of the work's historical conception, noting what technologies were unavailable which necessitated their creation or understanding in what ways human-technology relationships were being critiqued. Will the work

---

[35] DOCAM Conservation and Preservation Committee, "About the Conservation Guide: A Preservation GUide for Technology-Based Artworks" http://www.docam.ca/en/conservation-guide.html, accessed May 2014.

continue to invoke a historical moment related to the work's inception or will it become an entirely new work altogether, understanding that the original relations between the audience and technology are necessarily redefined? DOCAM has provided a useful structure for containing this information, although the information is again used more for the purposes of informing conservation decisions rather than adding to the scholarly access value of the piece. Can methods of arrangement be a document in and of itself and how can that document be granular yet flexible to adapt to a diverse set of researcher needs?

Here again we find ourselves back at the dilemma which immediately faces *The Brotherhood* and its likelihood to be included as an historical example of artistic and technological innovation in the early 1990s. The work needs to be fully understood and appreciated through its exploration of human-machine interaction and how it pushed the limits of conventional computation power and capabilities. In order to do this, descriptive frameworks *for* the documentation should consider what aspects of the work are unique such that they are not overlooked or disregarded, understanding that aspects which connect separate documents are not necessarily self-evident. The models that we have examined are great at helping us to understand relationships between behaviors and materials in interactive installations but it does not begin to address how a work such as *The Brotherhood* which lived in an entirely unique technological universe is to survive or, put another way, how we as archivists are to describe the universe that they created. 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 - Woody’s organic, autonomous machine which nonetheless behaved based off of inscribed, predictable commands (albeit after much trial and error).

In order to engage a scholarly community, we must think in terms of how scholarly collections are typically described and accessed such that the collection of *The Brotherhood* can be discovered. For example, what if the documents from DOCAM were treated more like a library record, going beyond the basics of controlled vocabulary to describe operating system, programming language, and so on to create relationships between all the documents (i.e. the digital files) within the holistic artwork? 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 keywords which describe technological processes and not just the environment. However, museum methodologies like DOCAM are still extremely vital for an archival context since researchers still need a framework for understanding how individual components affected the piece as a whole and contributed to the holistic functionality of the artwork. If research on device drivers for servo motors led them to the collection, they would still need to grasp how MIDI was used as the primary abstraction language, for example. 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. Additionally, examining the technological conditions and restrains of the time are necessary for dissecting the software that united the work and connected stimuli with responses, helping the scholar or conservator to better understand the direct relationships of stimuli and responses rather than simply accepting them as glitchy or unpredictable.

Media Archaeology is a field for examining how technologies affect culture and our social reality with a particular focus on how technologies developed. It is at once a refined examination of how technologies progressed over time while also placing a history of technologies within an assessment of how human interaction and media literacy are affected. For the purposes of *The Brotherhood*, the media archaeology field represents a community of scholars who may appreciate how the history of innovative technology helps to depict what social and political desires lay behind the technologies that are engineered and consumed. Most interestingly they also examine why certain technologies have been left behind, also striving to document a history of emerging media tools which are otherwise left out of the story in light of technology's constant acceleration. Naturally, media archaeologists are greatly attracted to collections of legacy hardware and of complex digital objects, and are interested in issues found in accessing the "authentic" look and feel of obsolete media. While not necessarily responding to this particular scholarly community, libraries have a long tradition of stewardship to protect the nature of the original (e.g. manuscripts and paper-based works) and the principles of stewardships continue to be applied as they turn their attention to the nature of digital collections.

For digital objects, one particular model has recently emerged which seeks to describe the interdependencies of complex works, thus facilitating a better understanding of their native environment as a way of enabling a more authentic rendering. In 2013, the University of Michigan issued their second version of "Describing Digital Materials in Finding Aids", for the purpose of describing the logical order and interdependent nature of filesystems and storage devices. This approach was developed as a means of making digital collections more accessible and searchable by researchers, allowing them to locate digital collections in the archive (e.g. series grouped by file type and listing any interrelated files or software) and to use this information to gain a sense for the technical dependencies within a set of objects. In this case, the finding aid predominately serves the function of describing and revealing the structure of digital storage devices by allowing the researcher to study a work's directory structure and the interdependent nature of bundles of files. However, contextual information for understanding these relationships is described quite generally, assuming that file formats and datasets are functioning as is and don't require a more robust description of the system upon which all the related media must run. Most works which are being indexed and catalog in the Michigan Finding Aid are much simpler than the *The Brotherhood* data that is specific to how the software package is configured and compiled and what necessary hardware is involved to enact the piece would require their own unique data fields..[36] Other data fields in the Michigan Finding Aid include the filetype and extension (e.g. .sh); the file size; its location in a directory; notes on other files which are directly required or relied upon for renderability; and a narrative about that

[36] Michigan Historical Collection, University Archives and Records Program, and Digital Curation "Describing Digital Materials in Finding Aids." Version 2.0, March 2013. http://deepblue.lib.umich.edu/bitstream/handle/2027.42/96874/BHL_DigitalDescriptionGuidelines_20130314.pdf?sequence=1, accessed May 2014.

file's significance in the overall scheme, but information on a systems overall configuration (for the purposes of triage or awareness of conservation risks pertaining to obsolete or endangered formats) is not contained as its own element.

