



National Aeronautics and
Space Administration

KSC-DPV-0102

REVISION A

DATA PRESENTATION AND VISUALIZATION (DPV) INTERFACE CONTROL DOCUMENT

Revision A

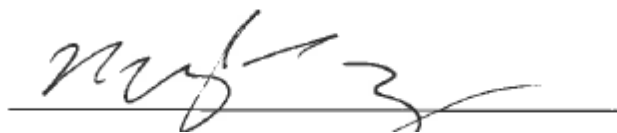
Data Presentation and Visualization (DPV) Interface Control Document (ICD)

Prepared By:

A handwritten signature in black ink, appearing to read 'Rebecca Mazzone', written over a horizontal line.

Rebecca Mazzone
Project Manager
Kennedy Space Center

Approved By:

A handwritten signature in black ink, appearing to read 'Michael Conroy', written over a horizontal line.

Michael Conroy
Modeling and Simulation Technology Manager
Kennedy Space Center

RECORD OF REVISIONS

REV	DESCRIPTION	DATE
Basic	Document Baseline	March 23, 2009
A	Updated to reflect DPV tool upgrades and document the MPC3 specification	March 9, 2015

TABLE OF CONTENTS

1. Introduction	5
1.1. Background	5
1.2. Purpose	5
1.3. Scope	5
2. Terminology	5
3. Roles	6
4. Generic Data Flow	6
5. Model Process Control Telemetry	7
5.1. History	7
5.2. MPC Specification	8
5.3. Mission File.....	8
5.3.1. Mission Level Element Definitions	10
5.3.2. Scene Section Element Definitions	11
5.3.3. Telemetry Section Element Definitions	32
5.4. Data File	33
5.4.1. Data Level Element Definitions.....	34
5.4.2. Initialization Section Element Definitions	35
5.4.3. Time Section Element Definitions	40
6. The Distributed Observer Network.....	45
7. Exploration Visualization Environment.....	45
APPENDIX A. Acronyms and Abbreviations	46
APPENDIX B. XML Schema for MPC3 Mission File	47
APPENDIX C. XML Schema for MPC3 Data File.....	54
APPENDIX D. Sample Mission File.....	58
APPENDIX E. Sample Data File	60
APPENDIX F. Data Flow Use Cases	62
F.1 Single Source, Graphical	62
F.2 Single Source, Non-graphical	63
F.3 Distributed Simulation Activity.....	64
F.4 Integration Activity	65

Revision A

1. Introduction

1.1. Background

Data Presentation and Visualization (DPV) is a subset of the modeling and simulation (M&S) capabilities at Kennedy Space Center (KSC) that endeavors to address the challenges of how to present and share simulation output for analysts, stakeholders, decision makers, and other interested parties. DPV activities focus on the development and provision of visualization tools to meet the objectives identified above, as well as providing supporting tools and capabilities required to make its visualization products available and accessible across NASA.

1.2. Purpose

The DPV Interface Control Document (ICD) outlines the formats and processes required to move simulation data from an authoritative source into a DPV tool. It is intended as a guide for external project developers who need to build DPV-compatible export functionality into their applications.

1.3. Scope

This document will provide a detailed explanation of the inputs required to recreate a simulation within a DPV visual environment. It is intended for developers of authoritative analysis tools to assist them in creating data export applications.

2. Terminology

A common set of terms is required to foster understanding among simulation providers, DPV teams, and DPV end users. With respect to all DPV products and processes, the following definitions should be applied.

Data Presentation: The display and distribution of data for review and discussion. Data presentation occurs to present the final results of a test, study or concept. Data cannot be added to, removed, modified or manipulated.

Data Visualization: The integration and visual representation of data. Data visualization occurs to develop a meaningful understanding of the data under evaluation during a test, study or concept. The data may be combined with other data sets and manipulated as needed.

Mission: All scene elements and telemetry files required to recreate the results of a specific simulation activity within a DPV environment.

Model: The geometric representation of an object. Specifically, it is a computer graphics file that can be loaded and processed by a software application. A model serves as the visual representation of a simulated element within a visualization environment.

Scene: A specific collection of models to be displayed in a visualization environment. A single scene may have multiple simulations associated with it.

Revision A

Simulation: Time-dependent data to be visualized within a DPV environment.

Simulation Activity: The collective design, development and execution of work performed in an authoritative simulation tool for a specific concept development or analysis task. Results from simulation activities are provided to DPV teams for recreation within their visualization tools.

Telemetry: Data provided by an authoritative source to identify how the dynamic objects in a scene behave or how time-dependent parameters change as the simulation progresses.

3. Roles

The process to move data from an authoritative simulation environment into a DPV-provided tool requires the communication and sharing of data between several parties. Each party holds one or more of the roles identified below.

Mission Builder: The person or group who integrates one or more sets of simulation data with models and scene data to create an overall mission. Items such as cameras and lighting are integrated based on feedback from the simulation developers and users to properly convey desired data or issues.

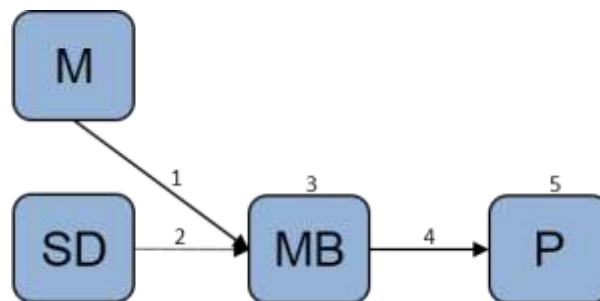
Modeler: The person or group who creates the individual graphics elements for inclusion in a graphics scene.

Presenter: The person or group responsible for obtaining the final data from the Mission Builder for presentation purposes.

Simulation Developer: The person or group who develops the simulation and generates telemetry for inclusion in a mission.

4. Generic Data Flow

The flow of data across the roles defined in Section 3 varies with the simulation activity requiring DPV support. In some instances the data exchanged between roles exists as physical files. In others the data exchanged is tool or interface information. The generic process for moving a simulation into or through DPV tools is provided in this section. Use cases illustrating specific applications of this data flow are provided in Appendix F.



Revision A

- (1) The **Modeler (M)** provides geometric models to the **Mission Builder (MB)** for integration with telemetry data. It is critical that all reference frames and units are clearly understood and conveyed from the Modeler to the Mission Builder.
- (2) The **Simulation Developer (SD)** provides simulation data to the **Mission Builder (MB)** for integration with geometric models and any other applicable mission telemetry. It is critical that all reference frames and units are clearly understood and conveyed from the Simulation Developer to the Mission Builder.
- (3) The **Mission Builder (MB)** integrates models and telemetry from all required providers to create the overall Mission. The integrated Mission data could itself become a simulation source, generating additional data based on the data received.
- (4) The **Mission Builder (MB)** exports data from the Mission and provides the files to the **Presenter (P)** for sharing with Mission stakeholders. The form of this data may include the MPC3 specification utilizing both the mission and data file formats for ingestion into an interactive data presentation system, or could include sets of animations and pictures for incorporation into static presentation. For the purposes of the DPV ICD, the presenter is an interactive system.
- (5) The **Presenter (P)** imports the Mission data and provides stakeholder access to the interactive system.

5. Model Process Control Telemetry

To best support the variety of simulation tools within NASA and ensure their continued support as simulation and DPV tools evolve, DPV has chosen to define and standardize on a single telemetry format called Model Process Control (MPC). Data Presentation tools will be capable of receiving and processing telemetry in the MPC format. Data Visualization tools will be capable of receiving, processing and providing telemetry in the MPC format.

5.1. History

The original MPC telemetry format was a comma-delimited text file. Each line entry was associated with a unique object and point in time, and contained 14 fields defining object position, orientation, scale, visibility, and any commands to be executed by the visualization environment. Each object in the simulation had to be represented at each time step and all 14 fields were required to form a valid line of telemetry.

The second version of the MPC file format, MPC2, began utilizing XML to enable better readability, usability, and expandability. It retained the same data for each object (position, orientation, scale and visibility) but dropped the command section, and adopted a state change philosophy so that objects not moving or changing from one time step to the next did not need to be included for that given time step. MPC2 also introduced new telemetry concepts of metadata and parenting. Metadata, declared in a new initialization section, allowed the user to capture "extra" time dependent data that was pertinent to the mission but not generally used for visualizing the motion of objects.

Revision A

Parenting allowed the data stream to dynamically alter the parent object of another object as a function of time.

MPC3 is the latest generation of the XML-based format created to transfer dynamic simulation information. It contains several changes to improve the usability of the format by software developers, improve the definition of data, and expand the data incorporated in the format. As a result of this effort, MPC3 is now comprised of two files: a mission setup file for defining the objects within the simulation, and a data file that contains the dynamic data for a given simulation/visualization.

5.2. MPC Specification

The following sections provide a general syntax for each file and a detailed description for each element found within the specification. For each element the following is provided: a general description is given to explain what the data stored within the element represents, the specific XML format for the element, a breakdown for each attribute available for the element, and a listing of any child elements that can be provided. For each attribute the following is provided: a description of the attribute is provided to explain the data associated with the attribute, the occurrence requirement for the attribute, the type of the value, the valid values, and the default value if not found. For each child element, the number of instances of that child is provided, ranging from "0" for not required, to "1" if it can occur only once, and "N" if the child can be provided any number of times.

5.3. Mission File

The MPC mission setup file contains the data needed to construct the visualization. This data includes the object hierarchy, lighting, and any cameras for viewing the scene. Also included in the mission file is a list of valid telemetry data sets for use with the scene. The general syntax is shown below, followed by descriptions of the elements and attributes.

```
<mpcMission version="3.0" epoch=" YYYY-MM-DDTHH:mm:ss.ssZ " author="string"
baseFilePath="dir/">
  <note>Mission description</note>
  <scene name="string" baseFilePath="dir/">
    <note>Scene description</note>
    <units time="unit" distance="unit" angle="unit" force="unit"/>
    <environment type="type" up="0 0 1" forward="1 0 0">
      <!-- Background options -->
      <backgroundColor color="r g b"/>
      <backgroundTexture tiling="1 1" offset="0 0" filePath="filepath"/>
      <skybox>
        <front tiling="1 1" offset="0 0" filePath="front.tif"/>
        <back tiling="1 1" offset="0 0" filePath="back.tif"/>
        <top tiling="1 1" offset="0 0" filePath="top.tif"/>
        <bottom tiling="1 1" offset="0 0" filePath="bottom.tif"/>
        <left tiling="1 1" offset="0 0" filePath="left.tif"/>
        <right tiling="1 1" offset="0 0" filePath="right.tif"/>
      </skybox>
    </environment>
  </scene>
</mpcMission>
```


