AEROSPACE REPORT NO.
ATR-2015-02436

Specification for Radiation Effects Kernels for Use with AE9/AP9

June 30, 2015

T. P. O'Brien
Space Science Applications Laboratory
Physical Sciences Laboratories

Paul Whelan
Atmospheric and Environmental Research, Inc.
Lexington, MA 02421

Prepared for:

Vice President
Technology and Laboratory Operations

Authorized by: Engineering and Technology Group

FOR PUBLIC RELEASE

Abstract

We specify the mathematics and the XML schema for generic radiation effects kernels for use with AE9/AP9. Kernels capture the relationship of incident flux outside a spacecraft to transient and cumulative effects inside the spacecraft. Kernels are presently applicable only to linear effects (i.e., effects that are proportional to the input spectrum). It is expected that users and third-party developers will be able to generate their own kernels, which a future version of AE9/AP9 will be able to ingest and use without modification of the AE9/AP9 source code. Examples of such kernels include: radiation dose or damage inside complex shielding, single-event effects for specific parts inside custom shielding, and charge accumulation (internal charging) for specific material geometry.

Acknowledgments

The authors acknowledge useful discussions with the AE9/AP9 team.

Contents

Table

1. Introduction

The AE9/AP9 team has developed the concept of "kernels," which will be defined external to AE9/AP9 and can be "read in" by the AE9/AP9 library and used to compute user-defined radiation and plasma effects. Kernels can be used to define any effect that is a linear convolution of the flux or fluence with a response or transform function. Examples of such effects include: ionizing dose or dose rate behind custom shielding geometry, displacement damage or damage rate, single-event effects rate (for a part that does not get progressively noisier), internal charging current (neglecting electrostatic repulsion from built-up potential), and internal charging potential (assuming no discharges). The concept of a kernel was described and prototyped in an Aerospace report[*] for displacement damage in silicon at the center of spherical aluminum shields.

The kernel definition is stored in an Extensible Markup Language (XML) file with the option to store large data tables in one or more separate HDF5 (Hierarchical Data Format version 5) files. This document provides the specification for the XML file. We include two example kernel files, and an XSD file that can be used to validate new kernel XML files.

Because one of the most common uses of kernels will be to estimate effects as a function of depth of shielding, we have included the ability to approximate different shielding materials using the stopping-power equivalence (ratio of mass densities). This approximation has its limitations, but can provide reasonably accurate results for comparing material substitutions, such as aluminum for a tungsten alloy. A master XML library of shielding materials and densities accompanies the kernel definition. The AE9/AP9 application will also be able to support user-defined shielding libraries.

[*] O'Brien, T. P., and B. P. Kwan, *Using pre-computed kernels to accelerate effects calculations for Ae9/AP9: A displacement damage example*, Aerospace Report No. TOR-2013-00529, The Aerospace Corporation, El Segundo, CA, 2013.

2. Kernel Evaluation Algorithm

Many radiation effects have a linear response to input particle flux. The underlying physics of the interaction may be nonlinear in many other ways, but if multiplying the input flux *j(E)* by a constant *c* effectively multiplies the effect by that same constant, then the radiation effect is linear for the purposes of a kernel. Let us denote the radiation effect *y{}*. Then the linearity condition is

$$y\{cj(E)\} = cy\{j(E)\}. \tag{1}$$

Such processes can be written (and are often represented in effects codes) as the convolution of a Green's function and the input flux:

$$y\{j(E)\} = \int_0^\infty G(E)j(E)dE. \tag{2}$$

The Green's function is the impulse response for the radiation effect because

$$G(E) = y\{\delta(E)\} = \int_0^\infty G(E')\delta(E' - E)dE'. \tag{3}$$

In other words, *G(E)* is the effect of a monoenergetic, omnidirectional flux spectrum with energy *E*.

In practice, flux is defined on an energy grid, so we have $j_i = j(E_i)$. Similarly the response is usually given in terms of a dependence on an additional variable *d*, such as depth of shielding, with its own discrete grid (giving d_k). The continuous form with the *d*-dependence is:

$$y_d\{j(E)\} = \int_0^\infty G(E;d)j(E)dE, \tag{4}$$

and the discrete form is:

$$y_k = \sum_i G(E_i;d_k)j(E_i)\Delta E_i. \tag{5}$$

In matrix vector form, we have

$$\vec{y} = \underline{\underline{K}}\vec{j}, \tag{6}$$

where

$$K_{k\square} = G(E_i;d_k)\Delta E_i. \tag{7}$$

Up to now, we have assumed that *j(E)* is an omnidirectional differential flux; that is, a flux per unit energy (such as #/cm^2/s/MeV). (Note: omnidirectional means integrated over all angles, thus there are no steradians in the flux units). However, it is common to express flux in terms of integral flux, and AE9/AP9 can run this way as well. Therefore, it is sometimes helpful to specify a separate kernel for integral flux. We denote integral flux above a threshold E as $j_>(E)$, defined as:

$$j_>(E) = \int_E^\infty j(E')dE'. \tag{8}$$

We can manipulate Eq. (2) via integration by parts to obtain

$$y\{j(E)\} = \int_0^\infty \left.\frac{dG}{dE}\right|_E j_>(E)dE, \tag{9}$$

assuming, quite reasonably, that zero-energy particles have no effect and that the flux $j(E)$ goes to zero before E reaches infinity. Thus, we can also produce a kernel appropriate for integral flux via:

$$\vec{y} = \underline{\underline{K}}_>\vec{j}_>, \tag{10}$$

where

$$K_{>ki} = \left.\frac{\partial G(E;d_k)}{\partial E}\right|_{E_i} \Delta E_i. \tag{11}$$

It is equally appropriate to represent $K_{>ki}$ as

$$K_{>ki} = G(E_i+\Delta E_i/2; d_k) - G(E_i-\Delta E_i/2; d_k). \tag{12}$$

This form is especially helpful when G is derived by binning results from a Monte Carlo effects simulation. If the output quantity is always non-negative, care must be taken to ensure that all the entries in $\underline{\underline{K}}$ and $\underline{\underline{K}}_>$ are non-negative.

The AE9/AP9 library will ingest the kernel file and perform either Eq. (6) or Eq. (10), depending on the contents of the file and the type of spectrum being produced by the requested AE9/AP9 run. The kernel's input energy grid may be finer than that of AE9/AP9, in which case AE9/AP9 will interpolate in log(E) and log($j(E)$), except where $j(E)$ is zero at one end of the local interpolation, in which case interpolation will be done in log(E) and $j(E)$. Similarly, the user may request a different output grid than d_n, in which case a similar log-log or log-linear interpolation will be used (if allowed by the kernel's XML file).

We note that the output grid need not be a real-valued grid. It can simply be a list of case numbers 1, 2, 3, etc., that represent discrete design options under consideration. For example, each case might represent a different critical location on circuit boards inside a box. In such cases, interpolation should be disabled (OutputInterp set to None), which will also tell AE9/AP9 to disallow a user-specified output grid.

3. Description of XML format

In order to use a generic kernel, the AE9/AP9 library must have the kernel's energy grid E_i, the transform $\underline{\underline{K}}$ and/or $\underline{\underline{K}}_>$, and the output grid d_n. The library can optionally compute ΔE_i if it is not already included in $\underline{\underline{K}}$ and/or $\underline{\underline{K}}_>$. The library also requires some labeling and flags relating to the appropriate use of the kernel. Because many kernels will deal with shielding materials, the library understands properties specific to shielding depth so that it can convert between shielding units and materials based on their mass densities. Table 1 provides a list of the entries expected in a kernel XML file.

Several asterisks (*) appear in Table 1. Where several values are separated by "/" in the "Type" column, it indicates that the entry must be a members of that list. All numeric list, grid, vector, table, or matrix data can be specified either as a set of <Values> or as a reference to a variable in an HDF5 file. When a reference is used, the XML has a <File> block that contains a <FileName> entry and a

Table 1. Fields of a Kernel XML File (*see text)

