Annotated sample XML schema (file sample.xsd)

The quantum-simulation.org sample XML Schema defines the allowed content of sample documents according to the http://www.quantum-simulation.org specification. Validating XML parsers (such as Apache Xerces-C) use the XML Schema file “sample.xsd” to verify the correctness of sample documents.

XML header

```
<?xml version="1.0"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
  xmlns:fpmd="http://www.quantum-simulation.org/ns/fpmd/fpmd-1.0"
  targetNamespace="http://www.quantum-simulation.org/ns/fpmd/fpmd-1.0">

  <annotation>
    <documentation> $Id: sample.xsd,v 1.12 2008-03-07 20:05:01 fgygi Exp $
      http://www.quantum-simulation.org
      FPMD sample XML Schema specification.
      Copyright (c) 2006-2008 The Regents of the University of California.
    </documentation>
  </annotation>
```

The header section contains the XML declaration of the `<schema>` element and the definition of the XMLSchema and fpmd namespaces. The `<annotation>` section describes the version of the species Schema specification. The current version is `1.12` .

Include species.xsd

```
  <include schemaLocation="species.xsd"/>
```

The included file “species.xsd” contains the XML Schema for species documents, i.e. documents describing atomic species and pseudopotentials. It is included in the sample XML Schema since `<species>` elements are used in `<sample>` elements.

<sample> element

```
  <element name="sample" type="fpmd:sampleType"/>
  <complexType name="sampleType">
    <sequence>
      <element name="description" type="string" minOccurs="0"
maxOccurs="1"/>
      <element name="atomset" type="fpmd:atomsetType" minOccurs="0"/>
      <element name="wavefunction" type="fpmd:wavefunctionType"
minOccurs="0"/>
      <element name="wavefunction_velocity"
         type="fpmd:wavefunctionType" minOccurs="0"/>
    </sequence>
  </complexType>
```

The `<sample>` element consists of a sequence of a `<description>` element, an `<atomset>` element, a `<wavefunction>` element and a `<wavefunction_velocity> e`lement. All these sub-elements are optional. The `<description>` element may contain any information about the sample in the form of a character string of arbitrary length (it can include newline characters). The choice of contents is left to application implementers. The `<description>` may typically include the name and version of the code that generated the sample as well as its creation date. Code-specific parameters used to generate the sample may also be included. There is no constraint on the contents of this element apart from the XML rules regarding character strings.

The `<atomset>` element is declared to be of type `atomsetType`. Both `<wavefunction>` and `<wavefunction_velocity>` elements are declared to be of type `wavefunctionType`. These types are described below.

atomsetType definition

```
  <complexType name="atomsetType">
    <sequence minOccurs="0">
      <element name="unit_cell" type="fpmd:domainType"
               minOccurs="0"/>
      <element name="species" type="fpmd:speciesType"
               minOccurs="0" maxOccurs="unbounded"/>
      <element name="atom" type="fpmd:atomType"
               minOccurs="0" maxOccurs="unbounded"/>
    </sequence>
    <attribute name="href" type="anyURI" use="optional"/>
  </complexType>
```

The `<atomset>` element is of type `atomsetType`. It is used to define the unit cell parameters, the atomic species names, and the atomic positions and velocities. An `<atomset>` is defined as a sequence of a `<unit_cell>` element followed by an arbitrary number of `<species>` elements, and an arbitrary number of `<atom>` elements. An optional `href` attribute can be used to replace the contents of the `<atomset>` with a link to another document. [Note to implementers: The contents of the `<atomset>` elements should be ignored if an `href` attribute is present.] The type `speciesType` is defined in the species.xsd document which is included in this document using the `<include>` element described above. The `<species>` element may consist of a complete definition of the species, or point to the URI of a species document using the `href` attribute, for example a local file:

```
<species name=”carbon” href=”C_LDA.xml”/>
```

or a web-accessible document:

```
<species name=”carbon” href=”http://example.com/potentials/C/C_LDA.xml”/>
```

atomType definition

```
  <complexType name="atomType">
    <sequence>
      <element name="position" type="fpmd:d3vectorType"/>
      <element name="velocity" minOccurs="0" type="fpmd:d3vectorType"/>
    </sequence>
    <attribute name="name" type="NMTOKEN" use="required"/>
    <attribute name="species" type="NMTOKEN" use="required"/>
  </complexType>
```

