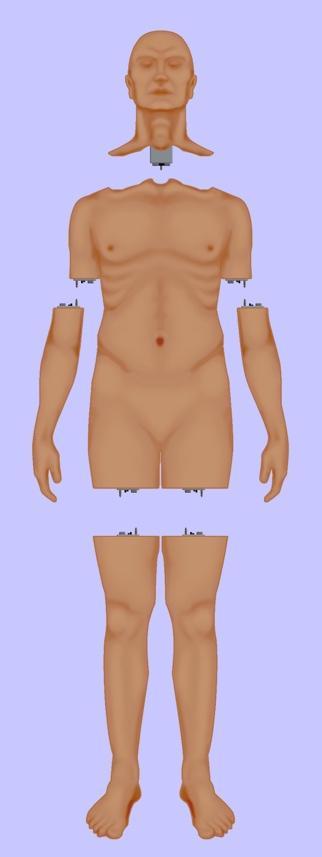
CDRL A002

Software Product Specification (SPS) for

Advanced Modular Manikin Project

Phase II Program

Contract # W81XWH-14-C-0101



Date: February 3, 2020

Revision: Rev-2

Unclassified

PRINCIPAL INVESTIGATOR: ROBERT M. SWEET, MD, FACS

REGENTS OF THE UNIVERSITY OF MINNESOTA

OFFICE OF SPONSORED PROJECTS

200 OAK ST SE

MINNEAPOLIS MN 55455-2009

This document is unclassified, and APPROVED FOR PUBLIC RELEASE; DISTRIBUTION IS UNLIMITED

Revision History

|  |  |
| --- | --- |
| Date of Revision | Description of Changes |
| 24 Dec 2018 | Original Draft Submitted |
| 26 Aug 2019 | Document Submitted |
| 15 Nov 2019 | Revised Document Submitted |
| 03 Feb 2020 | Distribution Updated |

**Distribution Statement**

This document was created under contract funding W81XWH-14-C-0101 from:

**DEPARTMENT OF THE ARMY**

**US ARMY MEDICAL RESEARCH ACQUISITION ACTIVITY**

**820 CHANDLER STREET**

**FORT DETRICK MD 21701-5014**

It contains no proprietary information, trade secrets, copyrighted material or classified information. This document is unclassified, and APPROVED FOR PUBLIC RELEASE; DISTRIBUTION IS UNLIMITED.

Table of Contents

[1 Scope 4](#_Toc24720848)

[1.1 Identification 5](#_Toc24720849)

[1.2 System overview 5](#_Toc24720850)

[1.3 Document Overview 5](#_Toc24720851)

[2 Referenced Documents 6](#_Toc24720852)

[2.1 Industry Documents 6](#_Toc24720853)

[2.2 Government Documents 6](#_Toc24720854)

[2.3 Related Contract Documents 6](#_Toc24720855)

[3 Requirements 6](#_Toc24720856)

[3.1 Executable Software 6](#_Toc24720857)

[3.2 Source Files 6](#_Toc24720858)

[4 Qualification 7](#_Toc24720859)

[5 Software Support Information 7](#_Toc24720860)

[5.1 As Built Software Design 7](#_Toc24720861)

[5.2 Compilation/Build Procedures 7](#_Toc24720862)

[5.3 Modification Procedures 7](#_Toc24720863)

[5.4 Computer Hardware Resource Utilization 8](#_Toc24720864)

[6 Requirements Traceability 8](#_Toc24720865)

[7 Notes 9](#_Toc24720866)

[8 Appendix 1: Acronyms 9](#_Toc24720867)

[Figure 1: Functional Overview of AMM Platform 4](#_Toc17472291)

[Table 1: Requirements Traceability Matrix 8](#_Toc17472310)

# Scope

This document defines the standards for 1.0 release of the Advanced Modular Manikin (AMM) platform and its formal deliverables. The formal deliverables consist of the platform specification, an open source\* Reference Implementation (RI) of the Computer Software Configuration Items (CSCIs), a reference implementation of the Universal Segment Connector (USC) and other hardware defined by the Hardware Configuration Items (HWCIs), the data models that ensure interoperability between the core and modules, and the documents that describe their design, operation, and extensibility through the addition of AMM Modules. Modules are defined as independent building blocks that provide incremental capabilities to the core, or provide training opportunities for different medical and trauma related conditions. The focus of this specification is on the platform, a much broader definition than a physical manikin, as illustrated in Figure 1, and on how it can be extended by medical simulation developers by adding:

* Modules that provide incremental capabilities to the core, including authoring tools, after action review tools, different physiology engines.
* Modules that add training opportunities, including IV/IO arms, intubation heads, laparotomy abdomens, virtual stethoscopes. These can be physical, virtual, or hybrid part task trainers.

