UNIVERSITY OF CALIFORNIA, IRVINE

# MCM

M O T I O N   C A P T U R E   M U S I C

---

## PHASE 1

## REQUIREMENTS SPECIFICATION DOCUMENT

---

Submitted by

Cayci Suitt
Gene Wie
Jimar Garcia
Salvador Ledezma

April 26, 2001
Revised May 17, 2001

# TABLE OF CONTENTS

# 1 Introduction

The Motion Capture Music System (MCM) is a software system that will be developed by the MCM Project Group for the University of California, Irvine (UCI) Dance Department. Its purpose shall be to interpret three-dimensional (3D) motion data and generate music based on the motion.

The project sponsors are Ms. Lisa Naugle, Assistant Professor, and Mr. Christopher Dobrian, Associate Professor, in the Dance and Music Department, respectively, of the UCI Claire Trevor School of the Arts (SOTA). Mr. Dobrian is also a professor in the Information and Computer Science (ICS) Department at UCI.

Currently, the project sponsors work with a Vicon Motion Systems (Vicon) Vicon8 Optical Motion Capture (Vicon8) System located in the School of the Arts Music Media Center. The Vicon8 System resides in the motion capture studio. In this room, there are eight infrared cameras that capture motion data from a predefined area of the room. The capture subject will don 1 to 48 spheres and move about within a cylindrical portion of the 3D space in the room. A circle is marked off on the ground to represent the projection of the 3D cylindrical space. Each of the spheres represents a point in 3D space. The cameras capture the points in motion at a variable rate specified by the user, typically 30 frames per second. At the present time, the motion capture is not used as input for creating new information. It records motion data via the network of cameras, but does not allow the simultaneous generation of a music or visual accompaniment based on the motion.

The project sponsors would like to expand the capabilities of the existing system so that in can be used in new and innovative ways, such as music composition. The MCM will interpret the motion data captured by the network of infrared cameras, and generate music in Musical Instrument Digital Interface (MIDI) format in real-time. This will allow the subject's specific movements to generate different pitches, change the intensity of the music, or change the voice of the instrument. The mapping of specific movements to pitch, tone, volume, and instrument will be customizable and it will allow the composer to load and save different mapping configurations.

Key features of the MCM system include real-time interpretation and a Graphical User Interface (GUI). MCM will translate the 3D motion data based on a user-specified mapping. This translation will be outputted in MIDI data in real-time. The GUI will allow the user to specify a mapping. It will allow the user to customize, save, and load files, which will contain specific mappings of motion to music.

The purpose of this requirements document is to define the specifications for the MCM software system and its interaction with the existing Vicon8 System. In addition to this introduction, this document details the requirements within the context of the following sections:

2. Project Plan – The project schedule and resources will be addressed in this section and it will include an identification of risks and how they will be mitigated.

3. Requirements Specification – This section will give a detailed overview of all the system requirements, including environmental constraints, use-case scenarios, and domain-specific rules that must be met.

4. Lifecycle Considerations – This section provides the context in which the development will take place and describes the process that the team will use for development.

5. Acceptance Requirements – This section provides the minimum acceptance requirements that must be met in order for the system to be acceptable by the project sponsors.

6. System Test Plan – In this section, the plan for testing will be presented. The test plan and its implementation will ensure that the software meets the requirements that will be stated in this document.

7. Project Deliverables – This section will describe the items that will be presented to the project sponsors upon completion of the project.

8. Glossary – Technical terms and acronyms presented in this document will be defined in the glossary.

9. Minutes – Minutes of all the meetings held by the project team in support of the development of this document will be included in this section.

10. Appendix – The appendix will contain any additional information that was used by the project team to develop this document.

11. Revisions – Revisions to this document will be listed in this section. They will include date of revision, the nature of the revision, the team member that made or requested the revision, and the sections of this document that are affected by the revision.

# 2 Project Plan

## 2.1 Project Schedule

The MCM system will be developed during the 11 Week Spring 2001 Trimester according to the following schedule:

| Task No. | Week | List of Tasks | Assigned Members[1] |
|---|---|---|---|
| 1 | 1 | Team Introduction and preliminary project review | All |
| 2 | 2 | Review existing documentation | All |
| 3 | 3 | Format Requirements document | Sal |
| 4 | 3 | Contact Vicon for technical details | Gene |
| 5 | 3 | Create use case scenarios | Cayci |
| 6 | 3 | Create test cases | Jimar |
| 7 | 3 | Gather individual team skills set | All |
| 8 | 3 | Contact Clients for meeting/presentation | Cayci |
| 9 | 3 | Finalize Requirements Document | All |
| 10 | 3-4 | Develop Team Web Page | All |
| 11 | 4-5 | Research Vicon Camera System | All |
| 12 | 4-5 | Research Java2/Swing | Cayci |
| 13 | 4-5 | Research Windows MFC, Socket Programming | Gene, Jimar, Sal |
| 14 | 4-5 | Research MIDI commands/interface | Gene, Jimar, Sal |
| 15 | 4-5 | Research Streaming Audio technologies | Gene, Jimar, Sal |
| 16 | 5 | Research Architectural Styles | All |
| 17 | 5-6 | Architectural Design | All |
| 18 | 6 | Produce Design Specifications | All |
| 19 | 6 | Develop capture/translation algorithms | All |
| 20 | 6 | Prepare Technical Presentation | Gene, Sal |
| 21 | 6 | Give Technical Presentation, Submit Report | All |
| 22 | 6-7 | Implement Mapping Functionality | Cayci |
| 23 | 6-7 | Implement Translation Functionality | Gene, Jimar, Sal |
| 24 | 8 | Test Mapping Functionality | Cayci |
| 25 | 8 | Test Translation | Gene, Jimar, Sal |
| 26 | 8 | Integration Testing | All |
| 27 | 8 | Usability Testing | All |
| 28 | 8-9 | Performance Analysis | All |
| 29 | 10 | Final Demonstration | All |
| 30 | 11 | Maintenance, Revision | All |

[1] All the team members are formally introduced in Section 2.3.1 Development Team

## 2.2 Project Risks

The project team has identified some potential risks that may affect the development of the MCM System. These risks will be under constant monitoring in order to eliminate them or minimize their effect. The risks identified are the following:

### 2.2.1 Software Speed

Description:

In order for the MCM system to be considered real-time, it must respond to the subject's movements within 33 to 50 milliseconds (ms). The delay must not be perceptible to the human ear.

Consequences:

If the system does not respond within the required time window, it will not be considered real-time. The subject will not be able to accurately tell which sounds correspond to which movements.

Likelihood of Occurrence:

It is highly likely that the software will not be able to respond within the required window of time.

Risk Avoidance:

Research and analysis will be done to ensure that the most efficient data structures and algorithms are used to minimize the running time of the translation of motion to music. The high-level programming language used for development will also be evaluated for its effect on response time.

Risk Monitoring:

Each implementation decision will be made based on its effect on the response time.

### 2.2.2 Hardware Unavailability or Not Fast Enough

Description:

The MCM system must run on a fast machine to be considered real-time. Any delay should be imperceptible to the human ear.

Consequences:

If the system does not respond within the required time window, it will not be considered real-time. The subject will not be able to accurately tell which sounds correspond to which movements.

Likelihood of Occurrence:

It is not likely that the software will run on an inadequate machine.

Risk Avoidance:

SOTA has committed to providing the appropriate hardware for the MCM System to run on.

Risk Monitoring:

Team members will keep in constant contact with the sponsors to make sure that the hardware is purchased and delivered on time.

### 2.2.3 Development Schedule

Description:

The project is on an accelerated schedule because it must be completed within the 11-week time constraints of the ICS Software Systems Development course.

Consequences:

If the team falls behind schedule, there is a possibility that the system will not be completed with the required amount of functionality.

Likelihood of Occurrence:

The nature of a trimester course does not preclude the possibility that the team will run out of time.

Risk Avoidance:

The team is committed to the project and has created management and accountability mechanisms to make sure that it does not fall behind schedule.

Risk Monitoring:

The team will be in constant contact with each other so that any potential delays can be identified in advance. The phase manager will also track the progress of the current phase and reassign tasks as needed to ensure that the schedule is met.

### 2.2.4 Future Changes

Description:

There is a possibility that there will be future enhancements requested by the sponsors after the trimester has ended.

Consequences:

The system will become obsolete if it is not maintained or improved.

Likelihood of Occurrence:

The sponsor had already indicated an interest in seeing a future version of the MCM capture multiple subjects and in doing non-linear mapping of multiple axes from the same spatial point to one MIDI command. The sponsor may also want to take into account subject movement velocity and acceleration.

Risk Avoidance:

Since the original team members will no longer be available, the source code will contain extensive documentation to ensure that future developers will understand it. The same coding style will be used for all parts of the system and extensive comments will be included. This, along with the phase documents, will make future development easier and more manageable.

Risk Monitoring:

The implementation phase manager, along with the team's input, will define a common coding style. The team will also conduct code reviews.

### 2.2.5 Lack of Proprietary System Knowledge

Description:

The MCM system will interface directly with the Vicon8 System, a proprietary motion capture hardware and software system. The team is not yet familiar with all the aspects and intricacies of this system.

Consequences:

If the team does not fully understand how the system works, then it may make incorrect assumptions about the Vicon8's interface. This may result in bugs in the system.

Likelihood of Occurrence:

If the manufacturer does not respond to our requests, then the team will not learn more about the Vicon8 system.

Risk Avoidance:

Efforts have been made to contact the company for information. Some team members have more knowledge about the system and they will educate the remaining team members, as they need it.

Risk Monitoring:

Team members are encouraged to ask questions as much as possible. Vicon will be continually called until it is able to provide the information that the team needs in order to proceed.

### 2.2.6 Not User-Friendly User Interface

Description:

The GUI may be developed in such a way that it is not intuitive to the user and it is difficult to navigate through the program's functions.

Consequences:

If the GUI is not user-friendly, users will not want to use the system, they will use it incorrectly, or it will not be used to its full potential.