For the purposes of maintaining intellectual control over the objects as well as their contextual data, the model will need to go further than this. This structure does not necessarily position the collection as discoverable based on the related technologies and how their dependencies themselves might be a point of interest for researchers. Furthermore, a structure that simply reveals a set of directories and files imagines that the scholar is attracted to the collection based on the creator or the subject (the Vasulkas) rather than the types of objects therein (MIDI programming and interactive robotics) and how they are being exploited within the work. Of course, the University of Michigan finding aid is not necessarily designed for the purposes of a more granular understanding of the technologies within a work but rather assists the researcher in navigating a hard drive-based work that is otherwise less intuitive to interpret than, for examples, papers which, more often than not, arrive at the archive in a more orderly state. Scholars may be attracted to a collection by virtue of its technologies and not necessarily based on and interest in the creator or subject. For example with *The Brotherhood*, understanding how the x-y plotter arm responds to a visitor's hand across the sensor table may be of interest to scholars who are doing work on complex robotics and interactive technologies; the analysis could pinpoint the exact responses in the code and, in so doing, leverage the significance of the experimentation in the work. These scholars may not be familiar with the Vasulkas such that this connection could be made, so a collection such as *The Brotherhood* in a scholarly environment such as a University library should respond to these types of researcher needs. How can data around a work articulate both the behavior intent of a work while also stating all of its technical requirements and the uniqueness of the configuration in a way that is agnostic to researcher needs?

For a work like *The Brotherhood* it a matter of tagging these unique computational properties and adopting the model of system concepts and configuration diagrams to create data fields that both outline the overall system concept and contain information pertaining to the necessary timing and interdependencies of each piece of hardware and software. As I have argued several times, works of a highly complex and innovative nature require an extra step in order to encapsulate its unique functionality along with its contextual significance. While this encapsulation will greatly aid researchers, for example in media archaeology, who may help to cement a work's place in history, an approach which combines behavioral and evidential descriptive techniques may also become the initial steps of a conservation plan within the library/archival setting. At a base level to embark on this process of encapsulation within an archival environment, one must document a work such as *The Brotherhood* in a manner that satisfactorily describes the work's original state and the artist's intent in creating it. This approach towards documentation also requires a means of making the information accessible such that these noteworthy concepts and experiments within a work can continue to be studied and understood; in other words, in consideration of the work's context within a history of related technologies. The missing piece that media archaeologists can offer is to shape a history of innovation within *The Brotherhood* that has been to this point inadequately recorded in light of emerging forms of technology which can often erase this material breadcrumb trail, helping us to understand how specific lines of code or commands, microprocessors, and BIOS programming are interrelated and the extent to which they affect timing, interactivity, and responsiveness. These are the elements that need more clarity in a descriptive system - how these individual nodes can be more distinctly noted.

If one starts from the point of view of Woody's idea of a technical curriculum – which is the intent of the Vasulkas' donation of their artistic work to the archives at the University of Colorado – one can document the piece so that it reveals the historical context within which it was made, which will further propel analysis of a deeper conceptual and technical understanding of the work. For example, what roles are actually being performed by each component part (MIDI instruments as interactive controllers rather than music instruments) and in what manner the software is behaving (C commands/MIDI note space and their interaction with BIOS control). Perhaps most importantly, how can the source code be used to relate and link specific commands with their designated hardware outputs, also noting what interdependencies lie within the software package which link specific scripts and database files? These questions help us to identify what the main purposes are for a new descriptive model, bearing in mind who are main stakeholders are how to simultaneously pay tribute to their respective work in developing more ideal documentation models. Through a collection of this nature, we can engage in archival partnerships between the stakeholders to balance the behavioral/aesthetic methods in museum conservation, the documentary/evidentiary methods in library sciences, and the pursuit of deeper engagement in legacy technology as is seen in the work of media archaeologists. This hybridization of methods can help us to create a new model for how collections such as *The Brotherhood* can be described, stored, and made accessible for further study and exploration.

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 (C) leave unexplained traces of what functionality (or potentially lack thereof) was allowed through its syntax. Identifying those examples, such as unique characters, syntax, 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.

Coincidentally, Boulder is a leading university in the emerging area of study of media archaeology, seen most presciently in the work of Lori Emerson who established the Media Archaeology Lab (MAL). The lab is an environment where students can engage with legacy hardware and gain a deeper understanding of how certain technologies were used by actually playing with the hardware and with the inherent programming languages that were used for designing and manipulating older software. According to Emerson, "What the MAL does best is that it provides direct access to defining moments in the history of computing and digital literature...The MAL is also a kind of thinking device in that providing access to the utterly unique, material specificity of these computers, their interfaces, platforms, and software makes it possible to defamiliarize or make visible for critique contemporary, invisible interfaces and platforms. It's an approach to media of the present via media of the past that aligns the lab with the vibrant field of 'media archaeology.'"[37]