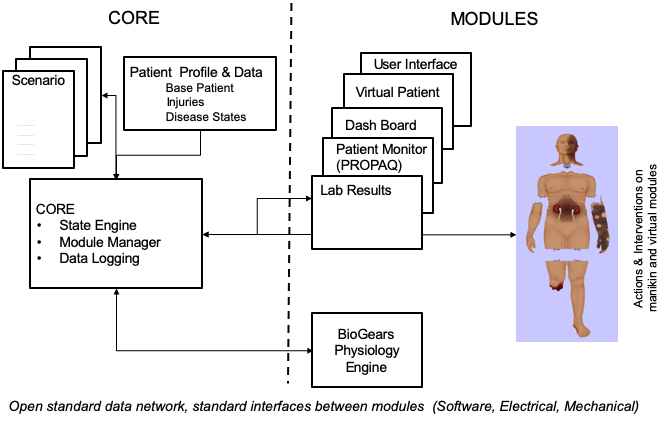


Figure : Functional Overview of AMM Platform

## Identification

This is the Advanced Modular Manikin (AMM) Software Product Specification (SPS) CDRL Item A002 of Contract # W81XWH-14-C-0101, Phase II. This SPS describes the “as built” design of the AMM Reference Software Computer Software Configuration Items (CSCIs) and describes the compilation, build, and modification procedures.

This CDRL is formatted to the requirements of Data Item Description Number DI-IPSC-81441A as required. It has been tailored to reflect the fact that the AMM Reference Implementation has been designed to run on a wide range of Linux and Windows systems.

## System overview

The AMM platform is a modular, distributed, interoperable system that enables physical, virtual, augmented and hybrid modules to work together as an integrated system. The traditional “core”, i.e. computer and state engine, can be in any one of the traditional manikin segments, i.e. torso, leg etc., or external to the human form, as it would be if the system is only running a virtual instance or if the targeted scenario, i.e. patient case, does not allow them to be internal due to the set of interventions that have to be performed on the body. The platform is architected as a system of systems that allow modules to function either as part of an integrated, whole body simulation or as autonomous part task trainers.

The published AMM standards guide the development and integration of AMM compatible modules. The reference designs provided for the final demo including electronics and central supplies were created to demonstrate the operation of the platform and are published as a developer’s tool kit with sources to acquire them from.

The developers of the platform have agreed to publish the AMM platform under the following open source licensing option:

\* *Creative Commons Attribution 4.0 International (CC BY 4.0)* [*https://creativecommons.org/licenses/by/4.0/deed.ast*](https://creativecommons.org/licenses/by/4.0/deed.ast)*.*

*Share — copy and redistribute the material in any medium or format*

*Adapt — remix, transform, and build upon the material for any purpose, even commercially.*

*The licensor cannot revoke these freedoms as long as you follow the license terms.*

This document does not cover modules that were created under separate funding and by other entities to demonstrate the functionality of the AMM Platform under separate funding and are not part of the Open Source agreement.

## Document Overview

Software Product Specification (SPS) CDRL Item A002 of Contract # W81XWH-14-C-0101, Phase II. The outline and subject matter content are based on DID DI-IPSC-81441A as required by the contract. The DID has been tailored as appropriate. This document is unclassified and contains no proprietary information, trade secrets, copyrighted material or classified information. Unlimited distribution.

# Referenced Documents

## Industry Documents

|  |  |
| --- | --- |
| Doc. No. | Title |
| 802.3x | IEEE Standard for Ethernet |
| DDS | Data Distribution Service |

## Government Documents

|  |  |
| --- | --- |
| Document Number | Title |
| W81XWH-14-C-0101 | AMM Phase II Contract, DOD |
| DI-IPSC-81441A | Data Item Description |
|  |  |

## Related Contract Documents

|  |  |
| --- | --- |
| Document Number | Title |
| CDRL A001 | Software Design Description (SDD) |
| CDRL A007 | Interface Design Description (IDD) |

# Requirements

## Executable Software

All AMM CORE module software will be provided as source. See section 3.2.