Likelihood of Occurrence:

The GUI will be fairly simple. There is not a large selection of screens that the user will have to navigate, so it is unlikely that it will not be user-friendly.

Risk Avoidance:

Standard GUI formats will be used to the maximum extent possible and approval from the sponsor on the look and feel of the GUI will be obtained before it is finalized.

Risk Monitoring:

As new GUI functions are developed, they will be approved by the team and by the sponsor.

### 2.2.7 Personnel Shortfalls

Description:

There is a possibility that not all of the team members will be present throughout the entire quarter.

Consequences:

Some team members may be overwhelmed with work and in the worst case, the project may not be completed.

Likelihood of Occurrence:

It is certain that one team member will be gone for at least one month during the implementation phase of the project.

Risk Avoidance:

The tasks will be scheduled so that every team member will have an equal amount of work. If one team member will not be available during a period of time, then his/her tasks will be scheduled around the time when he/she will be gone. Also, there will still be phone and electronic mail (email) contact between team members throughout the quarter so nothing will be overlooked.

Risk Monitoring:

Team members will be in constant contact in order to assess progress of tasks and to identify any areas where extra effort must be placed.

### 2.2.8 Relationship Between Real-Time Emulator and Real-Time Engine is Unknown

Description:

Development is being planned on the assumption that the real-time emulator and the real-time engine work the same and provide the same functionality. In reality this may not be so.

Consequences:

The software developed may not work exactly as expected or it may contain bugs if the emulator and actual engine work differently.

Likelihood of Occurrence:

The emulator models the actual real-time engine, so it is unlikely that they will work differently or contain different interfaces.

Risk Avoidance:

Vicon will be contacted in order to get as much information as possible on the differences between the emulator and the engine. Testing of the MCM will be conducted on both to ensure that they are indeed compatible.

Risk Monitoring:

Integration testing will ensure that the MCM is compatible with the real-time engine.

### 2.2.9 Camera Hardware Uses Up Too Much Time in Capturing Data

Description:

The total system response time, including the Vicon8 capture hardware and the MCM system must be in the range 33 to 50 ms. It is possible that the hardware will take too much time in capturing and creating the data packets. The MCM system may not be able to reduce its execution time to meet the response requirements.

Consequences:

If the system does not respond within the required time window, it will not be considered real-time. The subject will not be able to accurately tell which sounds correspond to which movements.

Likelihood of Occurrence:

It is highly likely that the Vicon8 hardware motion capture time plus the MCM software time will be greater than 50 ms.

Risk Avoidance:

In addition to the research and analysis mentioned above in Risk 2.2.1, further research must be completed on the Vicon8 system to determine and benchmark its response time. Several different algorithms will be considered so that the best performing one will be used.

Risk Monitoring:

Extensive testing will be done to ensure that the response time is adequate and that the most efficient algorithm is chosen.

## 2.3 Project Resources

### 2.3.1 Development Team

The MCM Project Team is composed of four members. Each of the members has unique personal and technical strengths that will contribute to the success of the project. The team members are:

- Cayci Suitt
- Gene Wie
- Jimar Garcia
- Salvador Ledezma

The following is a description of each team member's background and qualifications for completing this project.

*Cayci Suitt* is a fifth year ICS and Drama double major. One of her personal strengths is her commitment to the team and to completing the project in a professional manner. She is a hard worker and strives for excellence. As she can still remember what it was like to be an Arts major with little computer knowledge, she will have the ability to design an User Interface that will be easily useable for those users whose specialty is in Dance or Music. Of her technical strengths Java programming will be a useful asset to the team. Cayci likewise has an appreciation for proven software engineering techniques and looks forward to using them to complete this software system.

*Gene Wie* is a fifth year ICS Major who plans to attend a graduate program in music performance following the completion of his undergraduate degree. His dual interests in music and technology support the MCM project on all fronts, from the musical concepts involved in the interface to the technical aspects of the translation system. His background in programming, Java and C++, both feature prominently in the project tasks. Gene's aptitude for technical documentation and presentations is his other contribution to the group talent pool.

*Jimar Garcia* is a third year ICS major, minoring in Management. His strengths include software system design and implementation. Jimar has 3+ years of programming experience including C++, Java, Visual Basic, and MFC.

*Salvador Ledezma* is a self-motivated, hard-working graduate student in ICS. His technical skills include Java, C++, and Visual Basic. He has good interpersonal skills and enjoys working with other people. Salvador has experience working with the Windows NT/2000, Me, and 9x platforms, as well as a working knowledge of the UNIX and Mac environment. He enjoys low-level programming at the software/hardware interface and he likes to get "close to the machine." After receiving his Master's degree, he plans to be a Software Engineer for a company specializing in embedded, real-time systems.

### 2.3.2 Hardware and Software Requirements

The MCM project team shall use standard, readily available development environments and workstations. The development of the MCM system will not require the purchase of any new hardware or software.

Hardware

The MCM team shall develop the system on the standard Windows NT workstations located in the ICS Labs. Alternatively, if some team members choose to work from home, they shall use their own personal machines to develop the system.

In addition to the development machines, the project team shall test the MCM software by installing it on the network on which the Vicon8 system resides. The Vicon8 system includes infrared cameras, a proprietary camera data-decoding box, a recording workstation, and the real-time system, which converts the motion data into video. In the absence of the real-time system, a real-time emulator shall be substituted. It will also be provided by Vicon.

Software

The MCM system will operate on a system running Microsoft Windows NT/2000 operating system. All development and document shall be produced using existing standardized software that is readily available to the project team, or that is installed on the Windows NT workstations located in the ICS Labs.

Some of the software that shall be used for development shall include, but shall not be limited too:

- Microsoft Visual C++ 6.0
- Symantec Visual Café 4.01c
- Ethereal Network Packet Analyzer
- Microsoft Word and PowerPoint

The above-mentioned software is recommended, but not required to be used by the project team. Any equivalent development environment or tools may be substituted at the discretion of the project team.

## 2.4 Project Team Organization

### 2.4.1 Team Structure

Each major phase of the project: Requirements Specification, Technical Review and Report, Design Specification, Implementation, and Integration and Testing will have a separate phase leader. This will give each member an opportunity to gain leadership and management experience and it will allow the tasks to be evenly distributed among all the team members.

In addition to a phase leader, each phase will have a phase clerk. The remaining team members will support the leader with the goal of completing the phase correctly and on schedule.

The phases and their respective leaders are listed below:

| **Phase No.** | **Phase** | **Leader** | **Clerk** |
|---|---|---|---|
| 1 | Requirements Specification | Salvador Ledezma | Cayci Suitt |
| 2 | Technical Report | Gene Wie | Salvador Ledezma |
| 3 | Design Specification | Jimar Garcia | Gene Wie |
| 4 | Implementation | Gene Wie | Salvador Ledezma |
| 5 | Integration and Acceptance | Cayci Suitt | Jimar Garcia |

### 2.4.2 Member Responsibilities

<u>Phase Leader</u>

The phase leader will guide the team members to ensure that the phase gets completed correctly and on schedule. He/she will have the responsibility of allocating resources and assigning tasks necessary for phase completion. The leader will set up meetings, as needed, and will track the status of the phase against the schedule. Decisions made by the leader are considered final, unless a majority of the team members vetoes the decision. In this case, the team will have to have a brainstorming session to come up with a solution that everyone agrees with. The phase leader is also responsible for putting together the final document to be submitted for the phase.

<u>Phase Clerk</u>

The Phase Clerk will have the responsibility of writing and distributing all the meeting minutes. The clerk is responsible for communicating via email any important announcements or issues that occur during this particular phase of the project development cycle. The clerk will also be responsible for updating the project website with announcements, if any, or relevant documents that will be produced in this phase.

<u>Other Team Members</u>

The remaining team members will perform tasks assigned by the Phase Leader in support of successful phase completion.

In addition to the above-mentioned responsibilities, each team member has been assigned various technology areas and research tasks for possible use in the implementation of the MCM System. The research areas have been defined as follows:

| No. | Technology | Member Responsible |
|---|---|---|
| 1 | Java2/Swing Components | Cayci |
| 2 | Windows MFC[2], Socket Programming | Gene, Sal |
| 3 | MIDI Commands and Interface | Gene, Cayci |
| 4 | Streaming Audio Technologies | Jimar |
| 5 | Algorithm Analysis and Development | Sal, Jimar |

## 2.5 Tracking and Control Mechanisms

Each phase manager is responsible for the timely completion of all the tasks required for the phase. He/she will assign, reassign, or complete all the tasks, as appropriate. The phase manager is also responsible for managing the documents produced in the phase, including revision tracking.

Meetings will be the time to review the current phase status, to review documentation, and to review source code. Meeting minutes will be kept to ensure that all the team members understand all assignments, assumptions, and action items. Email will be used to pass on important information to team members in between scheduled meetings.

Given the fact that all team members will work on the same set of source code and document files, the team needs to be especially careful when modifying electronic files to ensure that no detrimental modifications or deletions are made. The following are the mechanisms that the team has developed:

All files will reside in a set of hierarchical directories that are only accessible to team members. All phases of the project will have its own directory. Within this directory will be directories for each file. If a team member will work on a file, then he/she will make a copy of it and concatenate the old file name with his/her initials to indicate that the file is “checked out” by this person. Once modifications are complete, then the file is saved with its name plus the date of the revision. It will be saved in a new directory to indicate that it is a new version.

A text file log will also be kept in the file directory. Each team member is responsible for completing the log each time a file is modified. The log will track who made the modification, the date of the modification, and the nature of the modification.

Source code files will be written with the same style and will have the same naming convention. Comments will contain the name of the module, its purpose, and will also indicate any changes made to it, who made them, and the date of the change.

[2] Microsoft Foundation Classes

# 3 Requirements Specification

## 3.1 Overview of System Requirements

Currently, the Vicon8 system captures data and saves it to hard disk in a standardized C3D data format. This format allows the captured motion data to be used in developing animation and realistic computer modeling of body shapes and movement.