In transitioning the Vasulka's collection of software-based elements and the documentation of the component parts, the library and MAL could offer a site for a continued investigation of how the various models and case studies discussed above could overlap and could further develop a descriptive system for works like *The Brotherhood*. Though none of the models or case studies discussed above were necessarily designed for the purpose of scholarly access within a library and archive setting, the way in which the models structure documentation and related elements can be useful when applied to processing an archival collection to exploit a collection's potential as an innovative and complex set of materials worthy of further analysis and play. New forms of documentation, arrangement, and description can give the user of set of instructions for understanding the environment such that they can not just study the relations among the hardware and software elements but potentially engage with the original code and its intended configuration with external hardware in order to witness how its customized language is enacted. Whether the original RPT or a new version of a roving camera, certain elements of the code's machine language and the nuanced conditions that are contained therein may be best revealed through use of the original software elements manifested through interaction with physical hardware. These notions of "access" for further scholarly inquiry should also be considered in developing our descriptive model.

What follows in the next section is a proposal for explicating the history of *The Brotherhood* and its technical genealogy, whose concepts and physical materials we have

---

[37] Lori Emerson "Media Archaeology Lab - Why?" http://mediaarchaeologylab.com/about/why/, accessed May 2014.

explored above. In this section I will argue for a specific structure for inventorying the evidence that remains for the work, organizing documents, and building a data structure that will leverage the work's applicability for media archaeological study. In the process we should remember that what might be most useful and discoverable to scholars is not necessarily only information about the creators or subjects but the significant components of the work from a technological perspective.

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 a 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, 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 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.

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.).

Example of source code "evidence" in IN040, user manual for the RPT by Russ Gritzo

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 Automat (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 ahead. In consolidating such examples of clusters of code into a singular document (what I referred to as a "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 filetypes 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. At this stage of the report it will be useful to view the accompanying inventory which more clearly articulates and diagrams the hardware and software elements in an interrelated manner.

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 image	Automata; Friendly Fire; Rails;		VGA/EGA from Special Effects Generator		VGA/EGA				IN205
Laserdisc Playe	Hardware	Pioneer LD-V8000				75 ohm sync in	NTSC video; EFM au	Terminal Pin Jack;	RCA; Gen-Lock; BNC			IN205
MIDI Pitchrider	Hardware	Portman	Feeds into drun	The Maiden; Rails		Analog amplitude/fre	MIDI out to drum ma	XLR	MIDI	XLR male/female and cable		IN006; IN008; I
MIDI Drumpad	Hardware	Roland PAD-5	Controller fed ir	Friendly Fire	Pitchrider II	User strikes one of 5	MIDI OUT to PC/S b	Drum pad	MIDI			IN006; IN057; I
Portman PC/S I	Hardware	Portman	Multiplexes anc	Automata; Frier	Opto-isolator	MIDI IN from Pitchrid	Serial	MIDI	Serial/RS-232/485	DIN 5-pin femal	MIDI 1.0 Specification;	IN006; IN008; I
Relay Board	Hardware	Opto22, PB8	Solid State Relay Board, low logic level signals t			Binary from Intronics	PWM to Slide Projec	DB-25	OAC5 to Slide Proj	50-pin to DB25	Intronics Power Supply	IN205
External Brain E	Hardware	Opto22, Pamux	External brain board for commu		AC28 Adapter Card; PDRIVER.OBJ; PDRIVER.COM; Pamu			DB-50	DB-25	AC28 adapter c	Microsoft Visual C++, \	IN205
Relay Board Po	Hardware	Intronics SME 1000/5D				Hex value from termi	Binary to Opto	S0 to S3	S0 to S12			IN018; IN058
Bimba Pneuma	Hardware	D-13206-A-2	Controls 12 cylinders, driven by electric solenoi			Air supply from valve	Electrical Impulse					IN008; IN171

Screen grab of Hardware Inventory

Under the **Hardware** table of the inventory/abstraction, there are a number of fields that are relatively straightforward in their approach. **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 have an understanding of 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 cable.

Input/Output Signal (Analog)

IR (Infrared Range)
Voltages
Direct Current (DC)
RPM
Pulse-Width Modulation
VGA/EGA/EFM
Resistance/Impedance (ohms)
RF
Pressurized Gas
Frequency (Hz)/Amplitude/Decibels (dBs)
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
PID
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