An `<atom>` element is of type atomType. An `<atom>` element must have a `<position>` element and can optionally have a `<velocity>` element. Both `<position>` and `<velocity>` are 3-vectors, defined in the type `d3vectorType` below. The `<atom>` element must have a `name` attribute, and must have a `species` attribute. The atom name and species must be valid html NMTOKENs. This means that they must contain only alphanumeric characters and numbers, and must not start with a number. The most commonly used atom names are made of the element symbol followed by a number (e.g. “Na17”), although any valid NMTOKEN sequence is admissible.

wavefunctionType definition

```
  <complexType name="wavefunctionType">
    <sequence minOccurs="0">
      <element name="domain" type="fpmd:domainType"/>
      <element name="reference_domain" minOccurs="0"
type="fpmd:domainType"/>
      <element name="grid" type="fpmd:gridType"/>
      <element name="slater_determinant" type="fpmd:slater_determinantType"
               maxOccurs="unbounded"/>
    </sequence>
    <attribute name="ecut" type="fpmd:nonNegativeDouble" use="optional"/>
    <attribute name="nspin" type="fpmd:nspinType" use="required"/>
    <attribute name="nel" type="nonNegativeInteger" use="required"/>
    <attribute name="nempty" type="nonNegativeInteger" default="0"/>
    <attribute name="href" type="anyURI" use="optional"/>
  </complexType>
```

Both the `<wavefunction>` and `<wavefunction_velocity>` elements are of type `wavefunctionType`. A`<wavefunction>` element consists of a `<domain>` element delimiting the region of space on which the wavefunctions are defined. This domain may or may not correspond to the unit cell defined in the `<atomset>` element. The freedom to define the wavefunction domain independently of the atomset unit cell is given to allow for situations in which only a part of the system is described quantum mechanically. The optional `<reference_domain>` element defines a domain that encloses the `<domain>` during variable cell molecular dynamics simulations. The `<grid>` element defines a grid covering the `<domain>`. A `<wavefunction>` element contains at least a `<slater_determinant>` element. Each `<slater_determinant>` defines the one-electron wavefunctions at a given k-point and for a given spin. The optional `<ecut>` element is given as a hint for plane-wave codes to specify the size of a plane-wave basis that accomodates the dimensions of the `<grid>`. The remaining elements specify the total number of spin components of the wavefunction, the total number of electrons and the number of empty states.

domainType

```
  <complexType name="domainType">
    <attribute name="a" type="fpmd:d3vectorType" use="required"/>
    <attribute name="b" type="fpmd:d3vectorType" use="required"/>
    <attribute name="c" type="fpmd:d3vectorType" use="required"/>
  </complexType>
```

The type `domainType` consists of three 3-vectors and is used to define domains in real space. The unit cell is an example of element of type domainType.

gridType

```
  <complexType name="gridType">
    <attribute name="nx" type="positiveInteger" use="required"/>
    <attribute name="ny" type="positiveInteger" use="required"/>
    <attribute name="nz" type="positiveInteger" use="required"/>
  </complexType>
```

The type `gridType` defines the size of the grid covering the wavefunction domain.

slater_determinantType

```
<complexType name="slater_determinantType">
  <sequence minOccurs="0">
    <element name="density_matrix" type="fpmd:density_matrixType"/>
    <element name="grid_function" type="fpmd:grid_functionType"
             maxOccurs="unbounded"/>
  </sequence>
  <attribute name="spin" type="fpmd:spinType" use="optional"/>
  <attribute name="kpoint" type="fpmd:d3vectorType" use="required"/>
  <attribute name="weight" type="double" use="required"/>
  <attribute name="size" type="positiveInteger" use="required"/>
  <attribute name="href" type="anyURI" use="optional"/>
</complexType>
```

The type `slater_determinantType` is a collection of `<grid_function>` elements, and describes a set of one particle orbitals. It includes a `<density_matrix>` element that contains occupation numbers. The spin, kpoint, weight and size attributes further define the `<slater_determinant>` element. The element can be replaced by a reference to another URI using the `href` attribute.