Revision A

```

</skybox>
<light type="type" parent="ID" color="r g b" unitDistance="unit" pos="x y
z" forward="x y z" intensity="value" unitAngle="unit" spotCutoff="value"
range="value"/>
<camera name="name" unitAngle="unit" fov="value" unitDistance="unit"
pos="x y z" up="x y z" forward="x y z" parent="ID" track="ID2"/>
</environment>
<objects baseFilePath="dir/">
<terrain id="ID" name="name">
  <note>Description of terrain</note>
  <offset unitDistance="unit" pos="x y z" quat="x y z w"/>
  <geometry>
    <model type="type" unitDistance="unit" pos="x y z" quat="x y z w"
scale="1" unitModel="unit" filePath="dir1/base.tif" tiling="1 1"
offset="0 0" textureFilePath="dir2/base.png"/>
  </geometry>
</terrain>
<object id="ID" isStatic="1" name="name">
  <note>Description of object</note>
  <offset unitDistance="unit" pos="x y z" quat="x y z w"/>
  <geometry>
    <model unitDistance="unit" pos="x y z" quat="x y z w" scale="1"
unitModel="unit" filePath="dir1/base.tif"/>
  </geometry>
<object id="ID2" isStatic="1" name="name">
  ...
</object>
<joint id="ID3" type="type" name="name">
  <note> Description of Joint</note>
  <pin unitDistance="unit" pos="x y z" quat="x y z w"/>
  <geometry>
    <model unitDistance="unit" pos="x y z" quat="x y z w"
scale="1" unitModel="unit" filePath="dir1/base.tif"/>
  </geometry>
  <object id="ID4" isStatic="1" name="name">
    ...
  </object>
</joint>
<thruster id="ID5" name="name">
  <note> Description of Thruster</note>
  <offset unitDistance="unit" pos="x y z" quat="x y z w"/>
  <geometry>
    <model unitDistance="unit" pos="x y z" quat="x y z w"
scale="1" unitModel="unit" filePath="dir1/base.tif"/>
  </geometry>
</thruster>
</object>

```

Revision A

```

        </objects>
    </scene>
<telemetries baseFilePath="dir/">
    <telemetry name="string" type="type" format="format" epoch="YYYY-MM-
    DDTHH:mm:ss.ssZ" source="location">
        <note>Telemetry description</note>
    </telemetry>
</telemetries>
</mpcMission>

```

5.3.1. Mission Level Element Definitions

<mpcMission>

Description The primary element of the mission file. This element encompasses all other elements of the file.

Format <mpcMission version="3.0" epoch="YYYY-MM-DDTHH:mm:ss.ssZ" author="string" baseFilePath="dir/"> ... </mpcMission>

Attributes

version

Description: Defines the version of the MPC file.
 Occurrence: Required
 Type: String
 Values: 3.0
 Default: None

epoch

Description: Used to specify a Gregorian date/time value in UTC that should map to time "zero" of the simulation data, i.e. the double precision time value 0.0. This value is used when the data is being integrated with other data, including celestial body data.
 Occurrence: Optional
 Type: String, following ISO 8601 standard YYYY-MM-DDTHH:mm:ss.ssZ
 Values: User Defined
 Default: Application defined

author

Description: Name of the mission developer
 Occurrence: Optional
 Type: String
 Values: User Defined
 Default: Empty String

Revision A

baseFilePath

Description: Provides a default path location where all future filePaths within the mission may be relative to

Occurrence: Optional

Type: String

Values: User defined

Default: Mission file directory

Subelements **note (0..1)**
scene (1..1)

telemetries (0..1)

<note>

Description

A note is an optional element used to provide descriptions or additional information that may be of interest regarding the overall mission file. Only one note is expected in any given section that provides the ability to store a note. The data contained within the note is not designed to be used by software reading MPC, other than for display purposes, but should be maintained with the MPC file if the file is imported and exported.

Format

<note>string</note>

Attributes

None

Subelements

None

5.3.2. Scene Section Element Definitions

<scene>

Description

Scene is the main element tag for the section, enclosing all scene elements. Only one scene section is allowed within the mission file and is required.

Format

<scene name="string" baseFilePath="dir/">...</scene>

Attributes*name*

Description: Informative name can be provided for the scene

Occurrence: Optional

Type: String

Values: User Defined

Default: Empty String

baseFilePath

Description: Provides a default path location where all future filePaths within the Scene element may be relative to

Occurrence: Optional

Revision A

Type: String
 Values: User defined
 Default: Mission Base File Path

Subelements note (0..1)
 units (0..1)
 environment (1)
 objects (1)

<units>

Description The units element is used to set the units for each of the supported types throughout the scene and for each type not specified by telemetry.

Format <units time="unit" distance="unit" angle="unit" force="unit"/>

Attributes*time*

Description: Unit of time to be used in the absence of specification.
 Occurrence: Optional
 Type: String
 Values: millisecond, second, minute, hour, day
 Default: second

distance

Description: Unit of distance to be used in the absence of specification.
 Occurrence: Optional
 Type: String
 Values: millimeter, centimeter, meter, kilometer, AU, inch, foot, yard, mile
 Default: meter

angle

Description: Unit of angle to be used in the absence of specification.
 Occurrence: Optional
 Type: String
 Values: degree, radian
 Default: degree

force

Description: Unit of force to be used in the absence of specification.
 Occurrence: Optional
 Type: String

Revision A

Values: newton, pound
 Default: newton

Subelements None

<environment>**Description**

The environment element encompasses the properties used for setting up the general environment aspects of the scene.

Format

`<environment type="string" up=" x y z" forward=" x y z" baseFilePath="dir/">...</environment>`

Attributes*type*

Description: Determines the default tool specific background of the scene
 Occurrence: Optional
 Type: String
 Values: Earth, Moon, Mars, Space, Custom
 Default: Custom

up

Description: Specifies the default "up" direction of the scene
 Occurrence: Optional
 Type: Set of three doubles
 Values: User defined unit vector orthogonal to "forward" direction
 Default: 0 0 1

forward

Description: Specifies the default "forward" direction of the scene
 Occurrence: Optional
 Type: Set of three doubles
 Values: User defined unit vector orthogonal to "up" direction
 Default: 1 0 0

baseFilePath

Description: Provides a default path location where all future filePaths within Environment element may be relative to
 Occurrence: Optional
 Type: String
 Values: User defined
 Default: Scene Base File Path

Revision A

Subelements

backgroundColor (0..1)
 backgroundTexture (0..1)
 skybox (0..1)
 light (0..N)
 camera (0..N)

<backgroundColor>

Description The backgroundColor element specifies the color to use for a scene backdrop if no backgroundTexture element or skybox element is specified or supported.

Format <backgroundColor color="r g b"/>

Attributes*color*

Description: The color to use for the background
Occurrence: Required
Type: r g b
Values: User defined set of three doubles clamped at 0 and 1
Default: None

Subelements None

<backgroundTexture>

Description The backgroundTexture element specifies the texture for a scene backdrop if no skybox element is specified or supported. The specified texture is warped to the inner surface of the bounding volume of the scene. A backgroundTexture element overrides any specified backgroundColor element.

Format <backgroundTexture tiling="x y" offset="x y" filePath="string"/>

Attributes*tiling*

Description: The number of repetitions of the texture to use in the width and height directions
Occurrence: Optional
Type: Set of two integers
Values: User defined set of two non-negative non-zero integers
Default: 1 1

Revision A

offset

Description: The offset of the top left corner of the texture from the top left corner of the planar surface of the bounding volume of the scene

Occurrence: Optional

Type: Set of two integers

Values: User defined set of two non-negative integers

Default: 0 0

filePath

Description: Path to a texture file

Occurrence: Required

Type: String

Values: User defined file path, either absolute, or relative to Environment Base File Path

Default: None

Subelements None

<skybox>

Description The skybox element encompasses the specification of a set of six textures for a scene backdrop. The specified textures are applied to each of the 6 primary directions.

Format <skybox> ... </skybox>

Attributes None

Subelements front (1..1)
back (1..1)
top (1..1)
bottom (1..1)
left (1..1)
right (1..1)

<front>

Description Defines the texture to be applied to the front wall of the skybox.

Format <front tiling="x y" offset="x y" filePath="string"/>

Attributes*tiling*

Description: The number of repetitions of the texture to use in the width and height directions

Occurrence: Optional

Type: Set of two integers

Values: User defined set of two non-negative non-zero integers
Default: 1 1

offset

Description: The offset of the top left corner of the texture from the top left corner of the front plane of the skybox
Occurrence: Optional
Type: Set of two integers
Values: User defined set of two non-negative integers
Default: 0 0

filePath

Description: Path to a texture file
Occurrence: Required
Type: String
Values: User defined file path, either absolute, or relative to Environment Base File Path
Default: None

Subelements None

<back>

Description Defines the texture to be applied to the back wall of the skybox.

Format <back tiling="x y" offset="x y" filePath="string"/>

Attributes

tiling

Description: The number of repetitions of the texture to use in the width and height directions
Occurrence: Optional
Type: Set of two integers
Values: User defined set of two non-negative non-zero integers
Default: 1 1

offset

Description: The offset of the top left corner of the texture from the top left corner of the front plane of the skybox
Occurrence: Optional
Type: Set of two integers
Values: User defined set of two non-negative integers
Default: 0 0

filePath

Description: Path to a texture file

Occurrence: Required
Type: String
Values: User defined file path, either absolute, or relative to Environment Base File Path
Default: None

Subelements None

<top>

Description Defines the texture to be applied to the top wall of the skybox.

Format <top tiling="x y" offset="x y" filePath="string"/>

Attributes

tiling

Description: The number of repetitions of the texture to use in the width and height directions
Occurrence: Optional
Type: Set of two integers
Values: User defined set of two non-negative non-zero integers
Default: 1 1

offset

Description: The offset of the top left corner of the texture from the top left corner of the front plane of the skybox
Occurrence: Optional
Type: Set of two integers
Values: User defined set of two non-negative integers
Default: 0 0

filePath

Description: Path to a texture file
Occurrence: Required
Type: String
Values: User defined file path, either absolute, or relative to Environment Base File Path
Default: None

Subelements None

<bottom>

Description Defines the texture to be applied to the bottom wall of the skybox.

Format <bottom tiling="x y" offset="x y" filePath="string"/>

Revision A

Attributes*tiling*

Description: The number of repetitions of the texture to use in the width and height directions

Occurrence: Optional

Type: Set of two integers

Values: User defined set of two non-negative non-zero integers

Default: 1 1

offset

Description: The offset of the top left corner of the texture from the top left corner of the front plane of the skybox

Occurrence: Optional

Type: Set of two integers

Values: User defined set of two non-negative integers

Default: 0 0

filePath

Description: Path to a texture file

Occurrence: Required

Type: String

Values: User defined file path, either absolute, or relative to Environment Base File Path

Default: None

Subelements None

<left>

Description Defines the texture to be applied to the left wall of the skybox.

Format <left tiling="x y" offset="x y" filePath="string"/>

Attributes*tiling*

Description: The number of repetitions of the texture to use in the width and height directions

Occurrence: Optional

Type: Set of two integers

Values: User defined set of two non-negative non-zero integers

Default: 1 1

offset

Description: The offset of the top left corner of the texture from the top left corner of the front plane of the skybox

Occurrence: Optional

Revision A

Type: Set of two integers
 Values: User defined set of two non-negative integers
 Default: 0 0

filePath

Description: Path to a texture file
 Occurrence: Required
 Type: String
 Values: User defined file path, either absolute, or relative to Environment Base File Path
 Default: None

Subelements None

<right>

Description Defines the texture to be applied to the right wall of the skybox.

