VAMDC-XSAMS Reference Guide

Document Information

Editors:	M. Doronin, M.L. Dubernet
Authors:	C. Hill, M. Doronin, T. Marquart
Contributors:	VAMDC WP6 working group
Type of document:	standards documentation
Status:	draft
Distribution:	public
Work package:	WP6
Version:	0.2
Date:	08/06/2011
Document code:	
Document URL:	http://www.vamdc.org/documents/vamdc-xsams-guide_v0.2.pdf

Abstract: This document describes the layout of VAMDC modified version of International Atomic Energy Agency's XML Schema for Atomic, Molecular and Solid Data (XSAMS) version 0.1.1. The modified schema is the working data model for the VAMDC consortium. The adopted name for the modified schema is VAMDC-XSAMS. The internal version of the schema described is 0.2.
The document is based on the official IAEA XSAMS version 0.1.1 documentation.

Version History

Version	Date	Modified By	Description of Change
0.2-r1	23/05/2011	M.Doronin	first draft
0.2-r2	27/05/2011	M.Doronin	corrections based on feedback
0.2-r3	08/06/2011	M.Doronin	Changes in the abstract and document versioning

Acknowledgements

VAMDC is funded under the "Combination of Collaborative Projects and Coordination and Support Actions" Funding Scheme of The Seventh Framework Program. Call topic: INFRA-2008-1.2.2 Scientific Data Infrastructure. Grant Agreement number: 239108.

CONTENTS

CHAPTER

ONE

INTRODUCTION

1.1 Status of VAMDC-XSAMS

The VAMDC consortium include many databases providers from very different fields of atomic, molecular and solid spectroscopy commmunity. The VAMDC consortium has initially implemented the International Atomic Energy Agency's XML Schema for Atomic, Molecular and Solid Data (XSAMS) version 0.1.1 [XSAMS] has found that modifications/additions were necessary in order to meet the needs of implementation and queries. This effort has been carried out within the WP6 workpackage including contributions/requirements from all VAMDC partners. This effort has resulted in the so-called VAMDC-XSAMS schema that is used within VAMDC. Releases of VAMDC-XSAMS are planned every 3 months and it is planned that VAMDC-XSAMS and XSAMS will converge every year. The current version of VAMDC-XSAMS is version 0.2 and it is part of the 11.05 Release of VAMDC Standards. The VAMDC release documents for the VAMDC-XSAMS schema include: a tarball including the VAMDC-XSAMS schema files (main file is xsams.xsd) and some example files [VAMDC-XSAMS-SCHEMA], the current reference guide and its PDF file [VAMDC-XSAMS-PDF], the documentation created directly from the schema [VAMDC-XSAMS-DOC], the changelog document [VAMDC-XSAMS-LOG] that provides the log between the IEAE XSAMS version 0.1.1 [XSAMS] and the current VAMDC-XSAMS version 0.2 [VAMDC-XSAMS-SCHEMA], the documentation for the case-by-case description of molecular states [case-by-case].

1.2 Motivation of XSAMS (and VAMDC-XSAMS)

Many fields in astronomy, physics, energy production and industry depend on databases for atomic, molecular and particle-surface interaction processes (AMPSI). A reliable exchange of such data has been recognized as important for decades, and the Atomic and Molecular (A+M) Data Unit of the International Atomic Energy Agency established a Data Centre Network (DCN) in 1976 to address this vital issue in connection with fusion energy research [Lorenz]. Thus, the ALADDIN data exchange system was adopted in 1988 by the DCN for use by the fusion community in a variety of applications [Janev].

While developments in data exchange within the fusion community were deemed to be adequate two decades ago, a rapid expansion of the Internet resulted in an equally impressive expansion of electronic databases. Most DCN members maintain separate databases, each with a distinct interface and data formats. Independently, the ADAS project was initiated in support of the JET programme [ADAS], and the resulting system has developed into a large project with a dedicated internal method of data exchange. Until recently, difficulties were experienced in obtaining the underlying AMPSI data from ADAS without being a member of the group. Furthermore, the astrophysics community has its own set of dedicated databases and interfaces. Under these circumstances, there is clearly an urgent need for a new data exchange standard based on the latest technology. The International Virtual Observatory Alliance (IVOA) undertook some initial work from 2002 onwards to define and adopt an XML schema for data exchange [Hanisch]. There have been other formats for AMPSI data exchange involving selected data producers and databases, but they did not include global efforts to define a data exchange process.

During the course of the 2004 ICAMDATA meeting in Japan, group discussions focused on the need to develop a more comprehensive schema for AMPSI data for general use. This group consisted of Yu. Ralchenko of the National Institute of Standards and Technology (USA), D. Schultz of the Oak Ridge National Laboratory (USA), M.L. Dubernet of Universit'e Pierre et Marie Curie (France) and E. Roueff of the Observatoire Paris-Meudon

(France), and D. Humbert and R.E.H. Clark of the International Atomic Energy Agency (Austria), and they have met twice every year to develop such a schema. Recently, P. Loboda and S. Gagarin have joined the group from the All-Russian Institute of Technical Physics, Russia. Summary reports of the meetings are available as IAEA reports and can be downloaded from the A+M Unit web site [XSAMS]. The resulting XML Schema for Atoms, Molecules and Solids (XSAMS) can be found on the same A+M Unit website [XSAMS].

The primary objective of the XSAMS exchange standard is to set up a framework for the correct exchange of AMPSI data, rather than to exchange correct data. Important issues such as data verification and data quality are not addressed by XSAMS, which has been established solely as the data exchange procedure. Although XSAMS does require inclusion of documentation of the source and generation of the data, issues related to the correctness or applicability of the data are left for consideration by data producers.

The overall structure of XSAMS consists of specifying physical processes in terms of states. Specific rules are used to describe the physical states of atoms, molecules, ions, elementary particles and solids as unambiguously as possible. States can be specified in any of a number of levels of detail and in different coupling schemes. Processes between states can be described using a reference to initial and final state, as well as a process type. The numerical data for processes can be tabulated, or parameters for a fitting procedure with provision for specifying the exact nature of the fitting function. Different types of differential cross sections as well as total cross sections can be accommodated. Finally, since the origin and history of the data are necessary for data assessment, XSAMS imposes strict requirements on the traceability of the data, with the mandatory inclusion of information on data sources and the methods used to generate specific sets of data.

XSAMS was developed through the joint efforts of researchers from the International Atomic Energy Agency (Austria), National Institute of Standards and Technology (USA), Oak Ridge National Laboratory (USA), Universit'e Pierre et Marie Curie and Observatoire Paris-Meudon (France), with contributions from the All-Russian Institute of Technical Physics (Russia).

The schema for describing states of physical systems and processes connecting different states are detailed in the following sections. All required and optional entries are clearly described, and examples given.

1.3 Limitations of VAMDC-XSAMS

As of June 08, 2011 , some parts of the schema are well developed (e.g., on atomic and molecular structure or processes) while the others are only in the initial phase of development (e.g., solids, environments). It was nevertheless decided to add those less-advanced sections in order to provide some insight into future development of the schema.

1.4 Document goal

The goal of this document is not to create a reference schema documentation, but to define guidelines and examples on filling in data in schema. For reference documentation, see autogenerated HTML docs [VAMDC-XSAMS-DOC].

CHAPTER

TWO

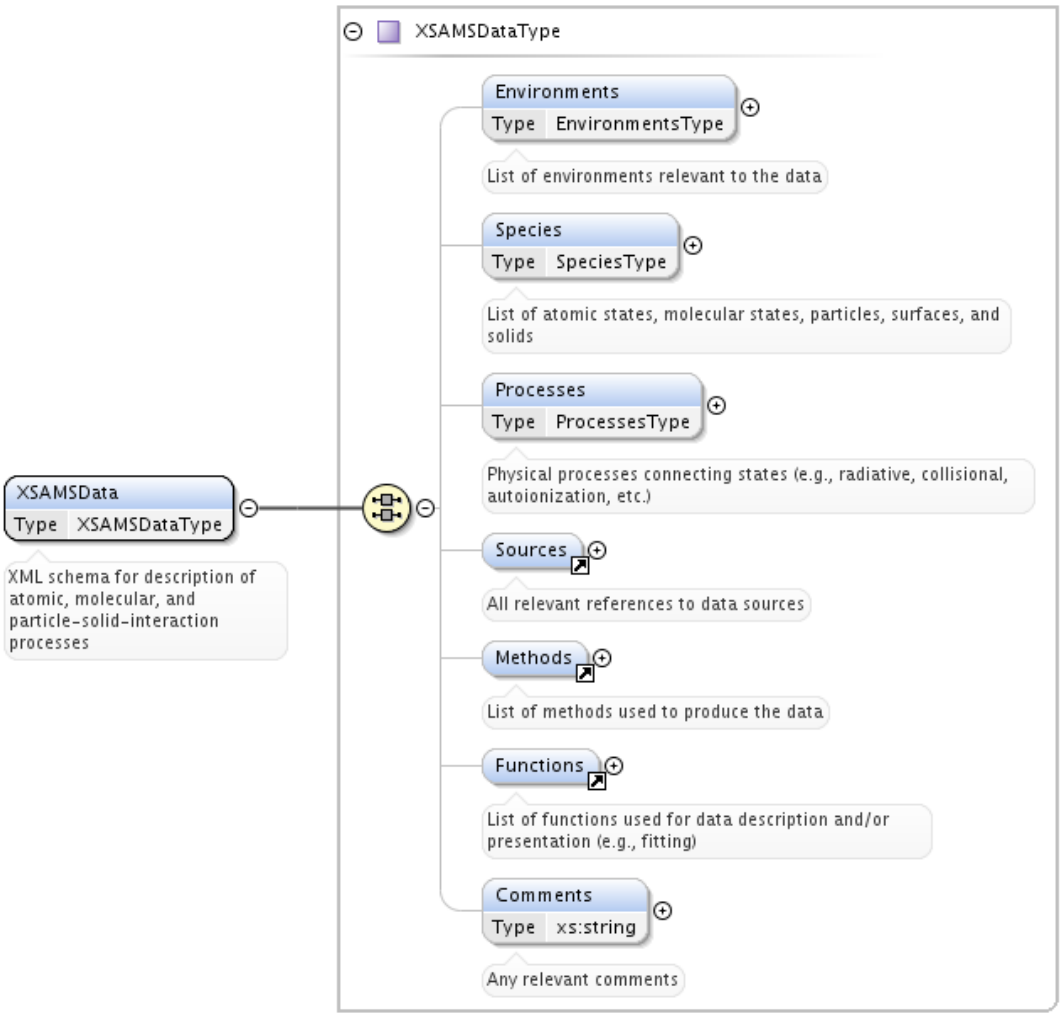
XSAMS STRUCTURE

2.1 Atomic, Molecular, Particle and Surface Interaction Data XML Schema Structure

The XSAMS schema provides a framework for a structured presentation of AMPSI data in an XML file. It is based on the physical representation of interaction between various objects through description of the physical states and interaction characteristics. Therefore, XSAMS provides rules for presentation of

- states of atoms, molecules, solids (surfaces) and some elementary particles,
- characteristics of interaction between physical objects,
- sources of the data.

2.2 XSAMSData



An *XSAMSData* element is defined by the type **XSAMSDataType** which in turn may consist of following optional elements:

- **Environments** of type **EnvironmentsType** that keeps data relevant to description of processes environments,
- *Species* of type **SpeciesType** which may include sub-elements *Species.Atoms*, *Species.Molecules*, *Species.Particles*, *Species.Solids*.
- **Processes** of type **ProcessesType** for state-to-state transitions and interaction processes data, with *Processes.Radiative*, *Processes.NonRadiative*, *Processes.Collisions* optional elements.
- *Sources* of type **SourcesType** for data origin description,
- *Methods* of type **MethodsType** that may contain descriptions of data production methods.
- *Functions* of type **FunctionsType** to keep information on functions used in different parts of schema
- **Comments** for arbitrary description of an XSAMS document.

Example XML instance follows. It has empty **Processes** and **States** branches, but is still a valid XSAMS document instance.

Contrary to IAEA version of XSAMS, VAMDC-XSAMS has a namespace URL defined, **http://vamdc.org/xml/xsams/0.2**

```
<?xml version="1.0" encoding="UTF-8"?>
<XSAMSData xmlns="http://vamdc.org/xml/xsams/0.2"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://vamdc.org/xml/xsams/0.2 xsams.xsd">
        <Species></Species>
        <Processes></Processes>
        <Sources>
                <Source sourceID="BSRC1">
                        <Category>journal</Category>
                        <SourceName>JnlName</SourceName>
                        <Year>2011</Year>
                        <Authors>
                                <Author>
                                        <Name>Smith J.</Name>
                                </Author>
                        </Authors>
                </Source>
        </Sources>
        <Methods>
                <Method methodID="MEX1">
                        <Category>experiment</Category>
                        <Description>High-precision measurements</Description>
                </Method>
        </Methods>
</XSAMSData>
```

CHAPTER

THREE

COMMON TYPES

For the sake of code reuse and reduction of complexity, most of the elements and element attributes of schema are defined as separate types.

3.1 Simple Types

Simple types are used mostly within attributes. For example, all reference types, defining **ID/IDREF** pairs value restriction, are defined as simple types. Here, only types that are used in multiple places of schema are indicated. Types, specific to branches: species(atoms, molecules),processes (radiative,collisions) are described in respective sections of this manual.

3.1.1 AngularMomentumProjectionType

Restriction of double type. Allowed values: integer or half-integer. Examples: **-11.5**, **2**, **+2.0**, **-0.5**.

3.1.2 AngularMomentumType

Restriction of double type. Allowed values: non-negative integer or half-integer. Examples: **11.5**, **2**, **2.0**, **0.5**.

3.1.3 DataDescriptionType

Descriptor for the type of a collisonal parameter, can take one of the following values:

- crossSection
- collisionStrength
- rateCoefficient
- probability
- effectiveCollisionStrength
- sputteringYield
- sputteredEnergyCoefficient
- particleReflectionCoefficient
- energyReflectionCoefficient
- meanPenetrationDepth

3.1.4 DoubleListType

A list of values of type double

3.1.5 ElementSymbolType

Symbol for a chemical element. Allowed values: an upper-case letter that may be followed by a lower-case letter. Examples: **D**, **Hf**, **P**.

3.1.6 EnvironmentIDType

Identifier for environments. May contain any symbols except spaces, first symbol must be **E**.

3.1.7 EnvironmentRefType

Reference for *EnvironmentIDType*

3.1.8 FunctionIDType

Identifier for functions. May contain any symbols except spaces, first symbol must be **F**.

3.1.9 FunctionRefType

Reference for *FunctionIDType*

3.1.10 MethodIDType

Identifier for methods of data production. May contain any symbols except spaces, first symbol must be **M**.

3.1.11 MethodRefType

Reference for *MethodIDType*

3.1.12 MixingClassType

Indicates the nature of the mixing coefficients in the expansion of a wave function in a specific basis. Allowed values:

- squared
- signed

3.1.13 NormalModeIDType

Identifier for vibrational normal modes for complex molecules.. May contain any symbols except spaces, first symbol must be **V**.

3.1.14 NormalModeRefType

Reference for *NormalModeIDType*

3.1.15 OrbitalAngularMomentumSymbolType

symbol for OrbitalAngularMomentumType, any lowercase letter is a valid value.

3.1.16 ParityType

Allowed values:

- even
- odd
- undefined

3.1.17 PrincipalQuantumNumberType

Allowed values: positive integer.

3.1.18 ReferenceFrameType

Descriptor for the frame of reference:

- CenterOfMass
- LaboratoryFrame
- TargetFrame

3.1.19 SourceIDType

Identifier for *Sources* of data. May contain any symbols except spaces, first symbol must be **B**.

3.1.20 SpeciesIDType

Identifier for species. Used in Atoms and Molecules. May contain any symbols except spaces, first symbol must be **X**.

3.1.21 SpeciesRefType

Reference for *SpeciesIDType*

3.1.22 StateIDType

Identifier for a specific state. Used in all species. May contain any symbols except spaces, first symbol must be **S**.

3.1.23 StateRefType

Reference for *StateIDType*

3.1.24 StringListType

Space-separated list of arbitrary strings.

3.1.25 UnitsType

Defines a list of allowed units within XSAMS document. This list is updated and extended as necessary on data producers request. [UnitsML] approach may be adopted in later versions of schema.

The positive powers are indicated by digits, the negative powers follow the backslash symbol **/**. For unitless (dimensionless) parameters, use **unitless**.

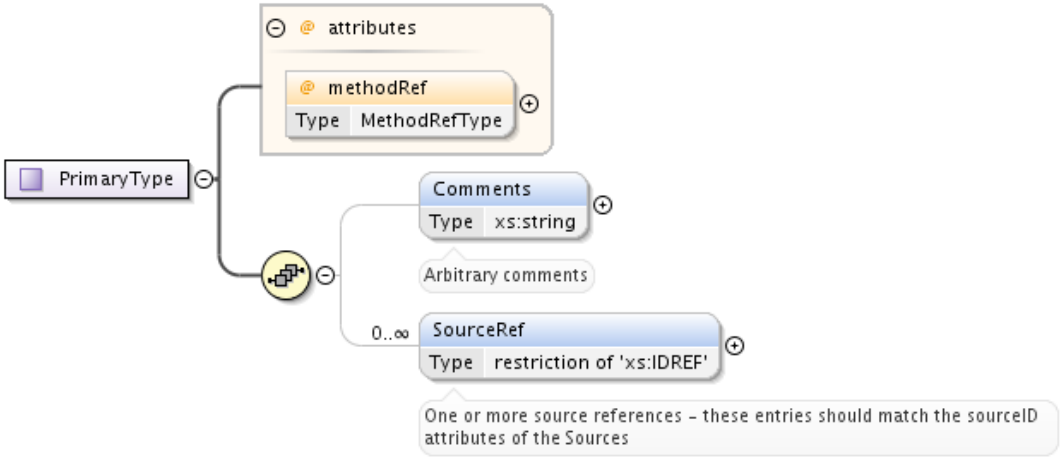
So far, this list contains the following values. If any additional values are required, contact responsibles for schema.

- undef
- eV/amu
- keV/amu
- MeV/amu
- eV
- keV
- MeV
- au
- 1/cm
- J
- Ry
- unitless
- kJ/mol
- kcal/mol
- K
- Hz
- kHz
- MHz
- m
- cm
- A
- nm
- deg
- rad
- srad
- s
- m3/s
- cm3/s
- cm6/s
- m2
- cm2
- b
- Mb

- 1/s
- C.m
- J/T
- C.m2
- m/s
- cm/s
- C
- electron
- g
- amu
- kg
- 1/m2/s
- 1/cm2/s
- J/m2/s
- J/cm2/s
- 1/m2
- 1/cm2
- 1/m3
- 1/cm3
- J/m2
- J/cm2
- W/m2
- W/cm2
- W
- atm
- km/mol
- 1/cm2/atm
- 1/cm/atm

3.2 Complex Types

3.2.1 PrimaryType

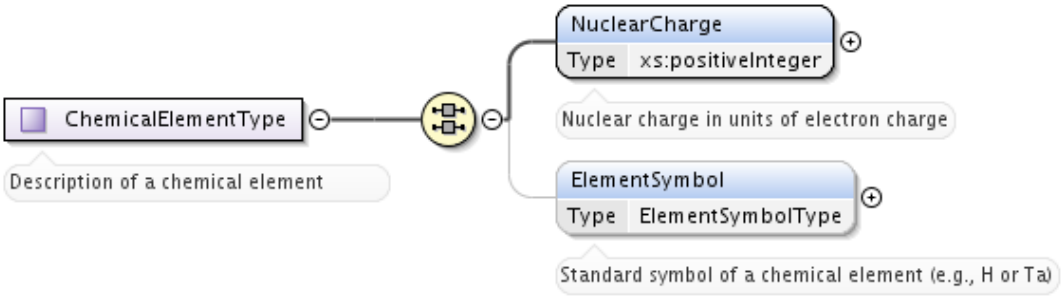


This is the fundamental elementary type used to built other types as extensions. It contains:

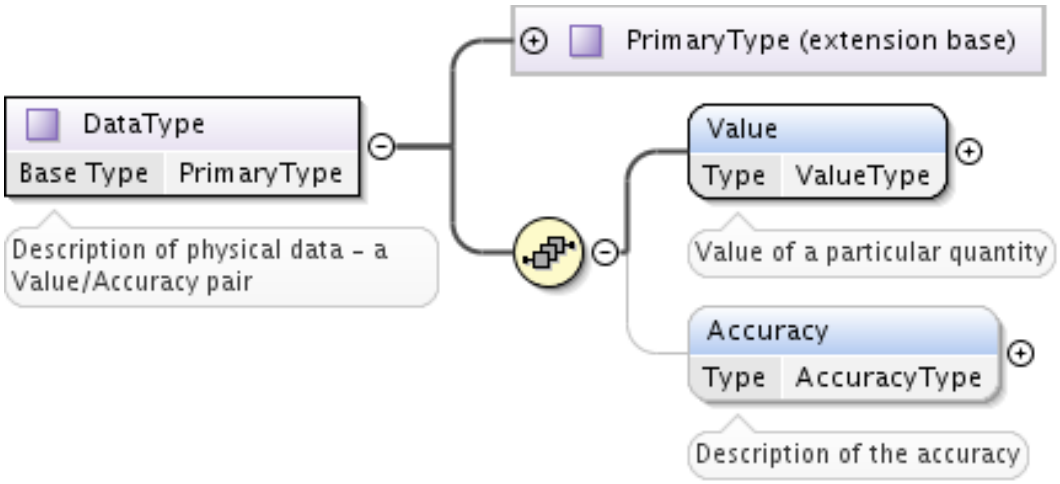
- optional **methodRef** attribute of type *MethodRefType*,
- one or more **SourceRef** elements, containing source identifiers,
- optional **Comments** element for arbitrary comments.

3.2.2 ChemicalElementType

The mandatory element **NuclearCharge** must be a positive integer. The optional element **ElementSymbol** must begin with an upper-case letter which may be followed by a lower-case letter. Examples: **P**, **T**, **Au**.

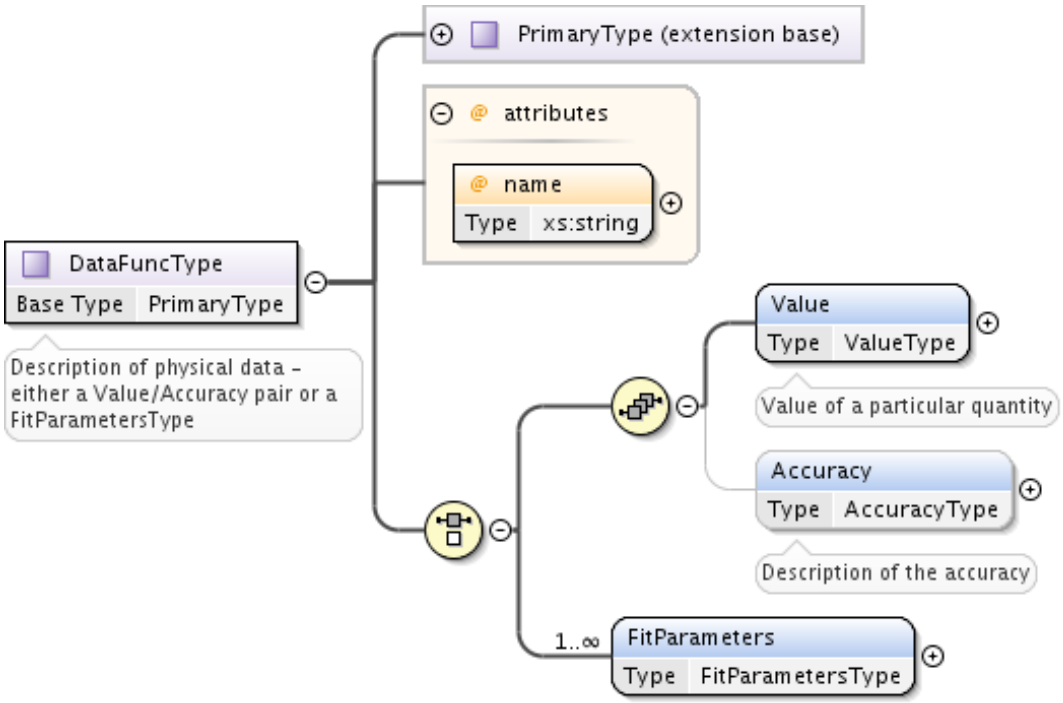


3.2.3 DataType



Extension of the *PrimaryType* which is used for description of numerical data, including units and accuracy. Contains a mandatory **Value** element of type *ValueType* and an optional **Accuracy** element, defined by *AccuracyType*.

3.2.4 DataFuncType

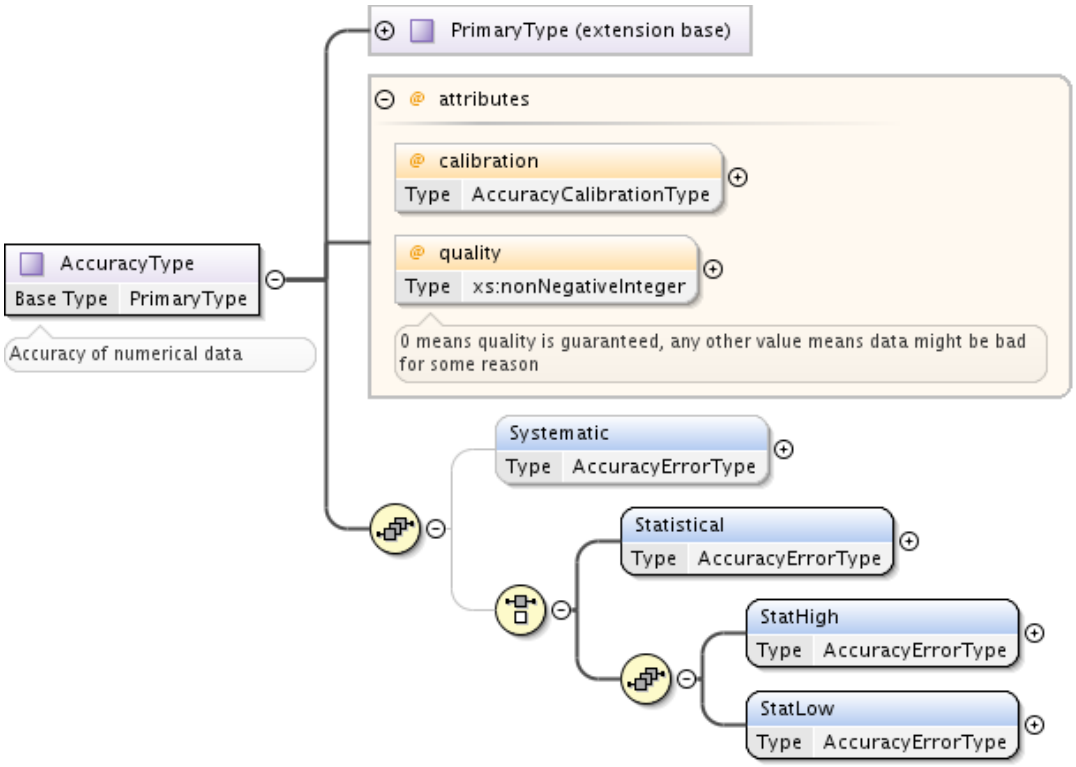


Defined in the similar way as *DataType*, **DataFuncType** has additionally

- mandatory **name** attribute,
- choice between a pair of **value / Accuracy** elements and a list of **FitParameters** elements, defined by *FitParametersType*, each containing sufficient set of parameters needed to calculate the value using some *Function*.

It is used in *Broadening* and *Shifting* sections of [XSAMS] to define lineshape parameters.

3.2.5 AccuracyType

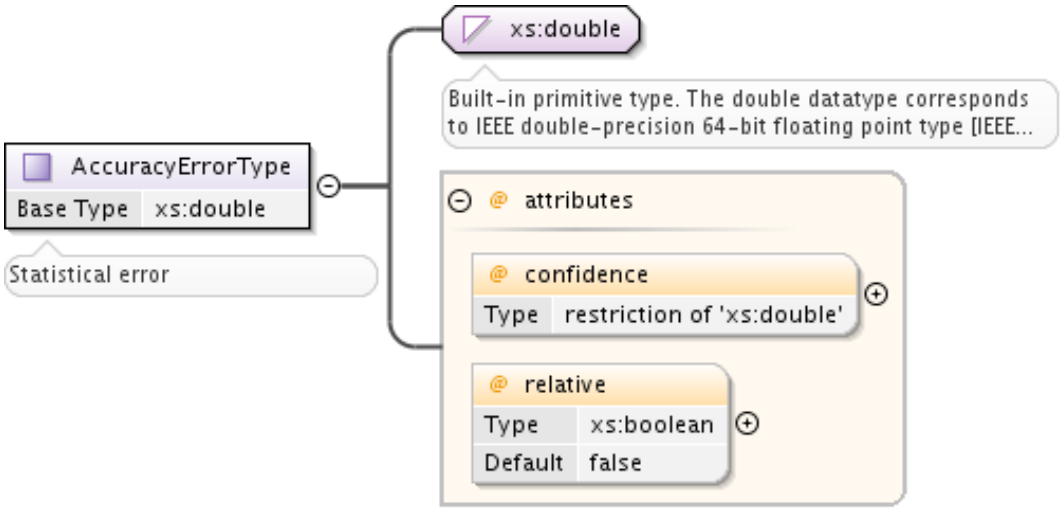


Extension of the *PrimaryType*, describing measurement/calculation accuracy of some physical quantity. Used in the **Accuracy** element of *DataType* and *DataFuncType*. The definition is inspired by [IVOA] *Spectral Data Model* and is still a draft and a subject to change or refine in future versions of [XSAMS].