nspinType

```
<simpleType name="nspinType">
  <restriction base="integer">
    <minInclusive value="1"/>
    <maxInclusive value="2"/>
  </restriction>
</simpleType>
```

spinType

```
<simpleType name="spinType">
  <restriction base="string">
  <enumeration value="up"/>
  <enumeration value="down"/>
  </restriction>
</simpleType>
```

d3vectorType

```
<simpleType name="d3vectorType">
  <restriction>
     <simpleType>
       <list itemType="double"/>
     </simpleType>
     <length value="3"/>
  </restriction>
</simpleType>
```

vectorType

```
<complexType name="vectorType">
  <simpleContent>
    <extension base="string">
    <attribute name="type" type="fpmd:numericType" use="required"/>
    <attribute name="size" type="positiveInteger" use="required"/>
    <attribute name="encoding" type="fpmd:encodingType" use="required"/>
    </extension>
  </simpleContent>
</complexType>
```

numericType

```
  <simpleType name="numericType">
    <restriction base="string">
    <enumeration value="double"/>
    <enumeration value="complex"/>
    </restriction>
  </simpleType>
```

encodingType

```
  <simpleType name="encodingType">
    <restriction base="string">
    <enumeration value="text"/>
    <enumeration value="base64"/>
    </restriction>
  </simpleType>
```

grid_functionType

```
  <complexType name="grid_functionType">
    <simpleContent>
      <extension base="string">
      <attribute name="type" type="fpmd:numericType" use="required"/>
      <attribute name="nx" type="positiveInteger" use="required"/>
      <attribute name="ny" type="positiveInteger" use="required"/>
      <attribute name="nz" type="positiveInteger" use="required"/>
      <attribute name="x0" type="nonNegativeInteger" default="0"/>
      <attribute name="y0" type="nonNegativeInteger" default="0"/>
      <attribute name="z0" type="nonNegativeInteger" default="0"/>
      <attribute name="encoding" type="fpmd:encodingType" use="required"/>
      <attribute name="href" type="anyURI" use="optional"/>
      </extension>
    </simpleContent>
  </complexType>
```

The type `grid_functionType` defines a one-electron orbital by giving its value on a subset of the `<grid>` defined in the `<wavefunction>` element. The numeric type (double or complex) is specified by the type attribute. The nx, ny, nz, x0, y0, z0 attributes define the subset of the grid on which the orbital is defined. The subset of the grid has its origin at position (x0,y0,z0) in the grid, and has a size (nx,ny,nz). The encoding attribute (text or base64) is required to specify the type of encoding used for the data. The body of the `<grid_function>` element consists of a string that contains either a list of formatted values (if encoding=”text”) or base64-encoded data (if encoding=”base64”). The order of the elements corresponds to the following loop order:

```
for ( int k = 0; k < nz; k++ )
  for ( int j = 0; j < ny; j++ )
    for ( int i = 0; i < nx; i++ )
      write(f(i,j,k));
```

The base64 encoded data must be consistent with a little-endian byte ordering. It is the responsibility of the application to convert the data on a big-endian architecture. This constraint ensure portability of sample documents across architectures of varying byte-ordering.

density_matrixType

```
  <complexType name="density_matrixType">
    <simpleContent>
      <extension base="fpmd:doubleListType">
        <attribute name="form" type="fpmd:density_matrix_formType"
use="required"/>
        <attribute name="size" type="positiveInteger" use="required"/>
        <attribute name="href" type="anyURI" use="optional"/>
      </extension>
    </simpleContent>
  </complexType>
```

The type `density_matrixType` defines the matrix of occupation numbers of the orbitals in a slater_determinant element. The `form` attribute can have the values “full” or “diagonal”, in which case the density matrix is defined by all its elements or only its diagonal elements, respectively.

density_matrix_formType

```
  <simpleType name="density_matrix_formType">
    <restriction base="string">
    <enumeration value="full"/>
    <enumeration value="diagonal"/>
    </restriction>
  </simpleType>

</schema>
```