The MCM system will enhance the current system by allowing the subject's movement to be immediately translated into audio signals. The mapping of movement to sound will be explicitly defined beforehand. The subject will hear the response to his/her movements and direct the nature of what sound is produced. In essence, the subject will be able to compose music only by the movement of his/her body in 3D space.

The MCM system will interpret data created by the Vicon8 system. The MCM system will be placed in the same network as the Vicon8 system and the workstation that records the motion data. The motion data is formatted into a C3D data stream. The MCM system will translate the C3D data into MIDI compatible statements for output to a MIDI device. MCM will intercept the packets sent by the Vicon8 system and process them.

User input through a GUI will define the mapping of the subject's movements to the appropriate MIDI musical response. For example, an arm motion can be defined to change pitch, volume, instrument type, or other MIDI-compatible instruction.

The MCM system will be based on the following specifications:

## 3.2 Environmental Characteristics

### 3.2.1 Hardware

The MCM system shall have its own workstation separate from the Vicon8 system. At a minimum, the workstation will have an Intel Pentium III processor running at 600 MHz or higher with at least 128 MB of RAM and at least 4 GB of hard disk space.

### 3.2.2 Software

The MCM system will operate on a system running Microsoft Windows NT/2000.

### 3.2.3 Network

The Vicon8 system and the MCM system will run on an isolated TCP/IP network. The bandwidth capacity of the network is 10 Megabits per second (Mbps).

### 3.2.4 Users

The MCM will be used exclusively by SOTA. It will be located in a workstation on an isolated network in a room not normally accessible by the public and it will not contain sensitive data. For this reason, no special security requirements or user restrictions will be implemented.

## 3.3 Non-functional Constraints

### 3.3.1 Correctness

The MCM system shall always output the MIDI commands as expected by the user per the choices input. If the correct outputs were not created, the music created would not be what the user expects and could lead to displeasing sounding compositions.

### 3.3.2 Reliability

MCM shall never crash as a result of lack of data or wrong input.

### 3.3.3 Robustness

Given the relatively small amount of information being transferred in the bandwidth available, MCM has no provisions for the minimization of data loss. If a user enters incorrect data in the mapping component of the system, it shall handle this error elegantly and notify the user that invalid input has been received.

### 3.3.4 Performance

The time latency between the motion of the capture subject and the output of MIDI data shall fall in the range between 33 and 50 ms, a threshold at which a human listener can perceive a "delay." One of the greatest constraints on the system design will be the speed at which MCM can capture data being sent through the network hub, which has a hardware bandwidth capacity of 10 Mbps. The overall performance of the system will depend heavily on the hardware it is running on (i.e., the power of the computer processor and speed of its components). The time latency shall never be perceptible to the human listener.

### 3.3.5 User-friendliness

Some users of the MCM system are not familiar with the use of computers. For this reason, the MCM shall be intuitive and easy to use.

### 3.3.6 Verifiability

The MCM shall be developed so that it can be easily verified against this requirements document. In addition, the MCM shall be demonstrated to the sponsor during the tenth week of the trimester to allow the sponsor to verify that the stated requirements have been met. The project team shall be available during week 11 to address any issues that may arise once the sponsor has taken control of the system.

### 3.3.7 Maintainability

The project team will be unavailable to maintain the system in the future once the course has been complete. For this reason, it is imperative that the system be as complete and error-free as possible. However, it shall also be implemented in such a way that it will be as easy as possible for other developers to add to and maintain the system. The system shall be fully documented and the code shall be fully commented to ensure that future developers of the system can understand the methodologies and techniques. Modules shall be designed to work independently of each other and so that they will communicate with each other only through well-defined interfaces.

### 3.3.8 Reparability

The system test plan shall be as comprehensive as possible to ensure that the system undergo heavy testing. This will identify as many faults as possible. The system shall implemented in such a way that any errors found are easy to repair.

### 3.3.9 Safety

The MCM does not have the ability to compromise the well being of any organic system.

### 3.3.10 Evolvability

Numerous other features may be added to the system, and Vicon may be interested in the MCM system in the future. Also, the base system may be expanded with additional functionality in the future. For this reason, the system shall be built in such a way that it may easily evolve.

### 3.3.11 Reusability

Vicon has expressed interest in this software system and it may be reused outside of a university setting. Therefore, it shall be designed to be reusable.

### 3.3.12 Portability

As the system is being developed for a workstation running Windows NT/2000, usage shall be limited to that platform and its emulators.

### 3.3.13 Understandability

This system shall be implemented in a manner that is easily understandable and changeable. It shall be designed so that users of the system will understand how it works and how to interact with the system.

### 3.3.14 Interoperability

The MCM shall interoperate with the Vicon8 system in order to intercept the 3D motion data. The entire system operates on a closed network with only three clients including the Vicon8 cameras, it's accompanying workstation, and MCM system. No outside access to the Internet or other networks exists, thus negating the possibility of any external influences.

### 3.3.15 Productivity

Due to the limited time constraints of the project, the project team must remain as productive as possible. Weekly meetings within the team and with the professor of the course shall ensure that productivity and quality is maintained. Furthermore, once completed, the MCM system will allow SOTA to use the Vicon8 system in new and innovative ways that were not possible before.

### 3.3.16 Size / Scalability

Although no estimates on the size of the system can be made as of this stage of development, it is expected that the program will be of a reasonable size for distribution on a single floppy disk of 1.44 MB capacity. It shall be designed to accommodate an increase in the amount of data generated by the Vicon8 system.

### 3.3.17 Timeliness

This is an important quality of the MCM project and its success depends on its timely completion. Strategic deadlines shall be created and met to ensure that the system will be developed on time.

### 3.3.18 Visibility

Any problems that occur during the development lifecycle of the MCM shall be brought forth and discussed with the sponsor and the professor of the course so that they are dealt with in a timely manner.

## 3.4 Domain Specific Rules

The domain of the MCM system is the Vicon8 system, which includes the motion capture cameras, the workstation that stores the C3D data, and the underlying network used for transporting the data packets. The MCM shall allow the system user to translate captured 3D motion data into MIDI music data in real-time, within certain limitations.

### 3.4.1 Introduction

The MCM system shall perform two main and discrete functions. First, it shall allow the user to define a mapping. A mapping is composed of MIDI commands and movement in 3D space. Specifically, it indicates what MIDI commands shall be carried out in response to a particular predefined movement in space. This functionality shall be referred to as the Mapping Component of the MCM system. Once the user has defined the mapping, it can be saved to hard disk. Alternatively, an existing mapping may also be loaded from disk.

The second important functionality is the implementation of the mapping component on actual motion data that is received. This shall be referred to as the Translation

Component of the MCM system. The Vicon8 system will generate the motion data. The Translation Component shall receive the data and, based on the specifications that are defined in a given mapping, it shall output the appropriate MIDI command. The MIDI command value is a direct function of the 3D motion data. This function that computes the MIDI command value is derived from the user-defined mapping.

### 3.4.2 Transfer of Data

The transfer of data in the system shall take place as follows:

The Vicon8 system includes an Application Programming Interface (API). The API provides a library of functions that allows other programs to "interact" with the Vicon8 system. Specifically, the MCM system shall use these functions to interact with the Vicon8 system and ask for the C3D motion data that is being generated. When the data is transferred over to the MCM system, the data shall be considered "captured" by the MCM system.

Typically, the Vicon8 system generates motion data at the rate of thirty frames per second. The Translation Component via the Vicon8 API shall continuously poll the Vicon8 system to capture the latest data. Each time that new capture data is retrieved shall be referred to as a capture event. The captured C3D motion data shall be stored in a continuously appended data structure in memory (or on disk, depending on available system resources).

The translation engine of the MCM shall read the C3D motion data and given the mappings specified by the user in the GUI, shall output the correlated MIDI command to the defined MIDI output device. For reference, a listing of MIDI commands is provided in the Appendix of this document. Section 3.8.1 describes the general format of the captured C3D data.

### 3.4.3 Motion Data Mapping

The mapping of the motion data to audio response data shall be one-to-one. The user shall define the range of actual movement that shall be measured and the range of the output MIDI command that shall correspond to the movement. Movement from a specific point along the x-, y-, or z-axis to another point along the same axis shall correspond to a predefined change in the MIDI command output.

The MIDI range shall be based on the range of movement for the particular axis it is defined for. It shall be calculated after a mapping has been defined and right before the translation will begin. It shall be considered to be part of the initializing of the Translation component of the system.

### 3.4.4 Data Point Placement

The number and placement of the data points/spheres on the capture subject is independent of the MCM system as long as they are specified when the mapping is defined by the user. For purposes of this document, these variables shall be assumed to be based on the standard setup for the Vicon8 system and is left to the discretion of the sponsor.

### 3.4.5 Data Range

The user shall specify values within the minimum and maximum data ranges for both input and output. These shall be selected from the available data ranges of the Vicon8 system and the MIDI output device.

The input ranges for each of the data points on the capture subject shall be within the absolute range of the 3D cylindrical space in the room captured by the Vicon8 cameras, and can include any portion equal to or smaller than the whole. The physical distance between the camera mounts and lens defines the exact range.

The output ranges shall be within the limits of pitch, volume, and selected instrument as defined by the MIDI standard**.** If the user attempts to input a number that is out of the required ranges then an error message shall be displayed. If a capture subject moves out of the specified ranges (either the absolute of the Vicon system or the user-specified range), then the information shall be ignored and discarded.

At the time that the translation is begun, the Translation Component shall compute a one-to-one functional relationship between the input range and the output range. A change from point A to point B in motion, along some axis, shall correspond to a change from some function of A MIDI output value to some function of B MIDI output value. This function shall be a linear function.

This has a direct cause-and-effect relationship on the output. A small movement by the capture subject between capture events shall cause a small change in the audio output. Similarly, a large movement between capture events shall cause a functionally related substantial change in the audio output.

Relating 3D space points directly to MIDI output values results in an absolute mapping of MIDI output values. For example, a point on the z-axis in 3D space shall always correspond to the same MIDI output value, regardless of how the capture subject arrived at the current point.