Property	Type	Description
<Description>	Free text	A long description of the quantity being produced
<Tag>	Letters, numbers, underscore	A short tag that identifies the kernel for use in generating file names
<Species>	e-/H+/He+/O+	Particle species appropriate for kernel
<EnergyGridMeV>	<Values> or <File>*	The list of energies E_i, in MeV. All energies must be positive
<Output>	three components listed below	
(Output grid)	<Values> or <File>*	A list of values d_n on the output grid
(Output) <GridUnits>	<Units> or <ShieldingMaterial>*	The units of d_n, or the special case of a shielding material (see text)
(Output) <OutputUnits>	Free text	The units of the kernel output
<Transform>	<Values> or <File>*	A transform matrix $\underline{\underline{K}}$ or $\underline{\underline{K}}_>$ (at least one Transform is required)
(Transform)<TransformType>	Diff/Integral	Type of transform
<ApplyDeltaE>	true/false	"true" indicates the differential transform matrix does not already include ΔE_i, so it must be included at run-time.
<OutputInterp>	None/Linear/Log	"None" indicates d_n is a discrete case number and cannot be interpolated. The user will not be given the option to select an alternate output grid at runtime. "Linear" indicates that linear interpolation in d is permitted, and the user will be allowed to specify an alternate output grid. "Log" indicates that the output should be log-interpolated when the user requests a grid other than d_n (linear interpolation will still be used when y at either end of the interpolation is zero).
<Uses>	One or more <Use> entry	
<Use>	Accumulation/Transient	Accumulation indicates that the kernel is appropriate for whole-mission accumulation, such as dose or damage. Transient indicates usable for transient worst case phenomena.
<Version>	Free text	A version identifier for the kernel. Recommend #.#.# format
<DevelInfo>	Free text	A description of who developed the kernel and when
<SchemaVersion>	Integer	Indicates XML schema version number. Version 1 is described in this report. Only the latest version (as defined in the XSD file) is allowed.

<VarName> entry. The <FileName> entry is a local file name or a full path. (The means of specifying the search path at runtime for local file names remains to be determined, but should have no effect on the kernel specification). The VarName can include forward slashes "/," which HDF5 uses to specify variables within a group hierarchy. When a set of <Values> is provided, each value is listed in its own <Value> block. For table or matrix data, the <Value> blocks should be enclosed in <Row> blocks as well. Additionally, the <Output> block's <GridUnits> entry can either be a <Units> block containing free text, or it can be the special <ShieldMaterial> block. A <ShieldMaterial> block consists of a <Material> block and a <DepthUnit> block, both of which should be chosen from the materials and units defined in the separate AE9AP9ShieldLibrary.xml file or in a user-defined supplemental material library.

The AE9AP9ShieldLibrary.xml file is provided with AE9/AP9, but it can be supplemented by user-defined materials libraries that follow the same format. A shielding library contains a set of material records (the <MatrialsDensityTbl> block) and unit records (the <DepthUnitsTbl> block). A material is defined by <Shielding> block, which contains a name (<Material> block) and a mass density in g/cm^3 (<Density_gPerCm3> block). A unit is defined by a <Units> block, which contains a name (<Unit> block) and conversion to millimeters (<PerMM> block). The underlying assumption is that materials with the same areal density provide the same effective shielding. This is a decent assumption for many proton effects, but is not as good for electron effects. A special case material of "g/cm2" is defined, which has a mass density of 1 g/cm^3. A corresponding depth unit also called "g/cm2" is defined as 0.1/mm. For example, a depth of 1 cm of Al has a mass density of 2.7 g/cm^3 and an areal density of 2.7 g/cm^2. That is equivalent to 2.7 "g/cm2" depth units of the material called "g/cm2". If the original effects calculation is thus done in terms of areal density rather than an actual material and depth, then it can be used as a kernel via the special material "g/cm2" and depth unit "g/cm2".

XML files can be validated with an XML Schema Definition (XSD) file. The AE9/AP9 project provides XSD files that can be used to validate kernel XML files and shielding library XML files.

4. Example Kernels

We provide two example kernels. The first uses only numerical <Values> entries in the file itself, while the second uses the <File> block to point to several numerical values in an external file. It is expected the users can mix-and-match <Values> and <File> entries in the same XML file.