Following attributes and child elements are defined:

- optional **calibration** attribute, describing the kind of the reference frame for data. It may take values:
  - **absolute** indicates that the values in the data are expected to be correct within the given uncertainty
  - **relative** indicates that although an unknown systematic error is present, the ratio and difference of any two values originating from the same source will be correct.
  - **normalized** indicates that the values, originating from this source, have been divided by a certain reference quantity. In this case units field of ValueType should be 'unitless'
  - **uncalibrated** indicates that not only an unknown systematic error is present in data, originating from that source, but also some unspecified value-dependant error. Thus, for example, for transitions frequencies only the order of transitions is guaranteed, neither frequencies, nor their difference/ratio are accurate.
- Optional **quality** attribute of integer type that may be used for distinguishing quality-assessed data. Zero value means data, accurate within their errors, other values means that there were some problems with data.
- Optional **Systematic** element of type *AccuracyErrorType* for systematic errors
- Optional **Statistical** element of type *AccuracyErrorType* for total statistical error, i.e. upper/lower range are equal.
- Optional **StatHigh** and **StatLow** elements group, also of *AccuracyErrorType*, may be specified instead of single **Statistical** element to indicate statistical errors in case of unequal upper and lower error ranges.

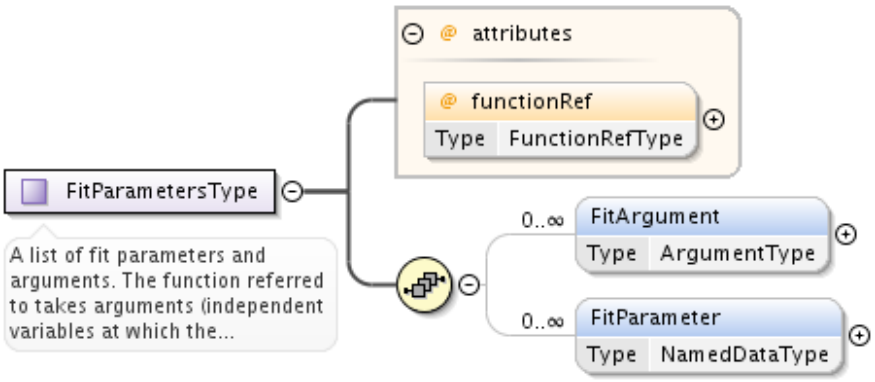
3.2.6 AccuracyErrorType



AccuracyErrorType is an extension of **xs:double** type, adding two optional attributes:

- **confidence** of type **xs:double**, with valid ranges from 0 to 1, indicating confidence level for which this accuracy is calculated. Ususal values would be like **0.95** or **0.99**.
- **relative** of type **xs:boolean**, indicating whether this accuracy value is absolute(**false**) or relative(**true**). By default, accuracy should be treated as absolute.

3.2.7 FitParametersType



FitParametersType defines a full set of data required to calculate some quantity using the predefined function. For a function definition, see *Functions* branch of XSAMS. Following attributes and elements are defined:

- **functionRef** attribute, of type *FunctionRefType*, defining the reference to a function used,
- list of **FitArgument** elements, of type *ArgumentType*, defining validity limits of arguments in particular fit,
- list of **FitParameter** elements, of type **NamedDataType**, giving the function parameters values, possibly with source references. The **NamedDataType** is an extension of *DataType* with a mandatory **name** string attribute.

3.2.8 LifeTimeType

Extension of the *DataType* that defines the additional attribute **decay** with possible values:

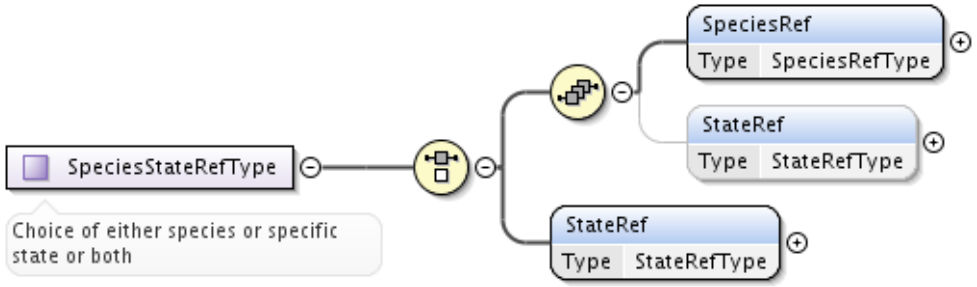
- total,

- totalRadiative,
- totalNonRadiative.

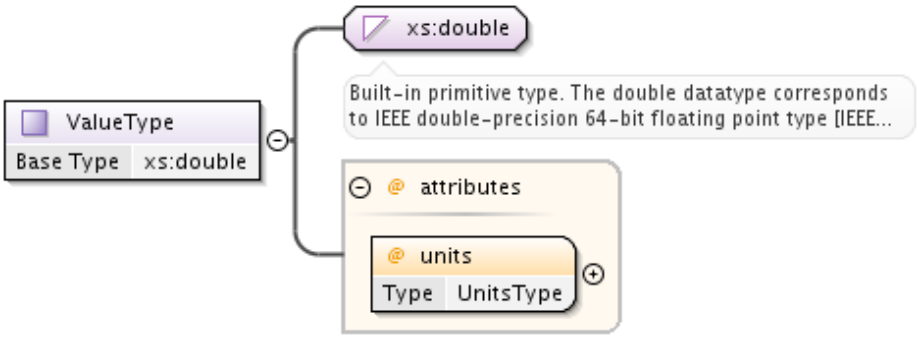
The type is used both in atomic and molecular states to define state lifetime.

3.2.9 SpeciesStateRefType

This type allows to define the **SpeciesRef** (*SpeciesRefType*) or **StateRef** (*StateRefType*) child elements (or both). At least one should be specified, but it is always kind to explicitly provide **SpeciesRef**.



3.2.10 ValueType



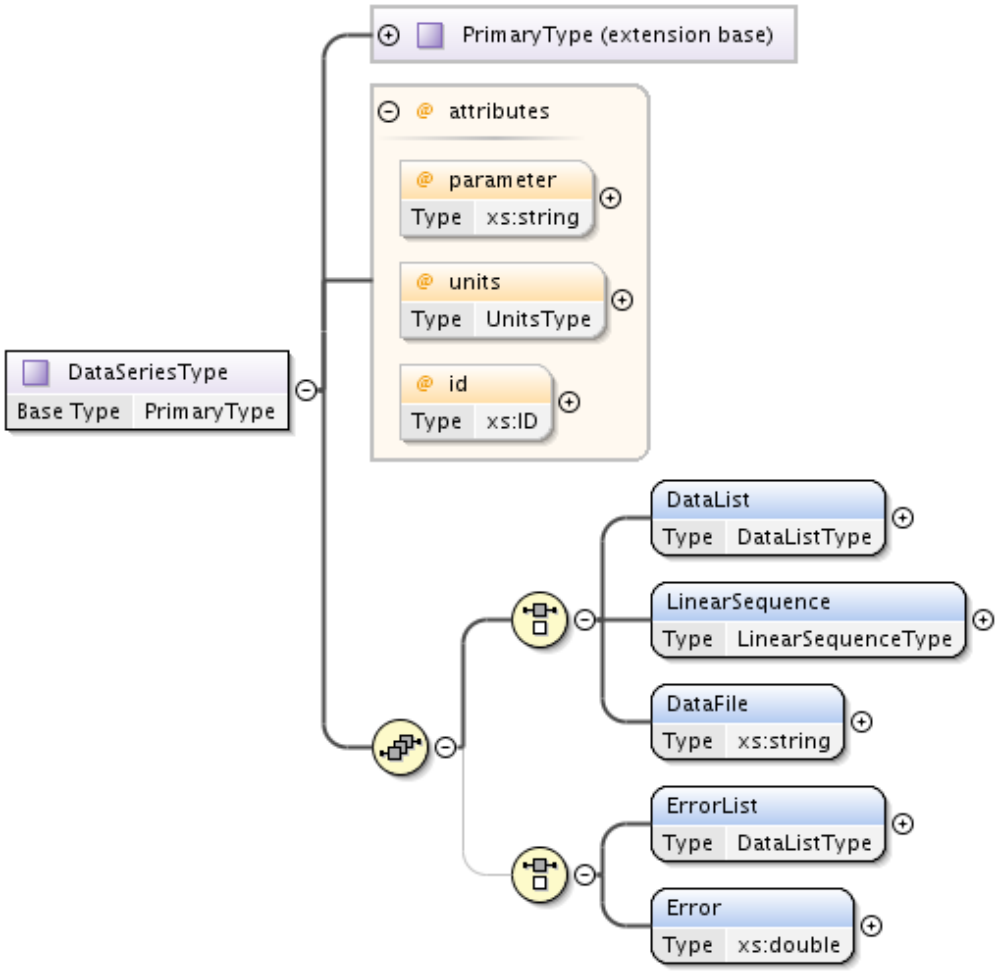
The data must be of type **xs:double** with the mandatory **units** attribute of type *UnitsType*

3.3 Data structures

In this section generic data structures that are used in multiple places in schema are described.

3.3.1 DataSeriesType

DataSeriesType allows description of data as a list of values or a linear sequence, including errors.

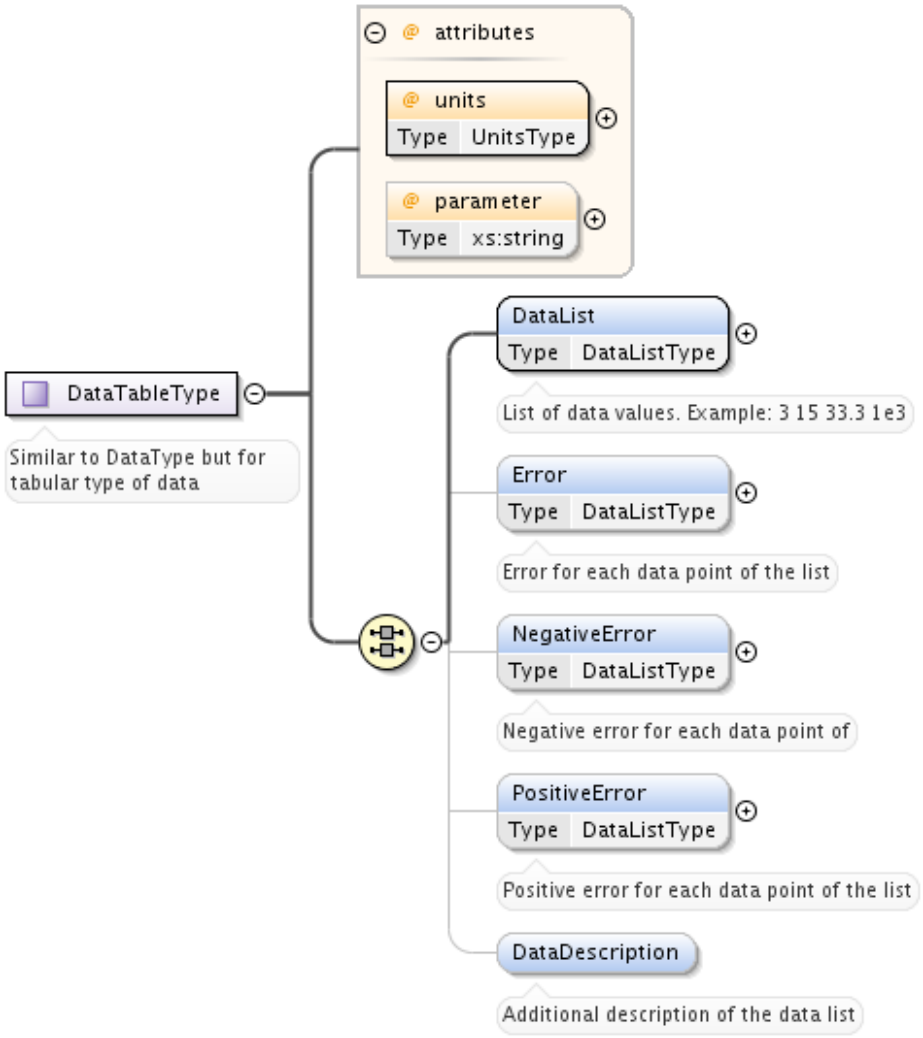


Extends *PrimaryType* to allow source references, adds following attributes and elements:

- optional string attribute **parameter** that may contain the function parameter name or just a description of meaning of sequence.
- optional attribute **units** of *UnitsType* to define the data units
- optional attribute **id** of type **xs:ID** that is reserved for future use.
- mandatory choice of
  - **DataList** element of *DataListType:* to represent a space-separated list of *double* values.
  - **LinearSequence** element of *LinearSequenceType*. Should be used if data is a linear sequence. Example would be to represent frequency/wavelength points for spectrum data.
  - **DataFile** string element that should contain a file name, containing space-separated or newline-separated set of data. This element is also reserved for future use, currently no software is able to pass data files along with XSAMS instance documents.
- optional choise of **Error** or **ErrorList** to define errors for data points. List may be used if error values are different for different points, otherwise single **Error** element should be used.

3.3.2 DataTableType

DataTableType is the type used in Collision's *DataXY* element.



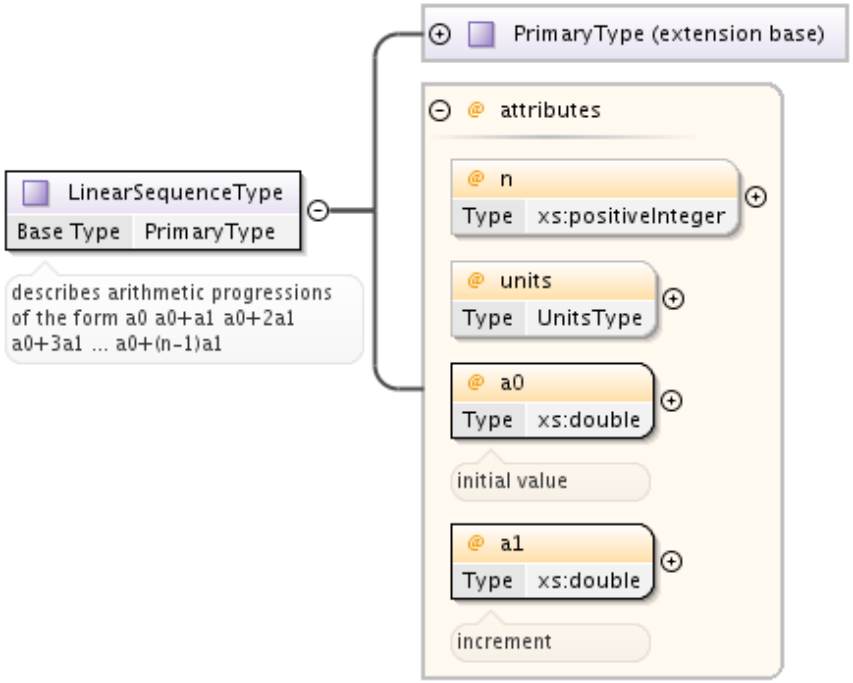
DataTableType has following attributes and elements defined:

- mandatory **units** attribute of type *UnitsType*,
- optional **parameter** attribute to describe the type of unit (e.g., energy, time, or surface...)
- mandatory **DataList** element, of type *DataListType:*, providing the numerical values as a list.
- optional **Error**, providing centered error bar, or
- **NegativeError** and
- **PositiveError** elements, providing asymmetric error bar;
- optional **DataDescription** string element.

All error-related elements have the same type *DataListType:* (list of values of type **xs:double**). All missing error values should be reported as **-1**.

3.3.3 LinearSequenceType

Allows to describe a linear sequence of data, in form of: $a_0, a_0 + a_1, a_0 + 2a_1, ..., a_0 + (n-1) \cdot a_1$

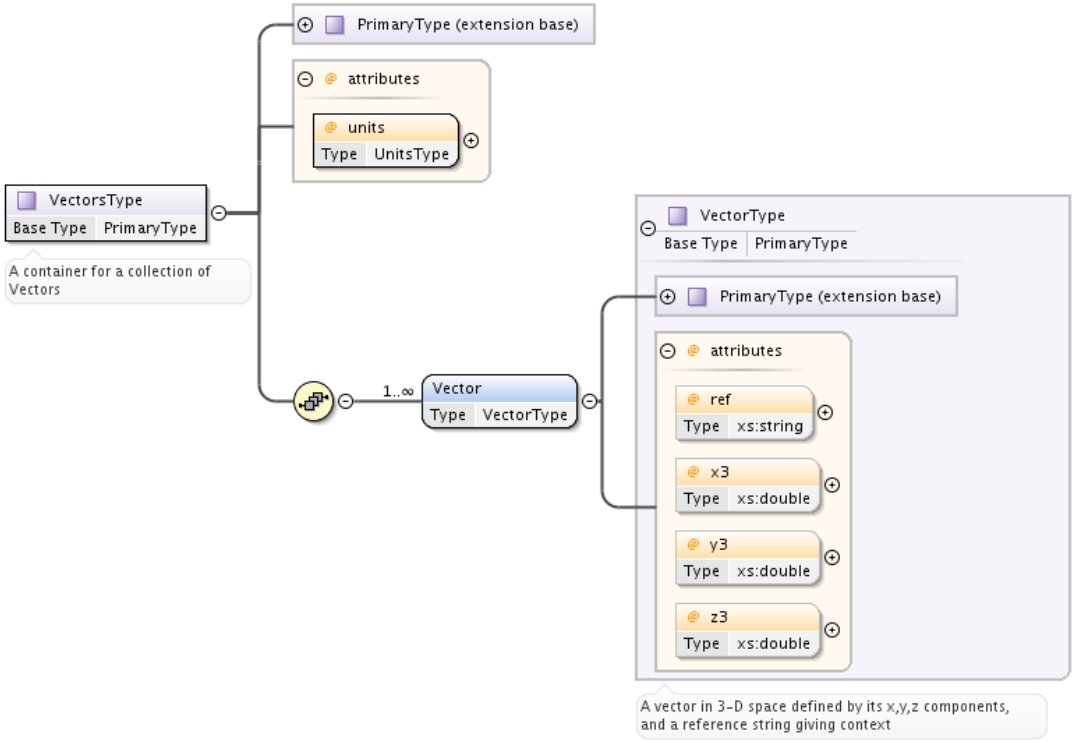


An extension of *PrimaryType*, defines following attributes:

- Mandatory **a0** and **a1** attributes of type **xs:double**
- Optional integer **n** attribute to indicate the count of sequence elements
- Optional **units** attribute of *UnitsType* to define units of data

3.3.4 VectorsType

Defines a collection of vectors in 3D space, specifying source reference, dimensions units and vector reference frame.



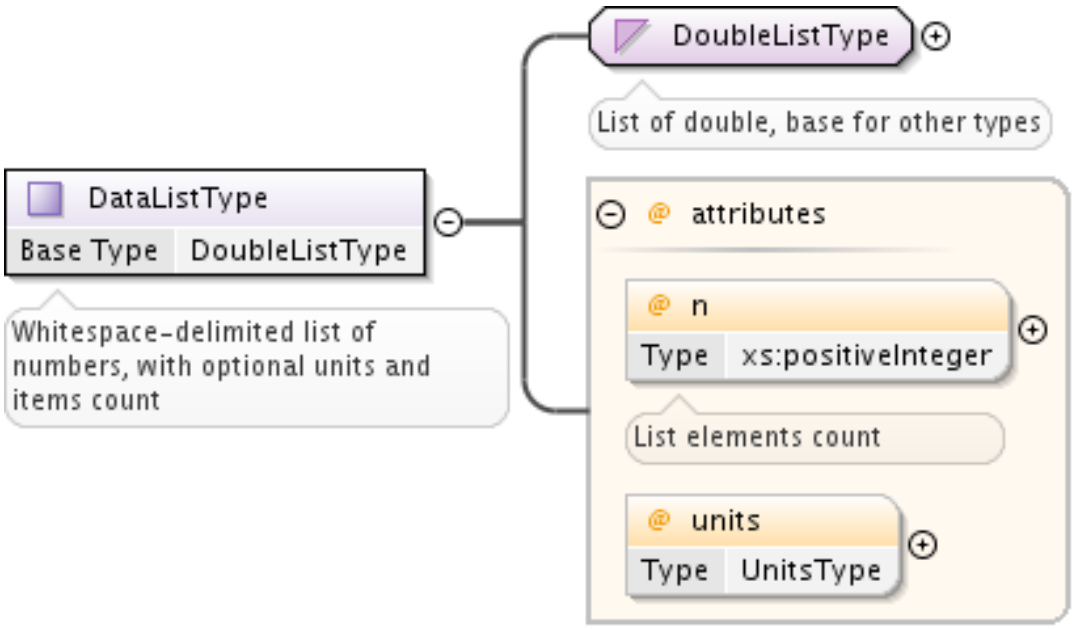
Both **VectorsType** and **Vector** are extending *PrimaryType*.

VectorsType defines **units** attribute of type *UnitsType* and a collection of **Vector** elements.

Vector has following attributes:

- string **ref** that is context-specific and defines the reference element to which the vector applies,
- three **xs:double** coordinate attributes: **x3**, **y3**, **z3**, with units defined in the **Vectors** container.

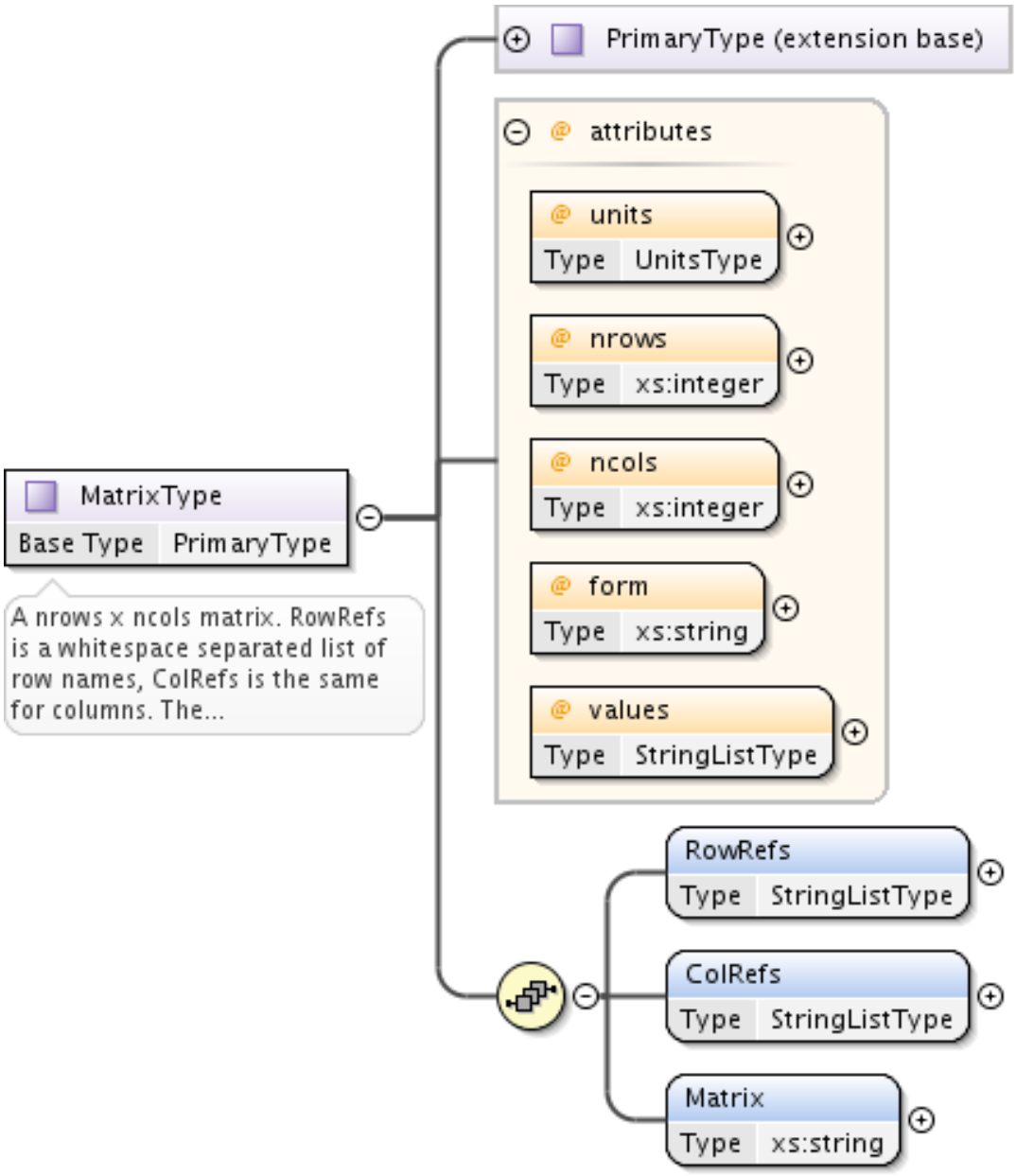
3.3.5 DataListType:



Defines a space-separated list of double precision floating-point numbers, with the following optional attributes:

- **units** of type *UnitsType* to define units of the list
- integer **n** to indicate the number of elements in a list

3.3.6 MatrixType:



A nrows x ncols matrix. Being an extension of the *PrimaryType*, **MatrixType** defines following elements and attributes:

- mandatory **RowRefs** element that contains whitespace separated list of row names,
- mandatory **ColRefs** element that contains the same list for columns.
- mandatory **Matrix** element that contains a whitespace separated list of values, either real or complex
- mandatory integer attribute **nrows** defining the number of rows,
- mandatory integer attribute **ncols** defining the number of columns,
- mandatory attribute **form** that identifies the type of matrix. It can be - *arbitrary* - *symmetric* - *asymmetric* - *diagonal* - *antidiagonal*
- mandatory attribute 'values' that may take values - *binary* - *integer* - *real* - *imaginary* - *complex*

For an **arbitrary** matrix, **Matrix** element has nrows*ncols entries. First you output elements of the first row, starting from the leftmost column.

	c1	c2	c3
r1	0	1	2
r2	3	4	5
r3	6	7	8

```
<MatrixData units="arbitrary" nrows="3" ncols="3" form="arbitrary" values="integer">
        <RowRefs>r1 r2 r3</RowRefs>
        <ColRefs>c1 c2 c3</ColRefs>
        <Matrix>
                0 1 2 3 4 5 6 7 8
        </Matrix>
</MatrixData>
```

For **diagonal** and **antidiagonal** matrix there are nrows=ncols entries, counting from left to right

	c1	c2	c3
r1	1	0	0
r2	0	2	0
r3	0	0	3

```
<MatrixData units="arbitrary" nrows="3" ncols="3" form="diagonal" values="integer">
        <RowRefs>r1 r2 r3</RowRefs>
        <ColRefs>c1 c2 c3</ColRefs>
        <Matrix>
                1 2 3
        </Matrix>
</MatrixData>
```

	c1	c2	c3
r1	0	0	3
r2	0	2	0
r3	1	0	0

```
<MatrixData units="arbitrary" nrows="3" ncols="3" form="antidiagonal" values="integer">
        <RowRefs>r1 r2 r3</RowRefs>
        <ColRefs>c1 c2 c3</ColRefs>
        <Matrix>
                1 2 3
        </Matrix>
</MatrixData>
```

For a **symmetric** matrix there are nrows(nrows+1)/2 entries.

	c1	c2	c3
r1	1	2	3
r2	2	4	5
r3	3	5	6

```
<MatrixData units="arbitrary" nrows="3" ncols="3" form="symmetric" values="integer">
        <RowRefs>r1 r2 r3</RowRefs>
        <ColRefs>c1 c2 c3</ColRefs>
        <Matrix>
                1 2 3 4 5 6
        </Matrix>
</MatrixData>
```

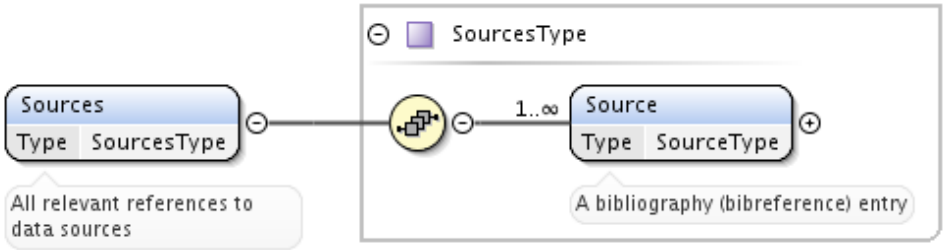
CHAPTER

FOUR

SOURCES

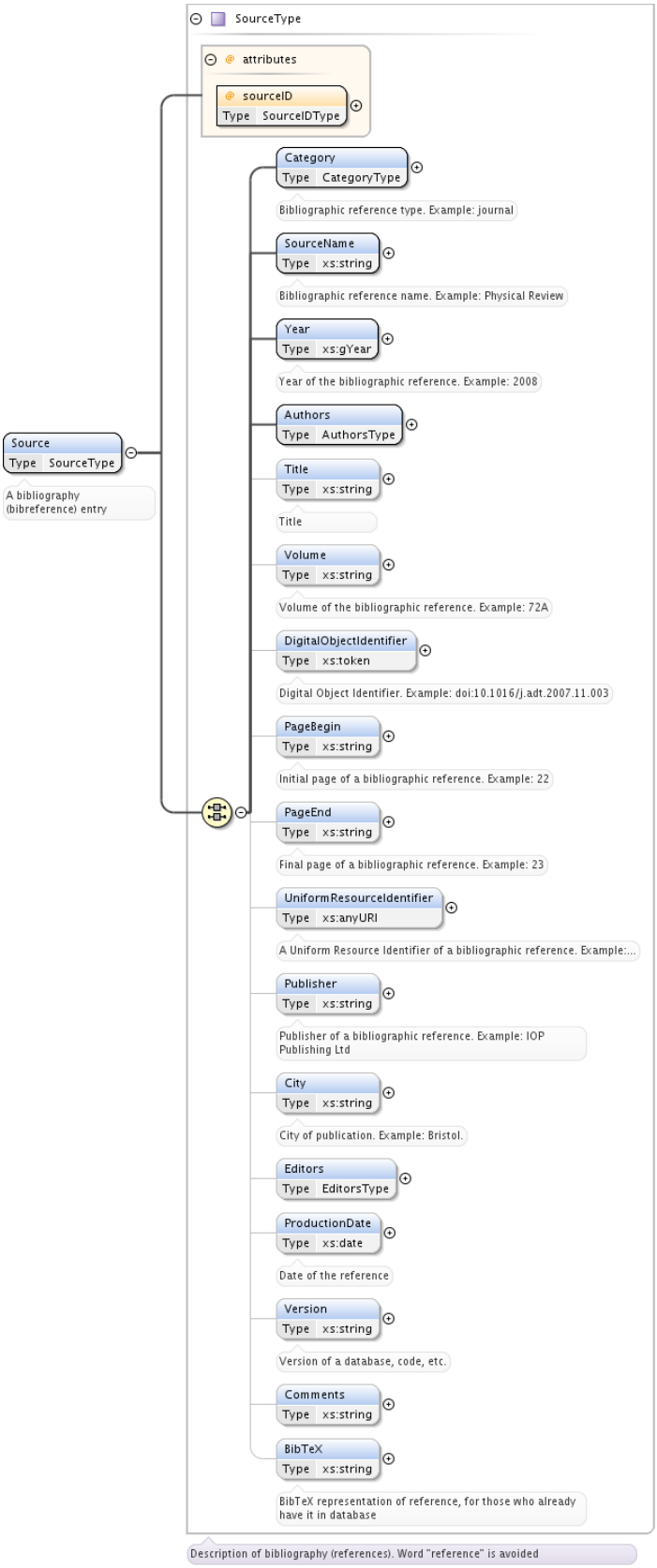
The **Sources** part of XSAMS describes the sources of data, i.e., journal articles, books, proceedings, personal communications, etc. The data sources described here are then referred to in other parts of an XML document by their references (see below).