### 3.4.6 Pairing of Data Point Axis with MIDI Command

Each pairing of a data point axis and MIDI command shall require one MIDI track.

### 3.4.7 MIDI Command Selection

The user shall be able to choose among pitch, pitch-bend, aftertouch, volume, continuous control, program change, and voice change to define an axis for a data point. This definition will allow changes along the axis to be interpreted as a change in the particular MIDI command. If pitch ("note") is chosen then the user shall also input the duration of the notes played.

### 3.4.8 Axis Mapping

The user shall be able to map only one axis for a data point to a MIDI command for each track. If the user would like to map multiple axes of a point, he/she will have to define each of them separately. These multiple axes for each data point may be mapped to different MIDI commands.

An individual axis may also be mapped to multiple tracks. For instance, the x-axis on the left hand may be interpreted as volume change and as pitch-bend, and the y-axis for the same data point may be mapped to pitch, a different MIDI command.

#### 3.4.9 Multiple Mapping Configurations

The user shall have the ability to save and load mapping configurations. It shall be necessary to save a new mapping or to choose an older mapping before the capture/interpretation can begin. The creation of mappings shall be handled by the separate mapping component. It shall be executed prior to running of the motion translation component.

#### 3.4.10 Note Velocity

The MIDI command for the velocity of each note played shall be standardized at the default value for the connected MIDI output device.

#### 3.4.11 Other Domain-Specific Rules

The following domain-specific rules are non-functional constraints, but are required in order for the MCM to work correctly with the Vicon8 system:

- The capture subject must stay within the 3 meter high by 3-meter diameter cylindrical 3D space in order for the Vicon8 cameras to capture the motion. The cylindrical space is projected into a marked circle on the floor so that the subject can gauge if he/she is in still in a viable area.
- The MCM must be connected to the same enclosed network as the Vicon8 system in order to capture and interpret the motion data in real-time. If not connected, the translation of motion to audio can still occur, but it will not be in real-time.
- The infrared cameras of the Vicon8 system capture motion and convert it to a standardized C3D data format. The MCM shall be able to read and translate the motion data in this format before it can convert it to MIDI audio data.

### 3.5 UI Model

The GUI shall allow the user to create, save, and/or load a mapping for the motion interpretation. The GUI shall present the following information to the user:

- The MCM GUI shall contain a menu of the standard Vicon8 data point labels from which the user can choose the parts of the body where the spheres shall be placed. The menu shall be called “Label”. This has the effect of selecting the number of data points and axes the system will interpret and tracks to output. Only available system resources to process the capture data limit the number of possible tracks.

- ?? Once a data point/body part has been selected from the “Label” menu, text fields and additional menus (if any) shall appear that allow the user to modify specific values for that specific data point. Each data point shall be in its own section and the section is designated as a row.

- ?? Included under “Label” menu shall be a menu item that the user can select to delete a row. When this item is chosen a pop-up dialog box shall appear that will allow the user to choose which existing row he/she wants to delete. It shall be called “Delete Row”.

- ?? Each row shall include a set of inputs associated with the 3D data input and shall be listed first from the left. There shall be a set of radio buttons from which the user must choose x, y, or z as the axis to be interpreted. There shall be text fields in which the user must input the minimum and maximum input values, based on the physical range of the capture area.

- ?? There shall also be a menu from which the user must choose the specific MIDI command to be associated with that axis and data point. Next from the left will be text fields in which the user must input the minimum and maximum values for the MIDI output. There shall then be a menu that will allow the user to input the channel associated with that line of interpretation. For example, if the user has chosen "pitch" from the list of MIDI commands, then another text field will appear on the far right in which the user must input the duration of the notes played in ms.

- ?? Under the standard “File” menu, there shall be a “Save” menu item, which the user can choose to save the mapping, and an “Open” menu item button, which the user can select to load a previously saved mapping.

- ?? The user shall be able to select a menu item that will start a translation or stop the current translation once it has begun. These menu items shall be under a “Translate” menu and they shall be called “Run Translation” and “Stop Translation”, respectively. This menu item shall control the start and stop of the translation component and control the generation of MIDI output based on the currently defined mapping.

- ?? The “File” menu shall have options for starting a new mapping and for closing the current mapping. If the current mapping is not saved, the user shall be given the option to save it.

- ?? There shall be a “Help” menu which will contain an “About” menu item and a “Help” menu item. The “Help” menu item shall open a general informational file that provides a basic description of the system. It shall also provide a navigational overview of how to use the system.

Below are some typical screenshots of the MCM GUI:


Figure 1: When the application is first started, the screen will be empty.

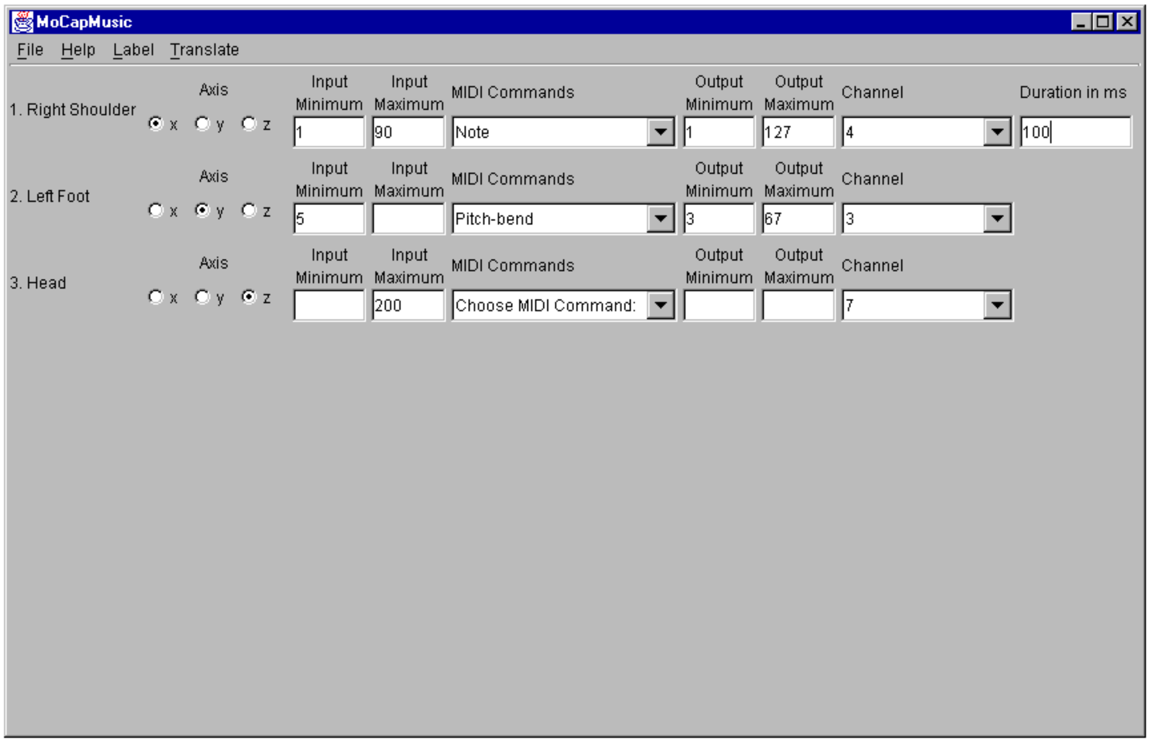


Figure 2: This screen shows some typical user selections for various body parts and their associated MIDI commands.

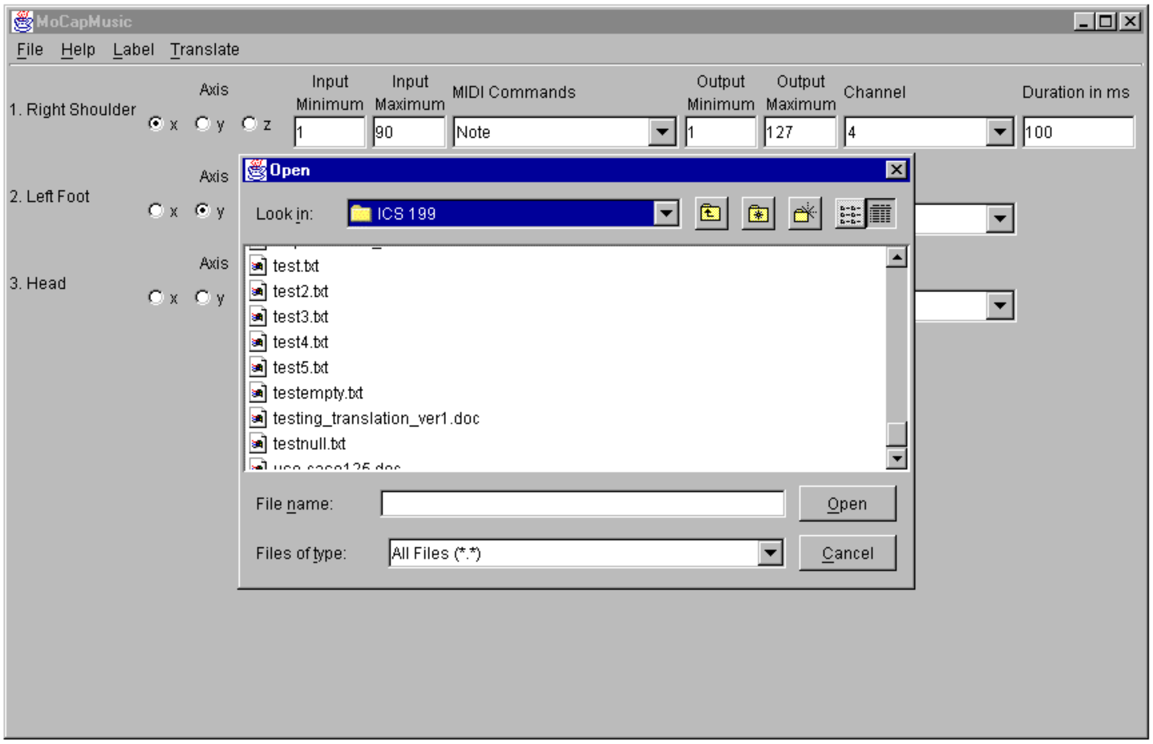


Figure 3: Opening and saving mapping files shall be done using standard Microsoft Windows dialog boxes.