Item Name	Directory	File Format	Extension	Abstract	Related Artwork	Related Hardware/ Software	I/O Protocol	Data Output	Device Driver	Variable
stealth_sensor2.db	unicom/actors_ICC/automata			Configuring laser beam detection an		mididd; stealth_	MIDI	MIDI Channel; I	MIDI; Serial; Lo	%frame, %ranc
stealth_sensor2.sh	unicom/actors_ICC/automata		.sh	Configuring laser be	Stealth	mididd; stealth_	MIDI	MIDI Channel; I	MIDI; Serial; Lo	%frame, %ranc
stealth_sensor3.db	unicom/actors_ICC/automata		.db	Configuring laser be	Stealth	mididd; stealth_	MIDI	MIDI Channel; I	MIDI; Serial; Lo	%frame, %ranc
stealth_sensor3.sh	unicom/actors_ICC/automata		.sh	Configuring laser be	Stealth	mididd; stealth_	MIDI	MIDI Channel; I	MIDI; Serial; Lo	%frame, %ranc
stealth_sensor4.db	unicom/actors_ICC/automata		.db	Configuring laser be	Stealth	mididd; stealth_	MIDI	MIDI Channel; I	MIDI; Serial; Lo	%frame, %ranc
stealth_sensor4.sh	unicom/actors_ICC/automata		.sh	Configuring laser be	Stealth	mididd; stealth_	MIDI	MIDI Channel; I	MIDI; Serial; Lo	%frame, %ranc
stealth_start.db	unicom/actors_ICC/automata		.db	Configuring laser be	Stealth	All subsequent	MIDI	Video Output: S	OMS; Serial	
stealth_start.sh	unicom/actors_ICC/automata		.sh	Configuring laser be	Stealth	All subsequent	MIDI	Video Output: S	OMS; Serial	
stealth_stop.db	unicom/actors_ICC/automata		.db	Configuring laser be	Stealth	All subsequent	MIDI	PID; Stepper M	Serial	
stealth_stop.sh	unicom/actors_ICC/automata		.sh	Configuring laser be	Stealth	All subsequent	MIDI	PID; Stepper M	Serial	
stealth_tjog.db	unicom/actors_ICC/automata		.db	Configuring laser be	Stealth	Address byte fc	MIDI	MIDI Channel; I	MIDI; OMS; Ser	%mnote
stealth_tjog.sh	unicom/actors_ICC/automata		.sh	Configuring laser be	Stealth	Address byte fc	MIDI	MIDI Channel; I	MIDI; OMS; Ser	%mnote
stealth_tlimit.db	unicom/actors_ICC/automata		.db	Configuring laser be	Stealth	Address byte fc	MIDI	MIDI Channel; I	MIDI; OMS; Ser	%mnote
stealth_tlimit.sh	unicom/actors_ICC/automata		.sh	Configuring laser be	Stealth	Address byte fc	MIDI	MIDI Channel; I	MIDI; OMS; Ser	%mnote

Unique Characters	Dependencies (OS, RAM/ baud rates, Peripherals, Processor)	Baud Rates	Risks	Documentation	Firmware	Programming Language	Operating System	Notes
Axdx\r; sp mf\r		9800 midi, 990(	stack registers;	OM037	PC-BIOS	C POSIX	GNU/Linux	hi/low hex values get progressively higher
Axdx\r; sp mf\r		9800 midi, 990(	stack registers;	OM037	PC-BIOS	C POSIX	GNU/Linux	hi/low hex values get progressively higher
Axdx\r; sp mf\r		9850 config, 98	stack registers;	OM037	PC-BIOS	C POSIX	GNU/Linux	hi/low hex values get progressively higher
Axdx\r; sp mf\r		9850 config, 98	stack registers;	OM037	PC-BIOS	C POSIX	GNU/Linux	hi/low hex values get progressively higher
Axdx\r; sp mf\r		9850 config, 98	stack registers;	OM037	PC-BIOS	C POSIX	GNU/Linux	hi/low hex values get progressively higher
Axdx\r; sp mf\r		9850 config, 98	stack registers;	OM037	PC-BIOS	C POSIX	GNU/Linux	hi/low hex values get progressively higher
Axdx\r; sp mf\r		9850 config, 98	stack registers;	OM037	PC-BIOS	C POSIX	GNU/Linux	hi/low hex values get progressively higher
Axdx\r; sp mf\r		9850 config, 98	stack registers;	OM037	PC-BIOS	C POSIX	GNU/Linux	hi/low hex values get progressively higher
p\r; Cxdx\r		9850 config; 99 stack registers; carriage returns			PC-BIOS	C POSIX	GNU/Linux	Main configuration shell command for Stealth, set t
p\r; Cxdx\r		9850 config; 99 stack registers; carriage returns			PC-BIOS	C POSIX	GNU/Linux	Main configuration shell command for Stealth, set t
p\r; Cxdx\r; F0\rD13-15@0\rG\r		9850 config; 99 stack registers; carriage returns			PC-BIOS	C POSIX	GNU/Linux	Main kill file for Stealth, PID output
p\r; Cxdx\r; F0\rD13-15@0\rG\r		9850 config; 99 stack registers; carriage returns			PC-BIOS	C POSIX	GNU/Linux	Main kill file for Stealth, PID output
		9850 config, 9800 MIDI, 9900 Device Driver			PC-BIOS	C POSIX	GNU/Linux	
		9850 config, 9800 MIDI, 9900 Device Driver			PC-BIOS	C POSIX	GNU/Linux	

Screengrab of Software Inventory

For **Software**, we also have a number of straightforward data columns which I like to refer to as “monolithic.” 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 C POSIX (a particular library structure within 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.

A **PID** (Process Identifier) is a temporary means of identifying a process within an operating system kernal.[38] In the case of *The Brotherhood* it is used to terminate a routine or command sequence which, enabled through ISR, would otherwise continue to accept new data in

---

[38] Microsoft Developer Network "GetIDProcessId function." http://msdn.microsoft.com/en-us/library/ms683215(v=vs.85).aspx, accessed May 2014.

order to continually accept new data as a string. With the PID the system in effect restarts and clears the data queue, allowing for the system to recallibrate. A PID allows the system to identifying a routine within the command line and "kill" the routine, often found in *The Brotherhood* within a shell script whose sole function is to kill the process of sending out data to the response hardware.