4.1 Sources



The element **Sources** of type **SourcesType** is a container for one or more **Source** elements each of type **SourceType**.

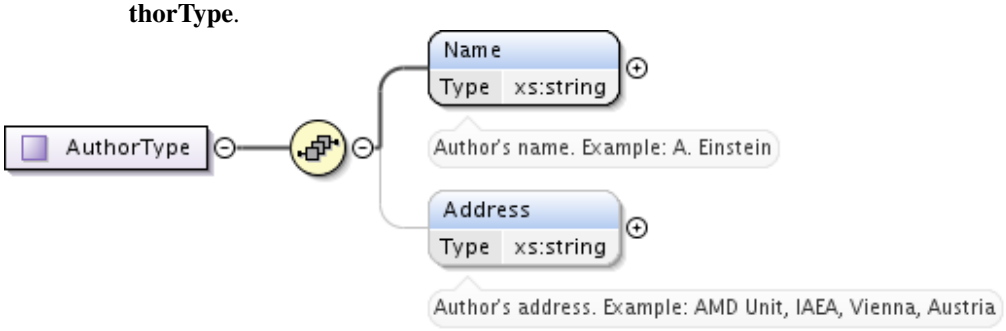
4.2 Source



Source element describes a data source, bibliography item or any other reference relevant to the data element.

It has the following attributes and child elements:

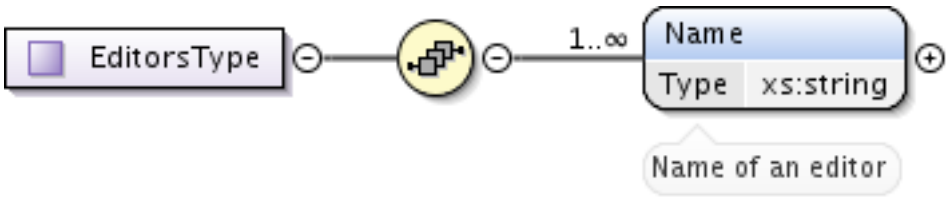
- Mandatory **sourceID** attribute, of type *SourceIDType*. This attribute assigns a unique ID to each data source. Must begin with the upper-case letter **B** followed by one or more characters. The **sourceID** is then referred to with the help of element **SourceRef** of the *PrimaryType*.
- Mandatory **Category** element, defined by **CategoryType**, that may take one of the following values:
  - book
  - database
  - journal
  - preprint
  - private communication
  - proceedings
  - report
  - theses
- Mandatory **SourceName** element. It's value should be a string containing the name of a journal, proceedings, book, etc. Examples: **Physical Review A**, **PhysRevA**, **PRA**, **Proceedings of the 10th Conference on something**.
- Mandatory **Year** element of type **xs:gYear**. Examples: **2009**
- Mandatory **Authors** element, containing unordered list of **Author** elements, each of type **AuthorType**.



AuthorType contains:

  - mandatory **Name** element that should contain author's name, in form of a string.
  - optional **Address** element, that may contain author's e-mail or institutional address.

- Optional **Title** element.
- Optional **Volume** element for journal volume.
- Optional **DigitalObjectIdentifier** [DOI] element.
- Optional **PageBegin** element for initial page of article.
- Optional **PageEnd** element for final page.
- Optional **UniformResourceIdentifier** element of type **xs:anyURI** that may contain reference to on-line copy of article, database access URI, or any other relevant URI.
- Optional **Publisher** element.
- Optional **City** element.
- Optional **Editors** element that may contain a list of **Name** elements.



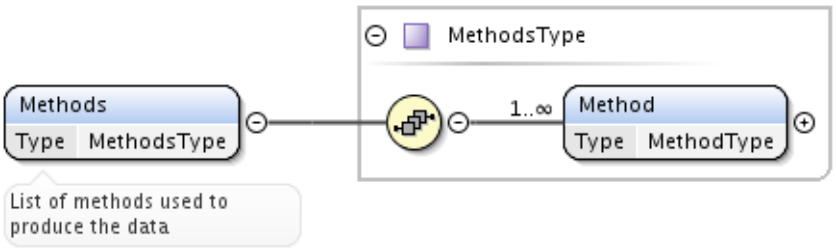
- Optional **ProductionDate** element of type **xs:date**,
- Optional **Version** element
- Optional **Comments** element
- Optional **BibTeX** element for the BibTeX representation of this reference. That element may be useful if you already have data in BibTeX format in your database and wish to export some fields that can't be represented in XSAMS.

CHAPTER

FIVE

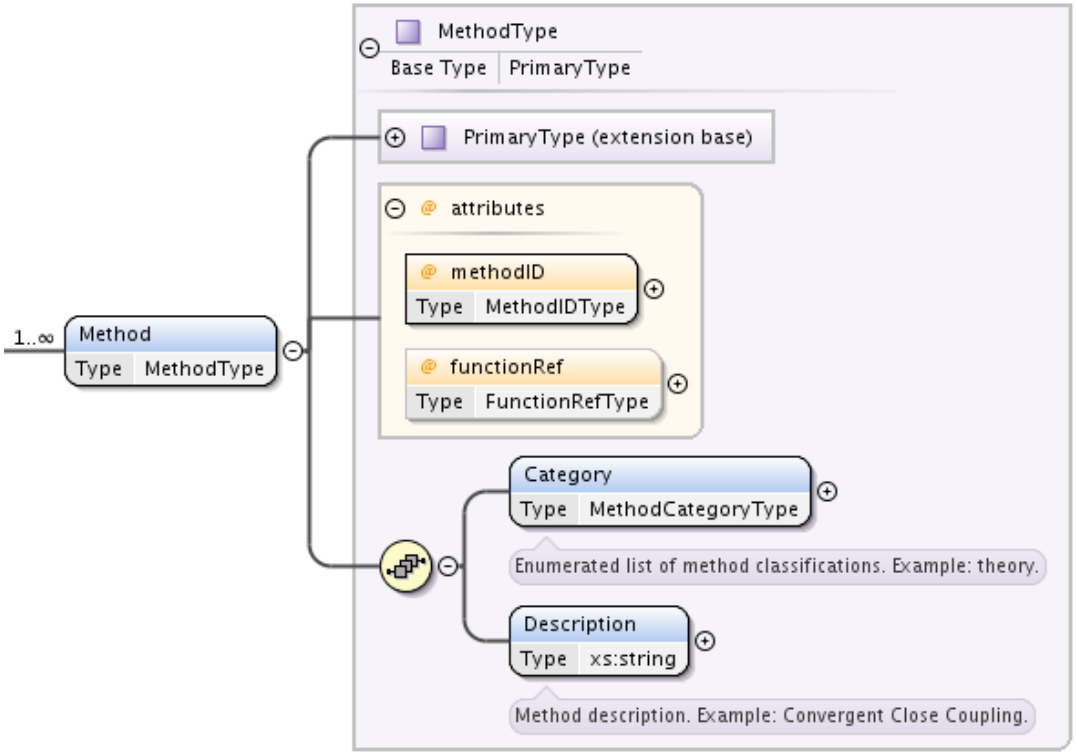
METHODS

The **Methods** branch provides the list of different methods used to produce numerical data. Defined by **MethodsType**, it contains one or more elements **Method**



5.1 Method

Element **Method** provides information on a specific method used to produce numerical data reported in the XML file. Examples of numerical data include energy levels for atoms and molecules, wavelengths for radiative transitions, cross sections for atomic and molecular collisions, etc. Each **Method** element is identified by a **methodID** which is then referred to in the data section.



Extending *PrimaryType*, **Method** has the following additional attributes and elements:

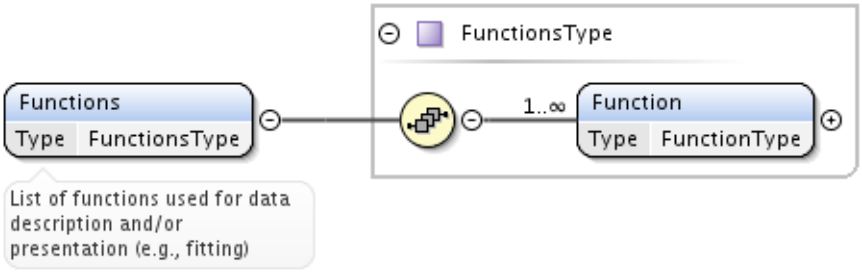
- mandatory attribute **methodID** of type *MethodIDType*, which is the method identification to which any numerical data can refer. This attribute must be composed of a capital letter **M** followed by one or more symbols.
- optional attribute **functionRef** of type *FunctionRefType*, which is a reference to a fit function that was used within this particular method.
- mandatory element **Category**, defined by type **CategoryType** as an item from the following list:
  - experiment
  - theory
  - ritz
  - recommended
  - evaluated
  - empirical
  - scalingLaw
  - semiempirical
  - compilation
  - derived
  - observed
- mandatory string element **Description**, which is a free format text to describe the method.

CHAPTER

SIX

FUNCTIONS

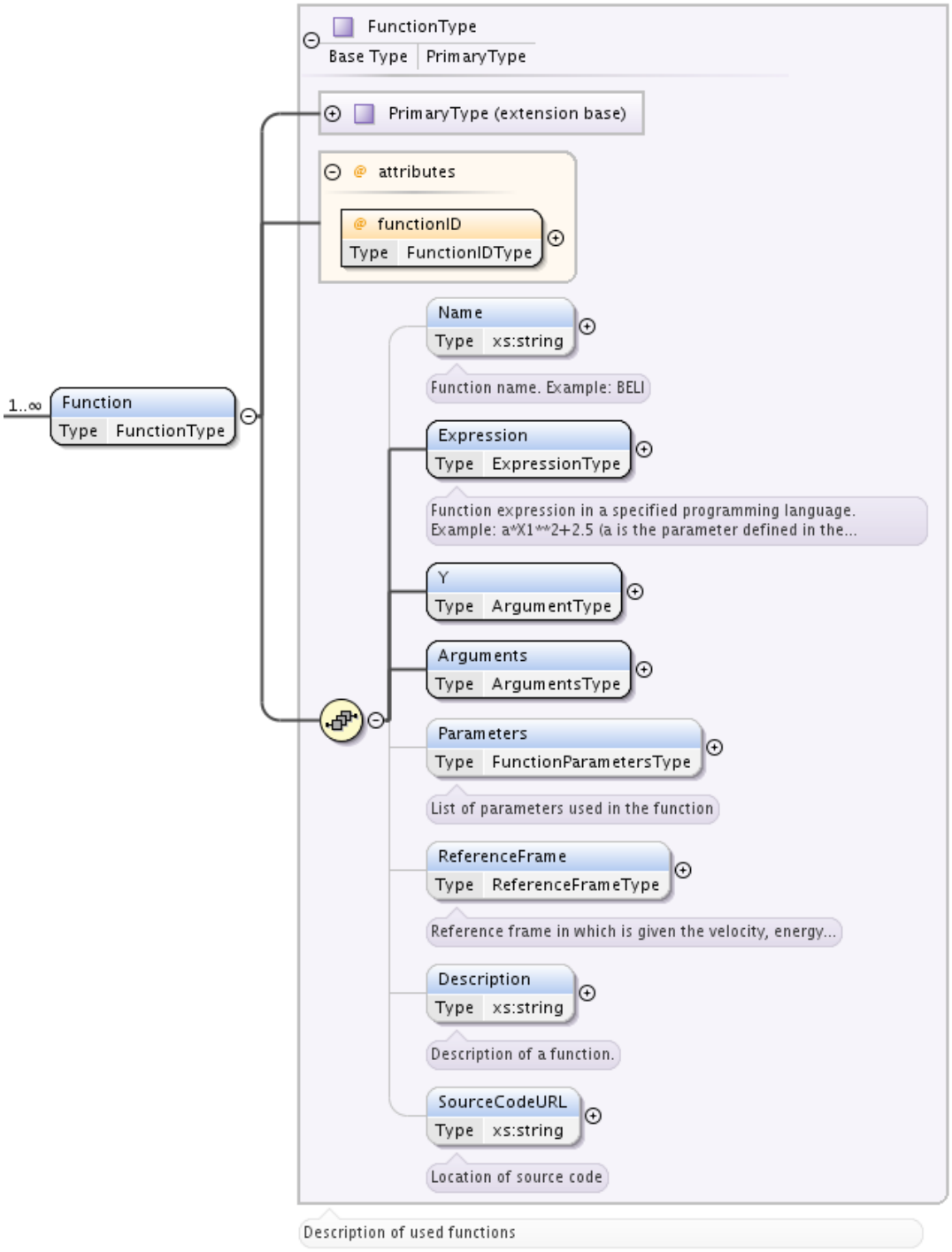
Functions branch of the schema provides the list and description of different functions used for data presentation. These functions can be fit functions to calculate numerical data or functions to describe specific methods. The **Functions** element is defined by type **FunctionsType** and contains one or more **Function** elements.



6.1 Function

Element **Function** of type **FunctionType** provides description of a function in terms of its value **y**, arguments x_1, x_2..., and additional parameters. Each function is identified by a **functionID**, that may be referred from *Methods*, *FitData*'s **FunctionParameters** elements, *Broadening*'s **Lineshape**. **FunctionType** is an extension of *PrimaryType*.

It is important to note that a fit function can be viewed just as a mathematical expression or as the representation of physical process. This second approach has been priviledged in XSAMS. Therefore, physical parameters, such as units or the reference frame for the energies are described in this section. Numerical data for the argument validity limits and the fit parameters are provided as fit data within the *FitParametersType*, wherever it is used in the schema.



Function element, extending *PrimaryType*, has the following elements and attributes:

- mandatory **functionID** attribute of type *FunctionIDType*, that is the function identification to which any numerical data or method can refer. This attribute is mandatory, it may contain any alphanumeric characters and the first character of the identifier should be **F**
- optional string **Name** element, which gives the name of the function;
- mandatory **Expression** element, defined as a string with an additional attribute **computerLanguage**, providing the expression used to calculate the function value;
- mandatory **Y** element of type *ArgumentType*, that describes the value of the function, providing its units and type;
- mandatory **Arguments** element, defining the list of **Argument** elements, each of *ArgumentType*,

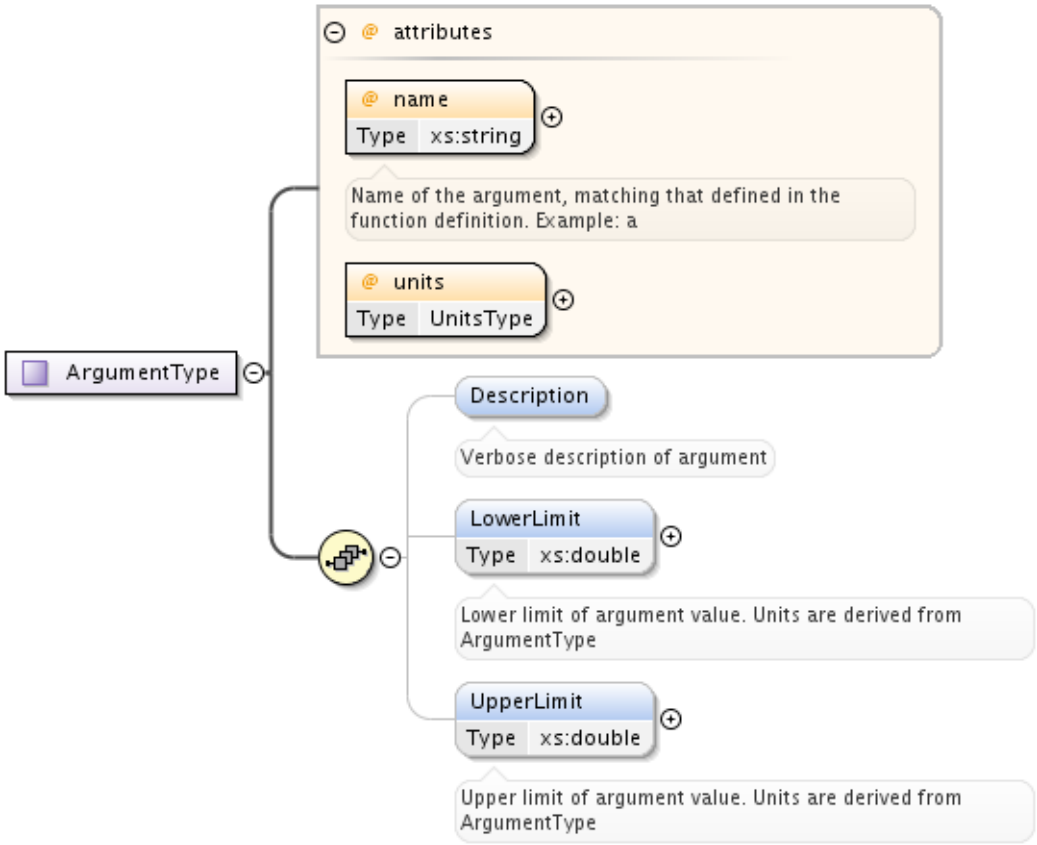
defining the function arguments, their units, types and the default validity limits;

- optional element **Parameters**, defining the list of *Parameter* elements, describing function parameters: their units and names;
- optinal element **ReferenceFrame**, with possible values
  - CenterOfMass
  - LaboratoryFrame
  - TargetFrame

  that is used when function is used in *FitData* within *Processes.Collisions*;
- optional string **Description** element, giving a description of the function;
- optional string **SourceCodeURL**, giving a location from where the function source code can be downloaded.

6.2 ArgumentType

ArgumentType is used in *Function* in **Y** and **Arguments** elements and in *FitParametersType* in **Arguments**.



Following attributes and elements are defined by ArgumentType:

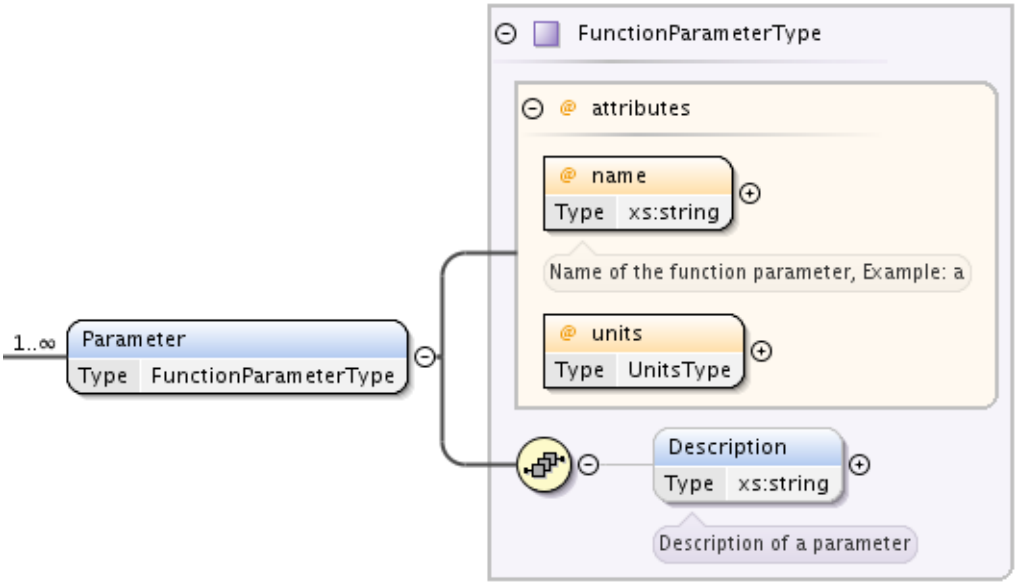
- mandatory string **name** attribute, defining the function argument name. Must be the same as defined in the *Function*'s element **Expression**.
- mandatory **units** attribute, of type *UnitsType*, defining the units of the argument,
- optional string **Description** attribute, providing the free-form description of the argument,
- optional **LowerLimit** element of type **xs:double**,

- optional **UpperLimit** element of type **xs:double**.

When **ArgumentType** is used within *Function*, **LowerLimit** and **UpperLimit** of the argument become the default range values. If required, they may be overridden later by **FitArgument** elements within *FitParametersType*. Attribute **units** should be ignored or checked for consistency when handling **FitArgument** from *FitParametersType*.

6.3 Parameter

The **Parameter** element, defined by **ParameterType**, provides the information about the function parameter.



Listed within **Parameters** element of *Function*, it defines following attributes and elements:

- mandatory string **name** attribute, that should match the one defined in the *Function*'s **Expression**,
- mandatory **units** attribute, of type *UnitsType*,
- optional string **Description** providing the free-form description of the parameter

6.4 Examples

Here is an example function branch instance, defining the function $k(T) = a * t/300^{b} * e^{-g/t}$, with temperature argument **T** and three parameters, **a**, **b**, **g**:

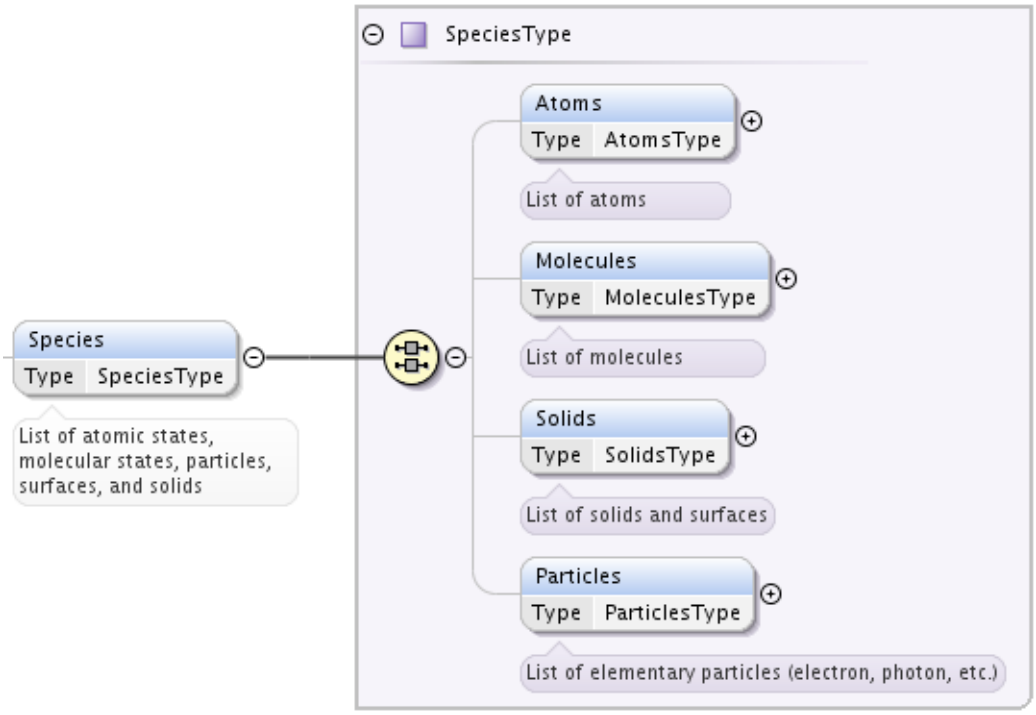
```
<Functions>
        <Function functionID="FKIDAKOOJI">
                <Comments>Kooij function</Comments>
                <SourceRef>BKIDAS1</SourceRef>
                <Name>Kooij</Name>
                <Expression computerLanguage="Fortran">
                        a * (T/300.) ** b * EXP(-g/T)
                </Expression>
                <Y name="k" units="cm3/s">
                        <Description>Rate coefficient vs temperature</Description>
                </Y>
                <Arguments>
                        <Argument name="T" units="K">
```

```
                        <Description>Temperature</Description>
                        <LowerLimit>20</LowerLimit>
                        <UpperLimit>200</UpperLimit>
                </Argument>
        </Arguments>
        <Parameters>
                <Parameter name="a" units="cm3/s">
                        <Description>alpha multiplier</Description>
                </Parameter>
                <Parameter name="b" units="unitless">
                        <Description>beta power</Description>
                </Parameter>
                <Parameter name="g" units="K">
                        <Description>gamma exponent</Description>
                </Parameter>
        </Parameters>
        <Description></Description>
        <SourceCodeURL></SourceCodeURL>
    </Function>
</Functions>
```

CHAPTER

SEVEN

SPECIES



Species branch of XSAMS, having four child elements, is used to represent the static properties of species, used in spectroscopy.

Species.Atoms, *Species.Molecules*, *Species.Particles*, *Species.Solids* elements are defined, each describing the corresponding kind of species. **Atoms** and **Molecules** may be identified by **speciesID** attribute, even if no state information is available, and all four kinds can be referenced by the **stateID** (*StateIDType*) attribute. In this case, for **Atoms** and **Molecules**, at least a dummy ground state needs to be created.

CHAPTER

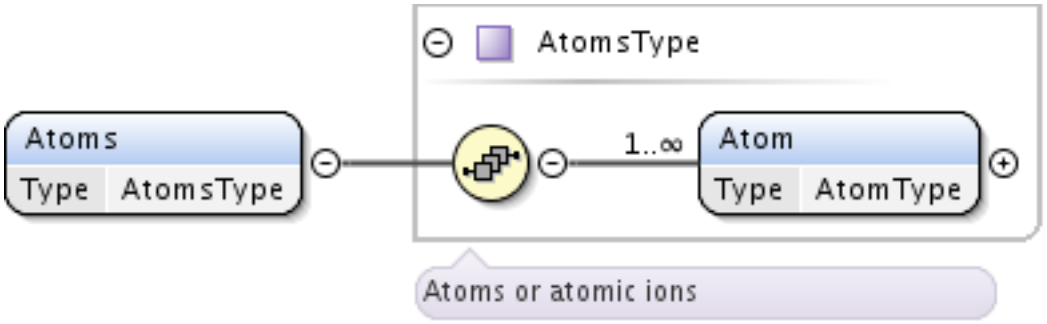
EIGHT

SPECIES.ATOMS

This section describes static properties of atoms and atomic ions such as state energy, total angular momentum, composition of the wave function with mixing coefficients, etc.

8.1 Atoms

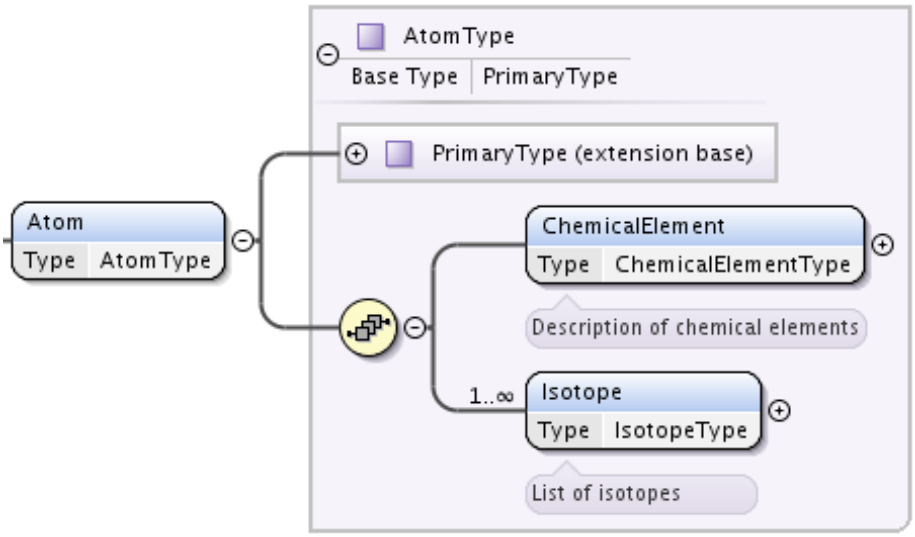
The element **Atoms** of type **AtomsType** is a container for one or more **Atom** elements of type **AtomType**.



8.2 Atom

AtomType is an extension of *PrimaryType*, additionally containing

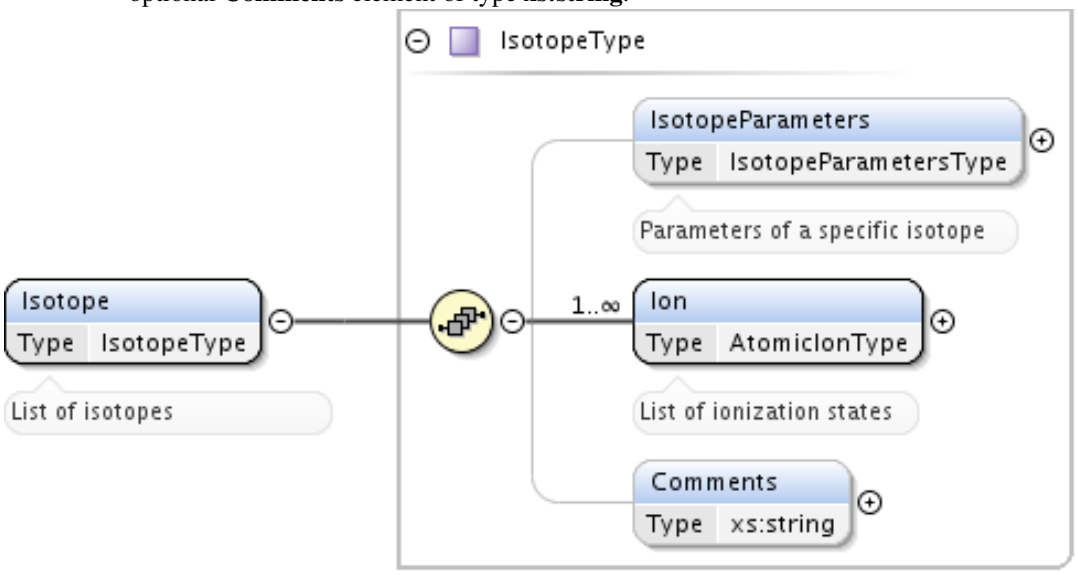
- mandatory element **ChemicalElement** of type *ChemicalElementType*
- one or more **Isotope** elements of type **IsotopeType**.