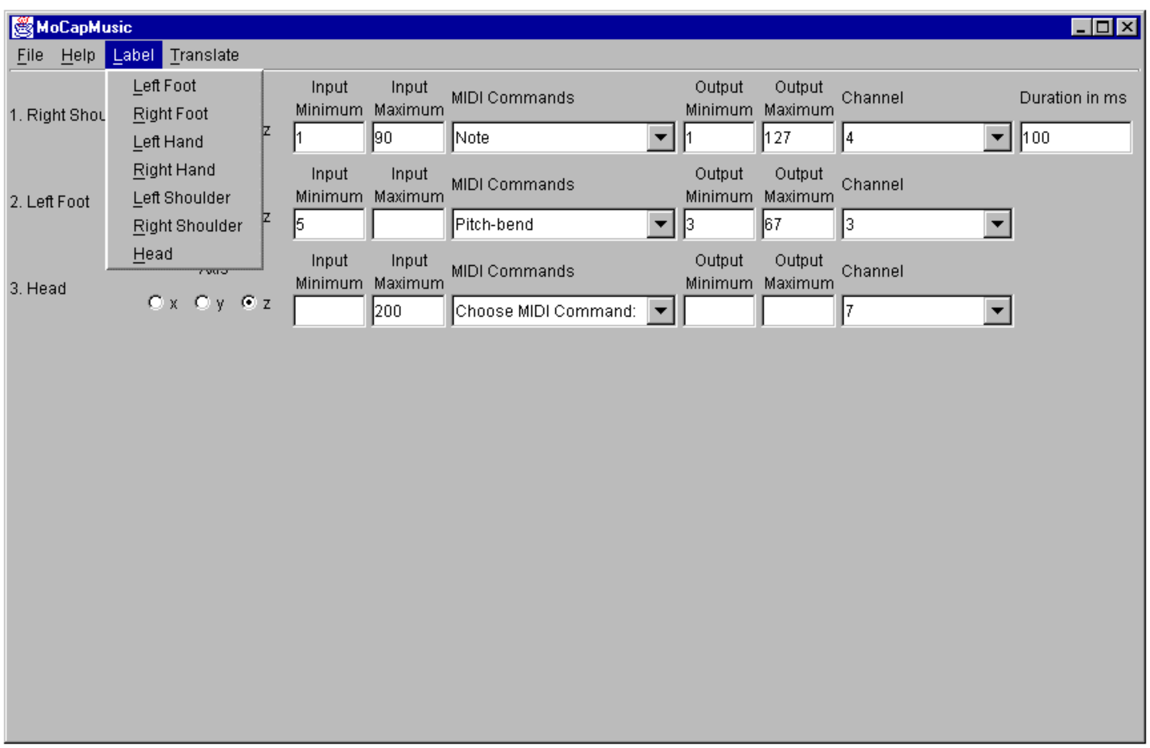

Figure 4: In addition to standard Microsoft Windows menus, additional menus will allow the user to select different body parts to be mapped to MIDI commands.

## 3.6 Systems and Dataflow Analysis

The MCM system shall be installed on its own workstation. The workstation shall be connected to the Vicon8 network, where it will be able to obtain the 3D motion data that the Vicon8 system of cameras has captured. The raw data is sent to the Vicon8 workstation where it is processed into a C3D format.

The Vicon8 workstation then sends the data stream to the real-time engine, which converts the data in C3D format into a video display of motion in real-time. The MCM shall perform a similar function, but instead of video, it shall convert the C3D data in MIDI audio. The C3D data format is discussed in Section 3.8.1.

The following is a high-level dataflow and hardware diagram of the entire Vicon8 system, including the MCM as defined in this document:

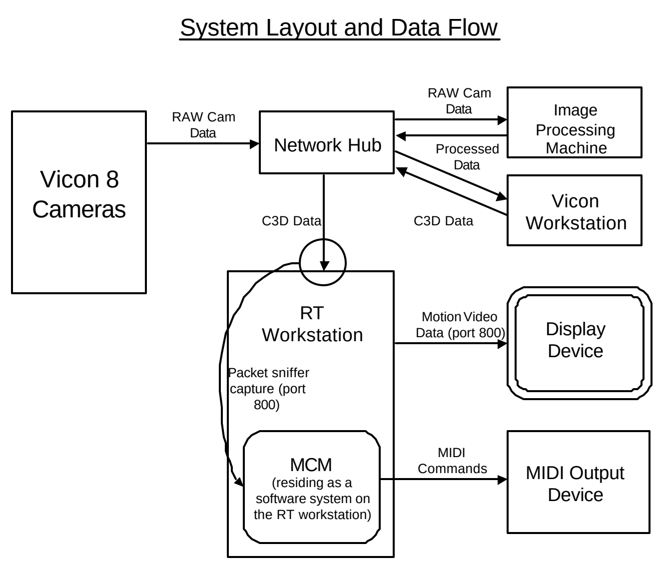


Figure 5: MCM System Layout and Data Flow

## 3.7 Use-Case Scenarios

Use-case Scenarios have been defined based on the specifications of Section 3.5, The UI Model. For more specific information on the components discussed here, refer to Section 3.5.

The following use-case scenarios will be handled by the MCM system GUI:

**Use-Case Scenario 1**

**Title:** "User chooses a body part from the Label Menu"

**Pre-conditions:** The mapping component of MCM is running.
**Post-conditions:** A row of inputs (UI components) has been added to screen.
**Priority:** High

**Scenario:**
The user clicks on the Label menu from the menu bar.
The user clicks on a particular body part (i.e. Left Shoulder).
A new row of UI components appears as the first row or under the last row added.

**Use-Case Scenario 2**

**Title:** "User chooses Save from the File menu with valid inputs"

**Pre-conditions:** The mapping component is running; user has entered valid information into the UI components.
**Post-conditions:** All the information from the UI components is saved to a file of the correct name and place.
**Priority:** High

**Scenario:**
The user chooses one or more body part labels, thus adding rows of UI components to the screen.
The user inputs valid information into some or all components. Valid information is defined as data that is within the defined ranges for the physical input and for the audio output.
The user chooses Save from the File menu.
A dialog box appears in which the user chooses the name of the file and the place to store that file, and clicks "Okay".
The information from the corresponding components are saved to a file of the correct name and place, and all empty components are considered "null" in the mapping file.

**Use-Case Scenario 3**

**Title:** "User chooses Open from the File menu"

**Pre-conditions:** MCM is running
**Post-conditions:** The correct file, and thus the correct UI components and corresponding information appear on the screen.
**Priority:** High

**Scenario:**
The user chooses Open from the File menu.
A dialog box appears in which the user chooses the file they wish to open, and clicks "Okay".
The corresponding file is "opened": the correct number of rows of UI components fills the screen and they reflect the correct information from the file.

**Use-Case Scenario 4**

**Title:** "User chooses Run Translation from the Translate menu with a valid file"

| | |
|---|---|
| **Pre-conditions:** | MCM is running; the MCM translation component exists on the same machine; a valid file is open. |
| **Post-condition:** | The translation component translates the movements of the capture subject per the specifications in the chosen file. |
| **Priority:** | High |

**Scenario:**
The user opens or creates a new file and enters complete and valid mapping information.
A capture subject is ready for a capture.
The user chooses Run Translation from the Translate menu.
The mapping component saves the file and calls the translation component with that file (and path) as an argument.
The translation component reads and parses the file.
The translation component sends commands to the MIDI port on the host machine as a capture subject moves.

**Use-Case Scenario 5**

**Title:** "User chooses Run Translation from the Translate menu with an invalid file"

| | |
|---|---|
| **Pre-conditions:** | MCM is running; the MCM translation component exists on the same machine; an invalid file is open. |
| **Post-condition:** | The user is asked to correct errors and to "fill in holes" in the mapping file. |
| **Priority:** | High |

**Scenario:**
The user opens or creates a new file and enters incomplete and/or invalid mapping information.
The user chooses Run Translation from the Translate menu.
The user is asked to correct errors in the format or range of the mapping information.
The user is asked to complete omissions in the mapping information.
Provided the user has complied, the mapping component saves the file and calls the translation component with that file (and path) as an argument.

**Use-Case Scenario 6**

**Title:** "User chooses Save from the File menu with invalid inputs"

| | |
|---|---|
| **Pre-conditions:** | The MCM mapping component is running; at least one row of UI components is on the screen and at least one contains information that is out of range or in the wrong format. |
| **Post-conditions:** | The user is asked to correct the errors; file will not be saved |
| **Priority:** | High |

**Scenario:**
The user changes the information in an open mapping or has created new rows of UI components.
The user has input invalid mapping information.

The user chooses Save from the File menu.
The user is asked to correct the specific errors in the mapping information.

**Use-Case Scenario 7**

**Title:** "User edits a mapping file"

| | |
|---|---|
| **Pre-conditions:** | The MCM mapping component is running; a mapping file is open |
| **Post-conditions:** | The correct new UI components and information are present on the screen |
| **Priority:** | High |

**Scenario:**
The user opens a mapping file.
The user edits the current UI components or adds a new row.
The components reflect the changes and are in the correct order.

**Use-Case Scenario 8**

**Title:** "User chooses Save from the File menu with no components on the screen"

| | |
|---|---|
| **Pre-conditions:** | The MCM mapping component is running; the UI has no rows created |
| **Post-conditions:** | An empty file is saved |
| **Priority:** | High |

**Scenario:**
The user chooses Save from the File menu.
A dialog box appears that asks the user to give a file name and choose where the file should be saved, and the user clicks "Okay."
An "empty" file is saved in the correct place under the correct name.

## 3.8 File Format

The MCM will work with two file formats. It will take as input a data packet in C3D file format and after processing, it will output a file in MCM format.