Format <right tiling="x y" offset="x y" filePath="string"/>

Attributes*tiling*

Description: The number of repetitions of the texture to use in the width and height directions
 Occurrence: Optional
 Type: Set of two integers
 Values: User defined set of two non-negative non-zero integers
 Default: 1 1

offset

Description: The offset of the top left corner of the texture from the top left corner of the front plane of the skybox
 Occurrence: Optional
 Type: Set of two integers
 Values: User defined set of two non-negative integers
 Default: 0 0

filePath

Description: Path to a texture file
 Occurrence: Required
 Type: String
 Values: User defined file path, either absolute, or relative to Environment Base File Path
 Default: None

Subelements None

Revision A

<light>

Description The light element is used to specify a light source for illuminating the scene.

Format <light type="string" parent="ID" color="r g b" unitDistance="unit" pos="x y z" forward="x y z" intensity="value" unitAngle ="unit" spotCutoff="value" range="value"/>

Attributes*type*

Description: Defines the type of the light

Occurrence: Required

Type: String

Values: point, directional

Default: None

parent

Description: Defines a parent object to which the light is attached such that if the parent object moves, the light moves with it.

Occurrence: Optional

Type: String

Values: Valid identification key of object defined within initialization section

Default: Empty String

color

Description: Color of the light

Occurrence: Optional

Type: Set of three doubles for Red, Green, and Blue

Values: User defined set of three doubles clamped at 0 and 1

Default: 1 1 1

unitDistance

Description: Units for distance to be applied to the light position

Occurrence: Optional

Type: String

Values: millimeter, centimeter, meter, kilometer, AU, inch, foot, yard, mile

Default: Scene unit of distance

pos

Description: Defines the position of the light in world coordinates or relative to the parent object, if specified.

Occurrence: Optional

Type: Set of three doubles

Values: User defined
Default: 0 0 0

forward

Description: Defines the forward direction for the light source in world coordinates or relative to the parent object, if specified.
Occurrence: Optional
Type: Set of three doubles
Values: User defined unit vector
Default: 0 0 1

intensity

Description: Defines the intensity of the light source
Occurrence: Optional
Type: Double
Values: 0 to 100
Default: 100

unitAngle

Description: Units for angle to be applied to this light's spotCutoff
Occurrence: Optional
Type: String
Values: degree, radian
Default: Scene unit of angle

spotCutoff

Description: For a point light source, defines the spot cutoff angle
Occurrence: Optional
Type: Double
Values: If in degrees, 0 to 90, or the special value of 180
If in radians, 0 to PI/2 or special value of PI
Default: 180 degrees or PI radians

range

Description: Defines the range of the light using the unitDistance
Occurrence: Optional
Type: Double
Values: User defined
Default: infinite

Subelements None

Revision A

<camera>

Description Provides the ability to define a camera, or view into the scene, that will be available to the user of a visualization system leveraging the mission file. Note, the first camera specified defines the initial view into the scene.

Format `<camera name="string" unitAngle="unit" fov="value" unitDistance="unit" pos="x y z" up="x y z" forward="x y z" parent="ID" track="ID2"/>`

Attributes*name*

Description: Provides a descriptive name for the camera
 Occurrence: Required
 Type: String
 Values: User defined
 Default: None

unitAngle

Description: Units for angle to be applied to this camera's data
 Occurrence: Optional
 Type: String
 Values: degree, radian
 Default: Scene unit of angle

fov

Description: Defines the field of view of the camera.
 Occurrence: Optional
 Type: Double
 Values: 0<FOV<180 if Angle Units are in degrees,
 0<FOV<PI if Angle Units are in Radians
 Default: Application Defined

unitDistance

Description: Units for distance to be applied to the camera's data
 Occurrence: Optional
 Type: String
 Values: millimeter, centimeter, meter, kilometer, AU, inch, foot, yard, mile
 Default: Scene unit of distance

pos

Description: Defines the position of the camera in world coordinates or relative to the parent object, if specified.
 Occurrence: Optional

Revision A

Type: Set of three doubles
 Values: User defined
 Default: 0 0 0

up

Description: Defines the up axis of the camera in world coordinates or relative to the parent object, if specified.
 Occurrence: Optional
 Type: Set of three doubles
 Values: User defined unit vector orthogonal to “forward” direction
 Default: 0 1 0

forward

Description: Defines the forward direction for the camera in world coordinates or relative to the parent object, if specified.
 Occurrence: Optional
 Type: Set of three doubles
 Values: User defined unit vector orthogonal to “up” direction
 Default: 0 0 1

parent

Description: Defines a parent object to attach the camera to such that if the parent object moves, the camera moves with it.
 Occurrence: Optional
 Type: String
 Values: Valid identification key of object defined within initialization section.
 Default: Empty String

track

Description: Defines an object that the camera will track.
 Occurrence: Optional
 Type: String
 Values: Valid identification key of object defined within initialization section.
 Default: Empty String

Subelements None

<objects>

Description Provides the container for all objects and the associated hierarchy.

Format <objects baseFilePath="dir/">

Attributes

baseFilePath

Description: Provides a default path location where all future filePaths may be relative to

Occurrence: Optional

Type: String

Values: User defined

Default: Scene Base File Path

Subelements object (0..N)
terrain (0..N)

<object>

Description The object element is used to identify an object to be included within the scene. Objects are one of the main building blocks of hierarchy within a scene.

Format <object id="ID" isStatic="value" name="string">

Attributes

id

Description: Unique identification key for the object within the scope of other objects, terrain, thrusters, and joints.

Occurrence: Required

Type: String

Values: User defined

Default: None

name

Description: Provides a descriptive name to the object.

Occurrence: Required

Type: String

Values: User defined

Default: None

isStatic

Description: Defines whether the object is static or dynamic. Static objects cannot be updated within time based telemetry stream, including parenting, visibility or scale.

Occurrence: Optional

Type: Integer

Values: 0 (False), 1 (True)

Default: 1

Revision A

Subelements note (0..1)
 offset (0..1)
 geometry (0..1)
 object (0..N)
 joint (0..N)
 thruster (0..N)
 terrain (0..N)

<offset>

Description The offset element is used to provide the location and orientation of the object relative to the parent object. This is used for static objects, thrusters, and terrain.

Format <offset unitDistance="unit" pos="x y z" quat="x y z w"/>

Attributes*unitDistance*

Description: Units for distance to be applied to the offset data
 Occurrence: Optional
 Type: String
 Values: millimeter, centimeter, meter, kilometer, AU, inch, foot, yard, mile
 Default: Scene unit of distance

pos

Description: Defines the static position of the object in world coordinates or relative to the parent object, if specified.
 Occurrence: Optional
 Type: Set of three doubles
 Values: User defined
 Default: 0 0 0

quat

Description: Defines the static orientation of the object relative to the parent object.
 Occurrence: Optional
 Type: Right Hand Unit Quaternion, specified in X Y Z W order
 Values: User defined
 Default: 0 0 0 1

Subelements None

Revision A

<geometry>

Description The geometry element is a container for any geometry that should be loaded for display of the object.

Format <geometry> ... </geometry>

Attributes None

Subelements model (1..N)

<model> (Object)

Description The model element provides the information needed to load in a geometry data file. The type of file (polygonal formats) can be determined from the file extension.

Format <model unitDistance="unit" pos="x y z" quat="x y z w" unitModel="unit" scale="1" filePath="dir1/base.tif"/>

Attributes*unitDistance*

Description: Units for distance to be applied to the offset data
Occurrence: Optional
Type: String
Values: millimeter, centimeter, meter, kilometer, AU, inch, foot, yard, mile
Default: Scene unit of distance

pos

Description: Defines the offset of the object in world coordinates or relative to the parent object, if specified.
Occurrence: Optional
Type: Set of three doubles
Values: User defined
Default: 0 0 0

quat

Description: Defines the static orientation of the object relative to the parent object.
Occurrence: Optional
Type: Right Hand Unit Quaternion, specified in X Y Z W order
Values: User defined
Default: 0 0 0 1

scale

Description: Provides the ability to scale the model
Occurrence: Optional
Type: Double

Values: User defined
Default: 1.0

unitModel

Description: Units for distance to be applied to the geometry data
Occurrence: Optional
Type: String
Values: millimeter, centimeter, meter, kilometer, AU, inch, foot, yard, mile
Default: Scene unit of distance

filePath

Description: Path to a model file
Occurrence: Required
Type: String
Values: User defined file path, either absolute, or relative to Objects Base File Path
Default: None

Subelements None

<joint>

Description A joint element is used to define a specialized object that represents a one degree of freedom joint within the hierarchy.

Format <joint id="ID3" type="type" name="string">...</joint>

Attributes

id

Description: Unique identification key for the object within the scope of other objects, terrain, thrusters, and joints.
Occurrence: Required
Type: String
Values: User defined
Default: None

name

Description: Provides a descriptive name to the joint.
Occurrence: Required
Type: String
Values: User defined
Default: None

type

Description: Defines the type of single degree of freedom.
Occurrence: Required

Revision A

Type: String
 Values: rotational, translational
 Default: None

Subelements note (0..1)
 pin (0..1)
 geometry (0..1)
 object (0..N)
 joint (0..N)
 thruster (0..N)

<pin>

Description The pin element defines the location and direction of the joint relative to the parent object. Based on the type of joint, the Z axis of the joint represents either the rotation vector or the translation axis.

Format <pin unitDistance="unit" pos="x y z" quat="x y z w"/>

Attributes*unitDistance*

Description: Units for distance to be applied to the pin data
 Occurrence: Optional
 Type: String
 Values: millimeter, centimeter, meter, kilometer, AU, inch, foot, yard, mile
 Default: Scene unit of distance

pos

Description: Defines the pin position relative to the parent object.
 Occurrence: Optional
 Type: Set of three doubles
 Values: User defined
 Default: 0 0 0

quat

Description: Defines the orientation of the pin relative to the parent object.
 Occurrence: Optional
 Type: Right Hand Unit Quaternion, specified in X Y Z W order
 Values: User defined
 Default: 0 0 0 1

Subelements None

Revision A

<thruster>

Description A thruster element is used to define a specialized object that represents a thruster plume within the scene hierarchy.

Format <thruster id="ID" name="string">

Attributes*id*

Description: Unique identification key for the object within the scope of other objects, terrain, thrusters, and joints.

Occurrence: Required

Type: String

Values: User defined

Default: None

name

Description: Provides a descriptive name to the joint.

Occurrence: Required

Type: String

Values: User defined

Default: None

Subelements note (0..1)
offset (0..1)
geometry (0..1)

<terrain>

Description The terrain element is used to identify topological data to be included within the scene.

Format <terrain id="ID" name="string">

Attributes*id*

Description: Unique identification key for the terrain within the scope of other terrain, objects, thrusters, and joints.