8.3 Isotope

Isotope element of **IsotopeType** defines a sequence containing

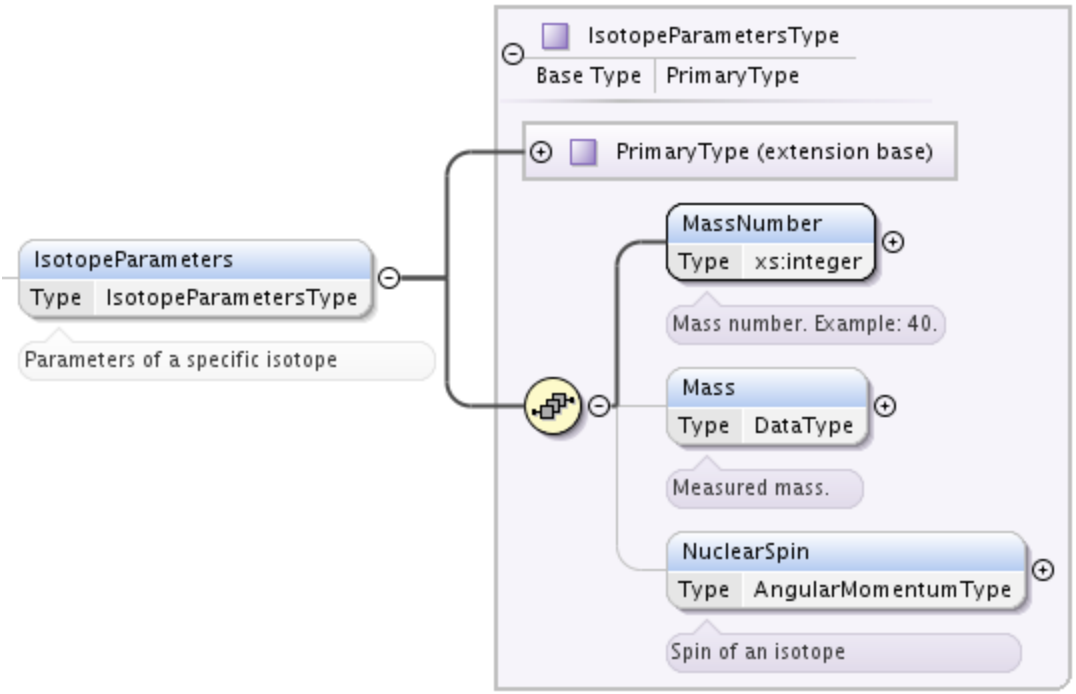
- optional **IsotopeParameters** element of type **IsotopeParametersType**,
- one or more *Ion* elements of type **AtomicIonType**,
- optional **Comments** element of type **xs:string**.



8.4 IsotopeParameters

IsotopeParameters element, having type **IsotopeParametersType**, is an extension of *PrimaryType*, defining additionally:

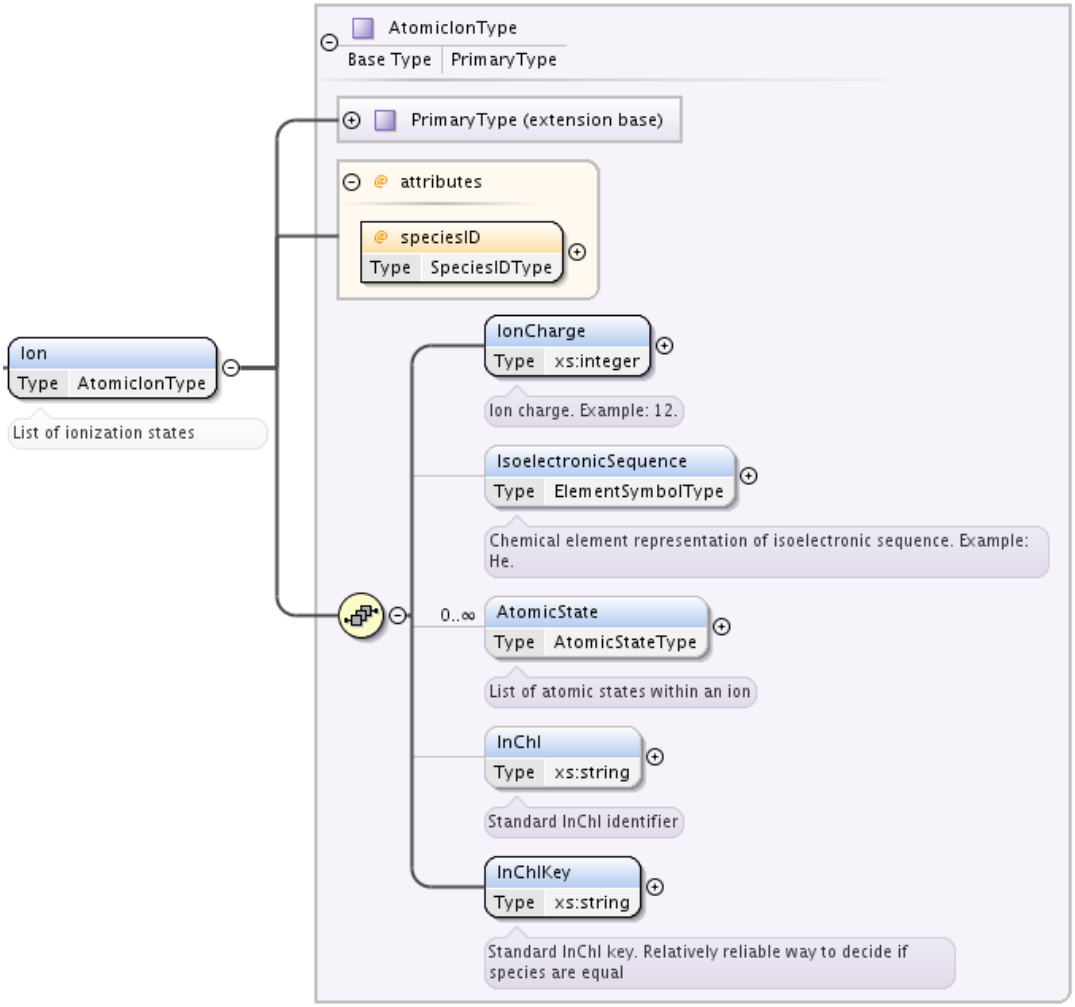
- mandatory integer element **MassNumber** that is the total number of nucleons (protons plus neutrons).
- optional element **Mass** of type *DataType*
- optional element **NuclearSpin** of type *AngularMomentumType*.



8.5 Ion

Ion element, with type **AtomicIonType**, is an extension of *PrimaryType*, defining additionally:

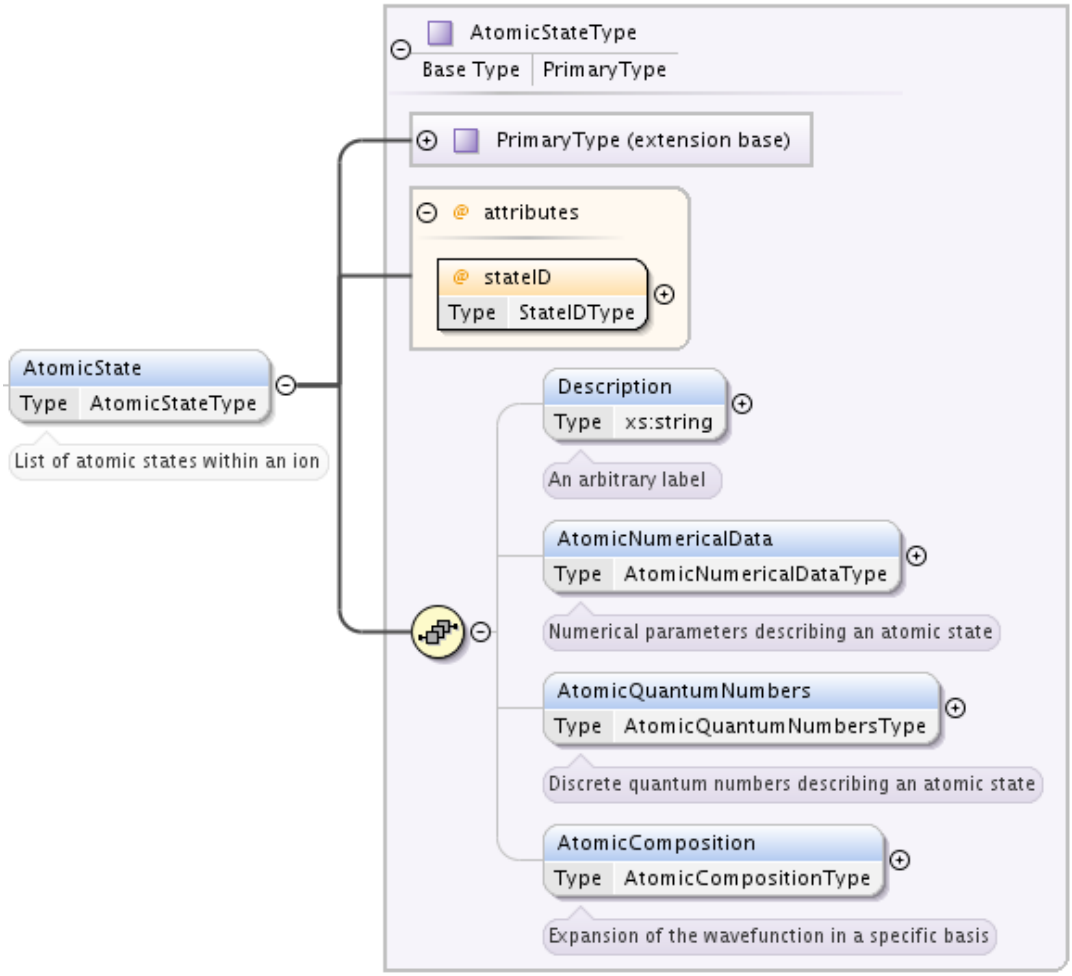
- mandatory **speciesID** attribute for referencing this atom from processes or environments
- mandatory **IonCharge** integer element
- optional **IsoelectronicSequence** element of type *ElementSymbolType*
- none or one or more *AtomicState* elements of type **AtomicStateType**
- optional **InChI** element to keep [INCHI] identifier for atom
- mandatory **InChIKey** element for [INCHIKey] of this ion. InChIKey element is mandatory since it is a fast and reliable way to see if we are talking about the same element, when comparing data originating from different sources.



8.6 AtomicState

AtomicState element of type **AtomicStateType**, again extending *PrimaryType*, holds all information about specific atomic states and defines following attributes and elements:

- mandatory **stateID** attribute of type *StateIDType* that is used to refer to this specific state from processes branch,
- optional string **Description** element that may contain short description or some label of state,
- optional *AtomicNumericalData* element of type **AtomicNumericalDataType** that may contain numerical information about state,
- optional *AtomicQuantumNumbers* element of type **AtomicQuantumNumbersType** that has elements for atomic quantum numbers describing the state,
- optional *AtomicComposition* element of type **AtomicCompositionType** that may contain wavefunction expansion in a specific basis.

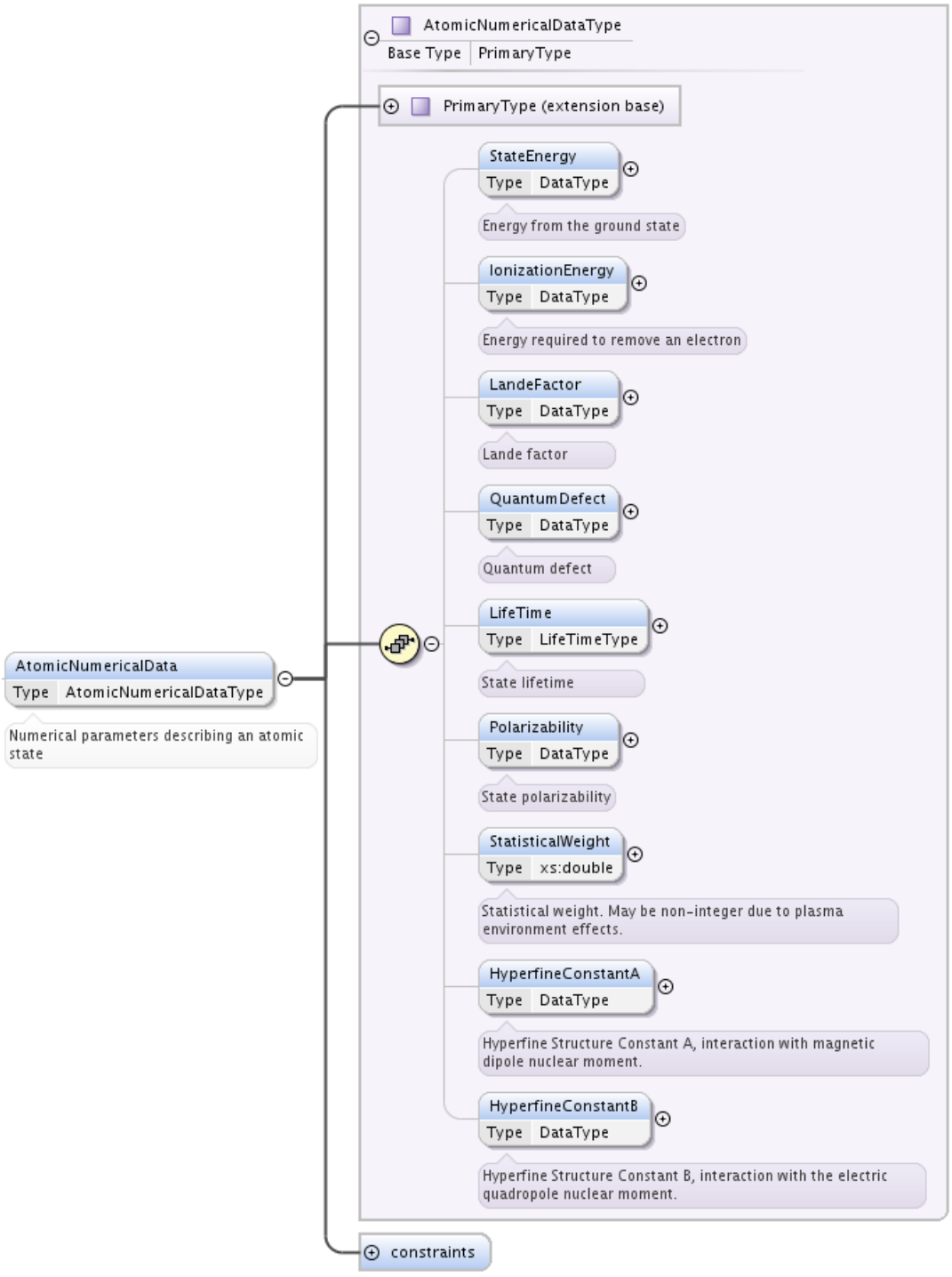


8.7 AtomicNumericalData

AtomicNumericalData element, extension of the *PrimaryType* that may contain the following optional elements:

- **StateEnergy** (above the ion ground state)
- **IonizationEnergy**,
- **LandeFactor**,
- **QuantumDefect**,
- **LifeTime** of *LifeTimeType*,
- **Polarizability**,
- **StatisticalWeight**,
- **HyperfineConstantA**,
- **HyperfineConstantB**.

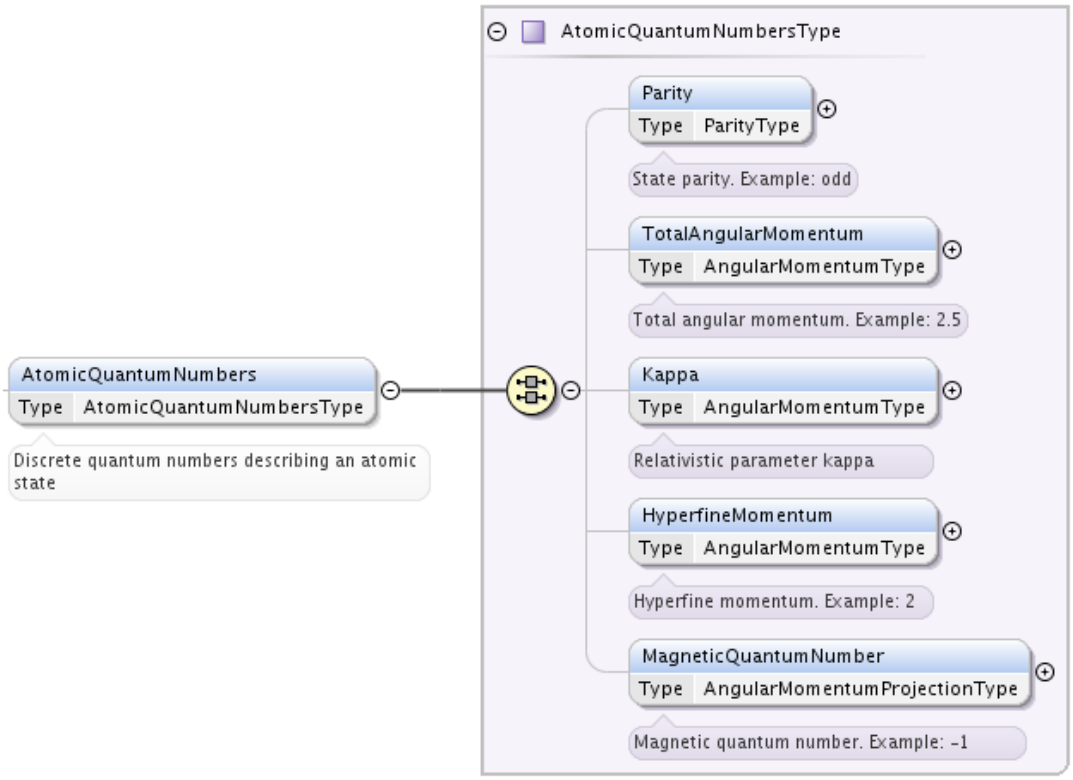
Most of these elements are of type *DataType*. State lifetime has an optional attribute, describing the decay mechanism. The statistical weight of a free atom/ion is an integer, however, in a plasma it can become non-integer and therefore here **StatisticalWeight** is assumed to be a real number.



8.8 AtomicQuantumNumbers

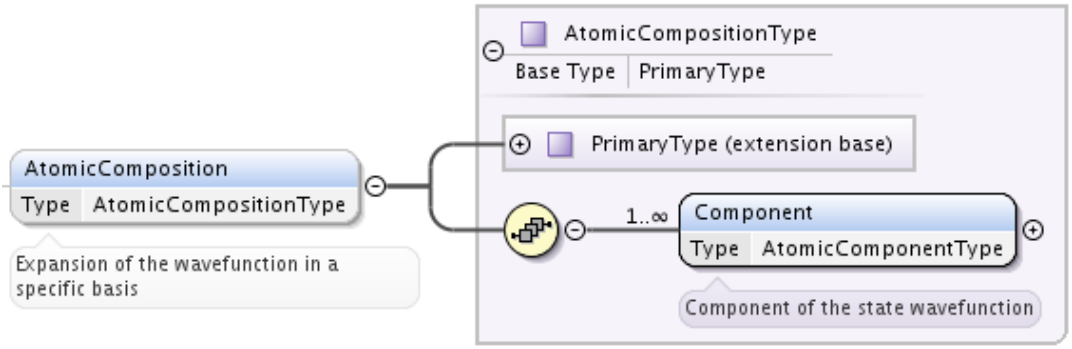
Represents quantum numbers of an atomic state. Following elements are defined:

- **Parity** of type *ParityType*,
- **TotalAngularMomentum**,
- **Kappa** relativistic parameter,
- and **HyperfineMomentum** of *AngularMomentumType*
- **MagneticQuantumNumber** of type *AngularMomentumProjectionType*.



8.9 AtomicComposition

Extension of the *PrimaryType*. Contains components of the atomic wavefunction in a specific basis. Each **Component** element is of type **AtomicComponentType**.



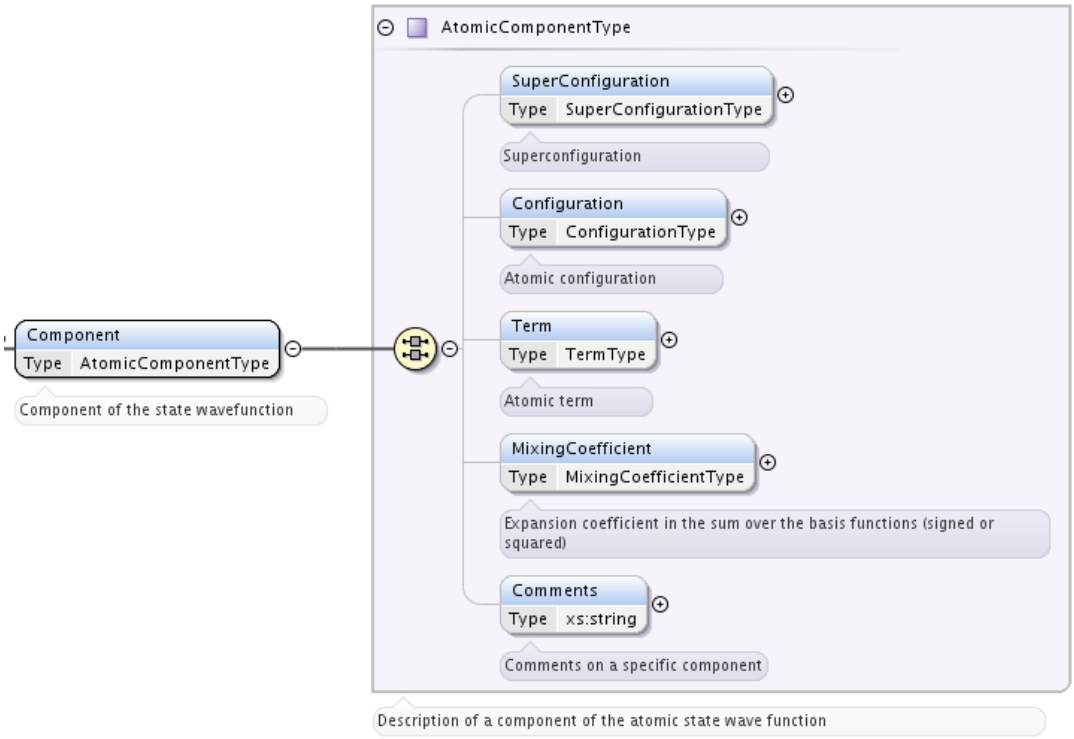
8.10 AtomicComponent

Describes individual components of the atomic wavefunction in a specific basis. May contain elements:

- *SuperConfiguration* of **SuperconfigurationType**,
- *Configuration* of **ConfigurationType**,
- **Term** of type *TermType*,
- **MixingCoefficient** of **MixingCoefficientType** MixingCoefficientType is an extension of XML type **double** with the mandatory attribute **mixingClass**, that indicates the nature of the mixing

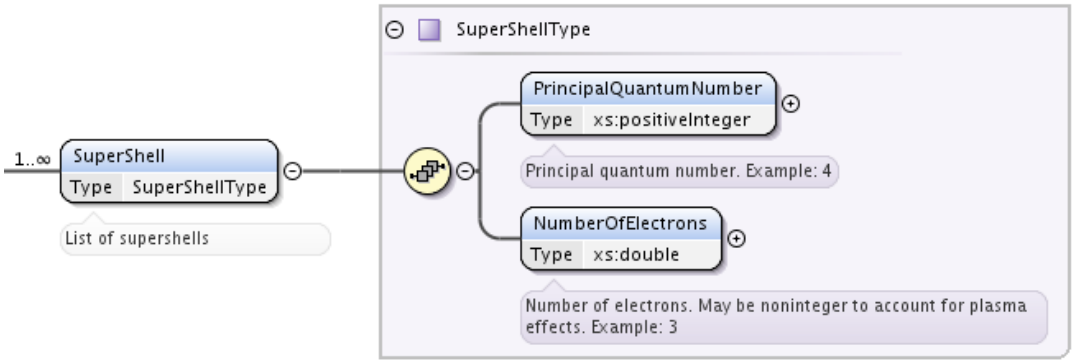
coefficients in the expansion of a wave function in a specific basis. Allowed values of **mixingClass** attribute are: - squared - signed

- **Comments** string element



8.10.1 SuperConfiguration

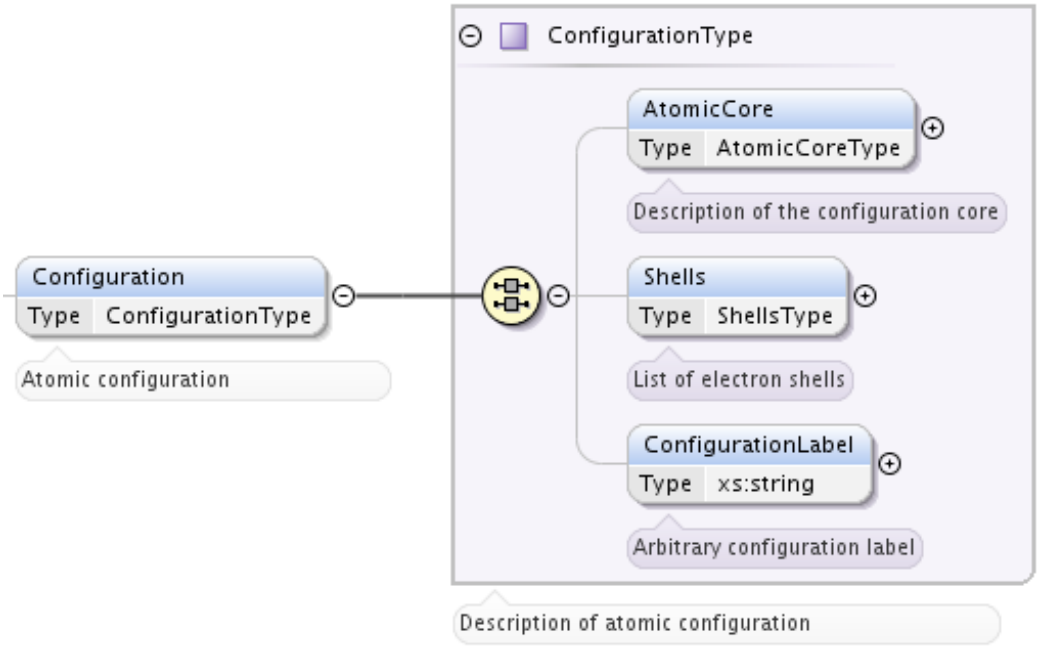
Contains one or more **Supershell** elements (type **SupershellType**), that is, electron distribution over atomic supershells.



Each **SuperShell** element describes how many electrons populate a specific supershell characterized by a positive-integer principal quantum number *n*. The element **NumberOfElectrons** of type **xs:double** can be non-integer to take into account possible plasma effects.

The element **PrincipalQuantumNumber** is of type **xs:positiveinteger**.

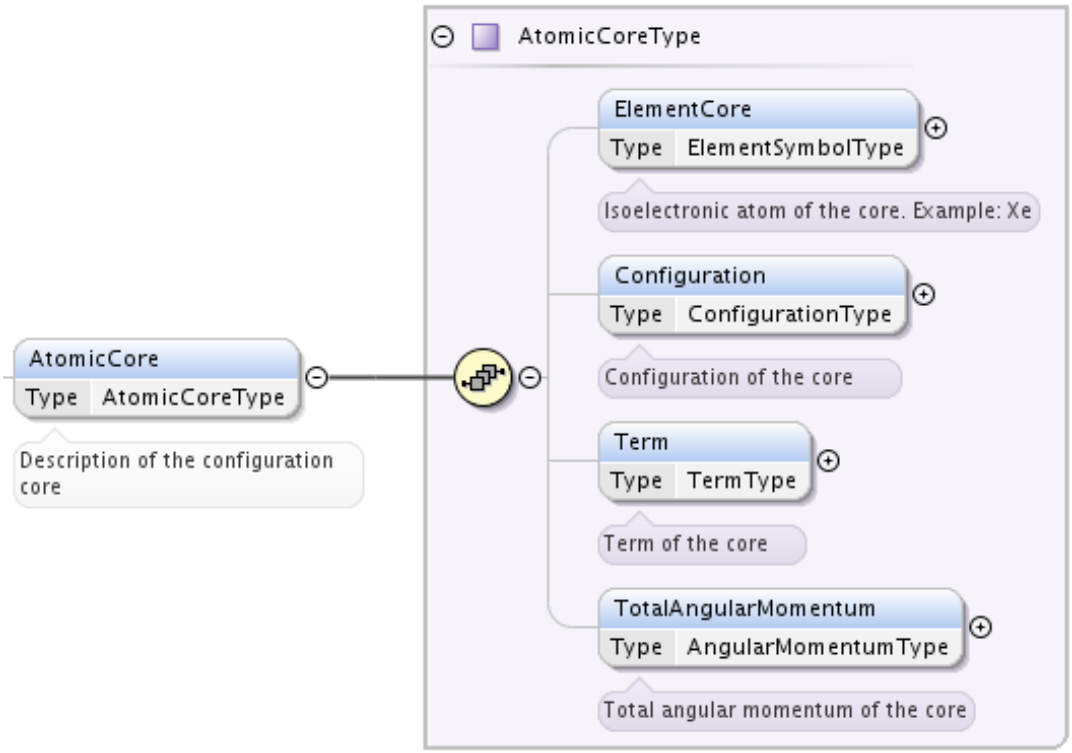
8.10.2 Configuration



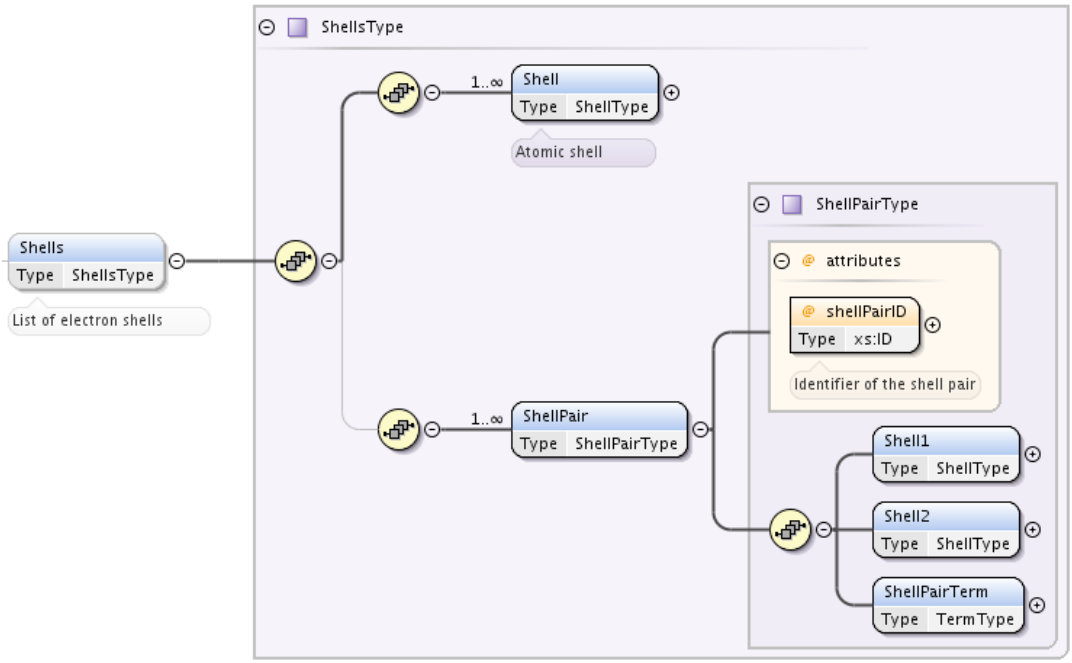
Describes how electrons are distributed over *nl* shells. May contain:

- optional **AtomicCore** element (type **AtomicCoreType**), that is used to compactly represent the atomic core. For instance, one may prefer to use notation **[Ne]3d** to describe the excited configuration $1s^2 2s^2 2p^6 3d$ in a Na-like ion. In this case, it would be sufficient to only indicate the **ElementCore** element, which will be **Ne**.
- optional *Shells* (type **ShellsType**) that contains list of electronic shells or shell pairs
- string **ConfigurationLabel**, that is used to represent configuration in a condensed form, if necessary. For instance, one may prefer to make use of a short configuration label *2s2.2p* instead of providing details of shell populations etc.