### 3.8.1 C3D Files

The design of the C3D file format was originally driven by the need for a convenient and efficient mechanism to store data collected during Biomechanics experiments. The C3D format stores 3D coordinate and numeric data for any measurement trial, with all the various parameters that describe the data, in a single binary file. This largely eliminates the need for trial data to travel around with additional notes and subject information in separate files (which usually gets separated from the data at some point in its travels).

The ability to store a multitude of information about the data is the feature that sets the C3D format apart from every other biomechanics format. Early in the design of the C3D format, it was realized that it was unlikely that one, ironclad, specification would fit every biomechanics need. As a result the C3D file usually stores a small number of common parameters that describe the 3D data and then allows the users to define, generate, and store within the file any number of user or lab defined data items so that anyone opening the C3D file can access them.

The file can be identified through its extension, which is .C3D. The data in .C3D files is written in 16-bit integer (INTEGER*2) format, or optionally floating point (REAL*4) format. The format of the data can be determined by reading the header of the C3D file at a binary level without making any assumptions about the data format. For compatibility with FORTRAN and various operating systems, all C3D files should be thought of as consisting of records that are 512 bytes long (or 256 16-bit words).

Each C3D file contains information about physical measurements and parameter information. Physical measurements are the 3D space coordinates along the x-, y-, and z-axis. They can also include analog data, such as force plate measurements, but for the purposes of the MCM system, this analog data shall be ignored. Parameter information gives more specific information about the data points. This includes, the measurement units and data point labels, such as "Left-hand" or "Right-hand."

The most common form of C3D file contains 3D and analog data and consists of three or more sections:

A single 512-byte header record
One or more Parameter records
One or more Label and Range records (optional).
One or more data records that contain 3D data and/or analog data.

The first section is the file header section, which always starts at record 1 at the beginning of the file. This header record contains pointers to the start of the other parameter and data sections in the C3D file. The parameter section usually starts at record number 2. This section is variable in length but is typically at least ten records long. The third section is optional and is not important and shall be ignored by the MCM system. The last section contains the actual data as 3D point coordinate data and/or analog data.

The common format of coordinate and analog data is the integer format. However, the facility for the data to be written in REAL*4 format is also available for the storing of filtered data in the case where the integer form may not provide sufficient precision.

The header contains pointers to the other sections. This has the advantage of accessing the data non-sequentially, which can be computationally expensive.

Data is composed of the x, y, and z coordinates of a point, as well as a bit string identifying which cameras of the Vicon8 system captured the point. Data is stored sequentially by frames. Within a frame, data is stored sequentially by point number. Finally, there is also a scaling factor that allows the conversion of the stored integer measurement into an accurate real number point in space based on a pre-calculated calibration volume.

The MCM will take a file in C3D format and convert it, using the MCM mapping format, into output to a MIDI device.

### 3.8.2 MCM Files

The MCM file format consists of a text file that represents the user inputs in the user interface and the information contained in them. It shall have a .mcm extension. The first element in the file is a whole positive integer that represents the total number of rows, and thus the number of individual mappings, in the file. This number is followed by a space, a carrot ("^") and another space.

Each row in the UI is represented by nine text elements, all delineated by a "space carrot space" combination. The elements are as follows:

1) Row number and the body part label;
2) Axis;
3) Minimum physical input value;
4) Maximum physical input value;
5) MIDI command;
6) Minimum MIDI command output;
7) Maximum MIDI command output;
8) Channel; and
9) Duration (which shall be "null" if "note" is not chosen as the MIDI command).

If the user has not entered anything into a text field or has not made a choice from a combo-box or radio group then that text element shall be listed as "null." An "empty" .mcm file shall contain the number 0 followed by a space and a carrot.

The .mcm file shall be used by the translation component to interpret the C3D data and generate output to the MIDI device.

# 4 Lifecycle Considerations

The MCM system shall be developed following the spiral lifecycle model. In addition, the project team has determined that there will be two concurrent, yet interconnected phases.

The first phase will build the mapping component of the system. This component takes users' inputs and creates a file that specifies which body parts correspond to which MIDI commands. It also defines the effect of moving that body part on the MIDI output. The second phase is building the translation component of the MCM. This component takes a defined mapping and uses it to convert C3D data into MIDI commands in real-time.

Both phases shall be built concurrently since their development does not depend on each other. The output of the mapping component will be the input for the translation component.

Within each major phase, there shall be sub-phases. These correspond to the major milestones of the project: requirements, technical review, design, implementation, testing, and maintenance.

Each sub-phase shall be characterized by the following events:

?? Defining of objectives, alternatives, and constraints;
?? Evaluation of alternatives and risk analysis;
?? Producing the deliverable product, such as the Requirements Document, the Design Document, the writing of the source code, etc.; and
?? Planning for the next phase

The requirements sub-phase is also characterized by the building of a rapid prototype of the GUI. The GUI prototype shall be non-functional. Its purpose is to help identify the interface requirements and to obtain the sponsor's approval on the "look and feel" of the user interface.

Each sub-phase shall be actively managed and documented by the phase leader and phase clerk, respectively.

Finally, during each sub-phase, the project team will obtain the project's sponsor approval either in writing or verbally to ensure that the project is progressing towards its goal. If at any point, it is determined that something was omitted, or that something is incorrect, the current sub-phase shall be reevaluated and the above steps shall be repeated. This will further ensure the success and quality of the final product.

# 5 Acceptance Requirements

The main purpose of the MCM project is to take motion data generated by the Vicon8 system and translate it into MIDI audio data in real-time. The minimum general acceptance requirements for the MCM system are as follows:

## 5.1 Mapping Component Functionality

5.1.1 Define mapping of body part motion to MIDI commands

5.1.2 Save a defined mapping to disk

5.1.3 Open a defined mapping from disk

5.1.4 Edit the current mapping

5.1.5 These functionalities shall be provided in an intuitive, easy-to-use GUI.

## 5.2 Translation Component Functionality

5.2.1 Read and parse motion data saved in C3D file format

5.2.2 Convert the motion data to MIDI format output based on user-defined mapping

## 5.3 Performance Requirements

5.3.1 The translation of motion data to MIDI output shall occur within 33 to 55 ms of the generation of the motion data

The MCM project team shall complete the project in accordance with the specifications outlined in this requirements document by the end of the Spring Trimester 2001. However, if there is a time constraint or other factor that may hinder the progress of the project, the project team agrees that completion of the above items, at a minimum, shall be sufficient to consider the project complete and a success.

# 6 System Test Plan

The purpose of the MCM system test plan is ensure that the requirements specified in this document have been met. The following test cases shall be performed to verify the system requirements:

## 6.1 Mapping Component

**Test Case ID:** Map-1
**Purpose of Test Case:** Test Label menu items
**Item(s) Being Tested:** Menu selection
**Input Specification:** User selects body part from Label menu
**Output Specification:** Row of UI components are added to screen
**Test Environmental Needs /Special Test Procedures:** None

**Test Case ID:** Map-2
**Purpose of Test Case:** Test File✍Save functionality
**Item(s) Being Tested:** Save feature
**Input Specification:** Valid inputs
**Output Specification:** Text file with properly populated inputs
**Test Environmental Needs /Special Test Procedures:** None

**Test Case ID:** Map-3
**Purpose of Test Case:** Test File✍Open functionality
**Item(s) Being Tested:** Open feature
**Input Specification:** Text file generated from File✍Save
**Output Specification:** Properly populated UI components
**Test Environmental Needs /Special Test Procedures:** None

**Test Case ID:** Map-4
**Purpose of Test Case:** Test Run Translation command with valid file
**Item(s) Being Tested:** Mapping component and Translation component interaction
**Input Specification:** Valid mapping file
**Output Specification:** MIDI port initialized
**Test Environmental Needs /Special Test Procedures:** Mapping component and translation component must exist on same machine

**Test Case ID:** Map-5
**Purpose of Test Case:** Test Run Translation command with invalid file
**Item(s) Being Tested:** Mapping component and Translation component interaction
**Input Specification:** Invalid mapping file
**Output Specification:** User is prompted to correct inconsistencies in mapping file
**Test Environmental Needs**

**/Special Test Procedures:** Mapping component and translation component must exist on same machine

**Test Case ID:** Map-6
**Purpose of Test Case:** Test File✍Save functionality against invalid inputs
**Item(s) Being Tested:** Save feature, error handling
**Input Specification:** Invalid mapping
**Output Specification:** User is prompted to correct inconsistencies in mapping
**Test Environmental Needs**
**/Special Test Procedures:** None

**Test Case ID:** Map-7
**Purpose of Test Case:** Test editing of mapping file
**Item(s) Being Tested:** Save feature
**Input Specification:** Valid mapping file, changes by user
**Output Specification:** Valid mapping file with proper changes
**Test Environmental Needs**
**/Special Test Procedures:** None

**Test Case ID:** Map-8
**Purpose of Test Case:** Test File✍Save functionality with empty mappings
**Item(s) Being Tested:** Save feature, error handling
**Input Specification:** Empty mappings
**Output Specification:** Empty file is saved
**Test Environmental Needs**
**/Special Test Procedures:** None

**Test Case ID:** Map-9
**Purpose of Test Case:** Data point axis mapping with track
**Item(s) Being Tested:** Mapping, error handling
**Input Specification:** Mapping with data point axes
**Output Specification:** Error message if not mapped
**Test Environmental Needs**
**/Special Test Procedures:** None

**Test Case ID:** Map-10
**Purpose of Test Case:** Mapping axis with data point
**Item(s) Being Tested:** Mapping, error handling
**Input Specification:** Mappings
**Output Specification:** Error message if not mapped
**Test Environmental Needs**
**/Special Test Procedures:** None

## 6.2 Translation Component

**Test Case ID:** Trans-1
**Purpose of Test Case:** Test the Vicon8 API
**Item(s) Being Tested:** Vicon8 API functions
**Input Specification:** Motion information from Vicon cameras
**Output Specification:** C3D motion data
**Test Environmental Needs**

**/Special Test Procedures:** The Vicon8 system and the MCM Translation component must be running and must be on the same enclosed network.