Solenoid Valves are air-pressure chambers which respond to electrical stimuli, creating the movement in the sculpture in *The Maiden*. The electrical stimuli is enabled through the Opto Brainboard, specific ports on the Opto board being hardwired to discrete appendages of the sculpture. In shell scripts we see this information represented as an alphanumeric string such as "FCK01" which corresponds to "headrest on", or FCL01 for "headrest off."

Excerpt from Opto notes, Russ Gritzo

wings(12)

	on	off	
relay 0	FCK01	FCL01	headrest
relay 1	FCK02	FCL02	breast rise
relay 2	FCK04	FCL04	pelvic roll
relay 3	FCK08	FCL08	breast situp
relay 4	FCK10	FCL10	hand left
relay 5	FCK20	FCL20	arm left
relay 6	FCK40	FCL40	arm right
relay 7	FCK80	FCL80	hand right

Text documents output ASCII characters whenever the system requires display text (such as the plotter arm in *Scribe*). These are often generated through Optical Character Recognition software that is found in the MicroDynII Speech Processing system.

All MIDI messages need to be conveyed in a manner in which a given computer system will understand them, and the protocol was built to be transposed as alphanumeric clusters which can be found in *The Brotherhood* scripts as **hexadecimals**. Hexadecimals are a positional numeral system where each digit represents four bits in which each bit must either be a 0 or a 1. This essentially assists in the notation of binary digit strings by reducing these numerical strings to more human-readable clusters. These values are expressed using numbers 0-9 (for bits 0-9) and letters a-f (for bits 10-15), expressing the total number of possible combinations based off of the 16-bit convention.[39] Thusly in computer scripts the hexadecimal is like a shorthand version of the binary representation of a byte which can be written more efficiently by a programmer and thus can be more broadly run off of any number of compilers. In using the hexadecimal it allowed Russ Gritzo a means of specifying the acceptable range of incoming values by virtue of the incoming bitstream and its binary information but without having to write out the entire line. Here is an example of how hexadecimals are programmed in the MIDI Nova System in the Brotherhood, a piece of hardware for triggering specific lighting configurations based off of a MIDI note:

Excerpt from NOVA MIDI lighting system user manual

```
All of these messages have no data bytes following (or they could
get interrupted themselves, obviously).  The messages:

0xf8   timing clock
0xf9   undefined
0xfa   start
0xfb   continue
0xfc   stop
0xfd   undefined
0xfe   active sensing
0xff   system reset
```

Hexadecimals are obviously a much more pervasive convention than simply for the purposes of interpreting MIDI streams. It is more generally used as a means of expressing the bit structure

[39] Tim Fischer, "Hexadecimals." PC Support, http://pcsupport.about.com/od/termshm/g/hexadecimal.htm, accessed May 2014.

and its intended use, as well as executing these functions. For example, a hex is output from the scripts in *The Brotherhood* as **pitch** and **velocity** of a MIDI note which, when present, will designate a specific response from the system. A **MIDI Channel** is also set so the system can differentiate simultaneous messages and ensure that they are being directed to the correct destination.

Example from "pad1.sh" in UNICOM

```
note_on
midi_channel 9
low   0x23
hi    0x23
low_velocity   0x01
hi_velocity    0x7f
end
```

In the above example from *Friendly Fire*, we see that an acceptable range of pitches and velocities must be precisely 0x23 for the pitch (MIDI bitstream = 00100011) and between 0x01 (1) and 0x7f (127) for velocity.[40] Depending on which pad is struck and how hard the visitor strikes the pad, the system will call up certain sequences of video and vary the speed with which sequences of video progress, respectively.

Similarly, **variables** are marked by the "%" sign in C, used to express an acceptable range of responses (often based on the presence of acceptable inputs represented in the Hex values for MIDI notes). These outputs will be formatted based on the native language of the peripheral hardware device.

Example from "stealth_xjog.sh" in UNICOM

---

40 Binary to decimal to hexadecimal convertor. http://www.mathsisfun.com/binary-decimal-hexadecimal-converter.html

```
  note_on
  midi_channel 1
  low   0x01
  hi    0x04
  low_velocity   0x00
  hi_velocity    0x7f
  end

if %mnote -lt 3
        10.5.1.17 9900 :ax jg200

if %mnote -gt 2
        10.5.1.17 9900 :ax jg-200
```

In the above example, the acceptable range of incoming MIDI notes (represented in hex) will result in the camera jumping greater than 200 spaces (jg200) on the x-y plotting table (generated by the laser-beam which strikes one of the four targets. When the laser hits one of the targets, for example, a MIDI note is generated that falls within the above hex values). Depending on if the note is less than 3 (lt 3) or greater than 2 (gt 2), the system will advance along the x axis either 200 spaces to the left or to the right. The other numbers refer to the IP address for the artwork (10.5.17) and the baud channel for delivering the message (9900, more information under the following section on Device Drivers).