## Source Files

Source for all AMM CORE modules is available from the Advanced Modular Manikin public git repositories. Source files are broken into multiple repositories to make development and usage easier.

**The repositories are laid out with the following structure:**

* AMM Standard Library - <https://github.com/advancedmodularmanikin/amm-library>
* AMM Core Meta-package (core-modules), which has submodules:
  + Module Manager - <https://github.com/advancedmodularmanikin/module-manager>
  + Simulation Manager - <https://github.com/advancedmodularmanikin/simulation-manager>
  + Physiology Engine Manager - <https://github.com/advancedmodularmanikin/physiology-manager>
* AMM Extended Core Meta-package (extended-core-modules), which has submodules:
  + REST Adapter - <https://github.com/advancedmodularmanikin/rest-adapter>
  + TCP Bridge - <https://github.com/advancedmodularmanikin/tcp-bridge>
* AMM Example Modules Meta-package (example-modules), which has submodules:
  + Virtual Equipment - <https://github.com/advancedmodularmanikin/virtual-equipment>
  + Command Line Module - <https://github.com/AdvancedModularManikin/command-line-module>
  + "Kitchen Sink" example - <https://github.com/advancedmodularmanikin/example-module>
  + Fluidics Manager - <https://github.com/advancedmodularmanikin/fluid-manager>
* AMM Documentation - <https://github.com/advancedmodularmanikin/docs>

Each module can be cloned independently or as part of a meta-package depending on the task a developer is working on. It is critical that the AMM Standard Library used for all modules within an AMM system exist on the same AMM major version.

The master branch of each repository shall always contain compatible versions. Tags will be used to mark semantically versioned releases.

# Qualification

Source can be verified by comparing the checked-out version of software with the SHA-1 hash on the GitHub repository pages.

# Software Support Information

## As Built Software Design

Refer to the AMM Interface Description Document (IDD, CDRL A007) and Software Description Document (SDD, CDRL A001) for software design information.

## Compilation/Build Procedures

The AMM CORE modules are written in C++ (using the C++14 standard) and utilize CMake, a cross-platform free and open-source software application for managing the build process of software using a compiler-independent method. This allows the source to be compiled on many different platforms with a variety of compilers and/or build environments.

Documentation for CMake is available at <http://www.cmake.org>

The complete compilation/build process is available documented as a README in markdown format within the git repository for each module.

Each module repository has a submodule with the AMM standard library in it - a developer has the option of installing the AMM Standard Library system-wide, or utilizing the submodule version within each module repository. The CMake build for each module will automatically identify which AMM standard library to build or utilize.

## Modification Procedures

Modifications to the AMM CORE modules can be made through the existing GitHub source repositories. Developers are expected to follow C++ development best-practices, using existing code as a guide for documentation and style.

A developer may fork their own copy of the repository, make the needed modifications and submit a pull request. A pull request is a method of submitting contributions to an open development project. A pull request occurs when a developer asks for changes committed to an external repository to be considered for inclusion in a project's main repository.

The Data Model, as described in the Software Design Description (AMM CDRL A001), is formally implemented in the AMM\_Standard.idl file in the standard-library repository. Changes to the data model are intended to be very rare, as a change to the Interface Description Language (IDL) will break compatibility with other AMM modules necessitating a new major version. As such, pull requests for the IDL should be very thorough and give design decisions and rationalization for any changes.

The AMM development team will review all submitted pull requests. After review, a pull request may be approved and merged into the repository, or it may be rejected. If a pull request is rejected, comments will be given addressing the nature of the rejection.

A full developer's guide, including API documentation and Doxygen generated documentation is available in the AMM Documentation (amm-documentation) repository. The developer's guide is also available within CDRL A005.

## Computer Hardware Resource Utilization

While the AMM CORE reference implementation software has been developed to be buildable on both Windows and Linux systems, performance and resource requirements will vary greatly depending on the module that is being developed. A full AMM CORE (including a Module Manager, Simulation Manager, Physiology Engine Manager with BioGears, REST Adapter and TCP Bridge) will require at least 1GB of RAM and a multi-core processor is highly suggested. A simple module that drives hardware, for example the fluidics manager, can exist on far fewer resources and a single processor.