**Test Case ID:** Trans-2
**Purpose of Test Case:** Test C3D motion data store
**Item(s) Being Tested:** Data store
**Input Specification:** C3D motion data from network
**Output Specification:** C3D motion data stored to local memory or file
**Test Environmental Needs /Special Test Procedures:** The Vicon8 system and the MCM Translation component must be running and must be on the same enclosed network.

**Test Case ID:** Trans-3
**Purpose of Test Case:** Test translation
**Item(s) Being Tested:** Translation functionality
**Input Specification:** C3D motion data from local memory
**Output Specification:** MIDI commands
**Test Environmental Needs /Special Test Procedures:** The Vicon8 system and the MCM Translation component must be running and must be on the same enclosed network.

**Test Case ID:** Trans-4
**Purpose of Test Case:** Make sure C3D motion data has a linear, one-to-one mapping with the output MIDI commands
**Item(s) Being Tested:** Mapping
**Input Specification:** C3D motion data and mappings
**Output Specification:** MIDI command values that are a function of the input value
**Test Environmental Needs /Special Test Procedures:** The Vicon8 system and the MCM Translation component must be running and must be on the same enclosed network.

**Test Case ID:** Trans-5
**Purpose of Test Case:** Test input ranges
**Item(s) Being Tested:** Translation, error handling
**Input Specification:** Motion Capture subject moves in and out of input ranges
**Output Specification:** Translation with no regard to movements outside of input ranges
**Test Environmental Needs /Special Test Procedures:** The Vicon8 system and the MCM Translation component must be running and must be on the same enclosed network.

**Test Case ID:** Trans-6
**Purpose of Test Case:** Test output ranges
**Item(s) Being Tested:** Translation, error handling
**Input Specification:** Mappings from mapping component
**Output Specification:** Valid MIDI output mappings within specified ranges
**Test Environmental Needs**

**/Special Test Procedures:** The Vicon8 system and the MCM Translation component must be running and must be on the same enclosed network.

**Test Case ID:** Trans-7
**Purpose of Test Case:** MIDI output
**Item(s) Being Tested:** Translation and MIDI interaction
**Input Specification:** C3D motion data from local memory store
**Output Specification:** Correct MIDI commands based on motion
**Test Environmental Needs**
**/Special Test Procedures:** The Vicon8 system and the MCM Translation component must be running and must be on the same enclosed network.

**Test Case ID:** Trans-8
**Purpose of Test Case:** Real-Time restriction
**Item(s) Being Tested:** Translation and MIDI interaction
**Input Specification:** Motion Capture subject moves, creating C3D information
**Output Specification:** MIDI commands execute within 50 ms (preferably 33 ms)
**Test Environmental Needs**
**/Special Test Procedures:** The Vicon8 system and the MCM Translation component must be running and must be on the same enclosed network.

# 7 Project Deliverables

Upon completion of the project, the MCM project team shall deliver the following to the sponsor:

?? Copy of the source code;
?? Compiled program in the form of two executable files (one for the mapping component and the other for the translation component);
?? Copies of all the documentation written in support of the project; and
?? The website link that hold the electronic version of the above-mentioned items.

The project shall be considered complete when all of the above items have been delivered to the sponsor.

# 8 Glossary And List Of Acronyms

?? After-touch

Decay of the sound after the initial force has been applied to the vibrating medium; the length of time it takes for the sound to "die out"

?? API

Application Programming Interface – a set of functions that allows the developer to use the functionality of an existing application. To use the application, the developer makes the appropriate function calls. The API usually includes documentation that defines what the functions are, what the parameters are, and what the return values are.

?? C3D File Format

The C3D format stores 3D coordinate and numeric data for any movement measurement, usually used in recording Biomechanics experiments.

?? Channel

There are 16 channels MIDI, and that's also the case in most sound cards and synthesizers. Different instruments can be assigned to different channels at the same time. So theoretically there can be a maximum of 16 instruments playing at the same time. The actual number can vary, because the sound card or synthesizer can only produce a fixed maximum number of notes at the same time. (eg: 32-voice polyphony means that you can only press 32 keys at the same time and hear them. The 33rd key you pressed will either be ignored, or the first key you pressed will be cut off) You can also use more than 16 instruments if you can "steal" a track (eg: Your harp only comes in at the introduction, so you can use the same track for another instrument that only appears later in the song on the same channel). All 16 channels are transmitted in one MIDI cable.

?? Ethereal

A network protocol analyzer. It allows the browsing of packet data interactively from a live network or from a previously saved capture file.

?? GUI

Graphical User Interface

?? ICS

UCI Department of Information and Computer Science

?? Intensity

The magnitude of a quantity (as force or energy or sound) per unit (as of area, charge, mass, or time).

?? Loudness

The attribute of a sound that determines the magnitude of the auditory sensation produced and that primarily depends on the amplitude of the sound wave involved

?? MCM

Motion Capture Music

?? MFC

Microsoft Foundation Classes. Standard Microsoft libraries for object-oriented Windows programming.

?? MIDI

Musical Instrument Digital Interface. MIDI is a standard protocol that was agreed upon by major manufacturers of Electronic Musical Instruments. It allows Keyboards, Synthesizers, Computers, Tape Decks and even Mixers & Stage Light Controllers to talk to each other.

?? Note-Velocity

Related to the force used to create a note; related to how hard a piano key is struck

?? Pitch

A musical note; a frequency in Hz of a vibrating medium

?? Pitch-Bend

A musical effect whereby the frequency of a note is modulated up or down.

?? RAM

Random Access Memory

?? SOTA

UCI Claire Trevor School of the Arts

?? TCP/IP

Networking communication protocols for transferring data from one computer to another. The routable protocol is the standard for the Internet and it ensures reliable data transfer and congestion control.

?? Tone

(See pitch)

?? Track

One can use as many tracks that the sequencer program allows you to. (Usually more than a hundred) A track contains the events (i.e. the things you did; eg: what key you pressed at when & for how long) you want to put inside. For example, you may want to do a very complicated Grand Piano part. You can record the right hand part in track 1, and then record the left hand part in track 2. If you make a mistake while recording the left hand part, you don't have to worry about the right hand's - it's stored separately in track 1. Then you can assign both tracks to channel 0 (Grand Piano in General MIDI - mentioned above). The end result will sound exactly like you played both hands at the same time! (*Note: percussion instruments are stored as a single "Instrument". Different keys correspond to different drums)

?? UCI

University of California, Irvine

?? Volume

The degree of loudness or the intensity of a sound

# 9 Meeting Minutes

The following are copies of the minutes from the project team meetings. These meetings were held in support of the development of the requirements and this specification document.

## 9.1 Date: Sunday 4/15/01

Attendees: Cayci Suitt, Gene Wie

Minutes:

Cayci and Gene spent a bit of time (quite a bit of time actually) putting together the previous revision of the previous requirements document. They requested that the two of you (Salvador and Jimar) get together to perform the necessary modifications and additions to have the document adhere to the layout specified by Prof. Ebert in lecture (and on his web page).

## 9.2 Date: Monday 4/16/01

Attendees: Cayci Suitt, Gene Wie, Salvador Ledezma

Minutes:

Jimar is sick and cannot attend.

Requirements Document
-We are working on changing the current document. We went over what was needed for the acceptance test.
-Our request that the presentation on the req.'s be given on the 26th so that Cayci could attend was granted.

Team Web page
-Ebert needs a link to our web space

Technical Report?
-it is actually a technical presentation. We will show what technical things we are learning or what we have to use to complete our project. It can be slides or lecture format.

Final Demo
-it can be 6/7 or 6/8, we need to email Chris and Lisa to schedule a time with them
-we also could use another meeting with the Vicon staff

## 9.3 Date: Wednesday 4/18/01

Attendees: Cayci Suitt, Gene Wie, Jimar Garcia, Salvador Ledezma

Minutes:

Meeting began at 7:20 pm. We discussed the holes in our current requirements document. Cayci attempted to show others the current interface, but had technical difficulties.

We created our task list, and Gene sent it to the rest of the group.

Cayci will send an email to the clients regarding various issues.
Gene will send Sal hardware and software details.
Jimar ate a burrito.

We talked over staff organization, team structure and tracking and control mechanisms.

Cayci will do use-case scenarios, Jimar will do test cases.
Cayci will provide screen shots of interface.
Gene will provide hardware diagram and dataflow diagram.

## 9.4 Date: Monday 4/23/01

Attendees: Cayci Suitt, Gene Wie, Salvador Ledezma

Minutes:

Jimar was absent, he was at work

Requirements issues
- we can hand in an unsigned draft initially
- music terms should go in glossary
- where to put contact info?
- no specs for MIDI device, C3D format is in the appendix
- answered question about axis mapping

## 9.5 Date: Wednesday 4/25/01

Attendees: Cayci Suitt, Gene Wie, Jimar Garcia, Salvador Ledezma

Minutes:

Final preparation for requirements presentation. Presentation responsibilities were assigned.

## 9.6 Date: Thursday 5/3/01

Attendees: Prof. Ebert, Gene Wie, Salvador Ledezma

Minutes:

Prof. Ebert provided comments on the Requirements Document. Overall he liked it, but he did have a couple of comments. The biggest was that the translation is vague and should be refined to include more details.

Ebert gave the team 2 weeks (until 5/17) to work on revisions and to resubmit the document with the appropriate clarifications.

# 10 Appendix – MIDI Command Definitions

The following table gives an overview of MIDI commands and what they mean. The MCM will generate these commands based on the given mapping of the subject's movements.