Occurrence: Required

Type: String

Values: User defined

Default: None

name

Description: Provides a descriptive name to the terrain.

Occurrence: Required

Type: String

Values: User defined

Default: None

Revision A

Subelements note (0..1)
 offset (0..1)
 geometry (0..1)
 object (0..N)
 terrain (0..N)

<model> (Terrain)

Description The model element provides the information needed to load in a geometry data file. Polygonal formats can be determined from the file extension.

Format `<model type="type" unitDistance="unit" pos="x y z" quat="x y z w" unitModel="unit" scale="1" filePath="dir1/base.tif" tiling="1 1" offset="0 0" textureFilePath="dir2/base.png"/>`

Attributes*type*

Description: Provides the type of the topological data
 Occurrence: Required
 Type: String
 Values: heightmap, polygonal
 Default: None

unitDistance

Description: Units for distance to be applied to the offset data
 Occurrence: Optional
 Type: String
 Values: millimeter, centimeter, meter, kilometer, AU, inch, foot, yard, mile
 Default: Scene unit of distance

pos

Description: Defines the offset of the object in world coordinates or relative to the parent object, if specified.
 Occurrence: Optional
 Type: Set of three doubles
 Values: User defined
 Default: 0 0 0

quat

Description: Defines the static orientation of the object relative to the parent object.
 Occurrence: Optional
 Type: Right Hand Unit Quaternion, specified in X Y Z W order
 Values: User defined

Default: 0 0 0 1

scale

Description: Provides the ability to scale the model
Occurrence: Optional
Type: Double
Values: User defined
Default: 1.0

unitModel

Description: Units for distance to be applied to the geometry data
Occurrence: Optional
Type: String
Values: millimeter, centimeter, meter, kilometer, AU, inch, foot, yard, mile
Default: Scene unit of distance

filePath

Description: Path to a model file
Occurrence: Required
Type: String
Values: User defined file path, either absolute, or relative to Objects Base File Path
Default: None

tiling

Description: The number of repetitions of the texture to use in the width and height directions
Occurrence: Optional
Type: Set of two integers
Values: User defined set of two non-negative non-zero integers
Default: 1 1

offset

Description: The offset of the top left corner of the texture from the texture origin of the geometry
Occurrence: Optional
Type: Set of two integers
Values: User defined set of two non-negative integers
Default: 0 0

textureFilePath

Description: Path to a texture file
Occurrence: Optional
Type: String

Revision A

Values: User defined file path, either absolute, or relative to
Objects Base File Path
Default: None

Subelements None

5.3.3. Telemetry Section Element Definitions

<telemetries>

Description Provides the container for all references to valid telemetry files that connect the specified scene to time based data.

Format <telemetries baseFilePath="dir/"> . . . </telemetries>

Attributes

baseFilePath

Description: Provides a default path location where all future filePaths within the telemetries element may be relative to

Occurrence: Optional

Type: String

Values: User defined

Default: Mission Base File Path

Subelements telemetry (1..N)

<telemetry>

Description Declares a given data source for use with the mission file.

Format <telemetry name="string" type="type" format="format" epoch="YYYY-MM-DDTHH:mm:ss.ssZ" source="location"> ... </telemetry>

Attributes

name

Description: Provides a shortened descriptive name of the telemetry.

Occurrence: Required

Type: String

Values: User defined

Default: None

type

Description: Defines the type of telemetry as either a discrete data file or a data set that will be streamed via the network.

Occurrence: Required

Type: String

Revision A

Values: file, network
 Default: None

format

Description: Defines the format of the data set.
 Occurrence: Required
 Type: String
 Values: MPC, MPC2, MPC3
 Default: None

epoch

Description: Used to specify a Gregorian date/time value in UTC that should map to time "zero" of the simulation data, i.e. the double precision time value 0.0. This value is used when the data is being integrated with other data.
 Occurrence: Optional
 Type: String, following ISO 8601 standard YYYY-MM-DDTHH:mm:ss.ssZ
 Values: User Defined
 Default: Mission Epoch

source

Description: Provides the location of the data set. File names are provided as paths relative to source file location.
 Occurrence: Required
 Type: String
 Values: IP:Port or user defined file path. File path is either absolute, or relative to Telemetries Base File Path
 Default: None

Subelements note (0..1)**5.4. Data File**

The data file contains all the dynamic data generated from an analysis or simulation that is to be visualized or integrated with other data. Contained within the file are two main sections, the initialization section, and the time based data section. The template is shown below, followed by a description for each element and attribute.

The general syntax is shown below.

```
<mpcData version="3.0">
  <!-- Data Initialization -->
  <init name="Sample" epoch="2013-05-17T19:22:10.00Z" author="author">
    <note>Description of the data contained</note>
    <units time="unit" distance="unit" angle="unit" force="unit"/>
  </init>
</mpcData>
```

Revision A

```

    <metaData id=" ID" type="type" unitLabel="label"/>
    <terrain id="ID"/>
    <object unitDistance="unit" unitAngle="unit" id="ID">
      <metaData id="ID" type="type" unitLabel="label"/>
    </object>
    <joint unitAngle="degree" id="ID">
      <metaData id="ID" type="type"/>
</joint>
<thruster unitForce="unit" id="ID">
  <metaData id="ID" type="type"/>
</thruster>
</init>
<!-- Time Based data -->
<time value="N">
  <event value="string"/>
  <metaData id="ID" value="value"/>
  <object id="ID" pos="x y z" vel="x y z" quat="x y z w" angularVel="rate x y
z" cg="x y z" scale="x y z" vis="1" parent="ID">
    <metaData id="ID" value="value"/>
  </object>
  <joint id="ID" value="value">
    <metaData id="ID" value="value"/>
  </joint>
  <thruster id="ID" value="value">
    <metaData id="ID" value="value"/>
  </thruster>
</time>
<time value = "N+1">
  ...
</time>
</mpcData>

```

5.4.1. Data Level Element Definitions

<mpcData>

Description All elements of the MPC data file are contained within the mpcData element. It encompasses the initialization and time data.

Format <mpcData version="3.0">...</mpcData>

Attributes

version

Description: Defines the version of the MPC file
 Occurrence: Required
 Type: String
 Values: 3.0
 Default: None

Revision A

Subelements init (1..1)
time (1..N)

5.4.2. Initialization Section Element Definitions**<init>**

Description Init is the main element tag for the section, enclosing all initialization elements. Only one init section is allowed within the data file and is required. All elements that are to be referenced within the Time section must be declared within the init section.

Format <init name="string" epoch="YYYY-MM-DDTHH:mm:ss.ssZ" author="string"> ... </init>

Attributes*name*

Description: Informative name can be provided for the data set
Occurrence: Optional
Type: String
Values: User Defined
Default: Empty String

epoch

Description: Used to specify a Gregorian date/time value in UTC that should map to time "zero" of the simulation data, i.e. the double precision time value 0.0. This value is used when the data is being integrated with other data, including celestial body data.
Occurrence: Optional
Type: String, following ISO 8601 standard
YYYY-MM-DDTHH:mm:ss.ssZ
Values: User Defined
Default: If used in conjunction with a mission file, the default epoch is inherited from the telemetry element, otherwise, application defined.

author

Description: Name of the data provider
Occurrence: Optional
Type: String
Values: User Defined
Default: Empty String

Subelements note (0..1)

<note>

Description A note is an element used to provide descriptions or additional information that may be of interest regarding the data. The data

Revision A

contained within the note is not designed to be used by software reading MPC, other than for display purposes, but should be maintained with the MPC file if the file is imported and exported. Default value for a note is an empty string.

Format <note>string</note>

Attributes None

Subelements None

<units>

Description The units initialization is used to set the units for each of the supported type throughout the data file.

Format <units time="unit" distance="unit" angle="unit" force="unit"/>

Attributes*time*

Description: Unit of time to be used in the absence of specification.

Occurrence: Optional

Type: String

Values: millisecond, second, minute, hour, day

Default: second

distance

Description: Unit of distance to be used in the absence of specification.

Occurrence: Optional

Type: String

Values: millimeter, centimeter, meter, kilometer, AU, inch, foot, yard, mile

Default: meter

angle

Description: Unit of angle to be used in the absence of specification.

Occurrence: Optional

Type: String

Values: degree, radian

Default: degree

force

Description: Unit of force to be used in the absence of specification.

Occurrence: Optional

Type: String

Revision A

Values: newton, pound
 Default: newton

Subelements None

<metaData>**Description**

Meta data is used to declare a variable/parameter that will contain a dynamic value as a function of time. Declaring this outside an object, thruster, or joint associates this parameter at the mission level. Declaring this inside an object, thruster, or joint associates the parameter with the parent element.

Format

`<metaData id="ID" type="type" unitLabel="label"/>`

Attributes*id*

Description: Unique identification key for the parameter within the scope of other meta data parameters within the object it is declared.

Occurrence: Required
 Type: String
 Values: User defined
 Default: None

type

Description: Defines the parameter type
 Occurrence: Required
 Type: String
 Values: string, double
 Default: None

unitLabel

Description: Provides a label to be displayed with the value, such as units
 Occurrence: Optional
 Type: String
 Values: User defined
 Default: Empty string

Subelements None

<object>**Description**

Object is used to declare an object that will contain, or will be referenced by, data within the time based data section. When used in conjunction with a mission file, an object declared as static in the mission file shall not be dynamically re-parented, scaled, positioned, or oriented within the time elements.

Format <object unitDistance="unit" unitAngle="unit" id="ID" ofInterest="value"> ... </object>

Attributes

id

Description: Unique identification key for the object within the scope of other objects, terrain, thrusters, and joints.
Occurrence: Required
Type: String
Values: User defined
Default: None

unitDistance

Description: Units for distance to be applied to this object's data
Occurrence: Optional
Type: String
Values: millimeter, centimeter, meter, kilometer, AU, inch, foot, yard, mile
Default: Init unit of distance

unitAngle

Description: Units for angle to be applied to this object's data
Occurrence: Optional
Type: String
Values: degree, radian
Default: Init unit of angle

ofInterest

Description: Identification of the object as an object of interest within the simulation
Occurrence: Optional
Type: Integer
Values: 0 (False), 1 (True)
Default: 0

Subelements metaData (0..N)

<terrain>

Description Terrain is used to declare topological data that will be referenced within the time based data section. Terrain objects shall not be dynamically reparented, scaled, positioned, or oriented within the time elements.

Format <terrain id="ID"/>

Revision A

Attributes*id*

Description: Unique identification key for the object within the scope of other terrain, objects, thrusters, and joints.

Occurrence: Required

Type: String

Values: User defined

Default: None

Subelements None**<thruster>**

Description Thruster is used to declare the visual representation of a thruster plume, where time based firing information of the thrust magnitude will be provided within the telemetry section. Thruster objects shall not be dynamically reparented or scaled within the time elements.

Format <thruster unitForce="unit" unitDistance="unit" id="ID"/>

Attributes*id*

Description: Unique identification key for the thruster within the scope of other objects, terrain, thrusters, and joints.