# Requirements Traceability

Table 1. provides traceability of each Computer Software Configuration Item (CSCI) to Specific Desired Capabilities identified in the AMM solicitation.

|  |  |  |
| --- | --- | --- |
| CSCI | Reference Paragraphs in Specific Desired Capabilities | Source Code Link |
| Module Manager | 2.b.ii. Extensible, 3.c.i. OS utilizing plug-play modular components | <https://github.com/advancedmodularmanikin/module-manager> |
| Simulation Manager | 3.c.iv. State Machine / Interactive Scenario Capability | <https://github.com/advancedmodularmanikin/simulation-manager> |
| Physiology Engine Manager | 2.b.v. DTME-PRP, 3.c.ii, Basic Physiology, 3.d.xii, Physiology System | <https://github.com/advancedmodularmanikin/physiology-manager> |
| REST Adapter | 1.d.i. Automated Data Collection | <https://github.com/advancedmodularmanikin/rest-adapter> |
| TCP Bridge | 2.b.ii. Extensible | <https://github.com/advancedmodularmanikin/tcp-bridge> |
| Virtual Equipment | 3.d.xiii. Electronic Monitoring Capability | <https://github.com/advancedmodularmanikin/virtual-equipment> |
| Command Executor | 2.b.iii, Command Line Interface | <https://github.com/advancedmodularmanikin/command-line-module> |
| Fluid Manager | 2.a.iii. Common Fluid System | <https://github.com/advancedmodularmanikin/fluid-manager> |

Table : Requirements Traceability Matrix

# Notes

None.

# Appendix 1: Acronyms

|  |  |
| --- | --- |
| **Acronym** | **Meaning** |
| **A** |  |
| ACDET | Company that provides modules to AMM |
| ACS | American College of Surgeons |
| AMM | Advanced Modular Manikin |
| AMMDK | Advanced Modular Manikin Development Kit |
| API | Application Programmer's Interface |
| **C** |  |
| CAE | CAE HealthCare Company |
| CDM | Common Data Model |
| CDRL | Contract Documentation Requirements List |
| CORBA | Common Object Request Broker Architecture |
| CSCI | Computer Software Configuration Item |
| **D** |  |
| DID | Data Item Definition |
| DDS | Data Distribution Service |
| DICOM | Digital Imaging and Communications in Medicine |
| DTME-PRP | Developer Tools for Medical Education Public Physiology Platform |
| **F** |  |
| FMA | Foundational Model of Anatomy |
| **G** |  |
| GUI | Graphical User Interface |
| **H** |  |
| HID | Human Interface Device |
| HLA | High Level Architecture |
| HWCI | Hardware Configuration Item |
| **I** |  |
| I2C | Inter-Integrated Circuit |
| ICD | Interface Control Document |
| ICD-10 | International Classification of Diseases 10 |
| IDD | Interface Design Description |
| IDL | Interface Definition Language |
| IGES | Initial Graphics Exchange Specification |
| IP | Internet Protocol |
| IVC | Inferior Vena Cava |
| **J** |  |
| JSON | JavaScript Object Notation |
| **L** |  |
| LMS | Learning Management System |
| LRS | Learning Record Store |
| **N** |  |
| NTP | Network Time Protocol |
| **O** |  |
| OMG | Object Management Group |
| OPB | Ontology of Physics for Biology |
| OS | Operating System |
| **P** |  |
| PhysDat | Physiology Data |
| PhysMod | Physiology Modification |
| PNG | Portable Network Graphics |
| PoE | Power over Ethernet |
| PSE | Power Sourcing Equipment |
| **Q** |  |
| QoS | Quality of Service |
| **R** |  |
| RenderMod | Render Modification |
| REST | Representational State Transfer |
| RI | Reference Implementation |
| RTPS | Real-time Publish-Subscribe |
| **S** |  |
| SDD | Software Design Description |
| SNOMED | International Health Terminology Standards Development Organization |
| SPI | Serial Peripheral Interface |
| SPS | Software Product Specification |
| SSS | System/Subsystem Specification |
| STL | Stereolithography |
| SUM | Software User's Manual |
| SWCI | Software Configuration Item (a subset of CSCI) |
| **U** |  |
| USB | Universal Serial Bus |
| UI | User Interface |
| UUID | Universally Unique Identifier |
| **X** |  |
| xAPI | Experiential API |
| XML | Extensible Markup Language |