AtomicCore



Shells

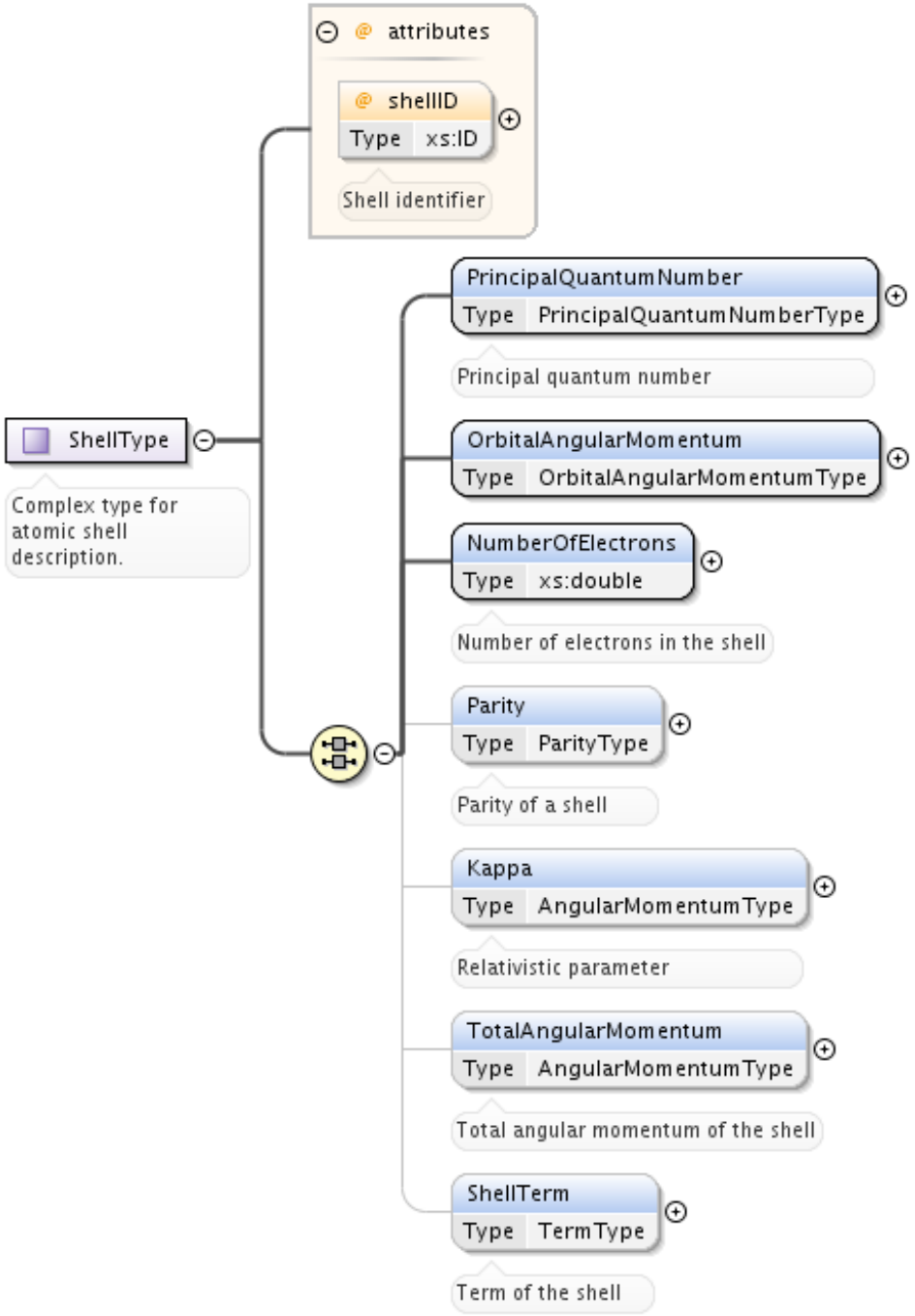


The shells in a configuration can be represented both individually and in terms of shell pairs. **ShellPair** describe a coupled pair of atomic shells. In addition to descriptors for each of the shells, it contains the mandatory attribute **shellPairID** to be referenced in a document.

8.11 Specific XML Types

Here, XML types, specific to Species.Atoms branch of XSAMS are described

8.11.1 ShellType



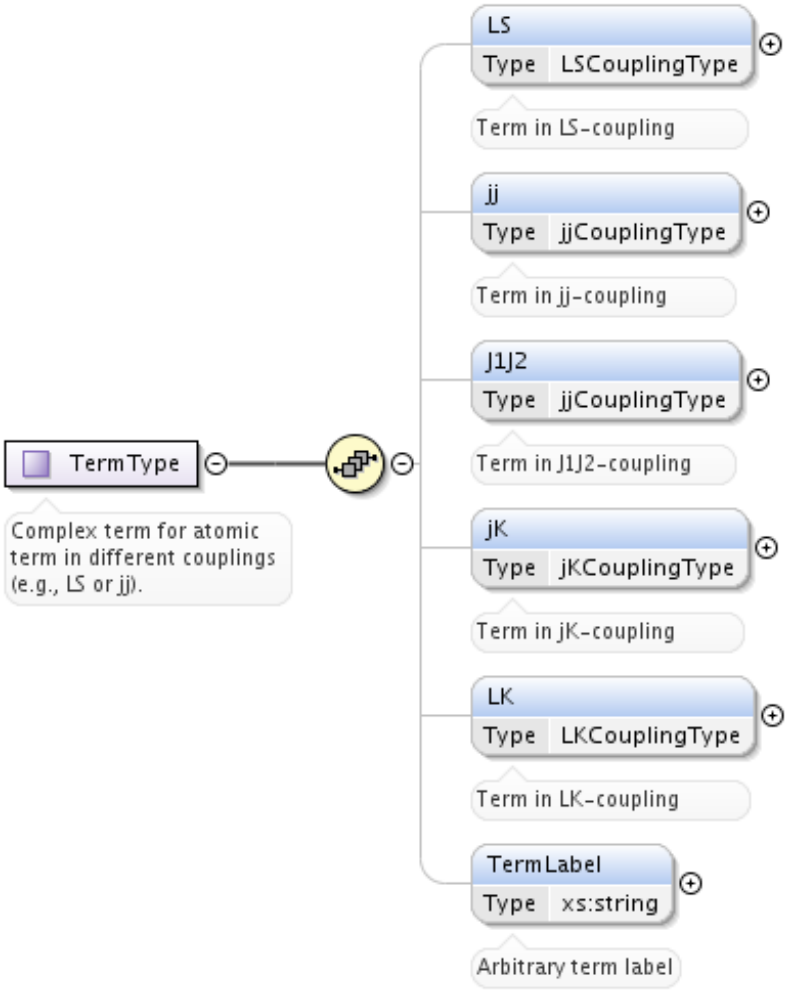
As an atomic shell is typically represented as nl^w , all three parameters:

- **PrincipalQuantumNumber** n,
- **OrbitalAngularMomentum** l, and
- **NumberOfElectrons** w are mandatory. **NumberOfElectrons** may be a non-integer decimal number in order to reflect plasma effects.

Other optional parameters include * **Parity** element (type **ParityType**), * **TotalAngularMomentum** element (type **AngularMomentumType**) * κ relativistic parameter (element **Kappa** of type **AngularMomentumType**), * **ShellTerm** element (type *TermType*) * optional attribute **shellID** (type **xs:ID**).

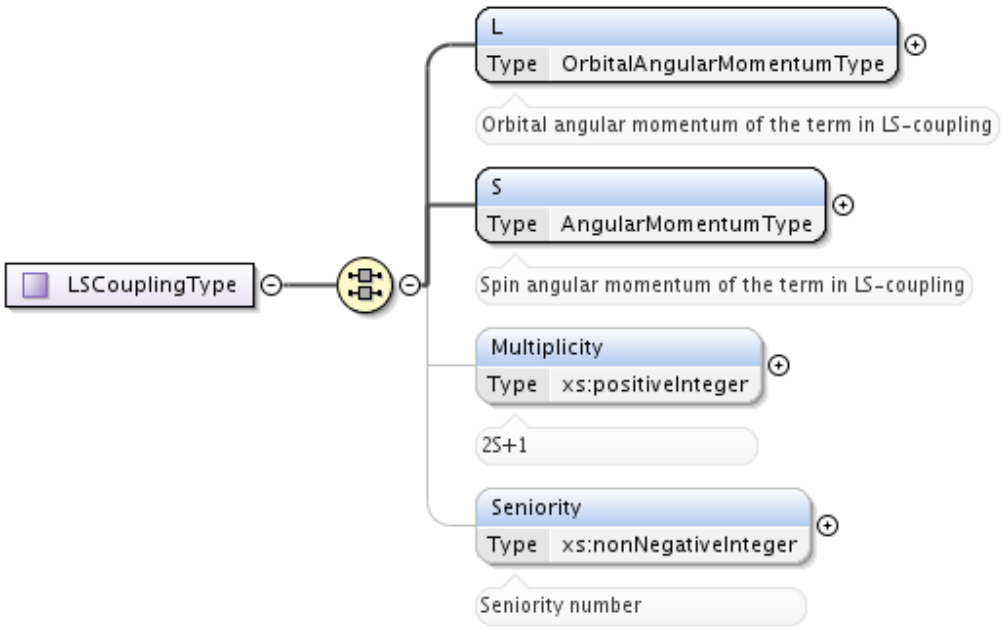
8.11.2 TermType

Currently an atomic term can be represented in five coupling schemes, i.e., LS, jj, J_1J_2, jK, and LK. For a detailed description of these and other schemes see, e.g., *Atomic Spectroscopy* at http://physics.nist.gov/Pubs/AtSpec/index.html

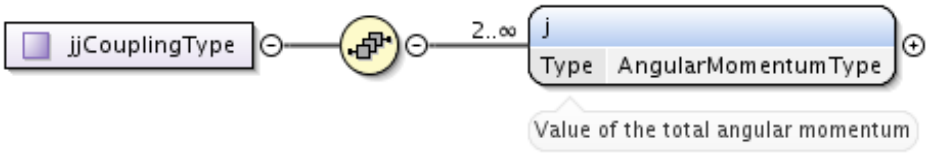


For each coupling, a separate XML type is defined:

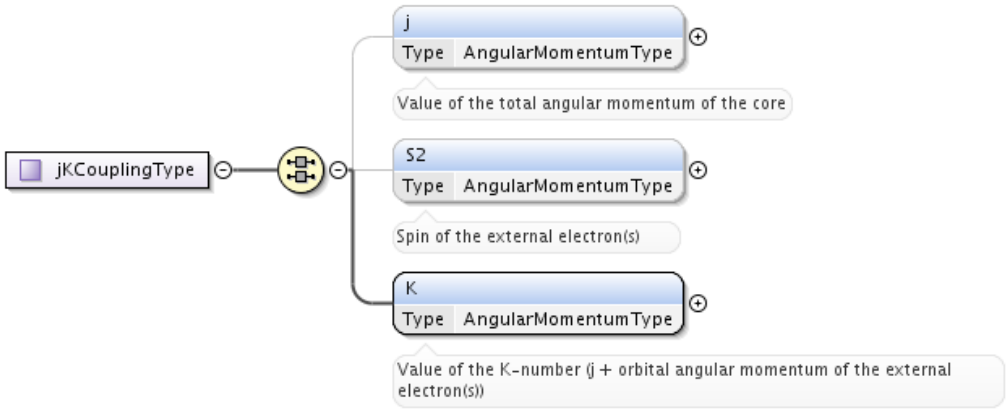
- **LSCouplingType** Describes LS-coupling in an atomic term in terms of total angular orbital momentum *L* (type *OrbitalAngularMomentumType*) and total spin *S* (type *AngularMomentumType*). Optional element **Multiplicity** is defined as $2S + 1$ and therefore always is a positive integer. Optional element **Seniority** is a non-negative integer with an optional attribute **soureceRef**.



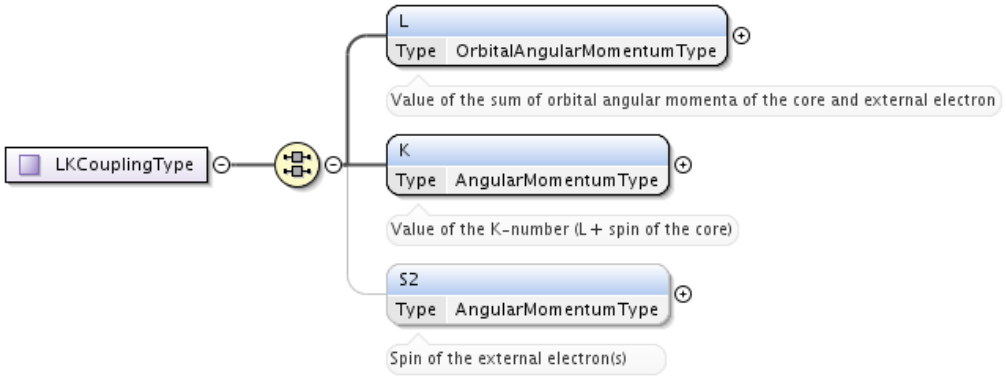
- **jjCouplingType** Describes jj-coupling in an atomic term as two or more **j**-values (each of *AngularMomentumType*).



- **J1J2CouplingType** Describes J_1J_2-coupling in an atomic term as two or more **j**-values (each of **AngularMomentumType**). Similar to the jj-coupling.
- **jKCouplingType** Describes *jK*-coupling in an atomic term typical in, e.g., noble-gas-like ions. The **K** quantum number is mandatory, while the core momentum **j** and the spin of outer electron(s) **S2** are optional; each of them is of *AngularMomentumType*.



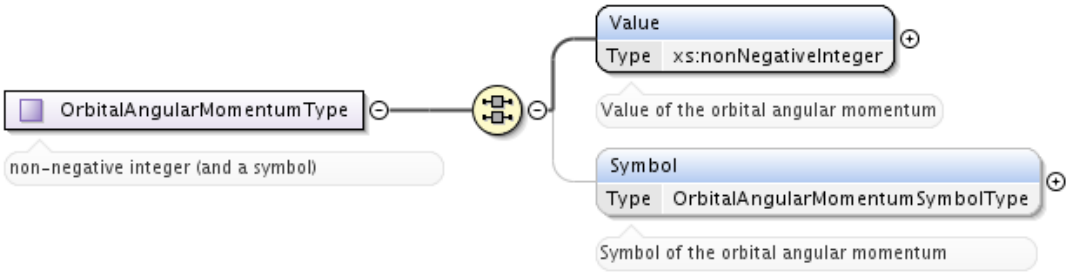
- **LKCouplingType** Describes *LK* - coupling (or LS_1) in an atomic term in terms of atomic core angular momentum **L** and its sum with the total spin of the core: $\overrightarrow{K} = \overrightarrow{L} + \overrightarrow{S_1}$. The spin of outer electron(s) **S2** is an optional element of type *AngularMomentumType*.



- **TermLabel** This string element is used to represent an atomic term in a condensed form, if necessary. For instance, one may prefer to make use of a term label **3P** instead of separately indicating the term **S** and **L** values.

8.11.3 OrbitalAngularMomentumType

Represents an orbital angular momentum as a mandatory non-negative integer **Value** and an optional lower-case letter **Symbol** (example: **p**).



8.12 Example XML instances

Example instances of Species.Atoms branch subtrees:

8.12.1 Vald Fe example

Example from VALD database for Fe, with two states present:

```
<Atom>
  <ChemicalElement>
    <NuclearCharge>59</NuclearCharge>
    <ElementSymbol>Pr</ElementSymbol>
  </ChemicalElement>
  <Isotope>
    <IsotopeParameters>
      <MassNumber>141</MassNumber>
    </IsotopeParameters>
    <Ion speciesID="Xvald-1714">
      <IonCharge>2</IonCharge>
      <AtomicState stateID="Svald-1086">
        <AtomicNumericalData>
          <StateEnergy>
            <SourceRef>Bvald-ISAN</SourceRef>
            <SourceRef>Bvald-RRKB</SourceRef>
            <SourceRef>Bvald-WTCR</SourceRef>
```

```
          <Value units="1/cm">18693.6490</Value>
        </StateEnergy>
        <LandeFactor>
          <SourceRef>Bvald-ISAN</SourceRef>
          <SourceRef>Bvald-RRKB</SourceRef>
          <SourceRef>Bvald-WTCR</SourceRef>
          <Value units="unitless">0.63</Value>
        </LandeFactor>
      </AtomicNumericalData>
      <AtomicQuantumNumbers/>
      <AtomicComposition>
        <Component>
          <Configuration/>
        </Component>
      </AtomicComposition>
    </AtomicState>
    <AtomicState stateID="Svald-1115">
      <AtomicNumericalData>
        <StateEnergy>
          <SourceRef>Bvald-ISAN</SourceRef>
          <SourceRef>Bvald-RRKB</SourceRef>
          <SourceRef>Bvald-WTCR</SourceRef>
          <Value units="1/cm">52026.9010</Value>
        </StateEnergy>
        <LandeFactor>
          <SourceRef>Bvald-ISAN</SourceRef>
          <SourceRef>Bvald-RRKB</SourceRef>
          <SourceRef>Bvald-WTCR</SourceRef>
          <Value units="unitless">1.32</Value>
        </LandeFactor>
      </AtomicNumericalData>
      <AtomicQuantumNumbers/>
      <AtomicComposition>
        <Component>
          <Configuration/>
        </Component>
      </AtomicComposition>
    </AtomicState>
    <InChI>InChI=1S/Pr/q+2/i1+0</InChI>
    <InChIKey>BXZAMIARVDIHCY-IGMARMGPSA-N</InChIKey>
  </Ion>
 </Isotope>
</Atom>
```

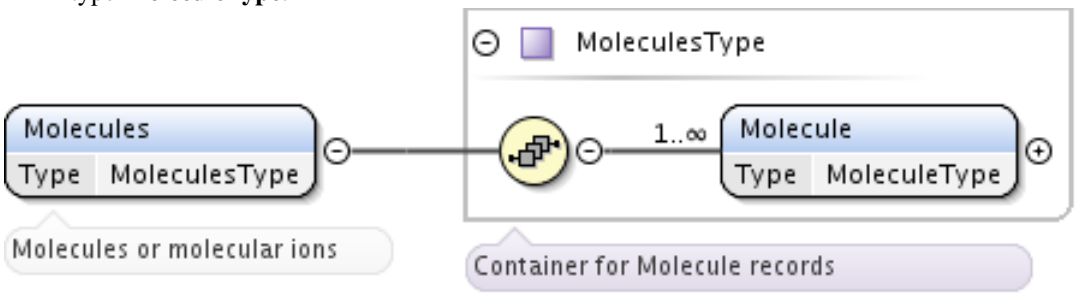
CHAPTER

NINE

SPECIES.MOLECULES

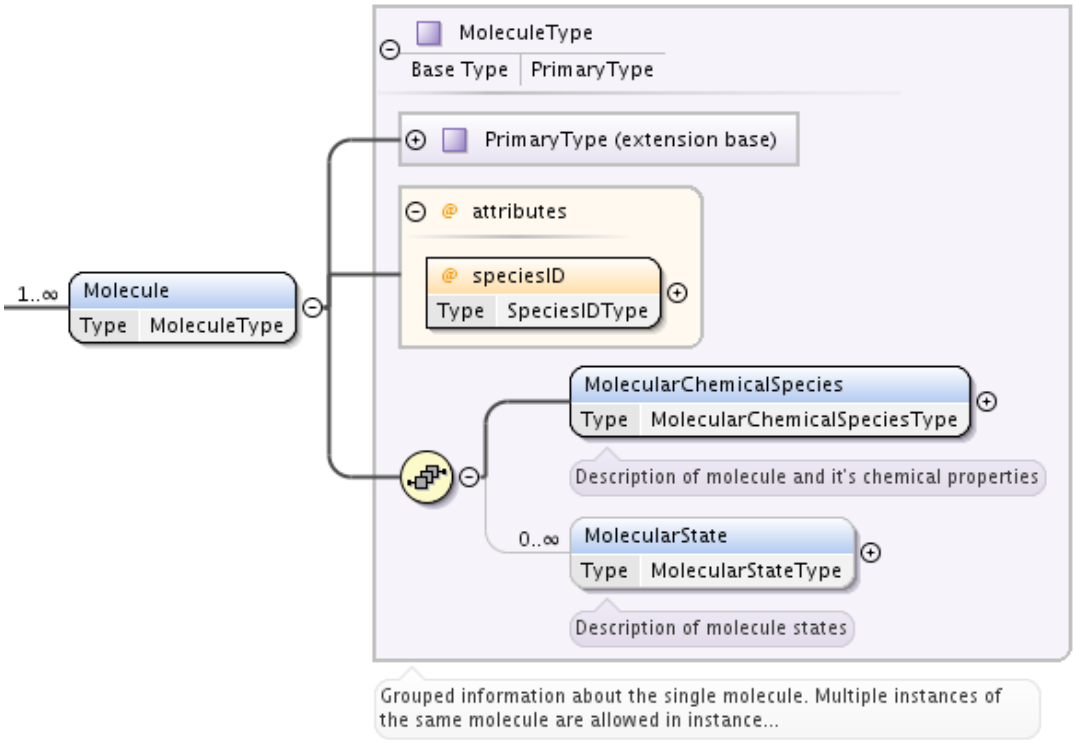
9.1 Molecules

The element **Molecules** of type **MoleculesType** is a container for one or more **Molecule** elements of type **MoleculeType**.



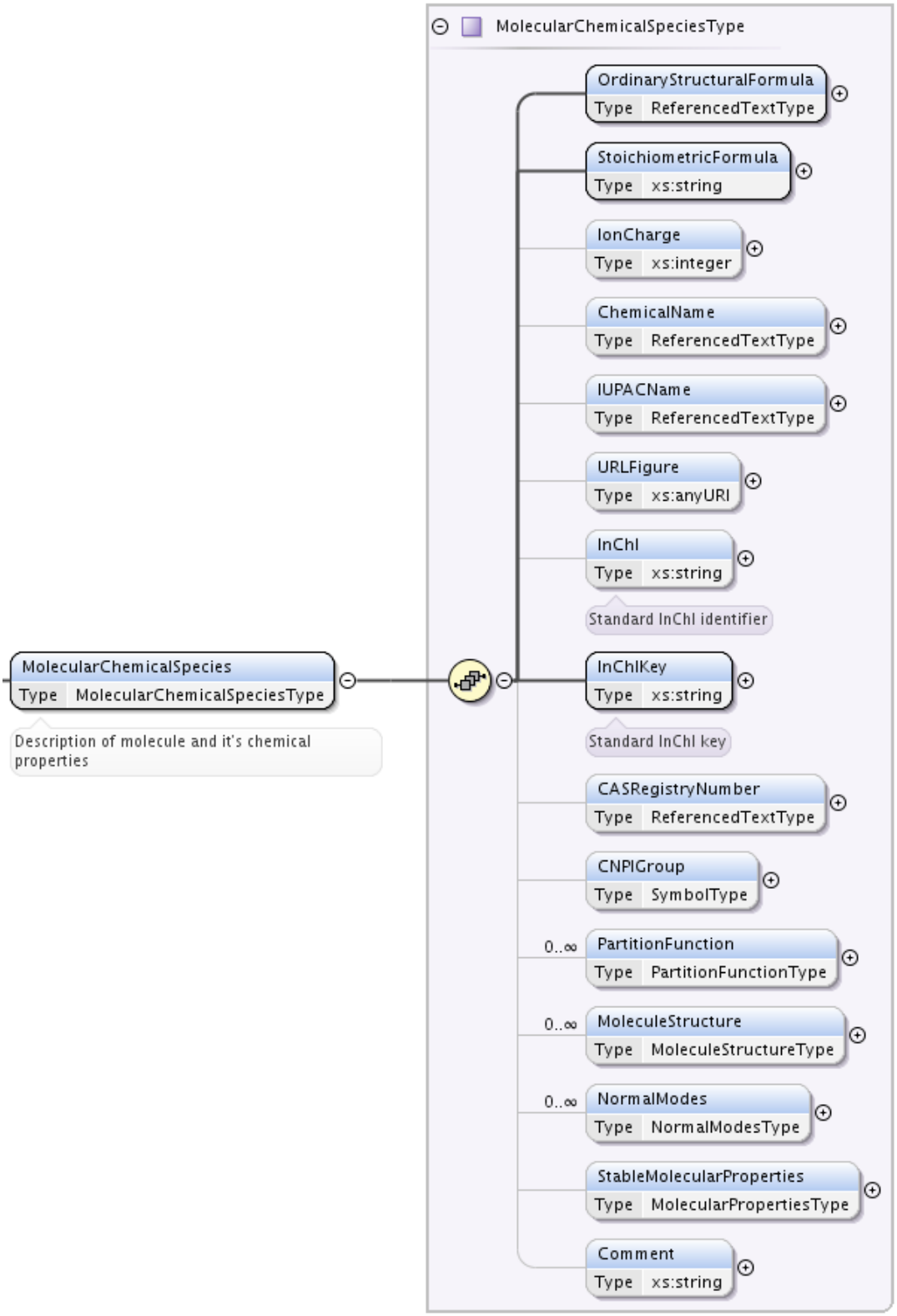
9.2 Molecule

This element is defined as **MoleculeType**, which is an extension of *PrimaryType*. It provides all information on molecular chemical species through **MolecularChemicalSpecies** of type **MolecularChemicalSpeciesType** and one or more elements **MolecularState** of type **MolecularStateType**. These molecular species may be involved in various processes that are described in the **Processes** section. Referencing is done either by **speciesID** or **stateID** attributes, or both.



9.3 MolecularChemicalSpecies

The element **MolecularChemicalSpecies** describes a simple model to identify the chemical molecule involved in the description of **States** and **Processes**.



NOTE: Recommendations on Organic and Biochemical Nomenclature, Symbols and Terminology are available on the International Union of Pure and Applied Chemistry [IUPAC] webpage.

MolecularChemicalSpecies element has following child elements defined:

- **OrdinaryStructuralformula** mandatory element, of type *ReferencedTextType*. Standard formula describing the chemical complex written in Latex (molecule or molecular ion). For the time being, the **ordinaryStructuralformula** element can not be used for search, because a worldwide consensus among molecular physicists is not reached yet. Here are some guidelines

on how to fill this element, based on kind of molecule:

- For simple molecules involving several atoms, the formula should reflect the order of the chemical bonds involved.

  Isotopic atomic species should be described by the usual chemical element name (see *List of Atomic Elements*) with the *atomicMassNumber* as an upper left subscript. For example: Carbon 13 would be ^{13}C (^{13}C).

  For the special case of Hydrogen, isotopic atomic species have specific symbols and names: Symbol= *D* and Name= *Deuterium* for core with 1 neutron and 1 proton, Symbol= *T* and Name= *Tritium* for core with 2 neutrons and 1 proton.

- When one or several radicals, such as methyl, CH_3, are involved, they should be indicated in bracket followed by the number of their occurence as a lower subscript. They are placed in the formula following the order of the chemical bonds. Sometimes the formula of the radical is replaced by an alias such as *(Me)* for methyl or *(Et)* for ethyl.

- The conformation (cis, trans, etc ...) should be indicated before the part of the formula giving the various atomic components.

- For more complex molecules, no general rules are provided for now in the document and any string describing the molecule can be used.

- Molecular ions should be described by their chemical formula followed by a plus/minus sign and the number of charges (when different from one). For example, the dyazenylium molecular ion would be N_2H^+ (N_2H^+) and the carbonyl doubly charged ion CO^{2+} (CO^{2+}).

- **StoichiometricFormula** mandatory string element. For molecules it is constituted by an alphabetical suite of the atomic constituents followed by the total number of their occurence (purely ASCII). For example: CH2O2 (CH_2O_2) corresponding to formic acid whose formula is $t/c-HCOOH$ ($t/c - HCOOH$). This is useful for a primary search of resources.