The **PIC-Servo Motor** and **Stepper Motor** Controllers are what drive these mechanical robotic responses, specifically managing the successful completion of commands and registering preset commands such as home mode, reset, set motion/gain/baud rate, and control of other I/O peripherals. Electrical outputs for driving the motors is in the form of Pulse-Width Modulation (PWM) which stipulates specific duration of the voltage along with its intensity.[41] The two basic components of controlling the voltage signal is based on supply and load, varying in degrees

[41] Michael Barr, "Introduction to Pulse Width Modulation." Embedded, August 2001. http://www.embedded.com/electronics-blogs/beginner-s-corner/4023833/Introduction-to-Pulse-Width-Modulation, accessed May 2014.

with which the pulse is turned on or off. Servo and Stepper Motors differ slightly in that servo uses rotary motion which drives acceleration, velocity, and angular positions/targets for fluid motion whereas Stepper Motors use a rotary cog system for less fluid control of acceleration.[42] Based on their differences, Servo Motors require quite a bit more power (and computer logic for controlling) so Stepper Motors are used in situations where less precise robotic control will suffice. For example, Stepper Motors are used in *Automata* to drive the RPT head (much simpler motion control for finding anchor points based on directional commands) whereas *Translocations/Rails* uses Servo Motors (more complex control of x-y crossbars and rail/screen system.)

Example from "automata_midi3.sh" in UNICOM

```
#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
        # light 6
        }
```

In the above example we see how position aa, when invoked, will direct the RPT head through the stepper motor controller to velocity positions 350, 350, and 350 along the x-y-z axis. If the velocity of the midi note equals 6, then the **MIDI light** command will be sent to the NOVA lighting systems for configured light settings (seen above in the command sequence

---

[42] Festo, "10 tips for servos and steppers." http://www.festo.com/net/SupportPortal/Files/10257/Festo_10_Tips_servos_steppers.pdf

“ml12000,0,0). The commands for Servo Motors differ slightly, using the command “vel” for velocity rather than “vl.”

Continuing to use the above example from “automata_midi3.sh” we also see the command “15070 se” which, based on evidence from IN040 indicates that these are commands specific to a **Laserdisc Sequence**. The numbers pertain to a specific time-coded section on a laserdisc that can be commenced, in the case of *Automata* for mixing a 3D rendering of the space (recorded on laserdisc) with a **live-camera feed** (indicated by a device-specific prompt such as “mr” above). Finally, in some instances (again, as seen in the above example), MIDI is also used in its intended form as audio output. Prompt “mn 90 7f 07” indicates a **MIDI Note** which is used to provide an audio score for the installation, perhaps as a means of creating a more direct connotation with the initial user input such that the viewer notices a one-to-one connection between the musical instruments as controller (e.g. drum pad in *Friendly Fire*) and a generated tone.

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 creating an exhaustive list of **Device Drivers** that were created for the piece, 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 this bit of evidence:

Excerpt from OM037 on Device Drivers

OPTODD 9600
PICDD 9700
MIDIDD 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 &
```

This correlated very directly to numbers that appeared ubiquitously throughout the shell scripts as various hardware-specific commands were often preceded by four-digit numbers starting with 9. 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.[43] 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

43 “Modem Speeds,” Brainbell. http://www.brainbell.com/tutors/A+/Hardware/Modem_Speeds.htm, accessed May 2014.

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 such as Boulder's MAL. 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.

The Related Documentation field is not the only area of the inventory which invites further inquiry. Many unique prompts appeared within the scripts again and again, commands that do not readily appear in C POSIX libraries or existing manuals for programming in C. I decided that these commands should not be overlooked despite the fact that thus far we cannot find evidence within the documentation to support our claims. I also decided that if I was unable to interpret certain commands after consulting with experts and referencing programming guides, that more knowledge needs to be generated around the nature of these commands. Perhaps pointing to these clusters of code could be the nucleus for a greater understanding of their functions. In other words, would it not be possible that another researcher such as myself also wished to learn what a certain command in C was designed to do? If this collection exploited further study of these commands by tagging them as keywords within the archival record, perhaps the researchers would be led to script in *The Brotherhood* that contains these code clusters and be able to make further sense of their use in relation to their own source material. Maybe hypotheses can be drawn by comparing use of the same command within two different software programs which nevertheless are aiming for a similar goal. This would indeed be a superior example of data-gathering through effective tagging and crowd-sourcing.

In the example of *The Brotherhood*, these mysterious code clusters are often found within variable messages (marked by the "%" character) for expressing a range of acceptable inputs/outputs as well as the type of the data relative to what hardware is to be engaged through a given device drivers ("%vel" for velocity followed by a hex value is a common example in many of the shell scripts). The variables types are generally quite intelligible and human-readable, though many do not necessarily explain what type of action is to occur (e.g. "%compass" or "%running" are too vague to make any assumptions through reading alone). Additionally, there is frequent use of hash values (#) which are used in C in a number of ways. Hashes are largely used within Scribe for describing certain actions exploited through plotting arm (#rotate, #extend) and in the overall C syntax greatly resembles how variables are used throughout the other tables. This is unusual given that hashes are generally used in file headers as notations and are ignored when the code is enacted. However, in some popular examples of C in the early 90s it was also used as a flexible, customizable character which would be programmed to signify whatever the programmer intended throughout the entire script.[44] However, in examples from the scripts in *The Brotherhood* it is clear that they are related to executable commands.