**MIDI Messages**

==========================================================================

TABLE 2: Expanded Status Bytes List
(adapted from "MIDI by the Numbers" by D. Valenti, Electronic Musician 2/88)
Updated 1995 By the MIDI Manufacturers Association

| STATUS BYTE | | | | | DATA BYTES | |
|---|---|---|---|---|---|---|
| 1st Byte Value | | | Function | | 2nd Byte | 3rd Byte |
| Binary | Hex | Dec | | | | |
| 10000000= | 80= | 128 | Chan 1 | Note off | Note Number | Note Velocity |
| 10000001= | 81= | 129 | Chan 2 | " | (0-127) | (0-127) |
| 10000010= | 82= | 130 | Chan 3 | " | see | " |
| 10000011= | 83= | 131 | Chan 4 | " | Table | " |
| 10000100= | 84= | 132 | Chan 5 | " | 4 | " |
| 10000101= | 85= | 133 | Chan 6 | " | " | " |
| 10000110= | 86= | 134 | Chan 7 | " | " | " |
| 10000111= | 87= | 135 | Chan 8 | " | " | " |
| 10001000= | 88= | 136 | Chan 9 | " | " | " |
| 10001001= | 89= | 137 | Chan 10 | " | " | " |
| 10001010= | 8A= | 138 | Chan 11 | " | " | " |
| 10001011= | 8B= | 139 | Chan 12 | " | " | " |
| 10001100= | 8C= | 140 | Chan 13 | " | " | " |
| 10001101= | 8D= | 141 | Chan 14 | " | " | " |
| 10001110= | 8E= | 142 | Chan 15 | " | " | " |
| 10001111= | 8F= | 143 | Chan 16 | " | " | " |
| 10010000= | 90= | 144 | Chan 1 | Note on | " | " |
| 10010001= | 91= | 145 | Chan 2 | " | " | " |
| 10010010= | 92= | 146 | Chan 3 | " | " | " |
| 10010011= | 93= | 147 | Chan 4 | " | " | " |
| 10010100= | 94= | 148 | Chan 5 | " | " | " |
| 10010101= | 95= | 149 | Chan 6 | " | " | " |
| 10010110= | 96= | 150 | Chan 7 | " | " | " |
| 10010111= | 97= | 151 | Chan 8 | " | " | " |
| 10011000= | 98= | 152 | Chan 9 | " | " | " |
| 10011001= | 99= | 153 | Chan 10 | " | " | " |
| 10011010= | 9A= | 154 | Chan 11 | " | " | " |
| 10011011= | 9B= | 155 | Chan 12 | " | " | " |
| 10011100= | 9C= | 156 | Chan 13 | " | " | " |
| 10011101= | 9D= | 157 | Chan 14 | " | " | " |
| 10011110= | 9E= | 158 | Chan 15 | " | " | " |

| | | | | |
|---|---|---|---|---|
| 10011111= 9F= 159 | Chan 16 | " | " | " |
| 10100000= A0= 160 | Chan 1 | Polyphonic | " | Aftertouch |
| 10100001= A1= 161 | Chan 2 | Aftertouch | " | amount |
| 10100010= A2= 162 | Chan 3 | " | " | (0-127) |
| 10100011= A3= 163 | Chan 4 | " | " | " |
| 10100100= A4= 164 | Chan 5 | " | " | " |
| 10100101= A5= 165 | Chan 6 | " | " | " |
| 10100110= A6= 166 | Chan 7 | " | " | " |
| 10100111= A7= 167 | Chan 8 | " | " | " |
| 10101000= A8= 168 | Chan 9 | " | " | " |
| 10101001= A9= 169 | Chan 10 | " | " | " |
| 10101010= AA= 170 | Chan 11 | " | " | " |
| 10101011= AB= 171 | Chan 12 | " | " | " |
| 10101100= AC= 172 | Chan 13 | " | " | " |
| 10101101= AD= 173 | Chan 14 | " | " | " |
| 10101110= AE= 174 | Chan 15 | " | " | " |
| 10101111= AF= 175 | Chan 16 | " | " | " |
| 10110000= B0= 176 | Chan 1 | Control/ | See | See |
| 10110001= B1= 177 | Chan 2 | Mode change | Table | Table |
| 10110010= B2= 178 | Chan 3 | " | 3 | 3 |
| 10110011= B3= 179 | Chan 4 | " | " | " |
| 10110100= B4= 180 | Chan 5 | " | " | " |
| 10110101= B5= 181 | Chan 6 | " | " | " |
| 10110110= B6= 182 | Chan 7 | " | " | " |
| 10110111= B7= 183 | Chan 8 | " | " | " |
| 10111000= B8= 184 | Chan 9 | " | " | " |
| 10111001= B9= 185 | Chan 10 | " | " | " |
| 10111010= BA= 186 | Chan 11 | " | " | " |
| 10111011= BB= 187 | Chan 12 | " | " | " |
| 10111100= BC= 188 | Chan 13 | " | " | " |
| 10111101= BD= 189 | Chan 14 | " | " | " |
| 10111110= BE= 190 | Chan 15 | " | " | " |
| 10111111= BF= 191 | Chan 16 | " | " | " |
| 11000000= C0= 192 | Chan 1 | Program | Program # | NONE |
| 11000001= C1= 193 | Chan 2 | change | (0-127) | " |
| 11000010= C2= 194 | Chan 3 | " | " | " |
| 11000011= C3= 195 | Chan 4 | " | " | " |
| 11000100= C4= 196 | Chan 5 | " | " | " |
| 11000101= C5= 197 | Chan 6 | " | " | " |
| 11000110= C6= 198 | Chan 7 | " | " | " |
| 11000111= C7= 199 | Chan 8 | " | " | " |
| 11001000= C8= 200 | Chan 9 | " | " | " |
| 11001001= C9= 201 | Chan 10 | " | " | " |
| 11001010= CA= 202 | Chan 11 | " | " | " |
| 11001011= CB= 203 | Chan 12 | " | " | " |
| 11001100= CC= 204 | Chan 13 | " | " | " |
| 11001101= CD= 205 | Chan 14 | " | " | " |
| 11001110= CE= 206 | Chan 15 | " | " | " |
| 11001111= CF= 207 | Chan 16 | " | " | " |
| 11010000= D0= 208 | Chan 1 | Channel | Aftertouch | " |
| 11010001= D1= 209 | Chan 2 | Aftertouch | amount | " |
| 11010010= D2= 210 | Chan 3 | " | (0-127) | " |
| 11010011= D3= 211 | Chan 4 | " | " | " |
| 11010100= D4= 212 | Chan 5 | " | " | " |
| 11010101= D5= 213 | Chan 6 | " | " | " |
| 11010110= D6= 214 | Chan 7 | " | " | " |
| 11010111= D7= 215 | Chan 8 | " | " | " |

| | | | | |
|---|---|---|---|---|
| 11011000= D8= 216 | Chan 9 | " | " | " |
| 11011001= D9= 217 | Chan 10 | " | " | " |
| 11011010= DA= 218 | Chan 11 | " | " | " |
| 11011011= DB= 219 | Chan 12 | " | " | " |
| 11011100= DC= 220 | Chan 13 | " | " | " |
| 11011101= DD= 221 | Chan 14 | " | " | " |
| 11011110= DE= 222 | Chan 15 | " | " | " |
| 11011111= DF= 223 | Chan 16 | " | " | " |
| 11100000= E0= 224 | Chan 1 | Pitch | Pitch | Pitch |
| 11100001= E1= 225 | Chan 2 | wheel | wheel | wheel |
| 11100010= E2= 226 | Chan 3 | control | LSB | MSB |
| 11100011= E3= 227 | Chan 4 | " | (0-127) | (0-127) |
| 11100100= E4= 228 | Chan 5 | " | " | " |
| 11100101= E5= 229 | Chan 6 | " | " | " |
| 11100110= E6= 230 | Chan 7 | " | " | " |
| 11100111= E7= 231 | Chan 8 | " | " | " |
| 11101000= E8= 232 | Chan 9 | " | " | " |
| 11101001= E9= 233 | Chan 10 | " | " | " |
| 11101010= EA= 234 | Chan 11 | " | " | " |
| 11101011= EB= 235 | Chan 12 | " | " | " |
| 11101100= EC= 236 | Chan 13 | " | " | " |
| 11101101= ED= 237 | Chan 14 | " | " | " |
| 11101110= EE= 238 | Chan 15 | " | " | " |
| 11101111= EF= 239 | Chan 16 | " | " | " |
| 11110000= F0= 240 | System Exclusive | | ** | ** |
| 11110001= F1= 241 | MIDI Time Code Qtr. Frame | | -see spec- | -spec- |
| 11110010= F2= 242 | Song Position Pointer | | LSB | MSB |
| 11110011= F3= 243 | Song Select(Song #) | | (0-127) | NONE |
| 11110100= F4= 244 | Undefined | | ? | ? |
| 11110101= F5= 245 | Undefined | | ? | ? |
| 11110110= F6= 246 | Tune request | | NONE | NONE |
| 11110111= F7= 247 | End of SysEx (EOX) | | " | " |
| 11111000= F8= 248 | Timing clock | | " | " |
| 11111001= F9= 249 | Undefined | | " | " |
| 11111010= FA= 250 | Start | | " | " |
| 11111011= FB= 251 | Continue | | " | " |
| 11111100= FC= 252 | Stop | | " | " |
| 11111101= FD= 253 | Undefined | | " | " |
| 11111110= FE= 254 | Active Sensing | | " | " |
| 11111111= FF= 255 | System Reset | | " | " |

** Note: System Exclusive (data dump) 2nd byte= Vendor ID (or Universal Exclusive) followed by more data bytes and ending with EOX.

# 11 Revisions

| Date | Version | Status/Action | Revision By | Comments |
|---|---|---|---|---|
| 4/26/01 | 1.0 | Completed Requirements Specification | All | The Requirements were submitted and presented today. |
| 5/17/01 | 1.1 | Added more detail and clarification to the document. | All | ?? Made extensive modifications to Section 3.4 Domain Specific Rules.<br>?? The Use-Case Scenarios were moved to Section 3.7, after Section 3.5, which defines the components discussed in the Use-case scenarios.<br>?? Added more detail about the C3D data format to Section 3.8 File Format.<br>?? Updated Section 6 Test Plan to include environmental requirements.<br>?? Added several terms to the glossary.<br>?? Updated the Section 9 Meeting Minutes to show the latest Requirements meetings. |
| | | | | |
| | | | | |