Occurrence: Required

Type: String

Values: User defined

Default: None

unitForce

Description: Units for force to be applied to this object's data

Occurrence: Optional

Type: String

Values: newton, foot-pound

Default: Init unit of force

unitDistance

Description: Units for distance to be applied to this object's data

Occurrence: Optional

Type: String

Values: millimeter, centimeter, meter, kilometer, AU, inch, foot, yard, mile

Default: Init unit of distance

Subelements metaData (0..N)

Revision A

<joint>

Description Joint is used to declare a single degree of freedom joint that will change as a function of time. Joint objects shall not be dynamically re-parented, scaled, positioned, or oriented within the time elements.

Format <joint unitAngle="unit" unitDistance="unit" id="ID">

Attributes*id*

Description: Unique identification key for the joint within the scope of other objects, terrain, thrusters, and joints.

Occurrence: Required

Type: String

Values: User defined

Default: None

unitAngle

Description: Units for angle to be applied to this object's data

Occurrence: Optional

Type: String

Values: degree, radian

Default: Init unit of angle

unitDistance

Description: Units for distance to be applied to this object's data

Occurrence: Optional

Type: String

Values: millimeter, centimeter, meter, kilometer, AU, inch, foot, yard, mile

Default: Init unit of distance

Subelements metaData (0..N)

5.4.3. Time Section Element Definitions**<time>**

Description Each time step is enclosed in a "time" tag. Time tags shall be listed in time increasing order.

Format <time value="N">

Attributes*value*

Description: Value contains the simulation time offset from the Epoch for the data contained within the tag.

Occurrence: Required

Type: Double

Revision A

Values: User defined
 Default: None

Subelements event (0..1)
 object (0..N)
 metaData (0..N)
 joint (0..N)
 thruster (0..N)

<event>

Description The event tag is used to capture a key occurrence that happened during this time within the simulation/data. It is a specialized type of meta data.

Format <event value="string"/>

Attributes*value*

Description: Captures the name or description of the event.
 Occurrence: Optional
 Type: String
 Values: User defined
 Default: Last known set value

Subelements None

<object>

Description Objects are updated using the "object" tag and a series of attributes to define all data associated with an object.

Format <object id="ID" pos="x y z" vel="x y z" quat="x y z w"
 angularVel="r x y z" cg="x y z" scale="x y z" vis="1" parent="ID">
 ... </object>

Attributes*id*

Description: The unique identification key of the object being set.
 Occurrence: Required
 Type: String
 Values: Valid Identification Key of an object declared within the initialization
 Default: None

pos

Description: Defines the position of the object relative to the reference frame used to capture the data.
 Occurrence: Optional

Type: Set of three doubles
Values: User defined
Default: Last known set value

vel

Description: Defines the velocity of the object relative to the reference frame used to capture the data.
Occurrence: Optional
Type: Set of three doubles
Values: User defined
Default: Last known set value

quat

Description: Defines the orientation of the object relative to the reference frame used to capture the data.
Occurrence: Optional
Type: Right Hand Unit Quaternion, specified in X Y Z W order
Values: User defined
Default: Last known set value

angularVel

Description: Defines the angular velocity of the object relative to the reference frame used to capture the data.
Occurrence: Optional
Type: Set of 4 doubles defining Rate and Axis of rotation, specified in R X Y Z order
Values: User defined
Default: Last known set value

cg

Description: Defines the location of the center of gravity of the object relative to the design reference frame of the object as a function of time.
Occurrence: Optional
Type: Set of 3 doubles
Values: User defined
Default: Last known set value

scale

Description: Defines a scale value to be applied to the object at this time
Occurrence: Optional
Type: Set of 3 doubles
Values: User defined
Default: Last known set value

Revision A

vis

Description: Provides an ability to hide or show the object
 Occurrence: Optional
 Type: Single integer
 Values: 0 (hide), 1 (show)
 Default: Last known set value

parent

Description: Defines a new parent object or joint for the current object. All data for this object should now be provided relative to the new parent.
 Occurrence: Optional
 Type: String
 Values: Valid Identification Key of an object, joint, or terrain declared within the initialization
 Default: None

Subelements metaData (0..N)**<metaData>****Description**

Meta data element found within the time element is used to declare the value of a meta data parameter. Meta data elements can be found at the mission level or within an object, thruster, or joint element.

Format

<metaData id="ID" value="value"/>

Attributes*id*

Description: The unique identification key of the parameter being set.
 Occurrence: Required
 Type: String
 Values: User defined
 Default: None

value

Description: Defines the parameter type
 Occurrence: Required
 Type: String or Double based on initialization
 Values: User defined
 Default: Last known set value

Subelements None

Revision A

<thruster>

Description Thruster is used to update the value and position of a thruster declared within the initialization.

Format <thruster id="ID" value = "value" pos="x y z" quat="x y z w" >...</thruster>

Attributes*id*

Description: The unique identification key for the thruster being updated.
 Occurrence: Required
 Type: String
 Values: Valid Identification Key of a thruster declared within the initialization
 Default: None

value

Description: Value of force for the thruster
 Occurrence: Required
 Type: Double
 Values: User Defined
 Default: Last known value set

pos

Description: Defines the position of the thruster relative to the reference frame used to capture the data.
 Occurrence: Optional
 Type: Set of three doubles
 Values: User defined
 Default: Last known set value

quat

Description: Defines the orientation of the thruster relative to the reference frame used to capture the data.
 Occurrence: Optional
 Type: Right Hand Unit Quaternion, specified in X Y Z W order
 Values: User defined
 Default: Last known set value

Subelements metaData (0..N)

<joint>

Description Joint is used to update the value of a joint declared within the initialization.

Format <joint id="ID" value = "value">...</joint>

Attributes

id

Description: The unique identification key for the joint being updated.
Occurrence: Required
Type: String
Values: Valid Identification Key of a joint declared within the initialization
Default: None

value

Description: Value of offset for the joint
Occurrence: Required
Type: Double
Values: User Defined
Default: Last known value set

Subelements metaData (0..N)

6. The Distributed Observer Network

The Distributed Observer Network (DON) is the primary presentation tool for DPV and is designed to serve as a final publication tool for other simulators. The version of a simulation presented through DON is intended to be as close as visually possible to the original, therefore the tool is not allowed to perform calculations or generate new information. Analysis type activities are to be performed in tools other than DON, either by the original simulation tool or EVE (Exploration Visualization Environment).

DON supports models in OBJ format and textures in MTL. Specifications for these formats are available at <http://www.martinreddy.net/gfx/3d/OBJ.spec> and <http://paulbourke.net/dataformats/mtl/>, respectively.

7. Exploration Visualization Environment

EVE is the primary visualization tool for DPV and is designed for use early in the development process. Data can be provided from a variety of sources and combined within the EVE environment to visualize an integrated system. EVE also includes capabilities that can be used to help fill in missing data for preliminary analysis as needed. Visualizations created with its environment may be published to the DON for distribution.

EVE is built with OpenSceneGraph and can support models in OBJ, IVE, STL, FLT, DAE, and 3DS formats.

Revision A

APPENDIX A. Acronyms and Abbreviations

3DS	3D Studio Graphics Model Format
AU	Astronomical Units
DAE	COLLADA File
DON	Distributed Observer Network
DPV	Data Presentation and Visualization
EVE	Exploration Visualization Environment
FLT	OpenFlight Model Format
FOV	Field of View
ICD	Interface Control Document
ISO	International Standards Organization
IVE	OpenSceneGraph Model Format
KSC	Kennedy Space Center
M	Modeler
M&S	Modeling and Simulation
MB	Mission Builder
MPC	Model Process Control
MTL	Wavefront Material Template Library
NASA	National Aeronautics and Space Administration
OBJ	Wavefront Object Model Format
P	Presenter
PNG	Portable Network Graphics
POST	Program to Optimize Simulated Trajectories
SD	Simulation Developer
STL	Stereolithography Interface Format
UTC	Coordinated Universal Time
XML	Extensible Markup Language

Revision A

**APPENDIX B.
XML Schema for MPC3 Mission File**

```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:simpleType name="timeUnitsType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="millisecond"/>
      <xs:enumeration value="second"/>
      <xs:enumeration value="minute"/>
      <xs:enumeration value="hour"/>
      <xs:enumeration value="day"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="distanceUnitsType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="millimeter"/>
      <xs:enumeration value="centimeter"/>
      <xs:enumeration value="meter"/>
      <xs:enumeration value="kilometer"/>
      <xs:enumeration value="AU"/>
      <xs:enumeration value="inch"/>
      <xs:enumeration value="foot"/>
      <xs:enumeration value="yard"/>
      <xs:enumeration value="mile"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="angleUnitsType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="radian"/>
      <xs:enumeration value="degree"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="forceUnitsType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="newton"/>
      <xs:enumeration value="pound"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:complexType name="unitsType">
    <xs:attribute name="time" type="timeUnitsType" default="second"/>
    <xs:attribute name="distance" type="distanceUnitsType" default="meter"/>
    <xs:attribute name="angle" type="angleUnitsType" default="degree"/>
    <xs:attribute name="force" type="forceUnitsType" default="newton"/>
  </xs:complexType>

  <xs:simpleType name="environmentType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="Earth"/>
      <xs:enumeration value="Moon"/>
    </xs:restriction>
  </xs:simpleType>
</xs:schema>
```

Revision A

```

<xs:enumeration value="Mars"/>
<xs:enumeration value="Custom"/>
</xs:restriction>
</xs:simpleType>

<xs:simpleType name="tripleDoubleType">
  <xs:restriction base="xs:string">
    <xs:pattern value="^[+-]?[0-9]*\.[0-9]+([eE][+-]?[0-9]+)?\s[+-]?[0-9]*\.[0-9]+([eE][+-]?[0-9]+)?\s[+-]?[0-9]*\.[0-9]+([eE][+-]?[0-9]+)?$"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="quadDoubleType">
  <xs:restriction base="xs:string">
    <xs:pattern value="^[+-]?[0-9]*\.[0-9]+([eE][+-]?[0-9]+)?\s[+-]?[0-9]*\.[0-9]+([eE][+-]?[0-9]+)?\s[+-]?[0-9]*\.[0-9]+([eE][+-]?[0-9]+)?\s[+-]?[0-9]*\.[0-9]+([eE][+-]?[0-9]+)?$"/>
  </xs:restriction>
</xs:simpleType >

<xs:simpleType name="rgbType">
  <xs:restriction base="xs:string">
    <xs:pattern value="^[0-1](\.[0-9]*)?\s[0-1](\.[0-9]*)?\s[0-1](\.[0-9]*)?$"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="doubleIntegerType">
  <xs:restriction base="xs:string">
    <xs:pattern value="^[0-9]+\s[0-9]+$"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="doubleIntegerTypeMin1">
  <xs:restriction base="xs:string">
    <xs:pattern value="^[1-9]+[0-9]*\s[1-9]+[0-9]*$"/>
  </xs:restriction>
</xs:simpleType>

<xs:complexType name="textureType">
  <xs:attribute name="tiling" type="doubleIntegerTypeMin1" default="1 1"/>
  <xs:attribute name="offset" type="doubleIntegerType" default="0 0"/>
  <xs:attribute name="filePath" type="xs:string"/>
</xs:complexType>

<xs:simpleType name="lightStyleType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="point"/>
    <xs:enumeration value="directional"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="spotCutoffType">
  <xs:restriction base="xs:double">
    <xs:minInclusive value="0"/>
    <xs:maxInclusive value="180"/>
  </xs:restriction>