```
10.5.1.9 9900 :FEK40
# rotate

shell msleep 500

10.5.1.9 9900 :FEK20
#extension

shell msleep 2500

10.5.1.9 9900 :FEK08
# vacuum
```

In the above example from "scribe_pageturn1.sh" we see a number of commands being sent out to specific solenoid valves via the Opto Board. Though, perhaps the specific valves being called out to (e.g. FEK40) is all that the computer needs in order to send this data to the Opto and activate the rotate function, the "#" character simply being a means of further linking certain actions with certain valves which carry out those robotic functions. Nonetheless, this will require further analysis and will be good to catalog such that these particular scripts can serve as good reference models for understanding configuration of the Opto Board.

Listing of Variables and Comments found in the source code for *The Brotherhood*

%mvel

%mnote

%loop

%compass

%activity

%frame

%random

---

[44] "Using hash (#) symbol." Stack Overflow. http://stackoverflow.com/questions/10242539/using-hash-symbol-to-indicate-input-newline-of-command-to-a-database-linkedl, accessed May 2014.

%switcher

%var1

%var2

%running

%video_mode

%sensor

rotate

extension

probe

vacuum

swing

Other examples of ambiguous code are found within the hardware-specific prompts through the device drivers, likely relating to specific names for pieces of hardware such as a specific video camera (e.g. in regards to the “mr” command for invocation of a live camera feed). While some deductions could be made based on the types of data reformatting that is present (which we definitively listed earlier) as well as the device driver in question, there is not enough empirical knowledge or evidence in the documentation in order to draw absolute conclusions. Thus my hope is that this inventory will not only be a guide to accessing the materials for scholarly research in Media Archaeology but will also solicit further participation and data gathering from the designated community.

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 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. Through engagement the Media Archaeology Lab in the study of the collection, 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.

Acknowledgements

Mona Jimenez, Associate Arts Professor and Assocation Director of the Moving Image Archiving and Preservation Program served as my advisor for this thesis project and was completely fundamental in developing this project. Not only did she tirelessly contribute her recommendations throughout the project and the editing of this paper, but she is also responsible for introducing me to the Vasulkas and embarking on this project in earnest. I am eternally indebted to her expert guidance and tutelage.

Russ Gritzo is truly a force to be reckoned with as is demonstrated in this monumantal work for which he served as engineer. Additionally, his voluminous notes on the process of creating *The Brotherhood* and contributions to interviews and correspondence made this project possible.

Deena Engel, Clinical Professor and Associate Director of Undergrade Studies for the Computer Science Minors programs at New York University, advised me througout the process of creating my inventory and leant her expert knowledge in computer programming to begin to deciper the UNICOM software package. Without her services I would not have been able to go nearly as deeply into the functionality of the programming and I am deeply appreciative of her patience, camraderie, and her incomparable genius.

Finally, I would like to acknowledge my superiors in the field of Archiving, Digital Preservation, and Museum Conservation who shared their thoughts and impressions on my project. Their feedback was all extremely formative throughout the process.

Howard Besser
Susanna Carlisle
Joshua Clayton
Michael Dulock
Lori Emerson
Ben Fino-Radin
Alex Gil
Dan Gillean
Andrea Goethals
Bruce Hamilton
Holley Long
Nancy McGovern
Don Mennerich
Bruce Montgomery
Peter Oleksik
Robert Rowe
Walker Sampson
Kari Smith
Dennis Tenen
Kara Van Malssen
Glenn Wharton

Bibliography

The Committee for Film Preservation and Public Access, “Preservation Without Access Is Pointless,” 1993. http://www.loc.gov/film/pdfs/fcmtefilmprespubaccess.pdf, last accessed 12/13/13

Society of American Archivists, “SAA Core Values Statement and Code of Ethics.” May, 2011. http://www2.archivists.org/statements/saa-core-values-statement-and-code-of-ethics, last accessed May, 2014.

American Institute of Conservation, “Code of Ethics and Guidelines for Practice.” August, 1994. http://www.nps.gov/training/tel/Guides/HPS1022_AIC_Code_of_Ethics.pdf%20, last accessed May, 2014.

Vincent Bonin, “Steina and Woody Vasulka fonds.” le fondation Daniel Langlois pour l’art, la science et la technologie, 2003. http://www.fondation-langlois.org/html/e/page.php?NumPage=422 last accessed May, 2014.