- **IonCharge** optional integer element. It gives the charge of the molecular ion. Examples: **+1** or **-1**

- **ChemicalName** element of type *ReferencedTextType*, a string (with reference) giving the name of the chemical complex. The **ChemicalName** element can not be used for search, but rather for information because different names might be associated to a single chemical complex.

- **IUPACName** element, of type *ReferencedTextType*, a string (with reference) giving the IUPAC name. It can be used for search.

- **URLFigure** optional element, that provides an URL to a figure showing the molecule in its stable configuration.

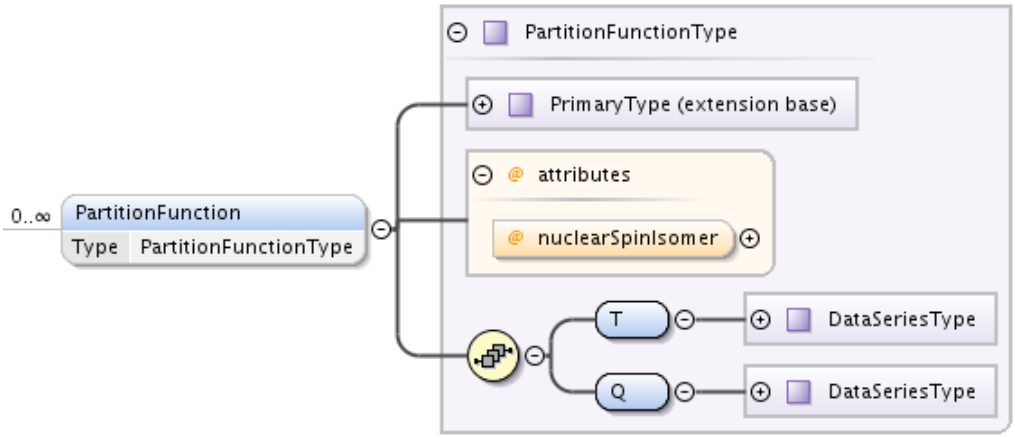
- **InChI** optional string element. Provides the InChI identifier. [InChI]

- **InChIKey** *mandatory* string element. Provides the InChIKey hash of InChI identifier. This element is mandatory because InChIKeys appeared to be the most convinient and relatively reliable way to identify if two Molecule blocks of data, originating from different databases, are describing the same species or not.

- **CASRegistryNumber** optional element of type *ReferencedTextType* that provides the CAS Registry Number.

- **CNPIGroup** element of type *SymbolType:*, that describes the Complete Nuclear Permutation Inversion Group.

- **PartitionFunction** element that may contain a temperature dependence of molecule partition function in a form of a list of points. May be specified multiple times.

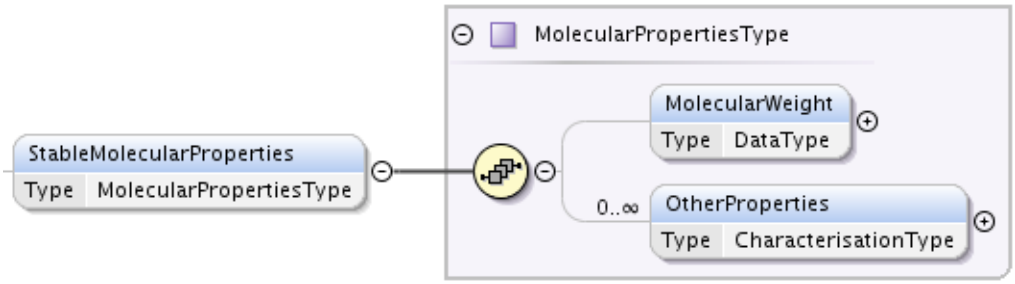


Extending *PrimaryType*, it has:

- **nuclearSpinIsomer** optional attribute to define nuclear spin isomer, the same way as in *MolecularStateCharacterisation*.
- **T** element of type *DataSeriesType* to define temperature points
- **Q** element of type *DataSeriesType* to define partition function values

- **MoleculeStructure** optional element that is a link to [CML] description of molecular structure. Extending *PrimaryType*, it defines additionally
  - optional **electronicStateRef** attribute of type *StateRefType* that may give a link to a separate description of electronic state,
  - mandatory **atomArray** element from [CML], that must occur at least once,
  - optional **bondArray** element from [CML], that may occur multiple times.

  This description should be sufficient to define the structure of any complex molecule.
- *NormalModes* element for description of vibrational normal modes of complex molecules.
- **StableMolecularProperties** element if type **MolecularPropertiesType** which provides information on properties of the molecule.



MolecularPropertiesType has two elements:

- **MolecularWeight**, of type *DataType*, that carries the sum of the individual isotopic masses of all the atoms in a molecule,
- multiple **OtherProperties** element of type *CharacterisationType* that allows to specify arbitrary properties of molecule in a form of a named dataset.

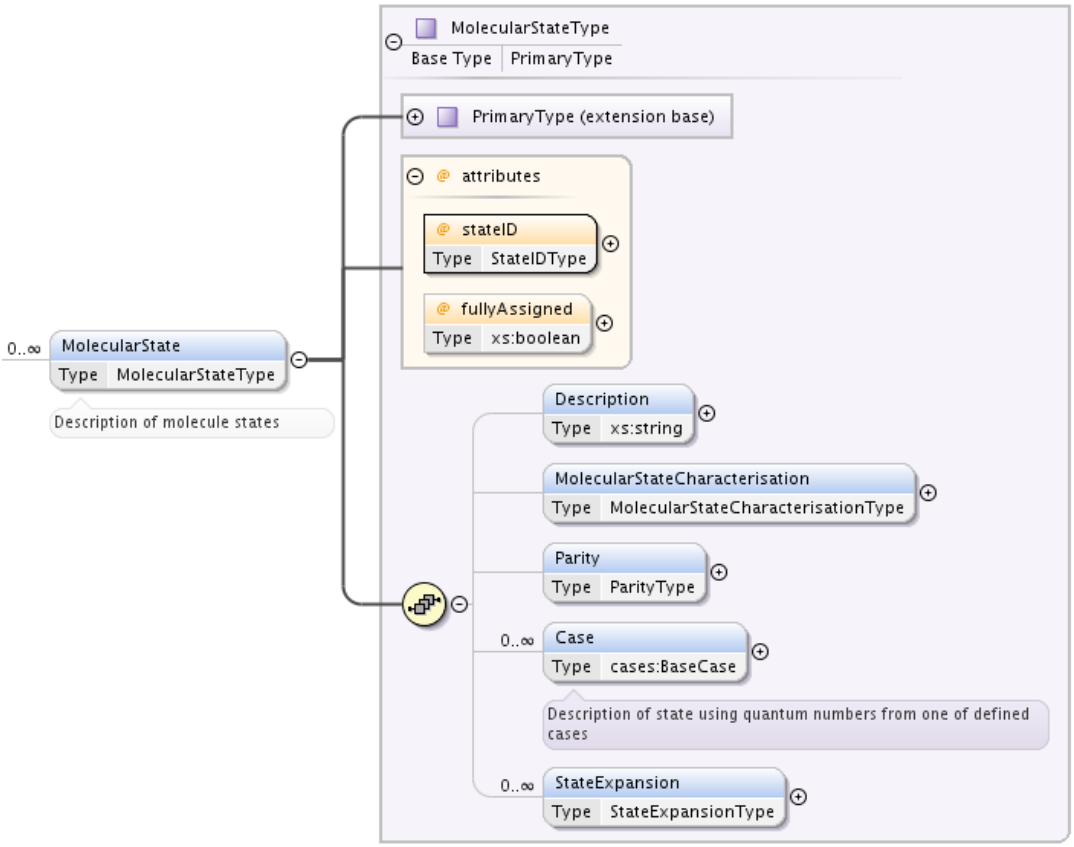
- **Comment** string element for arbitrary comments.

9.4 MolecularState

Formally a **MolecularState** element of type **MolecularStateType** is characterized by a single eigenvalue (possibly degenerate) and a single eigenstate (when non degenerate eigenvalue) of the hamilto-

nian describing the energy structure of the chemical compound. When the eigenvalue is degenerate, the quantum numbers associated to the degeneracy are not provided.

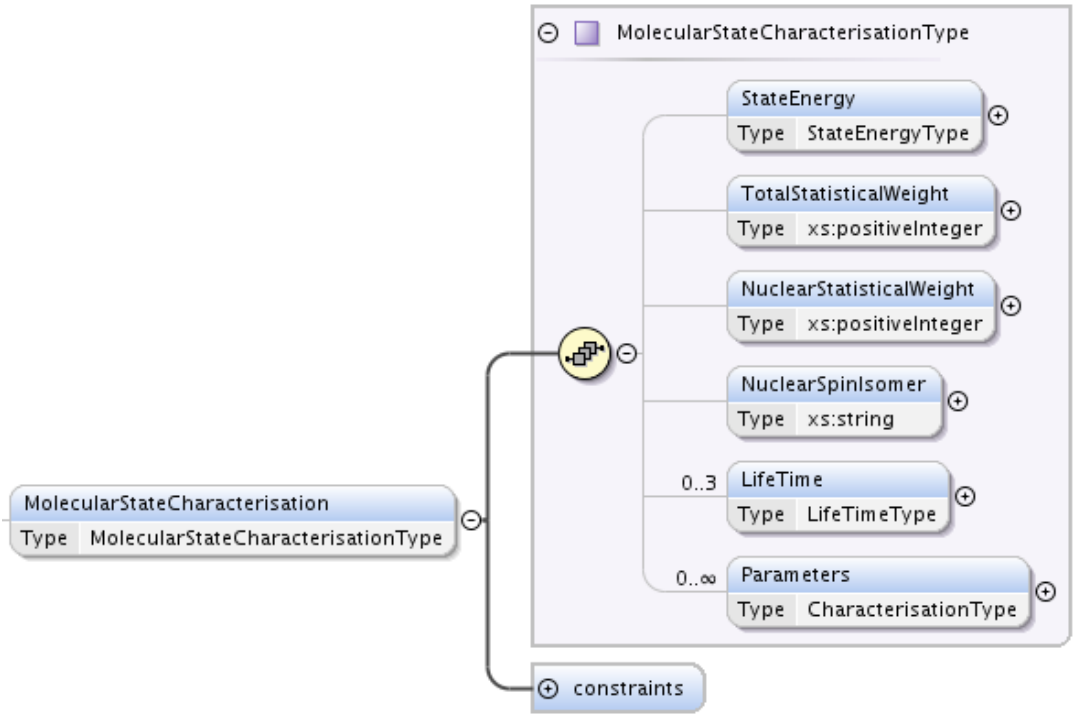
The eigenvalue corresponds to the **StateEnergy**, and is given relative to an **energyOrigin**. The eigenstate is characterized by a set of good quantum numbers, such as parity and total angular momemtum, and described by a wavefunction often expanded over some basis functions. The expansion is characterized by a coupling scheme between the quantum numbers identifying individual basis functions.



Following elements and attributes are defined for **MolecularState**

- **stateID** mandatory attribute, of type *StateIDType*, used for referencing the particular state.
- **fullyAssigned** optional boolean attribute, “true” defines that the state is fully described by a set of good quantum numbers.
- optional **Description** string element, where arbitrary name of state can be given.
- optional *MolecularStateCharacterisation* element of type **MolecularStateCharacterisationType** that describes all quantities related to the molecular state apart from quantum numbers (e.g. statistical weights, Land’e factors, radiative lifetime of the level and other properties).
- optional **Parity** element, that gives the total parity of the level. It is of type *ParityType*, a token taking the values *odd* or *even*.
- optional **Case** element of type *BaseCase*, that is a container for [case-by-case] definition of quantum numbers. *BaseCase* is explained below.
- optional *StateExpansion* element that allows to describe the state as a superposition of basis quantum numbers sets, each set with it’s own coefficient.

9.5 MolecularStateCharacterisation



MolecularStateCharacterisation element, defined as **MolecularStateCharacterisationType**, describes all quantities related to the molecular state apart from it's quantum numbers. Following optional elements are defined:

- **StateEenrgy**, of type **StateEnergyType**. Defined as an extension of *DataType* with an additional optional string attribute **energyOrigin**. **energyOrigin** should contain a human readable string indicating the nature of the origin taken for the energy scale. For example, **energyOrigin** might be *ground electronic*, *vibrational*, *rotational* state or could be *ionization energy* or *dissociation energy*. By physical meaning, state energy is the eigenvalue of the hamiltonian describing the species.
- **TotalStatisticalWeight** positive integer element, keeps statistical weight associated to the level, with all degeneracies (including nuclear spins).
- **NuclearStatisticalWeight** positive integer element, the same as **TotalStatisticalWeight** but for nuclear spins only.
- **NuclearSpinIsomer**, a string indicating the type of nuclear spin symmetry. Possible values can be *para*, *ortho*, *meta*, *A*, *E*. This element is a comfort element very often used to classify levels.
- **Lifetime**, of type **LifeTimeType**. Defined as an extension of *DataType* with an additional mandatory attribute **decay** that may take values *total*, *totalRadiative*, *totalNonRadiative*, allowing to represent total state lifetime, including radiative and non-radiative decay mechanisms, or define them separately.

  Maximum of three occurences of this element is possible, each must have different value in **decay** attribute.

  When only discrete radiative decay is involved, it is given by $\tau_i = \frac{1}{\sum_k A_{ik}}$
- **Parameters**, list of elements of *CharacterisationType*. It allows to add any additional characterisation of the molecular state.

  If data producer is intending to use this element, full description of data format and meaning should be given in one of the source references.

  As an example it can be used to describe a statistical weight associated to the level including

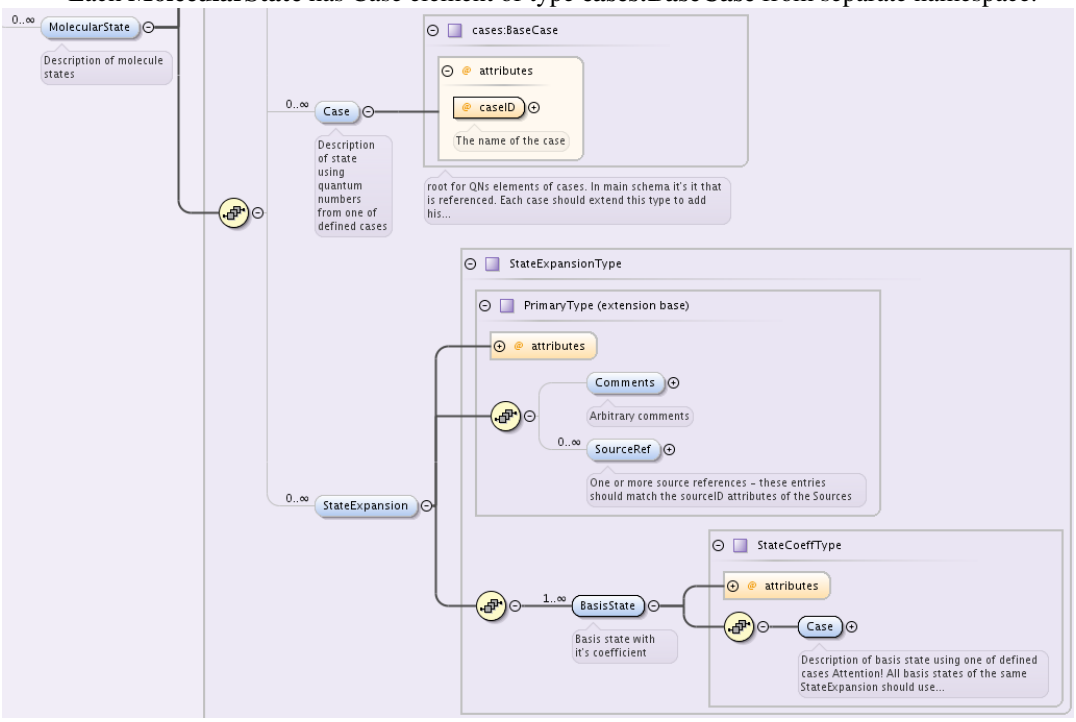
some degeneracies, but not all. In that case the element "Parameter.Name" takes the value **PseudoStatisticalWeight**. It does not include all degeneracies and is used in fractions. CDMS database provides such **PseudoStatisticalWeight**.

9.6 Concept of cases

The case-by-case XML description of molecular states within VAMDC-XSAMS is designed to provide a straightforward and flat data structure for representing the quantum numbers and symmetries that denote a molecular state. The reader is the refered to the [case-by-case] documentation for full description of the various cases.

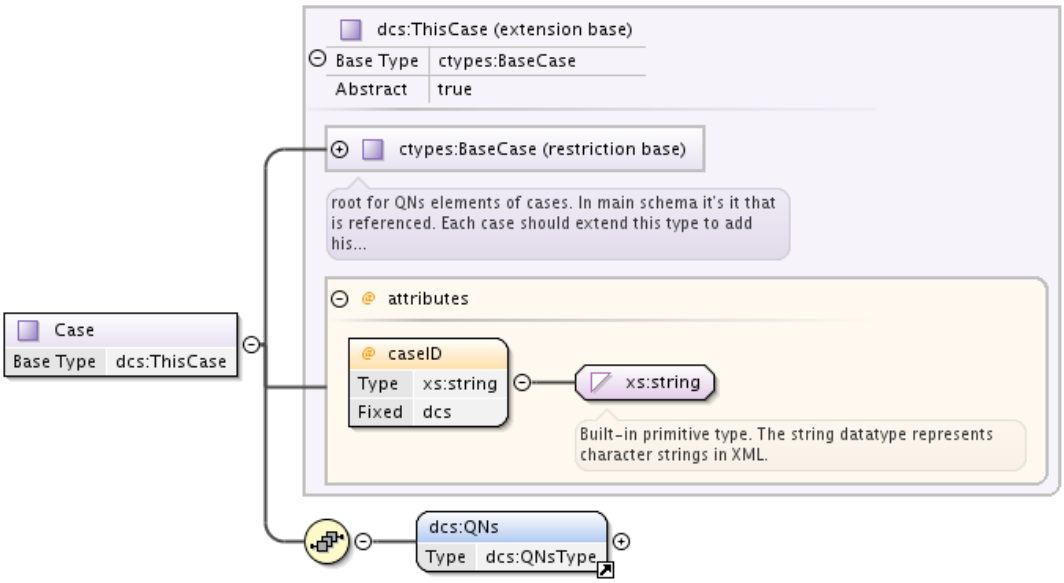
9.6.1 StateExpansion

Each **MolecularState** has Case element of type **cases:BaseCase** from separate namespace.



9.6.2 BaseCase

BaseCase type defines the single attribute, *caseID*, that denotes the case used.



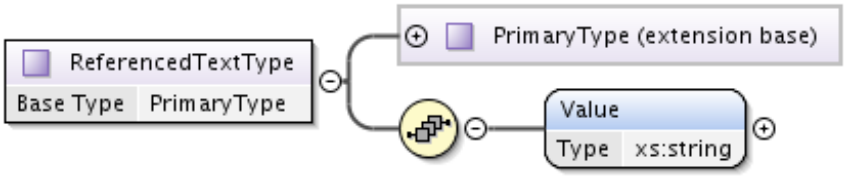
Each specific case, in turn, extends that **BaseCase** type, defining specific *caseID* attribute value and adding **QNs** element that contains a sequence of quantum numbers and symmetries.

9.7 Specific XML Types

Here, specific XML types, used only in Species.Molecules are described.

9.7.1 ReferencedTextType

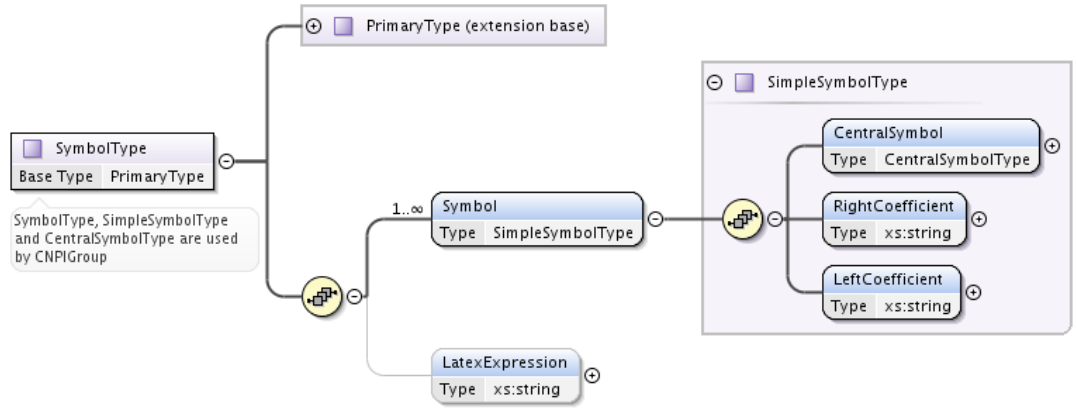
An extension of *PrimaryType* that has additional string **Value** element, is used to define strings with *Source* reference.



9.7.2 SymbolType:

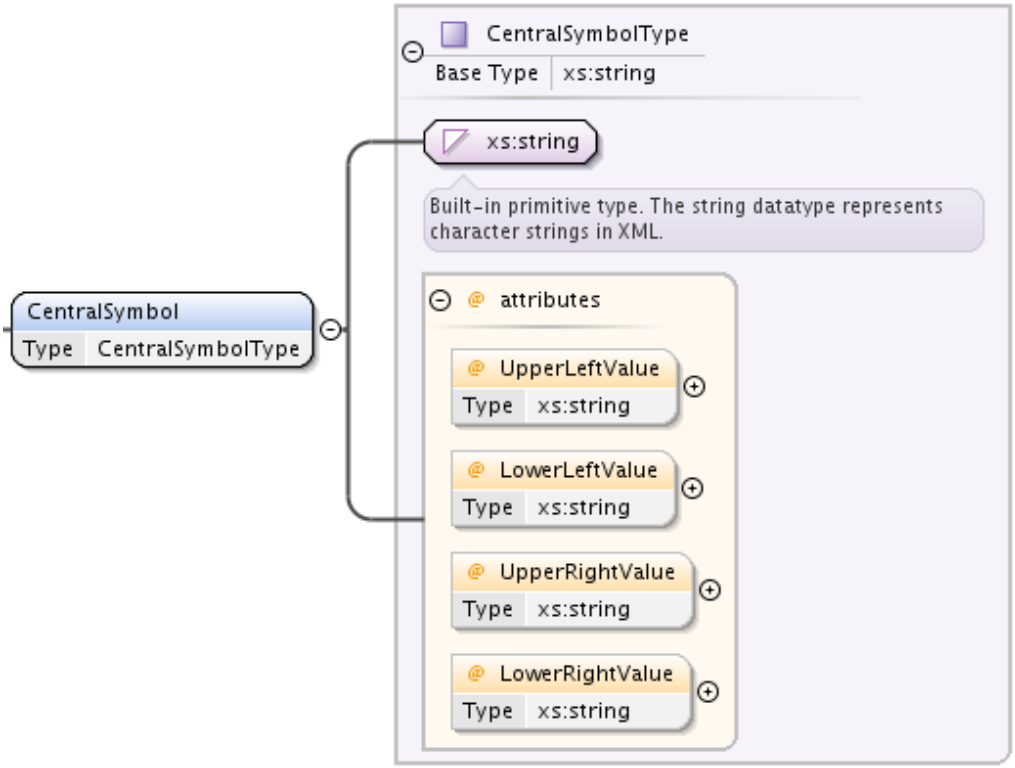
It is a model to describe symbols. Extending *PrimaryType*, it adds:

- a list of **Symbol** elements of type **SimpleSymbolType**
- a string element **LatexExpression**, that may contain latex representation of the symbol.



SimpleSymbolType defines three string elements,

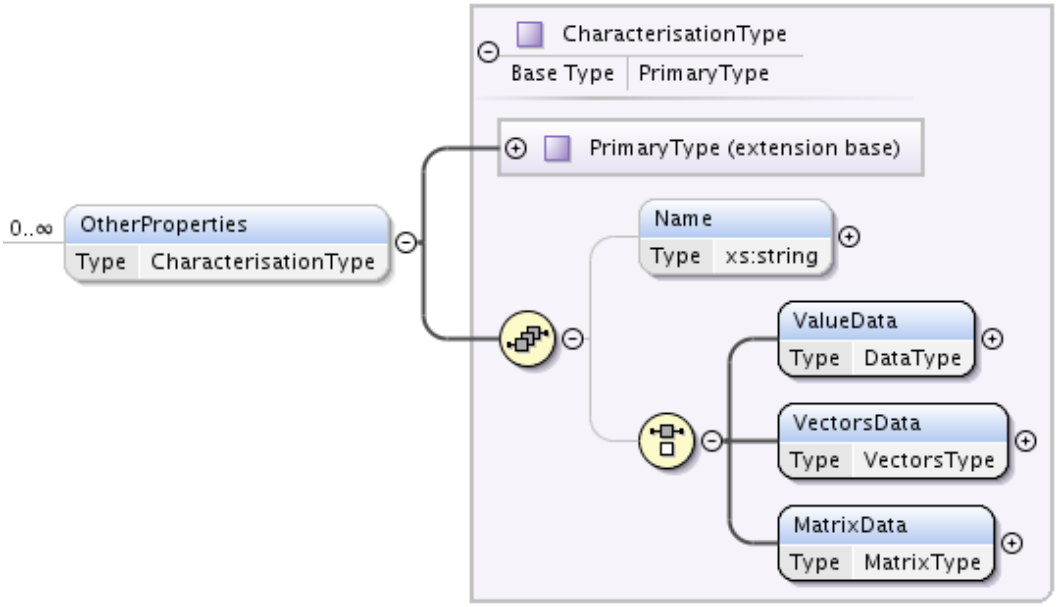
- **CentralSymbol** which is a string with 4 string attributes,
  - **UpperLeftValue**
  - **LowerLeftValue**
  - **UpperRightValue**
  - **LowerTightValue**



- **RightCoefficient**
- **LeftCoefficient**

Mathematical symbols and greek alphabet letters should be defined as unicode symbols, for example, ∞ or ∞ for infinity sign (∞)

9.7.3 CharacterisationType



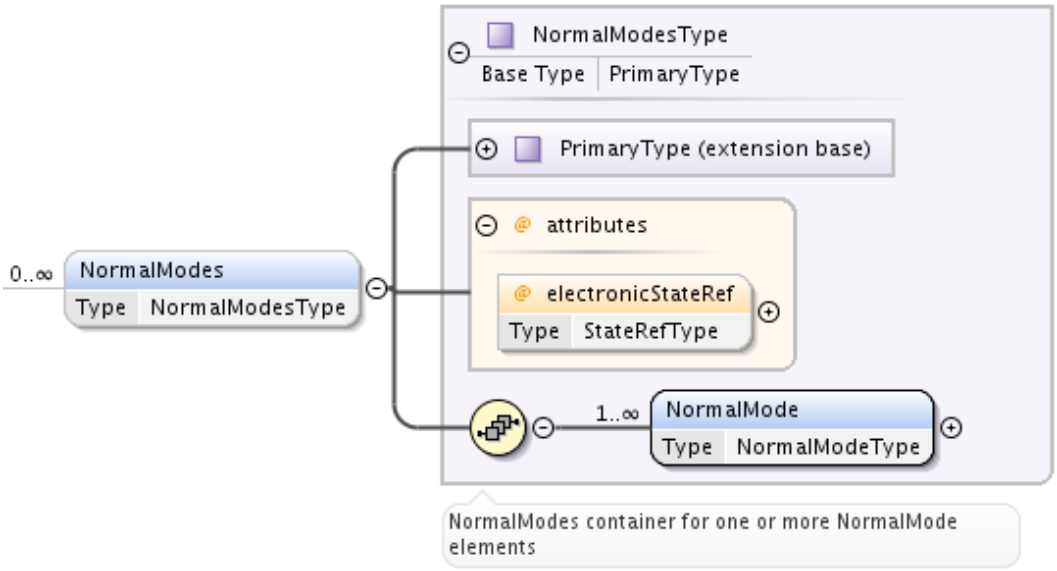
CharacterizationType is an extension of *PrimaryType*, adding a **Name** string element and a choice of one of:

- **ValueData** of type *DataType*,
- **VectorsData** of type *VectorsType* or
- **MatrixData** of type *MatrixType:*.

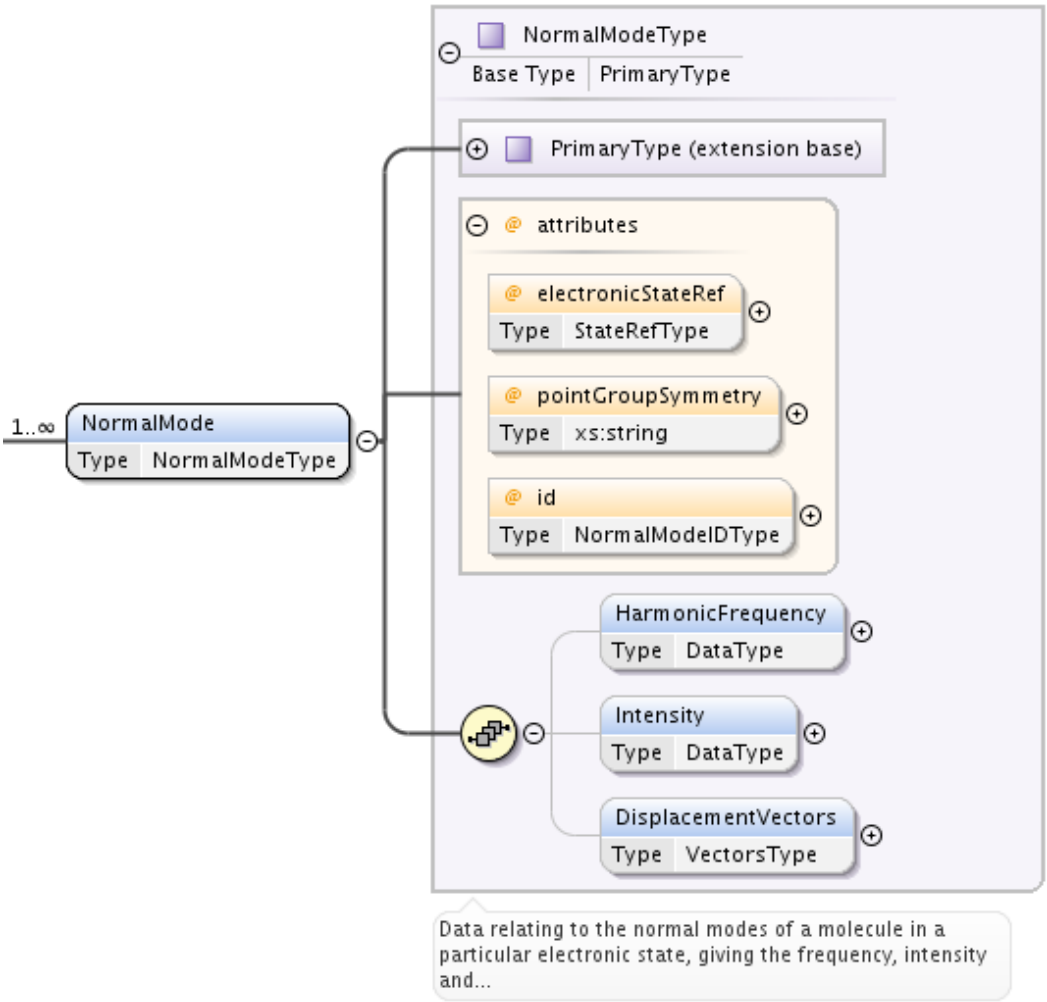
permitting representation of arbitrary data, relevant to state.