4.1 XML with <Values> Blocks

```
<?xml version="1.0" encoding="UTF-8" ?>
<!--  AE9AP9 sample value-based kernel definition file for effects
kernels -->
<!--  with the Ae9Ap9 radiation belt model. -->

<Ae9Ap9Kernel  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="AE9AP9Kernel.xsd">
<Description>
This kernel computes the displacement damage in Silicon at the cen-
ter of Al spherical shields. Input is differential proton flux (per
MeV) and output is damage versus depth in mils Al. Displacement Dam-
age is expressed in terms of equivalent 1-MeV neutron flux/fluence.
</Description>
<Tag>DispDamgSiSphAl</Tag>
<Species>H+</Species>
<EnergyGridMeV>
  <Values>
    <Value>1.5</Value>
    <Value>2.9</Value>
    <Value>6.7</Value>
    <Value>15.0</Value>
    <Value>50.0</Value>
    <Value>100.0</Value>
  </Values>
</EnergyGridMeV>
<Output>
  <Values>
    <Value>0.0</Value>
    <Value>233.1</Value>
    <Value>517.0</Value>
    <Value>1500.5</Value>
  </Values>
  <GridUnits>
    <Units>mm</Units>
  </GridUnits>
  <OutputUnits>#/cm^2</OutputUnits>
```

```
</Output>
<Transform>
  <TransformType>Integral</TransformType>
  <Values>
    <Row>
      <Value>0.53754</Value>
      <Value>2.96543</Value>
      <Value>0.06712</Value>
      <Value>0.00030</Value>
      <Value>1.12430</Value>
      <Value>0.85362</Value>
    </Row>
    <Row>
      <Value>21.5334</Value>
      <Value>22.6689</Value>
      <Value>16.6787</Value>
      <Value>15.0123</Value>
      <Value>5.03223</Value>
      <Value>10.2310</Value>
    </Row>
    <Row>
      <Value>0.11455</Value>
      <Value>0.42774</Value>
      <Value>0.74475</Value>
      <Value>0.00043</Value>
      <Value>0.35830</Value>
      <Value>0.00000</Value>
    </Row>
    <Row>
      <Value>0.00000</Value>
      <Value>0.00000</Value>
      <Value>0.00007</Value>
      <Value>1.12200</Value>
      <Value>0.12320</Value>
      <Value>0.14788</Value>
    </Row>
  </Values>
</Transform>
<ApplyDeltaE>true</ApplyDeltaE>
<OutputInterp>Log</OutputInterp>
<Uses>
  <Use>Accumulation</Use>
</Uses>
<Version>1.0.0</Version>
<DevelInfo>Created 13-Mar-2013 18:09:59 UTC by Paul O'Brien, The
Aerospace Corporation (with interpolation)</DevelInfo>
<SchemaVersion>1</SchemaVersion>
</Ae9Ap9Kernel>
```

4.2 XML with <File> blocks

```
<?xml version="1.0" encoding="UTF-8" ?>
<!--  AE9AP9 prototype kernel definition file using post-processing
kernels -->
<!--  with the Ae9Ap9 radiation belt model. -->

<Ae9Ap9Kernel  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="AE9AP9Kernel.xsd">
<Description>
This kernel computes the displacement damage in Silicon at the cen-
ter of Al spherical shields. Input is differential proton flux (per
MeV) and output is damage versus depth in mils Al. Displacement Dam-
age is expressed in terms of equivalent 1-MeV neutron flux/fluence.
</Description>
<Tag>DispDamgSiSphAl</Tag>
<Species>H+</Species>
<EnergyGridMeV>
  <File>
    <FileName>mulassis_dd_table.mat</FileName>
    <VarName>MeV</VarName>
  </File>
</EnergyGridMeV>
<Output>
  <File>
    <FileName>mulassis_dd_table.mat</FileName>
    <VarName>depth</VarName>
  </File>
  <GridUnits>
    <ShieldMaterial>
      <Material>Al</Material>
      <DepthUnit>mil</DepthUnit>
    </ShieldMaterial>
  </GridUnits>
  <OutputUnits>#/cm^2</OutputUnits>
</Output>
<Transform>
  <TransformType>Diff</TransformType>
  <File>
    <FileName>mulassis_dd_table.mat</FileName>
    <VarName>nFluence</VarName>
  </File>
</Transform>
<ApplyDeltaE>true</ApplyDeltaE>
<OutputInterp>Log</OutputInterp>
<Uses>
  <Use>Accumulation</Use>
  <Use>Transient</Use>
</Uses>
```

```
<Version>1.0.0</Version>
<DevelInfo>Created 13-Mar-2013 18:09:59 UTC by Paul O'Brien, The
Aerospace Corporation (with interpolation)</DevelInfo>
<SchemaVersion>1</SchemaVersion>
</Ae9Ap9Kernel>
```

4.3 Governing XSD for Kernel Files

The XSD file for the kernel XML files is given below. The XSD file is managed by the AE9/AP9 team and is not subject to user editing. The XSD file is called AE9AP9Kernel.xsd.