Woody Vasulka, “Rutt/Etra Scan Processor.” *Eigenwelt der Apparate Welt*, Ars Electronica, 1992. http://www.vasulka.org/archive/eigenwelt/pdf/136-139.pdf, accessed May 2014.

Vincent Bonin, “VCS3 (The Putney)” le fondation Daniel Langlois pour l’art, la science et la technologie, 2003. http://www.fondation-langlois.org/html/e/page.php?NumPage=446, accessed May 2014.

Steven Schoenherr, “The History of Magnetic Recording.” University of San Diego, 2005. http://www.aes.org/aeshc/docs/recording.technology.history/magnetic4.html, accessed May 2014.

Peter Manning, *Electronic and Computer Music,* p. 277, 1993. Clarendon Press: Oxford, New York.

Planet of Tune, “Index of Sequences,” 2013. http://www.planetoftunes.com/sequence/se_media/message.gif, last accessed 12/13/13

“The 8086 Interrupt Mechanism” http://ece425web.groups.et.byu.net/stable/labs/InterruptMechanism.html, accessed May 2014.

BonaFide OS Development “Interrupts, Exceptions, and IDTs: Part 1 - Interrupts, ISRs,” 2003. http://www.osdever.net/tutorials/interrupts.1.php, last accessed 12/13/13

Benj Edwards, “Birth of a Standard: The Intel 8086 Microprocessor,” http://www.pcworld.com/article/146957/article.html, last accessed 12/13/13

Russ Gritzo, personal interview, 08/05/13

Jeff Rothenberg, “Renewing *The Erl King*,” 2006. http://bampfa.berkeley.edu/about/ErlKingReport.pdf, last accessed 12/13/13

Tim Fisher, “Firmware.” PC Support. http://pcsupport.about.com/od/termsf/g/firmware.htm, accessed May 2014.

Russ Gritzo, “ExhFour,” http://www.vasulka.org/archive/ExhFOUR/Brno/brno.pdf, last accessed 12/13/13

Ray Kurzwell, “Law of Accelerating Returns.” Lifeboat Foundation Scientific Advisory Board. https://lifeboat.com/ex/law.of.accelerating.returns, accessed May 2014.

“Integrated Circuit Classification.” Integrated Circuit Help. http://integratedcircuithelp.com/classification.htm, accessed May 2014.
Gritzo, Russ. “User Manual for *The Brotherhood*” 1995.

Vickey Isley & Paul Smith, “In Homage of Change.” *The Preservation of Complex Objects Symposia (POCOS) Vol 2: Software Art*. JISC, November 2012. p. 56

Solomon R. Guggenheim Museum of Art, “Variable Media Initiative.” http://www.guggenheim.org/new-york/collections/conservation/conservation-projects/variable-media, accessed May 2014.

Variable Media Questionnaire, http://www.variablemedia.net/e/index.html, accessed May 2014.

Barbara Tillett, “What is FRBR? A Conceptual Model for the Bibliographic Universe.” Library of Congress Cataloging Distribution Service, http://www.loc.gov/cds/downloads/FRBR.PDF, accessed May 2014.

Richard Rinehard, “A System of Formal Notation for Scoring Works of Digital and Variable Media Art.” University of California, Berkeley. http://www.bampfa.berkeley.edu/about/formalnotation.pdf, accessed May 2014.

Michigan Historical Collection, University Archives and Records Program, and Digital Curation “Describing Digital Materials in Finding Aids.” Version 2.0, March 2013. http://deepblue.lib.umich.edu/bitstream/handle/2027.42/96874/BHL_DigitalDescriptionGuidelines_20130314.pdf?sequence=1, accessed May 2014.

DOCAM “Descriptive Table - Essential data-entry fields for documenting new media works and recommendations.” http://www.docam.ca/en/cataloguing-collections/descriptive-table-essential-data-entry-fields-.html, accessed May 2014.

Transcripted interview with Anne-Marie Zeppetelli, October 2009. http://www.docam.ca/en/bringing-together-information/three-approaches/168-entretien-avec-anne-marie-zeppetelli-archiviste-des-collections.html, accessed May 2014.

DOCAM Conservation and Preservation Committee, “About the Conservation Guide: A Preservation GUide for Technology-Based Artworks” http://www.docam.ca/en/conservation-guide.html, accessed May 2014.

Lori Emerson “Media Archaeology Lab - Why?” http://mediaarchaeologylab.com/about/why/, accessed May 2014.

Microsoft Developer Network “GetIDProcessId function.” http://msdn.microsoft.com/en-us/library/ms683215(v=vs.85).aspx, accessed May 2014.

Tim Fischer, “Hexadecimals.” PC Support, http://pcsupport.about.com/od/termshm/g/hexadecimal.htm, accessed May 2014.

Michael Barr, “Introduction to Pulse Width Modulation.” Embedded, August 2001. http://www.embedded.com/electronics-blogs/beginner-s-corner/4023833/Introduction-to-Pulse-Width-Modulation, accessed May 2014.

Festo, “10 tips for servos and steppers.” http://www.festo.com/net/SupportPortal/Files/10257/Festo_10_Tips_servos_steppers.pdf

"Modem Speeds," Brainbell. http://www.brainbell.com/tutors/A+/Hardware/Modem_Speeds.htm, accessed May 2014.

"Using hash (#) symbol." Stack Overflow. http://stackoverflow.com/questions/10242539/using-hash-symbol-to-indicate-input-newline-of-command-to-a-database-linkedl, accessed May 2014.