9.7.4 NormalModes

To represent vibrational normal modes of molecules, **NormalModes** element is used.



Each **NormalModes** element, extending *PrimaryType*, may have an attribute **electronicStateRef**, defining reference to electronic state, and must have at least one **NormalMode** element, each defining a single mode.



NormalMode element, also extending *PrimaryType*, has following attributes and elements:

- optional **electronicStateRef** attribute, of type *StateRefType*, defining electronic state;
- optional **pointGroupSymmetry** string attribute;
- optional **id** attribute of *NormalModeIDType*, defining unique identifier for this mode, to be referenced from radiative *CrossSection* band assignment;
- optional **HarmonicFrequency** element of *DataType*;
- optional **Intensity** *DataType* element;
- optional **DisplacementVectors** element of type *VectorsType* to define atoms displacement configuration of the mode.
  - **ref** attribute of **Vector** must contain the id of the atom in molecule's structure,
  - **x3**, **y3**, **z3** define atom's relative displacement against it's position in ground state.

Example:

Example XML block for **NormalModes** would look like:

```
<NormalModes electronicStateRef="SX_Azulene-1">
        <NormalMode id="V1" pointGroupSymmetry="A1">
                <HarmonicFrequency>
                        <Value units="1/cm">162</Value>
                        <Accuracy><Systematic>1</Systematic></Accuracy>
                </HarmonicFrequency>
```

```
                <Intensity>
                        <Value units="km/mol">0</Value>
                </Intensity>
                <DisplacementVectors units="A">
                        <Vector ref="C1" x3="0." y3="0.001" z3="0.0005"/>
                        <Vector ref="C2" x3="0.01" y3="-0.001" z3="0.0005"/>
                        <Vector ref="C3" x3="-0.005" y3="0.001" z3="0."/>
                        <!-- etc... -->
                </DisplacementVectors>
        </NormalMode>
        <NormalMode id="V2" pointGroupSymmetry="A2">
                <HarmonicFrequency>
                        <Value units="1/cm">214</Value>
                        <Accuracy><Statistical>5</Statistical></Accuracy>
                </HarmonicFrequency>
        </NormalMode>
        <NormalMode id="V3" pointGroupSymmetry="A1">
                <HarmonicFrequency>
                        <Value units="1/cm">1720.21</Value>
                        <Accuracy>
                                <Statistical>0.05</Statistical>
                        </Accuracy>
                </HarmonicFrequency>
        </NormalMode>
</NormalModes>
```

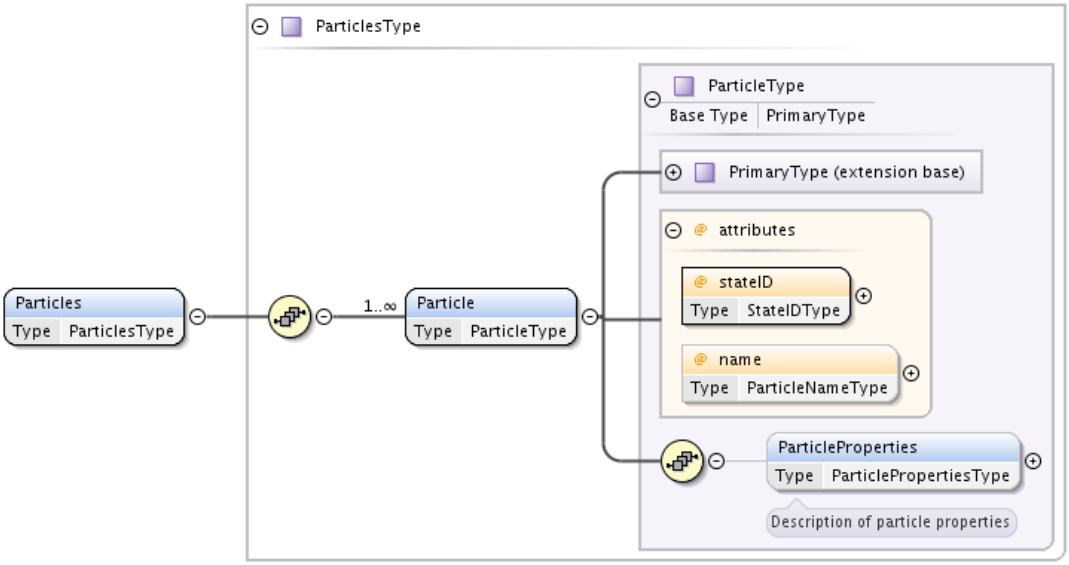
CHAPTER

TEN

SPECIES.PARTICLES

Describes a number of *elementary* particles, namely, **electron**, **photon**, **muon**, and **positron**, if such a description is essential for a particular process. An example can be provided by photoionization.

10.1 Particle

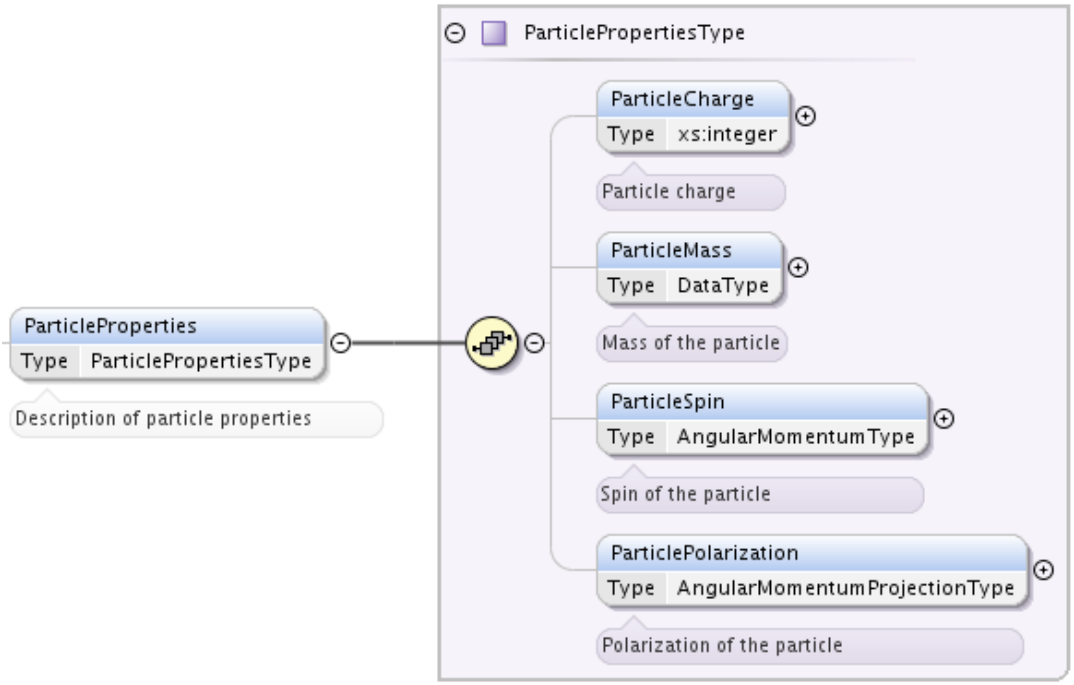


If a particle is needed in some of the processes, then it must have a **stateID** (*StateIDType*) identifier to enable referencing it. The attribute **name** of the type is optional. It is defined as a restriction of String, that may take following values:

- photon
- electron
- muon
- positron

Other particle names may be added in future versions, based on schema users requests.

ParticleProperties XML element may contain some static properties of the particle:



- **ParticleCharge** integer element,
- **ParticleMass** element of *DataType*,
- **ParticleSpin** (type *AngularMomentumType*), and
- **ParticlePolarization** element (type *AngularMomentumProjectionType*).

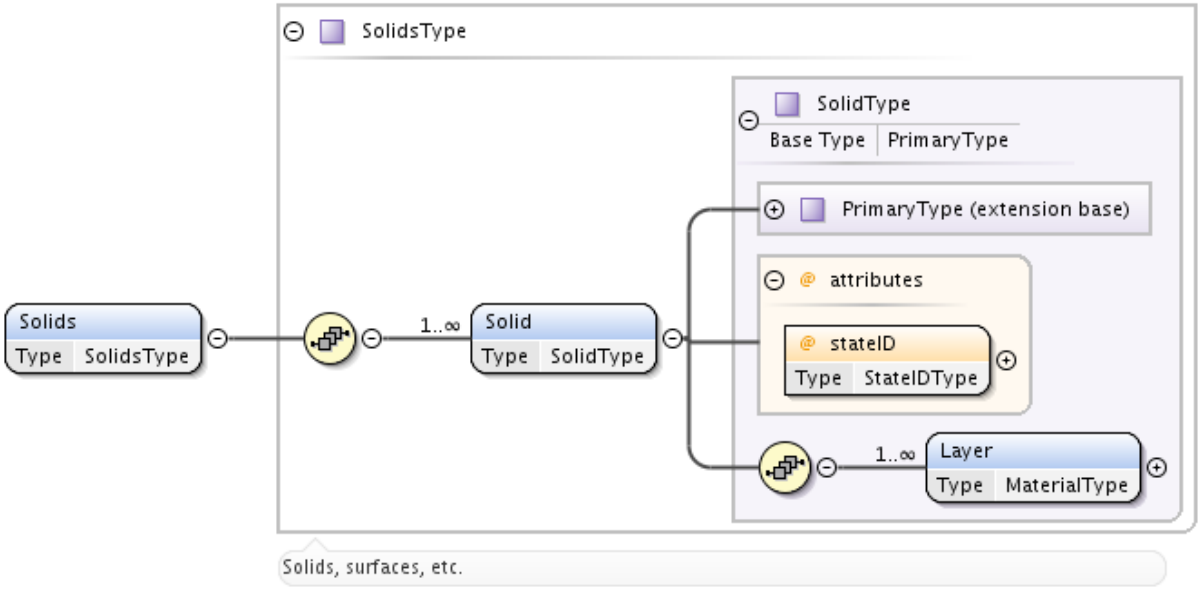
CHAPTER

ELEVEN

SPECIES.SOLIDS

The **Solids** branch of XSAMS that describes properties of solids and surfaces is still in the inital phase of development. However, it was decided to provide at least such rudimentary description in order to give users and developers a clear idea of the chosen approach.

The same way as with other **species**, **Solids** element is just a container for **Solid** elements.



11.1 Solid

Extension of the *PrimaryType*. Attribute **stateID** is mandatory. Must have one or more **Layer** elements of type **MaterialType**.

11.2 Layer

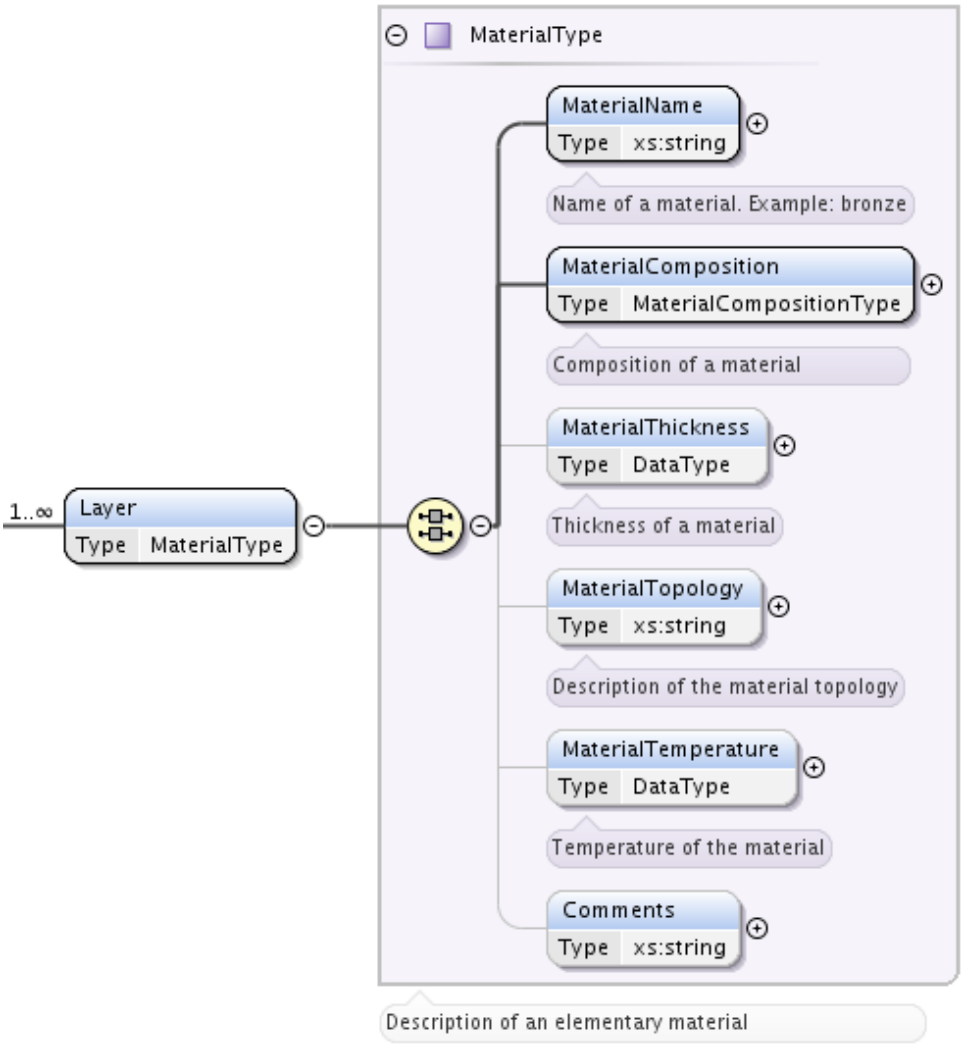
The layer is defined here as a *uniform macroscopic* part of a solid. It is characterized by the mandatory elements:

- **MaterialName** (type **xs:string**) and
- *MaterialComposition* of type **MaterialCompositionType**.

Other optional elements are:

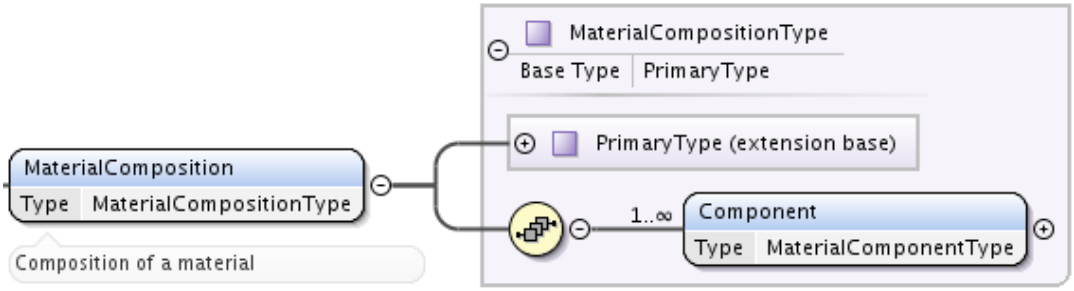
- **MaterialThickness** of *DataType*,
- string **MaterialTopology**,

- **MaterialTemperature** (type *DataType*),
- arbitrary string **Comments**.

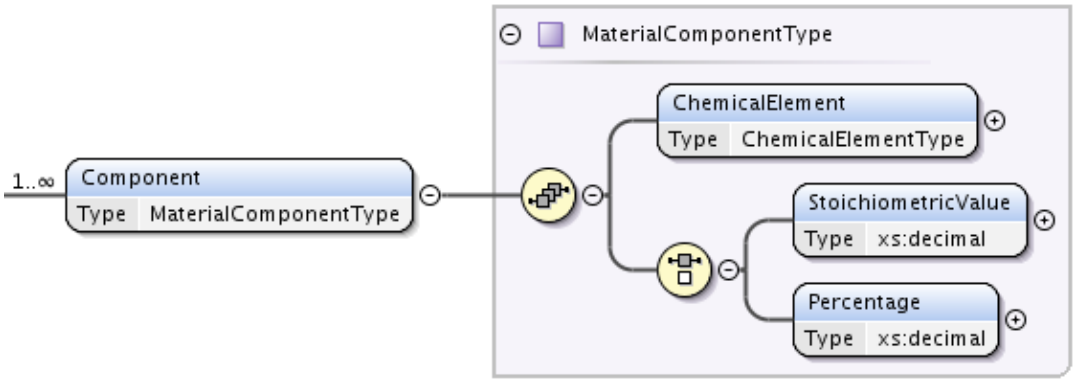


11.3 MaterialComposition

Extension of the *PrimaryType*, must have one or more of **Component** elements, each belonging to the **MaterialComponentType**.



Component element has **ChemicalElement** element (of *ChemicalElementType*) and either **StoichiometricValue** or **Percentage**, both **xs:decimal**. All these parameters are mandatory.

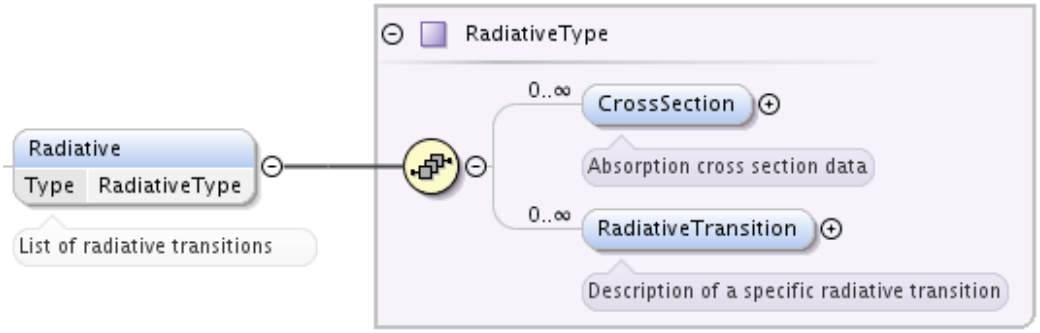

MaterialComponentType
ChemicalElement
Type ChemicalElementType
StoichiometricValue
Type xs:decimal
Percentage
Type xs:decimal
1..∞
Component
Type MaterialComponentType

CHAPTER

TWELVE

PROCESSES.RADIATIVE

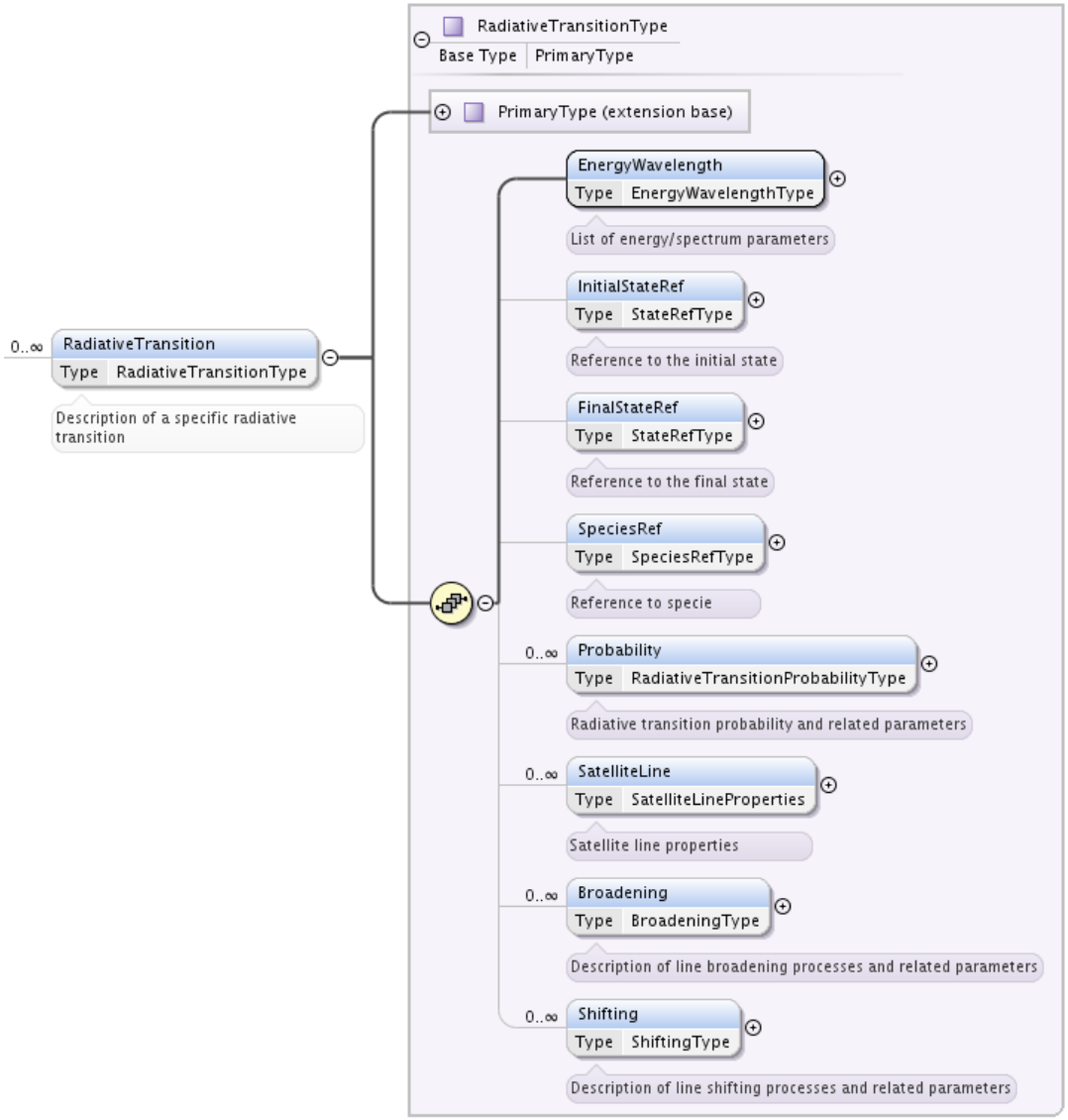
This part of the schema describes radiative processes including spontaneous radiative decays. Also photoabsorption *CrossSection* is represented here. The other way to describe Collisions between photons and various objects (e.g., molecules) is in the *Processes.Collisions* part.



12.1 RadiativeTransition

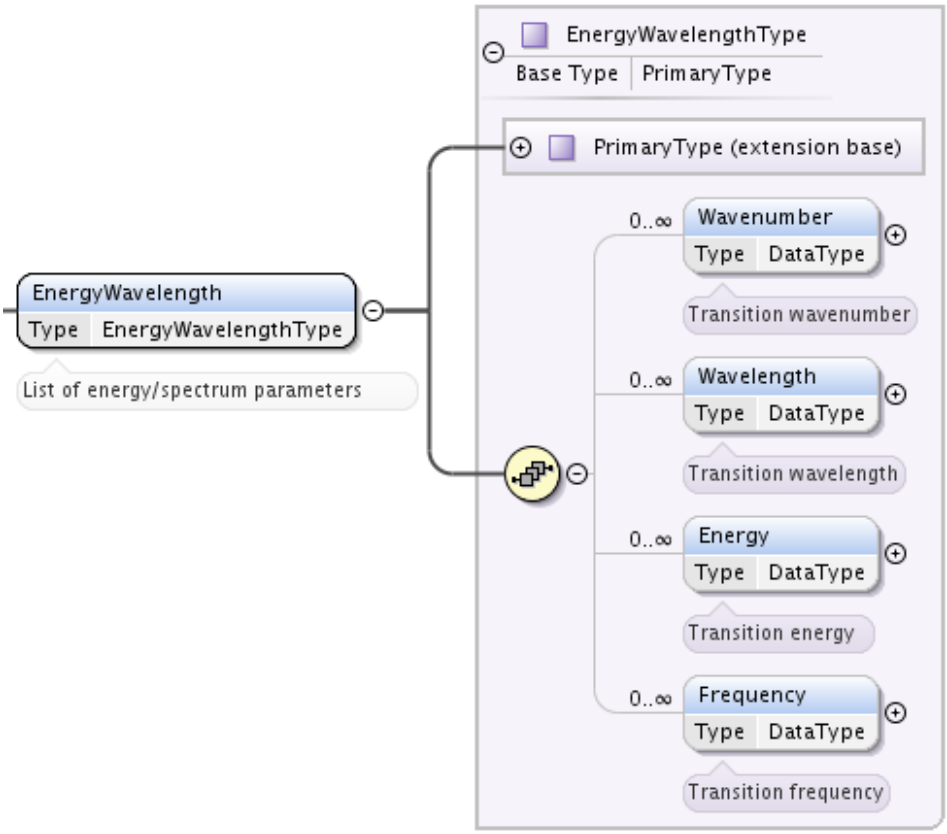
Extension of the *PrimaryType*. A transition is characterized by its energy/wavelength (element *EnergyWavelength*) and following optional elements:

- **InitialStateRef** of type *StateRefType*, reference to the initial state of the transition,
- **FinalStateRef**, reference to the final state of the transition, type *StateRefType*),
- **SpeciesRef** element of type *SpeciesRefType*, that may be used in place of the previous two in case of unknown states. All three references may be specified, in that case states should belong to the molecule/atom, indicated by **SpeciesRef**.
- list of *Probability* elements of type **RadiativeTransitionProbabilityType**. Different values of probabilities may be due to different multipole orders (e.g., M1 and E2 may be possible for the same initial and final states).
- multiple *SatelliteLine* elements,
- multiple *Broadening* elements,
- list of *Shifting*.



12.2 EnergyWavelength

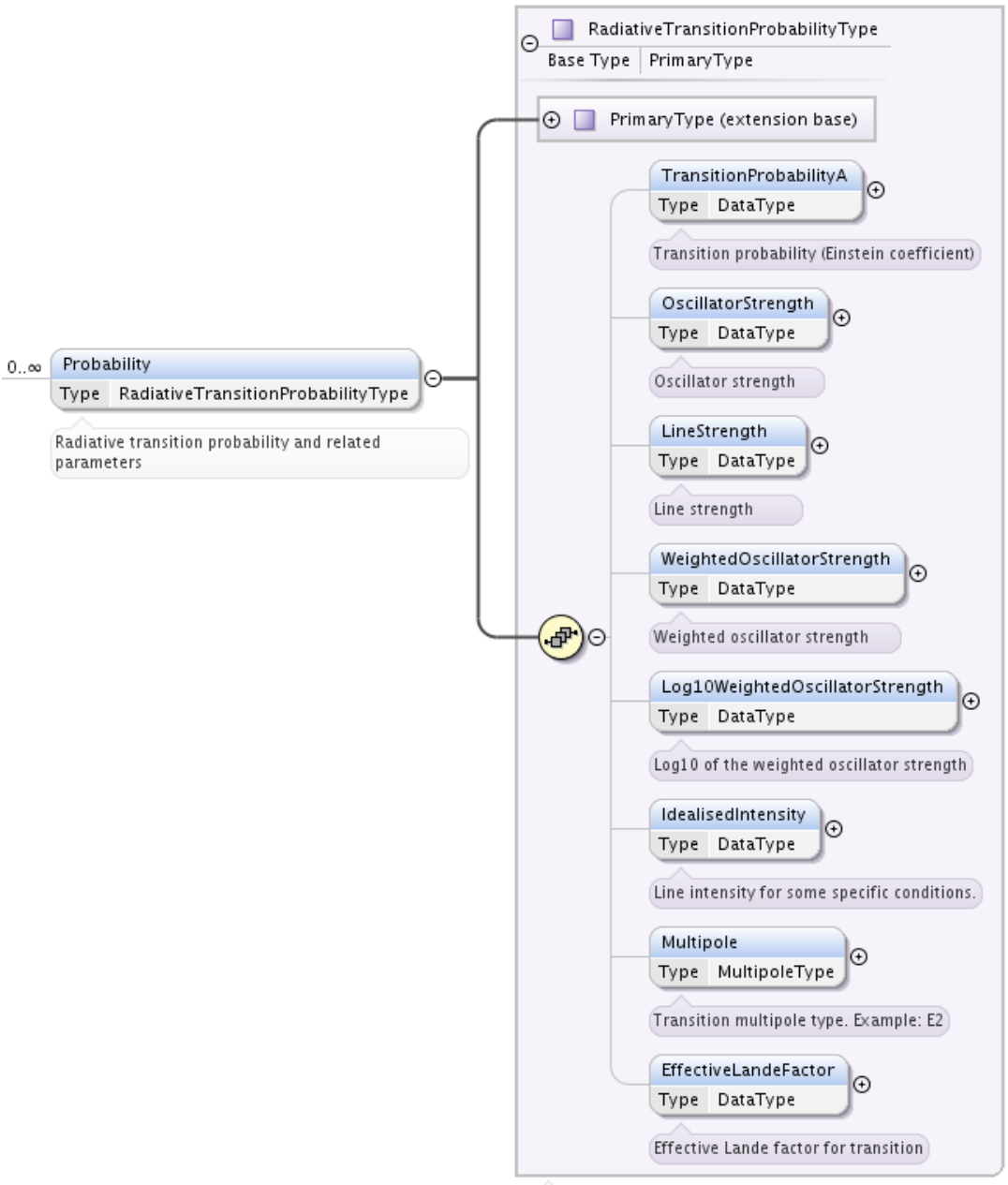
Extension of the *PrimaryType*, may contain multiple **Wavenumber**, **Wavelength**, **Energy** or **Frequency** elements, each of *DataType*. Exact nature of parameter must be determined through *Method*, with help of *PrimaryType*'s method reference. Method's **Category** element takes in this case values in (*experiment*, *theory*, *ritz*).