```


Revision A

```
</xs:simpleType>
```

```
<xs:simpleType name="intensityType">  
  <xs:restriction base="xs:integer">  
    <xs:minInclusive value="0"/>  
    <xs:maxInclusive value="100"/>  
  </xs:restriction>  
</xs:simpleType>
```

```
<xs:simpleType name="cameraFovType">  
  <xs:restriction base="xs:double">  
    <xs:minExclusive value="0"/>  
    <xs:maxExclusive value="180"/>  
  </xs:restriction>  
</xs:simpleType>
```

```
<xs:complexType name="geometryType">  
  <xs:sequence>  
    <xs:element name="model" minOccurs="1" maxOccurs="unbounded">  
      <xs:complexType>  
        <xs:attribute name="unitDistance" type="unitDistanceType"/>  
        <xs:attribute name="pos" type="tripleDoubleType" default="0 0 0"/>  
        <xs:attribute name="quat" type="quadDoubleType" default="0 0 0 1"/>  
        <xs:attribute name="unitModel" type="unitDistanceType"/>  
        <xs:attribute name="scale" type="xs:double" default="1"/>  
        <xs:attribute name="filePath" type="xs:string"/>  
      </xs:complexType>  
    </xs:element>  
  </xs:sequence>  
</xs:complexType>
```

```
<xs:complexType name="terrainGeometryType">  
  <xs:sequence>  
    <xs:element name="model" minOccurs="1" maxOccurs="unbounded">  
      <xs:complexType>  
        <xs:attribute name="unitDistance" type="unitDistanceType"/>  
        <xs:attribute name="pos" type="tripleDoubleType" default="0 0 0"/>  
        <xs:attribute name="quat" type="quadDoubleType" default="0 0 0 1"/>  
        <xs:attribute name="unitModel" type="unitDistanceType"/>  
        <xs:attribute name="scale" type="xs:double" default="1"/>  
        <xs:attribute name="filePath" type="xs:string"/>  
        <xs:attribute name="tiling" type="doubleIntegerTypeMin1" default="1 1"/>  
        <xs:attribute name="offset" type="doubleIntegerType" default="0 0"/>  
        <xs:attribute name="textureFilePath" type="xs:string"/>  
      </xs:complexType>  
    </xs:element>  
  </xs:sequence>  
</xs:complexType>
```

```
<xs:complexType name="offsetType">  
  <xs:attribute name="unitDistance" type="unitDistanceType"/>  
  <xs:attribute name="pos" type="tripleDoubleType" default="0 0 0"/>  
  <xs:attribute name="quat" type="quadDoubleType" default="0 0 0 1"/>  
</xs:complexType>
```

```
<xs:simpleType name="jointTypeType">
```

Revision A

```
<xs:restriction base="xs:string">
  <xs:enumeration value="rotational"/>
  <xs:enumeration value="translational"/>
</xs:restriction>
</xs:simpleType>

<xs:complexType name="jointType">
  <xs:sequence>
    <xs:element name="note" type="xs:string" default="" minOccurs="0" maxOccurs="1"/>
    <xs:element name="pin" type="offsetType" minOccurs="0" maxOccurs="1"/>
    <xs:element name="geometry" type="geometryType" minOccurs="0" maxOccurs="1"/>
    <xs:element name="object" type="objectType" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="joint" type="jointType" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="thruster" type="thrusterType" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="id" type="xs:string" use="required"/>
  <xs:attribute name="name" type="xs:string" use="required"/>
  <xs:attribute name="type" type="jointTypeType" use="required"/>
</xs:complexType>

<xs:complexType name="thrusterType">
  <xs:all>
    <xs:element name="note" type="xs:string" default="" minOccurs="0" maxOccurs="1"/>
    <xs:element name="offset" type="offsetType" minOccurs="0" maxOccurs="1"/>
    <xs:element name="geometry" type="geometryType" minOccurs="0" maxOccurs="1"/>
  </xs:all>
  <xs:attribute name="id" type="xs:string" use="required"/>
  <xs:attribute name="name" type="xs:string" use="required"/>
</xs:complexType>

<xs:complexType name="objectType">
  <xs:sequence>
    <xs:element name="note" type="xs:string" default="" minOccurs="0" maxOccurs="1"/>
    <xs:element name="offset" type="offsetType" minOccurs="0" maxOccurs="1"/>
    <xs:element name="geometry" type="geometryType" minOccurs="0" maxOccurs="1"/>
    <xs:element name="object" type="objectType" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="joint" type="jointType" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="thruster" type="thrusterType" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="terrain" type="terrainType" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="id" type="xs:string" use="required"/>
  <xs:attribute name="name" type="xs:string" use="required"/>
  <xs:attribute name="isStatic" default="1">
    <xs:simpleType>
      <xs:restriction base="xs:integer">
        <xs:minInclusive value="0"/>
        <xs:maxInclusive value="1"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:attribute>
</xs:complexType>

<xs:complexType name="terrainType">
  <xs:sequence>
    <xs:element name="note" type="xs:string" default="" minOccurs="0" maxOccurs="1"/>
    <xs:element name="offset" type="offsetType" minOccurs="0" maxOccurs="1"/>
```

Revision A

```
<xs:element name="geometry" type=" terrainGeometryType " minOccurs="0" maxOccurs="1"/>
<xs:element name="object" type="objectType" minOccurs="0" maxOccurs="unbounded"/>
<xs:element name="terrain" type="terrainType" minOccurs="0" maxOccurs="unbounded"/>
</xs:sequence>
<xs:attribute name="id" type="xs:string" use="required"/>
<xs:attribute name="name" type="xs:string" use="required"/>
</xs:complexType>

<xs:simpleType name="telemetryDeliveryType">
<xs:restriction base="xs:string">
<xs:enumeration value="file"/>
<xs:enumeration value="network"/>
</xs:restriction>
</xs:simpleType>

<xs:simpleType name="telemetryFormatType">
<xs:restriction base="xs:string">
<xs:enumeration value="MPC"/>
<xs:enumeration value="MPC2"/>
<xs:enumeration value="MPC3Data"/>
</xs:restriction>
</xs:simpleType>

<xs:complexType name="skyboxType">
<xs:all>
<xs:element name="front" type="textureType" minOccurs="1" maxOccurs="1"/>
<xs:element name="back" type="textureType" minOccurs="1" maxOccurs="1"/>
<xs:element name="top" type="textureType" minOccurs="1" maxOccurs="1"/>
<xs:element name="bottom" type="textureType" minOccurs="1" maxOccurs="1"/>
<xs:element name="left" type="textureType" minOccurs="1" maxOccurs="1"/>
<xs:element name="right" type="textureType" minOccurs="1" maxOccurs="1"/>
</xs:all>
</xs:complexType>

<xs:complexType name="backgroundColorType">
<xs:attribute name="color" type="rgbType" use="required"/>
</xs:complexType>

<xs:complexType name="lightType">
<xs:attribute name="type" type="lightType" use="required"/>
<xs:attribute name="parent" type="xs:string" default=""/>
<xs:attribute name="color" type="rgbType" default="1 1 1"/>
<xs:attribute name="unitDistance" type="unitDistanceType"/>
<xs:attribute name="pos" type="tripleDoubleType" default="0 0 0"/>
<xs:attribute name="forward" type="tripleDoubleType" default="0 0 1"/>
<xs:attribute name="intensity" type="intensityType" default="100"/>
<xs:attribute name="unitAngle" type="unitAngleType"/>
<xs:attribute name="spotCutoff" type="spotCutoffType" default="180"/>
</xs:complexType>

<xs:complexType name="cameraType">
<xs:attribute name="name" type="xs:string" use="required"/>
<xs:attribute name="unitAngle" type="unitAngleType"/>
<xs:attribute name="spotCutoff" type="cameraFovType" default="45"/>
<xs:attribute name="unitDistance" type="unitDistanceType"/>
<xs:attribute name="pos" type="tripleDoubleType" default="0 0 0"/>
```

Revision A

```

<xs:attribute name="up" type="tripleDoubleType" default="0 1 0"/>
<xs:attribute name="forward" type="tripleDoubleType" default="0 0 1"/>
<xs:attribute name="parent" type="xs:string" default=""/>
<xs:attribute name="track" type="xs:string" default=""/>
</xs:complexType>

<xs:complexType name="telemetryType">
  <xs:all>
    <xs:element name="note" type="xs:string" default="" minOccurs="0" maxOccurs="1"/>
  </xs:all>
  <xs:attribute name="name" type="xs:string" use="required"/>
  <xs:attribute name="type" type="telemetryDeliveryType" use="required"/>
  <xs:attribute name="format" type="telemetryFormatType" use="required"/>
  <xs:attribute name="epoch" type="xs:dateTime"/>
  <xs:attribute name="source" type="xs:string" use="required"/>
</xs:complexType>

<xs:complexType name="environmentType">
  <xs:sequence>
    <xs:element name="backgroundColor" type="backgroundColorType" minOccurs="0"
      maxOccurs="1"/>
    <xs:element name="backgroundTexture" type="textureType" minOccurs="0" maxOccurs="1"/>
    <xs:element name="skybox" type="skyboxType" minOccurs="0" maxOccurs="1"/>
    <xs:element name="light" type="lightType" minOccurs="1" maxOccurs="unbounded"/>
    <xs:element name="camera" type="cameraType" minOccurs="1" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="type" type="environmentType" default="Custom"/>
  <xs:attribute name="up" type="tripleDoubleType" default="0 0 1"/>
  <xs:attribute name="forward" type="tripleDoubleType" default="1 0 0"/>
  <xs:attribute name="baseFilePath" type="xs:string"/>
</xs:complexType>

<xs:complexType name="objectsType">
  <xs:sequence>
    <xs:element name="terrain" type="terrainType" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="object" type="objectType" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="baseFilePath" type="xs:string"/>
</xs:complexType>

<xs:complexType name="sceneType">
  <xs:sequence>
    <xs:element name="note" type="xs:string" default="" minOccurs="0"/>
    <xs:element name="units" type="unitsType" minOccurs="0"/>
    <xs:element name="environment" type="environmentType" use="required"/>
    <xs:element name="objects" type="objectsType" use="required"/>
  </xs:sequence>
  <xs:attribute name="name" type="xs:string" default=""/>
  <xs:attribute name="baseFilePath" type="xs:string"/>
</xs:complexType>

<xs:complexType name="telemetriesType">
  <xs:sequence>
    <xs:element name="telemetry" type="telemetryType" minOccurs="1"
      maxOccurs="unbounded"/>
  </xs:sequence>