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">

 <!-- SchemaVersion definition/restriction -->

 <xs:simpleType name="schemaVersionType">
  <xs:restriction base="xs:unsignedLong">
    <!-- Only latest version allowed -->
    <xs:enumeration value="1"/>
  </xs:restriction>
</xs:simpleType>


<!-- Base types -->

<xs:simpleType name="stringType">
  <xs:restriction base="xs:string"/>
</xs:simpleType>

<!-- Restrictions on base types -->

<xs:simpleType name="posValType">
  <xs:restriction base="xs:double">
    <xs:minInclusive value="0.0"/>
  </xs:restriction>
</xs:simpleType>

<!-- Vector data -->

<xs:complexType name="vectorType">
  <xs:sequence>
      <xs:element name="Value" type="posValType"
maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>

<!-- Matrix data -->
```

```
<xs:complexType name="matrixType">
  <xs:sequence>
      <xs:element name="Row" type="vectorType"
maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>

<!-- Complex types -->

<!-- Species type -->

<xs:simpleType name="speciesType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="e-"/>
    <xs:enumeration value="H+"/>
    <xs:enumeration value="He+"/>
    <xs:enumeration value="O+"/>
  </xs:restriction>
</xs:simpleType>

<!-- File-Variable pairs -->

<xs:complexType name="fileVarType">
  <xs:sequence>
    <xs:element name="FileName" type="stringType"/>
    <xs:element name="VarName" type="stringType"/>
  </xs:sequence>
</xs:complexType>

<!-- Energy and output grid sources -->

<xs:complexType name="gridType">
  <xs:sequence>
    <xs:choice>
      <xs:element name="Values" type="vectorType"/>
      <xs:element name="File" type="fileVarType"/>
    </xs:choice>
  </xs:sequence>
</xs:complexType>

<!-- Output grid parameters -->

<xs:complexType name="shieldingType">
  <xs:sequence>
    <xs:element name="Material" type="stringType"/>
    <xs:element name="DepthUnit" type="stringType"/>
  </xs:sequence>
</xs:complexType>
```

```
<xs:complexType name="gridUnitType">
  <xs:sequence>
    <xs:choice>
      <xs:element name="ShieldMaterial" type="shieldingType"/>
     <xs:element name="Units" type="stringType"/>
    </xs:choice>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="OutputType">
  <xs:sequence>
    <xs:choice>
      <xs:element name="Values" type="vectorType"/>
      <xs:element name="File" type="fileVarType"/>
    </xs:choice>
    <xs:element name="GridUnits" type="gridUnitType"/>
    <xs:element name="OutputUnits" type="stringType"/>
  </xs:sequence>
</xs:complexType>


<!-- Output Interpolation type -->

<xs:simpleType name="interpType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="None"/>
    <xs:enumeration value="Linear"/>
    <xs:enumeration value="Log"/>
  </xs:restriction>
</xs:simpleType>

<!-- Transform matrix source or data -->

<xs:simpleType name="transformStringType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Diff"/>
    <xs:enumeration value="Integral"/>
  </xs:restriction>
</xs:simpleType>

<xs:complexType name="transformType">
  <xs:sequence>
    <xs:element name="TransformType" type="transformStringType"/>
    <xs:choice>
      <xs:element name="Values" type="matrixType"/>
      <xs:element name="File" type="fileVarType"/>
    </xs:choice>
  </xs:sequence>
</xs:complexType>
```

```
<!-- Uses definition -->

<xs:simpleType name="useStringType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Accumulation"/>
    <xs:enumeration value="Transient"/>
  </xs:restriction>
</xs:simpleType>

<xs:complexType name="usesType">
  <xs:sequence>
    <xs:element name="Use" type="useStringType"
maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>

<!-- Kernel definition -->

<xs:element name="Ae9Ap9Kernel">
  <xs:complexType>
  <xs:sequence>
  <xs:element name="Description" type="stringType"/>
  <xs:element name="Tag" type="stringType"/>
  <xs:element name="Species" type="speciesType"/>
  <xs:element name="EnergyGridMeV" type="gridType"/>
  <xs:element name="Output" type="OutputType"/>
  <xs:element name="Transform" type="transformType" maxOccurs="2"/>
  <xs:element name="ApplyDeltaE" type="xs:boolean"/>
  <xs:element name="OutputInterp" type="interpType"/>
  <xs:element name="Uses" type="usesType" maxOccurs="2"/>
  <xs:element name="Version" type="stringType"/>
  <xs:element name="DevelInfo" type="stringType"/>
  <xs:element name="SchemaVersion" type="schemaVersionType"/>
  </xs:sequence>
  </xs:complexType>
</xs:element>

</xs:schema>
```

5. Example Materials and Depths library

The primary Materials and Depths library file is called AE9AP9ShieldingLibrary.xml, and an example is given here. The AE9/AP9 program will include a more extensive XML library in its distribution, and the user is free to add new materials and depth units via user-specified shielding libraries. The governing XSD file is also provided for validating changes to the shielding library. AE9/AP9 verifies that materials listed in the kernel XML file are also defined in the shielding library XML file at run-time and will report an error if the kernel uses a shielding material or depth unit that is not defined in one of the shielding libraries.

5.1 Example XML

Below we provide an example of AE9AP9ShieldingLibrary.xml. This library will be updated by the AE9/AP9 team, and user-defined libraries will also be supported.

```
<?xml version="1.0" encoding="UTF-8" ?>
<!--  AE9AP9 sample shielding data table definition file-->
<!--  for post-processing effects kernels. -->

<shieldingDataTables xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
          xsi:noNamespaceSchemaLocation="AE9AP9Shielding.xsd">

  <Description>Sample material density and depth unit tables for
Ae9Ap9 effects kernels</Description>

  <MaterialDensityTbl>
    <Shielding>
      <Material>g/cm2</Material>
      <Density_gPerCm3>1.0</Density_gPerCm3>
    </Shielding>
    <Shielding>
      <Material>Al</Material>
      <Density_gPerCm3>2.7</Density_gPerCm3>
    </Shielding>
    <Shielding>
      <Material>W</Material>
      <Density_gPerCm3>19.25</Density_gPerCm3>
    </Shielding>
    <Shielding>
      <Material>Kovar</Material>
      <Density_gPerCm3>8.35</Density_gPerCm3>
    </Shielding>
  </MaterialDensityTbl>
```

```
  <DepthUnitsTbl>
    <Units>
      <Unit>g/cm2</Unit>
      <PerMM>0.1</PerMM>
    </Units>
    <Units>
      <Unit>m</Unit>
      <PerMM>0.001</PerMM>
    </Units>
    <Units>
      <Unit>cm</Unit>
      <PerMM>0.1</PerMM>
    </Units>
    <Units>
      <Unit>mm</Unit>
      <PerMM>1.0</PerMM>
    </Units>
    <Units>
      <Unit>in</Unit>
      <PerMM>0.03937</PerMM>
    </Units>
    <Units>
      <Unit>mil</Unit>
      <PerMM>39.37</PerMM>
    </Units>
  </DepthUnitsTbl>
</shieldingDataTables>
```

5.2 Governing XSD

The XSD file for shielding library XML files is given below. The XSD file is managed by the AE9/AP9 team and is not subject to user editing. The XSD file is called AE9AP9Shielding.xsd, and it can be used to validate user-defined AE9/AP9 shielding libraries.

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">

<!-- Base types -->

<xs:simpleType name="stringType">
  <xs:restriction base="xs:string"/>
</xs:simpleType>

<xs:simpleType name="posRealType">
  <xs:restriction base="xs:double">
    <xs:minExclusive value="0.0"/>
  </xs:restriction>
</xs:simpleType>
```

```
<!-- Complex types -->

<xs:complexType name="materialDensityType">
  <xs:sequence>
    <xs:element name="Material" type="stringType"/>
    <xs:element name="Density_gPerCm3" type="posRealType"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="shieldingDensityTableType">
  <xs:sequence>
    <xs:element name="Shielding" type="materialDensityType"
maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="depthUnitsType">
  <xs:sequence>
    <xs:element name="Unit" type="stringType"/>
    <xs:element name="PerMM" type="posRealType"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="depthUnitsTableType">
  <xs:sequence>
    <xs:element name="Units" type="depthUnitsType"
maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>

<xs:element name="shieldingDataTables">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="Description" type="stringType"/>
      <xs:element name="MaterialDensityTbl"
type="shieldingDensityTableType"/>
      <xs:element name="DepthUnitsTbl" type="depthUnitsTableType"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

</xs:schema>
```

6. Acronyms

AE9/AP9 –A radiation climatology model for electrons, protons and plasma

HDF5 – Hierarchical Data Format version 5

XML – Extensible Markup Language

XSD – XML Schema Definition

AEROSPACE REPORT NO.
ATR-2015-02436

Specification for radiation effects kernels for use with AE9/AP9

Approved Electronically by:

Margaret W. Chen, ASSOC DIRECTOR
SPACE SCIENCES DEPARTMENT
SPACE SCIENCE APPLICATIONS LABORATORY
ENGINEERING & TECHNOLOGY GROUP

Charles L. Gustafson, SR VP ENG & TECH
ENGINEERING & TECHNOLOGY GROUP

Office of General Counsel Approval Granted Electronically by:

Danielle E. Sherrod, ATTORNEY SENIOR
OFFICE OF GENERAL COUNSEL & SECRETARY

SC2507