```

Revision A

```
<xs:attribute name="baseFilePath" type="xs:string"/>
</xs:complexType>

<xs:element name="mpcMission">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="note" type="xs:string" default="" minOccurs="0"/>
      <xs:element name="scene" type="sceneType" use="required" minOccurs="1"
        maxOccurs="1"/>
      <xs:element name="telemetries" type="telemetriesType" minOccurs="0" maxOccurs="1"/>
    </xs:sequence>
    <xs:attribute name="version" type="xs:string" use="required" fixed="3.0"/>
    <xs:attribute name="baseFilePath" type="xs:string"/>
    <xs:attribute name="epoch" type="xs:dateTime"/>
    <xs:attribute name="author" type="xs:string" default=""/>
  </xs:complexType>
</xs:element>
</xs:schema>
```

Revision A

**APPENDIX C.
XML Schema for MPC3 Data File**

```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <xs:simpleType name="timeUnitsType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="millisecond"/>
      <xs:enumeration value="second"/>
      <xs:enumeration value="minute"/>
      <xs:enumeration value="hour"/>
      <xs:enumeration value="day"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="distanceUnitsType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="millimeter"/>
      <xs:enumeration value="centimeter"/>
      <xs:enumeration value="meter"/>
      <xs:enumeration value="kilometer"/>
      <xs:enumeration value="AU"/>
      <xs:enumeration value="inch"/>
      <xs:enumeration value="foot"/>
      <xs:enumeration value="yard"/>
      <xs:enumeration value="mile"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="angleUnitsType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="radian"/>
      <xs:enumeration value="degree"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="forceUnitsType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="newton"/>
      <xs:enumeration value="pound"/>
    </xs:restriction>
  </xs:simpleType>

  <xs:complexType name="unitsType">
    <xs:attribute name="time" type="timeUnitsType" default="second"/>
    <xs:attribute name="distance" type="distanceUnitsType" default="meter"/>
    <xs:attribute name="angle" type="angleUnitsType" default="degree"/>
    <xs:attribute name="force" type="forceUnitsType" default="newton"/>
  </xs:complexType>

  <xs:simpleType name="metaDataTypeType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="string"/>
      <xs:enumeration value="double"/>
    </xs:restriction>
  </xs:simpleType>
</xs:schema>
```

Revision A

```

</xs:restriction>
</xs:simpleType>

```

```

<xs:complexType name="metaDataType">
  <xs:attribute name="id" type="xs:string" use="required"/>
  <xs:attribute name="type" type="metaDataTypeType" use="required"/>
  <xs:attribute name="unitLabel" type="xs:string"/>
</xs:complexType>

```

```

<xs:complexType name="metaDataValueType">
  <xs:choice>
    <xs:element name="metaData">
      <xs:attribute name="id" type="xs:string" use="required"/>
      <xs:attribute name="value" type="xs:double" use="required"/>
    </xs:element>
    <xs:element name="metaData">
      <xs:attribute name="id" type="xs:string" use="required"/>
      <xs:attribute name="value" type="xs:string" use="required"/>
    </xs:element>
  </xs:choice>
</xs:complexType>

```

```

<xs:simpleType name="boolType">
  <xs:restriction base="xs:integer">
    <xs:minInclusive value="0"/>
    <xs:maxInclusive value="1"/>
  </xs:restriction>
</xs:simpleType>

```

```

<xs:complexType name="tripleDoubleType">
  <xs:restriction base="xs:string">
    <xs:pattern value="^[+-]?[0-9]*\.[0-9]+([eE][+-]?[0-9]+)?\s[+-]?[0-9]*\.[0-9]+([eE][+-]?[0-9]+)?\s[+-]?[0-9]*\.[0-9]+([eE][+-]?[0-9]+)?$"/>
  </xs:restriction>
</xs:complexType>

```

```

<xs:complexType name="quadDoubleType">
  <xs:restriction base="xs:string">
    <xs:pattern value="^[+-]?[0-9]*\.[0-9]+([eE][+-]?[0-9]+)?\s[+-]?[0-9]*\.[0-9]+([eE][+-]?[0-9]+)?\s[+-]?[0-9]*\.[0-9]+([eE][+-]?[0-9]+)?\s[+-]?[0-9]*\.[0-9]+([eE][+-]?[0-9]+)?$"/>
  </xs:restriction>
</xs:complexType>

```

```

<xs:complexType name="objectInitType">
  <xs:sequence>
    <xs:element name="metaData" type="metaDataType" minOccurs="0"
      maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="id" type="xs:string" use="required"/>
  <xs:attribute name="unitAngle" type="unitAngleType"/>
  <xs:attribute name="unitDistance" type="unitDistanceType"/>
  <xs:attribute name="ofInterest" type="boolType"/>
</xs:complexType>

```

```

<xs:complexType name="thrusterInitType">
  <xs:sequence>

```

Revision A

```
<xs:element name="metaData" type="metaDataType" minOccurs="0"
  maxOccurs="unbounded"/>
</xs:sequence>
<xs:attribute name="id" type="xs:string" use="required"/>
<xs:attribute name="unitForce" type="unitForceType"/>
<xs:attribute name="unitDistance" type="unitDistanceType"/>
</xs:complexType>

<xs:complexType name="jointInitType">
  <xs:sequence>
    <xs:element name="metaData" type="metaDataType" minOccurs="0"
      maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="id" type="xs:string" use="required"/>
  <xs:attribute name="unitAngle" type="unitAngleType"/>
</xs:complexType>

<xs:complexType name="terrainInitType">
  <xs:attribute name="id" type="xs:string" use="required"/>
</xs:complexType>

<xs:complexType name="objectType">
  <xs:sequence>
    <xs:element name="metaData" type="metaDataValueType" minOccurs="0"
      maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="id" type="xs:string" use="required"/>
  <xs:attribute name="pos" type="tripleDoubleType"/>
  <xs:attribute name="vel" type="tripleDoubleType"/>
  <xs:attribute name="quat" type="quadDoubleType"/>
  <xs:attribute name="angularVel" type="quadDoubleType"/>
  <xs:attribute name="cg" type="tripleDoubleType"/>
  <xs:attribute name="scale" type="tripleDoubleType"/>
  <xs:attribute name="vis" type="boolType"/>
  <xs:attribute name="parent" type="xs:string"/>
</xs:complexType>

<xs:complexType name="jointType">
  <xs:sequence>
    <xs:element name="metaData" type="metaDataValueType" minOccurs="0"
      maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="id" type="xs:string" use="required"/>
  <xs:attribute name="value" type="xs:double" use="required"/>
</xs:complexType>

<xs:complexType name="thrusterType">
  <xs:sequence>
    <xs:element name="metaData" type="metaDataValueType" minOccurs="0"
      maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="id" type="xs:string" use="required"/>
  <xs:attribute name="value" type="xs:double" use="required"/>
  <xs:attribute name="pos" type="tripleDoubleType"/>
  <xs:attribute name="quat" type="quadDoubleType"/>
</xs:complexType>
```


Revision A

```
<xs:element name="mpcData">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="init" minOccurs="1" maxOccurs="1">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="note" type="xs:string" default="" minOccurs="0"/>
            <xs:element name="units" type="unitsType" minOccurs="0"/>
            <xs:element name="metaData" type="metaDataType" minOccurs="0"
              maxOccurs="unbounded"/>
            <xs:element name="terrain" type="terrainInitType"/>
            <xs:element name="object" type="objectInitType"/>
            <xs:element name="joint" type="jointInitType"/>
            <xs:element name="thruster" type="thrusterInitType"/>
          </xs:sequence>
          <xs:attribute name="name" type="xs:string" default=""/>
          <xs:attribute name="epoch" type="xs:dateTime"/>
          <xs:attribute name="author" type="xs:string" default=""/>
        </xs:complexType>
      </xs:element>
      <xs:element name="time" minOccurs="0" maxOccurs="unbounded">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="event" type="xs:string" minOccurs="0"/>
            <xs:element name="metaData" type="metaDataValueType" minOccurs="0"
              maxOccurs="unbounded"/>
            <xs:element name="object" type="objectType" minOccurs="0"
              maxOccurs="unbounded"/>
            <xs:element name="joint" type="jointType" minOccurs="0" maxOccurs="unbounded"/>
            <xs:element name="thruster" type="thrusterType" minOccurs="0"
              maxOccurs="unbounded"/>
          </xs:sequence>
          <xs:attribute name="value" type="xs:double" use="required"/>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
    <xs:attribute name="version" type="xs:string" use="required" fixed="3.0"/>
  </xs:complexType>
</xs:element>
</xs:schema>
```

Revision A

APPENDIX D. Sample Mission File

```

<?xml version="1.0" encoding="utf-8"?>
<mpcMission version="3.0" epoch="2013-05-17T19:22:10.00Z" author="author">
  <note>File Description Here</note>
  <scene name="Sample">
    <note>Scene description</note>
    <units time="second" distance="meter" angle="degree" force="newton"/>
    <environment type="Earth" up="0 0 1" forward="1 0 0">
      <!-- Background option
      <backgroundColor color="r g b"/>
      <backgroundTexture tiling="1 1" offset="0 0" filePath="filepath"/>
      -->
      <skybox>
        <front tiling="1 1" offset="0 0" filePath="textures/earth/front.tif"/>
        <back tiling="1 1" offset="0 0" filePath="textures/earth/back.tif"/>
        <top tiling="1 1" offset="0 0" filePath="textures/earth/top.tif"/>
        <bottom tiling="1 1" offset="0 0" filePath="textures/earth/bottom.tif"/>
        <left tiling="1 1" offset="0 0" filePath="textures/earth/left.tif"/>
        <right tiling="1 1" offset="0 0" filePath="textures/earth/right.tif"/>
      </skybox>
      <light type="directional" color="1 1 1" forward="0 0 1" intensity="100"/>
      <light type="point" parent="Rover" color="1 1 1" unitDistance="kilometer" pos="10 0 0"
        forward="0 0 1" intensity="100" spotCutoff="40"/>
      <camera name="mainCam" fov="20" pos="10 10 0" up="0 0 1" forward="0 1 0" />
      <camera name="trackCam" unitAngle="degree" fov="20" unitDistance="inch" pos="10 10
        0" up="0 0 1" forward="0 1 0" track="obj1"/>
      <camera name="objCam" fov="20" parent="Rover" pos="1 0 0" up="0 0 1" forward="0 1
        0"/>
    </environment>
    <objects baseFilePath="data/">
      <terrain id="Base" name="Serenity Base">
        <note>Terrain; terrain ignores any pos/quat telemetry</note>
        <offset unitDistance="meter" pos="0 1 0" quat="0 0 1 1"/>
        <geometry>
          <model type="heightmap" unitDistance="meter" pos="0 1 0" quat="0 0 1 1"
            scale="1" unitModel="feet" filePath="models/ base.tif " tiling="1 1" offset="0 0"
            textureFilePath="textures/base.png"/>
        </geometry>
      </terrain>
      <object id="Rover" name="Rover">
        <note>Normal Six Degree of Freedom Object</note>
        <geometry>
          <model unitDistance="meter" pos="0 1 0" quat="0 0 1 1" scale="1"
            unitModel="meter" filePath="models/rover.obj"/>
        </geometry>
        <thruster id="Booster" name="Main Booster">
          <offset unitDistance="meter" pos="0 1 0" quat="0 0 1 1"/>
          <geometry>

```

Revision A

```
        <model unitDistance="meter" pos="0 1 0" quat="0 0 1 1" scale="1"
          unitModel="meter" filePath="models/plume.obj"/>
      </geometry>
    </thruster>
    <joint id="Pivot" type="rotate" name="Dish Pivot">
      <note>Single Degree of Freedom Rotational Joint</note>
      <pin unitDistance="meter" pos="0 1 0" quat="0 0 1 1"/>
      <geometry>
        <model unitDistance="meter" pos="0 1 0" quat="0 0 1 1" scale="1"
          unitModel="meter" filePath="models/arm.obj"/>
      </geometry>
      <object id="Comm Dish" isStatic="1" name="Communications Dish">
        <note>Static objects inherit parent motion</note>
        <offset unitDistance="meter" pos="0 1 0" quat="0 0 1 1"/>
        <geometry>
          <model unitDistance="meter" pos="0 1 0" quat="0 0 1 1" scale="1"
            unitModel="meter" filePath="models/dish.obj"/>
        </geometry>
      </object>
    </joint>
  </object>
</objects>
</scene>
<telemetries>
  <telemetry name="Proposed Sorty" type="file" format="MPC3" epoch="2013-05-
    17T19:22:10.00Z" source="telemetry/t1.mpc3">
    <note>Telemetry file description</note>
  </telemetry>
  <telemetry name="Collaborative Sorty" type="network" format="MPC3" epoch="2013-05-
    17T19:22:10.00Z" source="111.111.111.111:1000">
    <note>Telemetry stream description</note>
  </telemetry>
</telemetries>
</mpcMission>
```

Revision A

APPENDIX E. Sample Data File

```

<?xml version="1.0" encoding="utf-8"?>
<mpcData version="3.0">
  <init name="Sample" epoch="2013-05-17T19:22:10.00Z" author="author">
    <note>Simulated Sorty at Serenity Base; Live Sensor Capture</note>
    <units time="second" distance="meter" angle="degree" force="newton"/>
    <metaData id="Status" type="string"/>
    <metaData id="Temperature" type="double" unitLabel="degC"/>
    <terrain id="Base"/>
    <object unitDistance="inch" unitAngle="radian" id="Rover">
      <metaData id="Status" type="string"/>
      <metaData id="Engine Temperature" type="double" unitLabel="degC"/>
    </object>
    <joint unitAngle="degree" id="Pivot">
      <metaData id="Status" type="string"/>
    </joint>
    <thruster unitForce="newton" id="Booster">
      <metaData id="Status" type="string"/>
    </thruster>
  </init>
  <time value="0">
    <event>Mission Start</event>
    <metaData id="Status">Starting Mission</metaData>
    <metaData id="Temperature">24.78</metaData>
    <object id="Rover" pos="0 0 0" vel="0 0 0" quat="0 0 0 1" angularVel="0 1 0 0" cg="0 0 0"
      scale="1 1 1" vis="1" parent="Base">
      <metaData id="Status">Cold Start</metaData>
      <metaData id="Engine Temperature">24.78</metaData>
    </object>
    <joint id="Pivot" value="42">
      <metaData id="Status">Hard Stop</metaData>
    </joint>
    <thruster id="Booster" value="0">
      <metaData id="Status">Warmup</metaData>
    </thruster>
  </time>
  <time value="10">
    <event>Precheck Complete</event>
    <metaData id="Status">Deploying Rover</metaData>
    <metaData id="Temperature">24.78</metaData>
    <object id="Rover" pos="1 1 0" vel="1 1 0">
      <metaData id="Status">All Systems Go</metaData>
      <metaData id="Engine Temperature">30.42</metaData>
    </object>
    <thruster id="Booster" value="1000">
      <metaData id="Status">Full Throttle</metaData>
    </thruster>
  </time>
  <time value="20">
    <event>Waypoint 1</event>
    <metaData id="Status">Deploying Rover</metaData>
    <metaData id="Temperature">24.78</metaData>
    <object id="Rover" pos="10 10 0" vel="0 0 0">

```

Revision A

```
      <metaData id="Status">Signal Loss</metaData>
    </object>
    <thruster id="Booster" value="0">
      <metaData id="Status">Shutdown</metaData>
    </thruster>
  </time>
  <time value="30">
    <object id="Rover" pos="10 11 0" vel="0 1 0"/>
  </time>
  <time value="40">
    <event>shutdown</event>
  </time>
</mpcData>
```

Revision A

APPENDIX F. Data Flow Use Cases

In the following examples, boxes will be colored to indicate who performs each roles as follows:

Red: Model Team

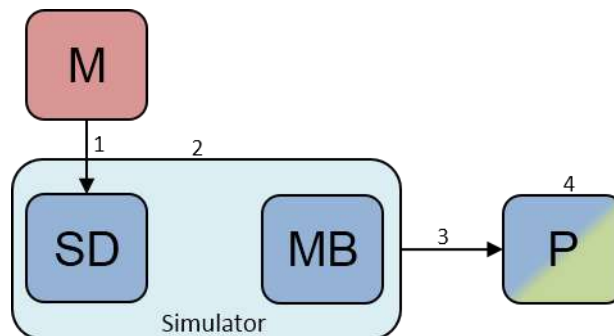
Blue: Simulator Team

Green: DPV Team

F.1 Single Source, Graphical

Scenario: The authoritative data source is a single simulation tool with a built-in graphics display capability.

Examples: Delmia, ROAMS



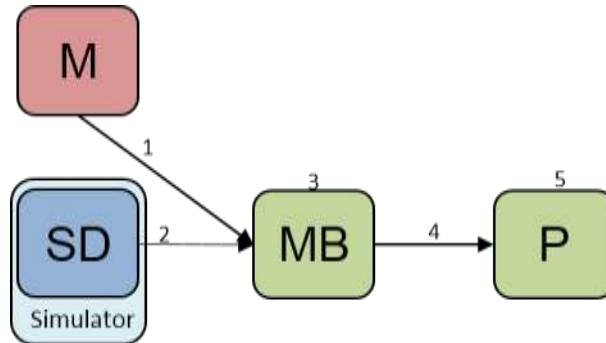
- (1) The design team(s) of the system(s) applicable to the simulation activity construct their 3D models and make them available to the simulation team.
- (2) The simulation team integrates the models within their simulator and conducts the simulation activity.
- (3) The simulation team exports the simulation results and associated models in the proper formats for presentation, and provides the presenter with a single folder containing all exported data.
- (4) The Presenter places the simulation folder containing the exported data into the DON Simulations folder, making it available to DON users. This step may be performed by the simulation team by providing the customer with a copy of DON containing the exported simulation, or it may be performed by the DPV team by placing the exported simulation on a server to make it available for download by existing DON Users.

Revision A

F.2 Single Source, Non-graphical

Scenario: The authoritative data source is a single simulation tool that lacks a built in graphics display capability.

Example: POST



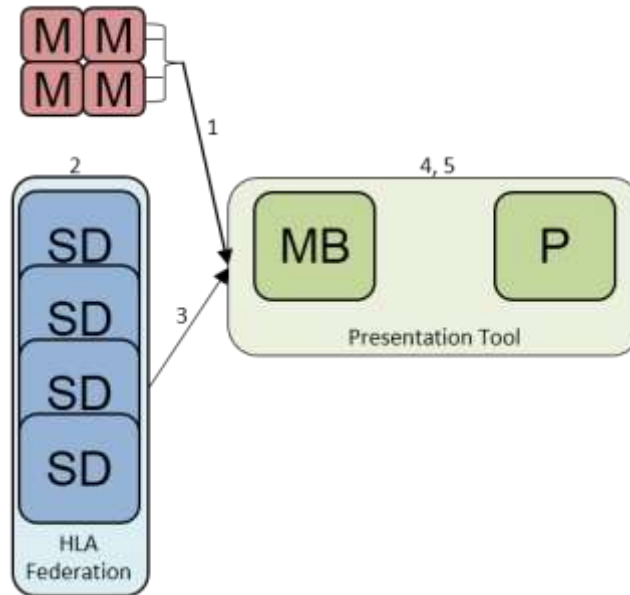
- (1) The design team(s) of the system(s) applicable to the simulation activity construct their 3D models and make them available to the DPV team.
- (2) The simulation team conducts the simulation activity, exports the simulation results in the proper format for presentation, and makes the results available to the DPV team.
- (3) The DPV team integrates the models with the simulation results to build the mission.
- (4) The DPV team exports the mission in the proper formats for presentation and provides the presenter with a single folder containing all exported data.
- (5) The DPV team places the simulation folder containing the exported data into the DON Simulations folder and either provides the customer with a copy of DON containing the exported simulation or places the exported simulation on a server to make it available for download by existing DON Users.

Revision A

F.3 Distributed Simulation Activity

Scenario: Multiple authoritative data sources are connected in an HLA federation, running together real-time, and using DPV tools to display the simulation as it occurs.

Example: integrated mission simulations



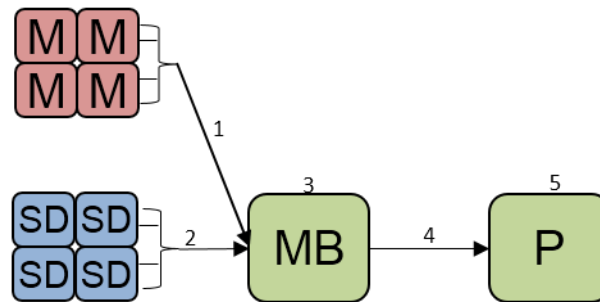
- (1) The design teams of the systems applicable to the simulation activity construct their 3D models and make them available to the DPV team.
- (2) The simulation teams join the HLA federation and conduct the simulation activity.
- (3) The DPV team joins the presentation tool to the HLA federation to receive data during the simulation activity.
- (4) The presentation tool integrates the data received from the HLA federation with the graphical models and updates the simulation in real-time.
- (5) The presentation tool saves all received data and preserves it for future playback and distribution.

Revision A

F.4 Integration Activity

Scenario: Multiple authoritative data sources are providing simulation results that must be integrated with each other.

Example: exploration campaign scenarios



- (1) The design teams of the systems applicable to the simulation activity construct their 3D models and make them available to the DPV team.
- (2) The simulation teams conduct their respective simulation activities, export their simulation results in the proper format for presentation, and make the results available to the DPV team.
- (3) The DPV team integrates the models with the simulation results to build the mission.
- (4) The DPV team exports the mission in the proper formats for presentation and provides the presenter with a single folder containing all exported data.
- (5) The DPV team places the simulation folder containing the exported data into the DON Simulations folder and either provides the customer with a copy of DON containing the exported simulation or places the exported simulation on a server to make it available for download by existing DON Users.