12.3 Probability

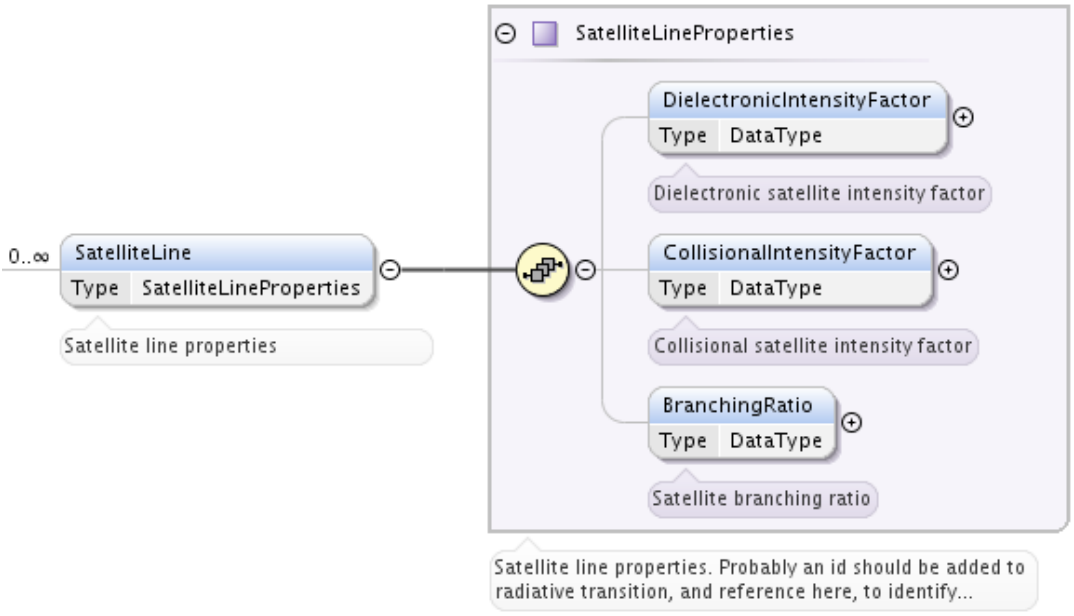
Probability element is defined as the extension of the *PrimaryType*. Describes parameters, relevant to transition probability. Following optional elements are possible, all of *DataType* if not stated otherwise:

- **TransitionProbabilityA** - Einstein coefficient, or transition probability.
- **OscillatorStrength** - Oscillator strength (dimensionless). Whether it is absorption or emission, is determined from the initial and final states of the transition.
- **LineStrength** - A symmetric quantity with respect to initial and final states of the transition.
- **WeightedOscillatorStrength** - Product of the oscillator strength and statistical weight of the **initial** state.
- **Log10WeightedOscillatorStrength** - Log_{10} of the **WeightedOscillatorStrength**.
- **IdealisedIntensity** - Line intensity under specific conditions. Due to its dependence on plasma parameters, this is not a universal property of a spectral line and thus should be used with care.
- **Multipole** of type **MultipoleType** - multipole order of a radiative transition (electric or magnetic). It is a string with the first upper-case symbol **E** or **M** followed by one or more digits, and the first digit cannot be **0**.
- **EffectiveLandeFactor** - Effective Lande factor, line intensity coefficient for magneto-sensitive atomic lines.



12.4 SatelliteLine

SatelliteLine element is used to keep properties of atomic optical electron satellite lines:

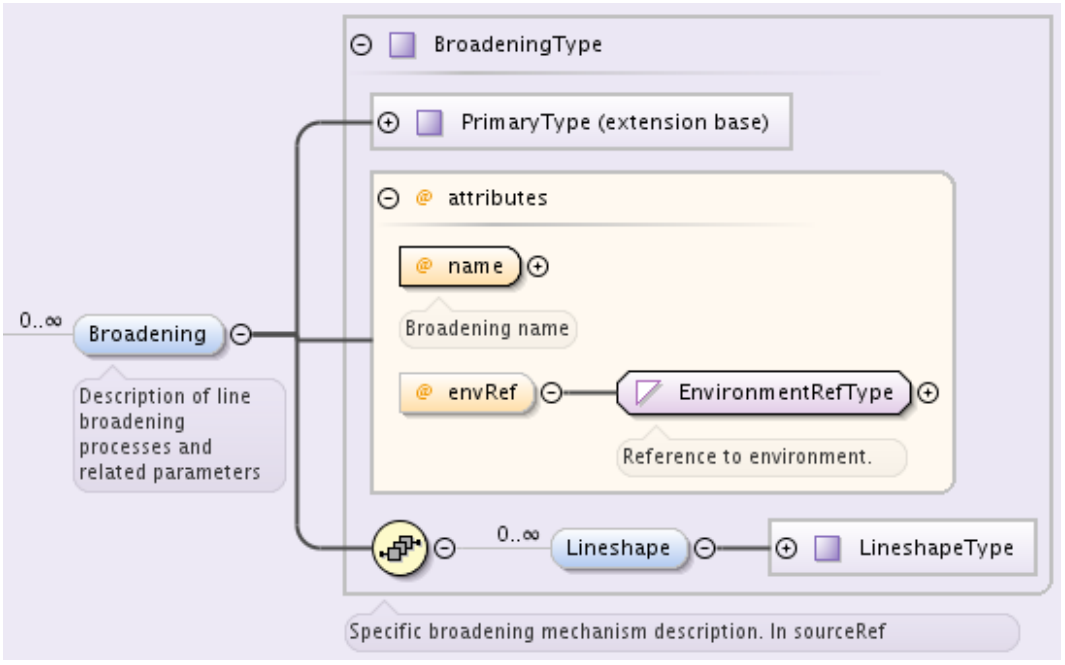


Multiple **SatelliteLine** elements may be specified, with three optional child elements:

- DielectronicIntensityFactor
- CollisionalIntensityFactor
- BranchingRatio

each of *DataType*.

12.5 Broadening



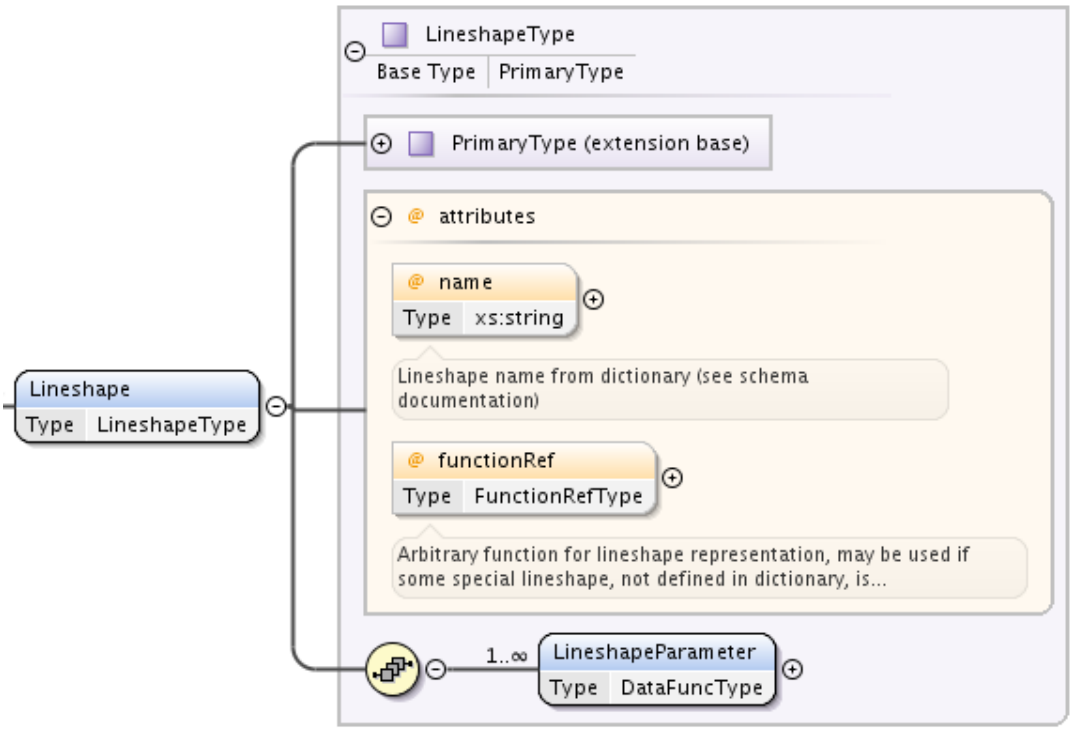
Each broadening element has

- **name** attribute, that must contain one of the names from *Lineshapes dictionary*
- **envRef** attribute, referencing the specific environment conditions, for example, for collisional broadening.

- one or more *Lineshape* elements, with their respective parameters.

Normally, one broadening record should be created for each broadening mechanism and for each source (data origin). Many Lineshape elements allow to represent, for example, processing of the same experimental data with different lineshapes. Usually, there will be only one Lineshape element.

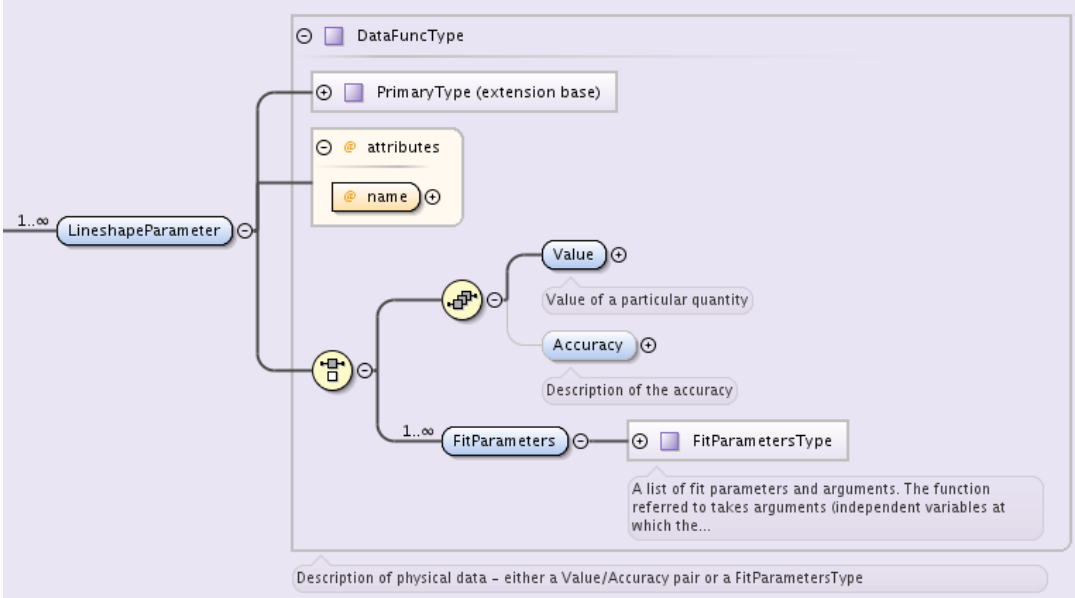
12.5.1 Lineshape



Lineshape in turn also has

- optional **name** attribute from *Lineshapes dictionary*, like Lorentz, Voigt, Doppler, etc.
- optional **functionRef** attribute of type *FunctionRefType*, with a reference to a *Functions* describing the lineshape. This attribute should be used only in case when the lineshape used in data fitting is absent in the *Lineshapes dictionary*
- one or more *LineshapeParameter*, each representing specific lineshape parameter as either a constant value or a function of environment parameters

12.5.2 LineshapeParameter



LineshapeParameter either **FitParameters** or **Value/Accuracy** pair must be specified

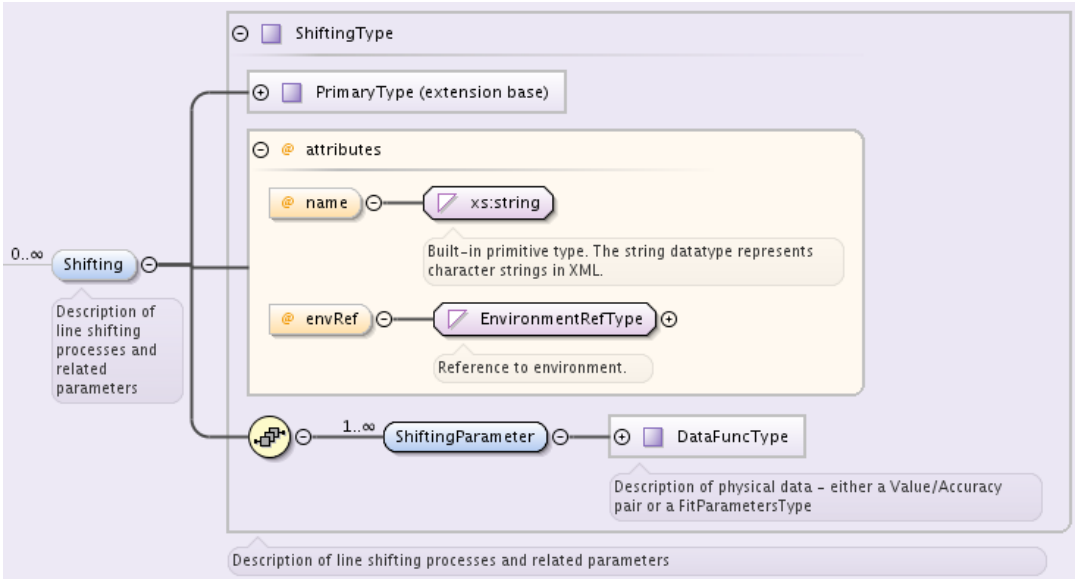
- **name** attribute corresponds to one defined in *Lineshapes dictionary* for specific lineshape
- **Value/Accuracy** elements are the same as in *DataType*,
- **FitParameters**, of type *FitParametersType*, that allows representation of this parameter as a function of environment parameters, for example:

```
<Broadening envRef="Eair-broadening-ref-env" name="pressure">
        <Lineshape name="Lorentzian">
                <Comments>The temperature-dependent pressure broadening
                Lorentzian lineshape</Comments>
                <LineshapeParameter name="gammaL">
                        <FitParameters functionRef="FgammaL">
                                <FitArgument units="K" name="T">
                                        <LowerLimit>240</LowerLimit>
                                        <UpperLimit>350</UpperLimit>
                                </FitArgument>
                                <FitArgument units="atm" name="p">
                                        <LowerLimit>0.</LowerLimit>
                                        <UpperLimit>1.2</UpperLimit>
                                </FitArgument>
                                <FitParameter name="gammaL_ref">
                                        <SourceRef>BHIT-B_HITRAN2008</SourceRef>
                                        <Value units="1/cm">0.0635</Value>
                                        <Accuracy>0.003175</Accuracy>
                                </FitParameter>
                                <FitParameter name="n">
                                        <SourceRef>BHIT-B_HITRAN2008</SourceRef>
                                        <Value units="unitless">0.75</Value>
                                        <Accuracy>0.15</Accuracy>
                                </FitParameter>
                        </FitParameters>
                </LineshapeParameter>
        </Lineshape>
</Broadening>
```

Where function FgammaL is defined as follows:

```
<Function functionID="FgammaL">
        <Comments>This function gives the pressure- and
        temperature-dependence of the Lorentzian component of the
        pressure-broadened line width (HWHM)</Comments>
        <Expression computerLanguage="Fortran">
                gammaL_ref * p * (296./T)**n
        </Expression>
        <Y name="gammaL" units="1/cm"/>
        <Arguments>
                <Argument name="T" units="K">
                        <Description>The absolute temperature, in K</Description>
                </Argument>
                <Argument name="p" units="atm">
                        <Description>The partial pressure of the broadening species,
                                                        in atm</Description>
                </Argument>
        </Arguments>
        <Parameters>
                <Parameter name="gammaL_ref" units="1/cm">
                        <Description>The Lorentzian HWHM of the line, broadened at
                                Tref = 296 K and broadening species partial pressure
                                pref = 1atm</Description>
                </Parameter>
                <Parameter name="n" units="unitless">
                        <Description>
                                The temperature exponent of the gammaL function
                        </Description>
                </Parameter>
        </Parameters>
</Function>
```

12.6 Shifting



Line shifting is defined by **name** and/or **envRef** attributes. In case of linear collisional shifting, only environment is sufficient.

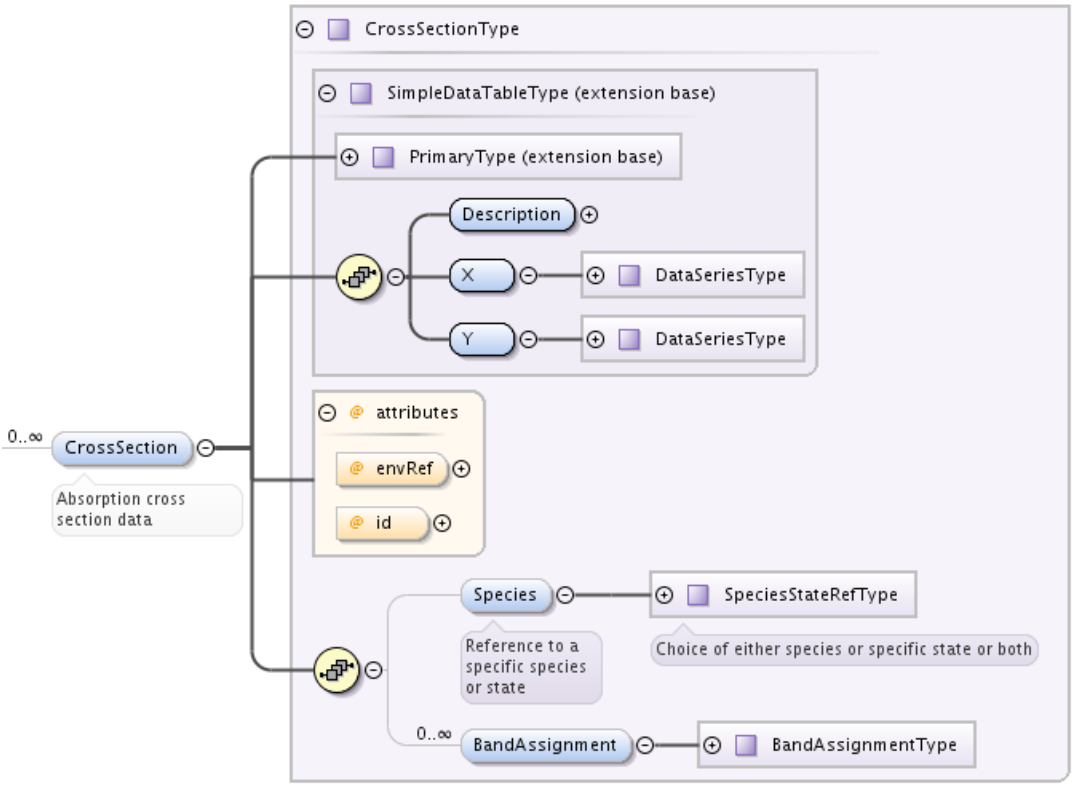
ShiftingParameter element is defined the same way as broadening *LineshapeParameter*, it is either value or function of environment parameters.

Example of a Shifting definition:

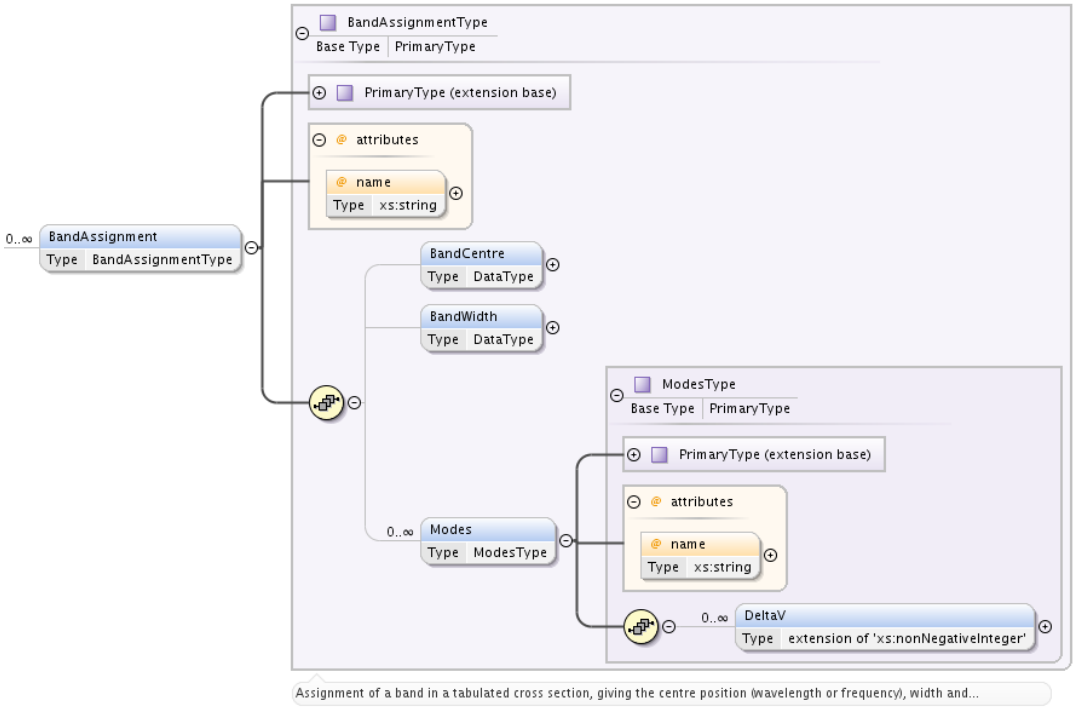
```
<Shifting envRef="Eair-broadening-ref-env">
        <ShiftingParameter name="delta">
                <FitParameters functionRef="Fdelta">
                        <FitArgument name="p" units="atm">
                                <LowerLimit>0.</LowerLimit>
                                <UpperLimit>1.2</UpperLimit>
                        </FitArgument>
                        <FitParameter name="delta_ref">
                                <SourceRef>BHIT-B_HITRAN2008</SourceRef>
                                <Value units="unitless">-0.001</Value>
                                <Accuracy>0.1</Accuracy>
                        </FitParameter>
                </FitParameters>
        </ShiftingParameter>
</Shifting>
```

12.7 CrossSection

Among with **RadiativeTransition** elements, *Processes.Radiative* processes block now has a **CrossSection** element, which allows description of absorption cross-section data and vibrational bands assignment in case of complex molecules.



- **Description**, **X** and **Y** elements describe cross-section data in tabular form, where **X** can be absorbed radiation frequency, wavelength or wavenumber in a form of a list of values or sequence. **Y** then represents a sequence of sigma values.
- **Species** element may have **StateRef** and/or **SpeciesRef** child elements, indicating species or specific states, to which crossection data applies.
- **BandAssignment** allows to indicate specific vibrational modes in cross-section data.



12.7.1 Example cross-sections record

Warning: **DataList** is truncated for clarity, originally it contains 880 space-separated numbers. You may see the original element in schema examples (*tests/valid/azulene-working.xml*).

```
<CrossSection>
        <SourceRef>B_NIST1</SourceRef>

        <Description>The IR transmittance cross section of azulene from the NIST
        Standard Reference Data Program Collection</Description>

        <X parameter="wavenumber" units="1/cm">
                <LinearSequence n="880" units="1/cm" a0="450." a1="4"/>
        </X>
        <Y parameter="sigma" units="arbitrary">
                <DataList n="880">
                        0 85 94 .. 102
                </DataList>
        </Y>

        <Species>
                <SpeciesRef>X-CUFNKYGDVFVPHO-UHFFFAOYAT</SpeciesRef>
                <StateRef>SX_Azulene-1</StateRef>
        </Species>

        <BandAssignment name="2v1+v2">
                <BandCentre>
                        <Value units="1/cm">410</Value>
                        <Accuracy>2</Accuracy>
                </BandCentre>
                <BandWidth>
                        <Value units="1/cm">40</Value>
                        <Accuracy>5</Accuracy>
                </BandWidth>
                <Modes>
                        <DeltaV modeID="V1">2</DeltaV>
                        <DeltaV modeID="V2">1</DeltaV>
```

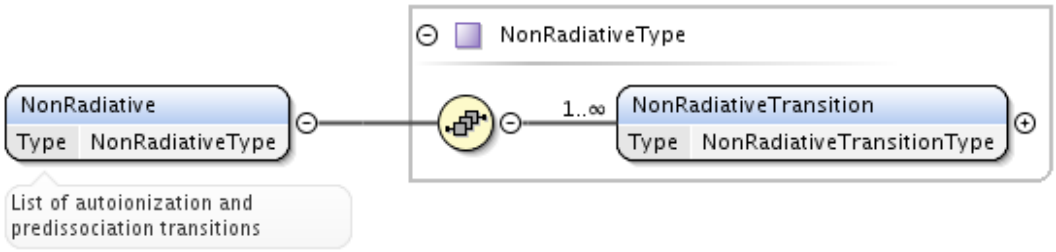
```
                </Modes>
        </BandAssignment>
        <BandAssignment name="3v4+2v5">
                <BandCentre>
                        <Value units="1/cm">1657</Value>
                        <Accuracy>10</Accuracy>
                </BandCentre>
                <BandWidth>
                        <Value units="1/cm">120</Value>
                        <Accuracy>15.5</Accuracy>
                </BandWidth>
                <Modes>
                        <DeltaV modeID="V2">3</DeltaV>
                        <DeltaV modeID="V3">2</DeltaV>
                </Modes>
        </BandAssignment>

</CrossSection>
```

CHAPTER

THIRTEEN

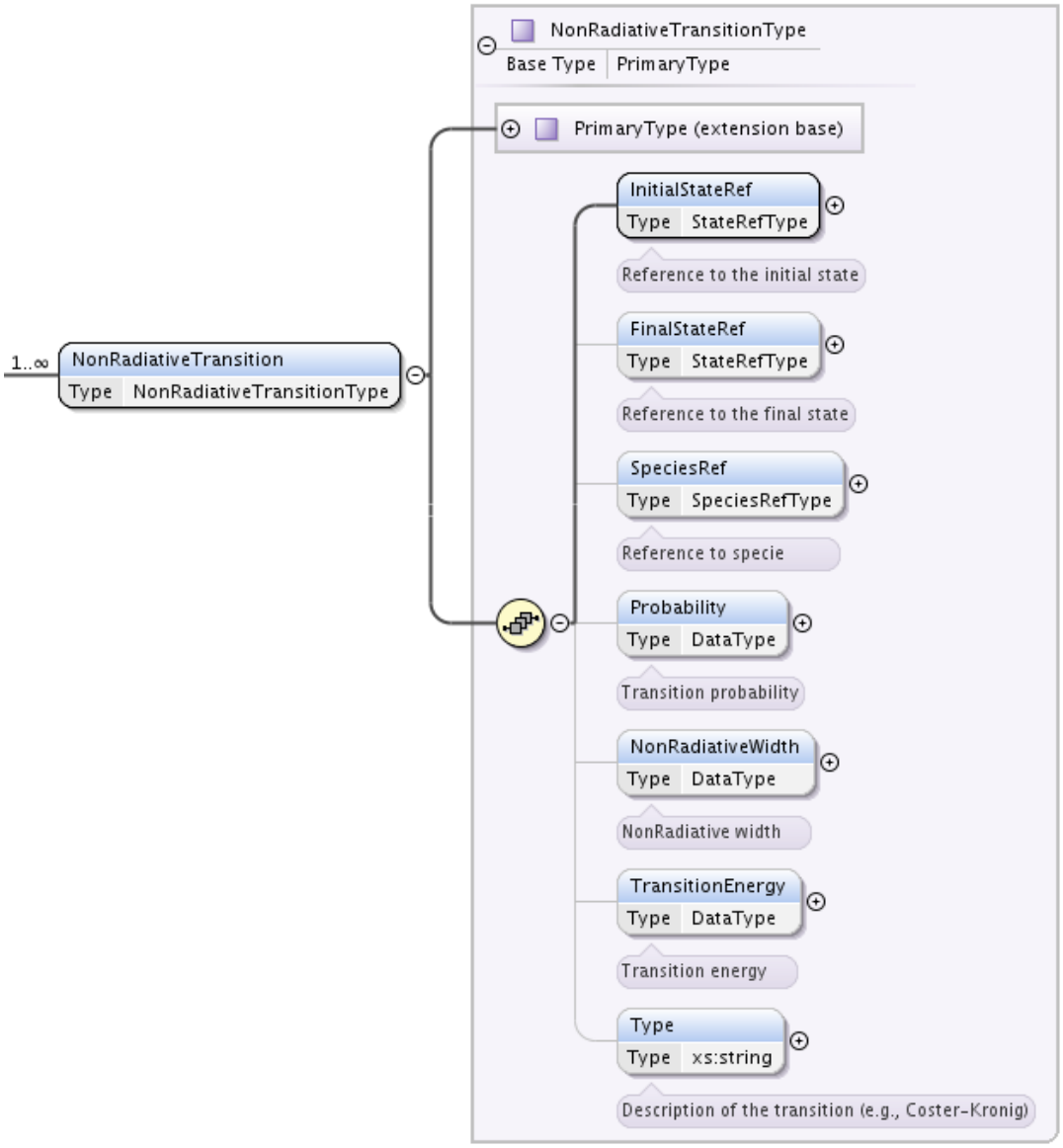
PROCESSES.NONRADIATIVE

Describes typical processes of a spontanous decay without photons, for instance, autoionization or predissociation.



NonRadiative element is a container for a list of **NonRadiativeTransition** elements

13.1 NonRadiativeTransition



NonRadiativeTransition element is defined in **NonRadiativeTransitionType** as the extension of the *PrimaryType*, containing following elements:

- mandatory **InitialStateRef** *StateRefType* reference,
- optional **FinalStateRef** of the same type,
- optional **SpeciesRef** of *SpeciesRefType*,
- optional **Probability** element (type *DataType*),
- optional **TransitionEnergy** of *DataType*, the energy difference between initial and final state.
- optional string **Type** element, that may provide additional details (e.g., indicate that this is a Coster-Kronig process).

CHAPTER

FOURTEEN

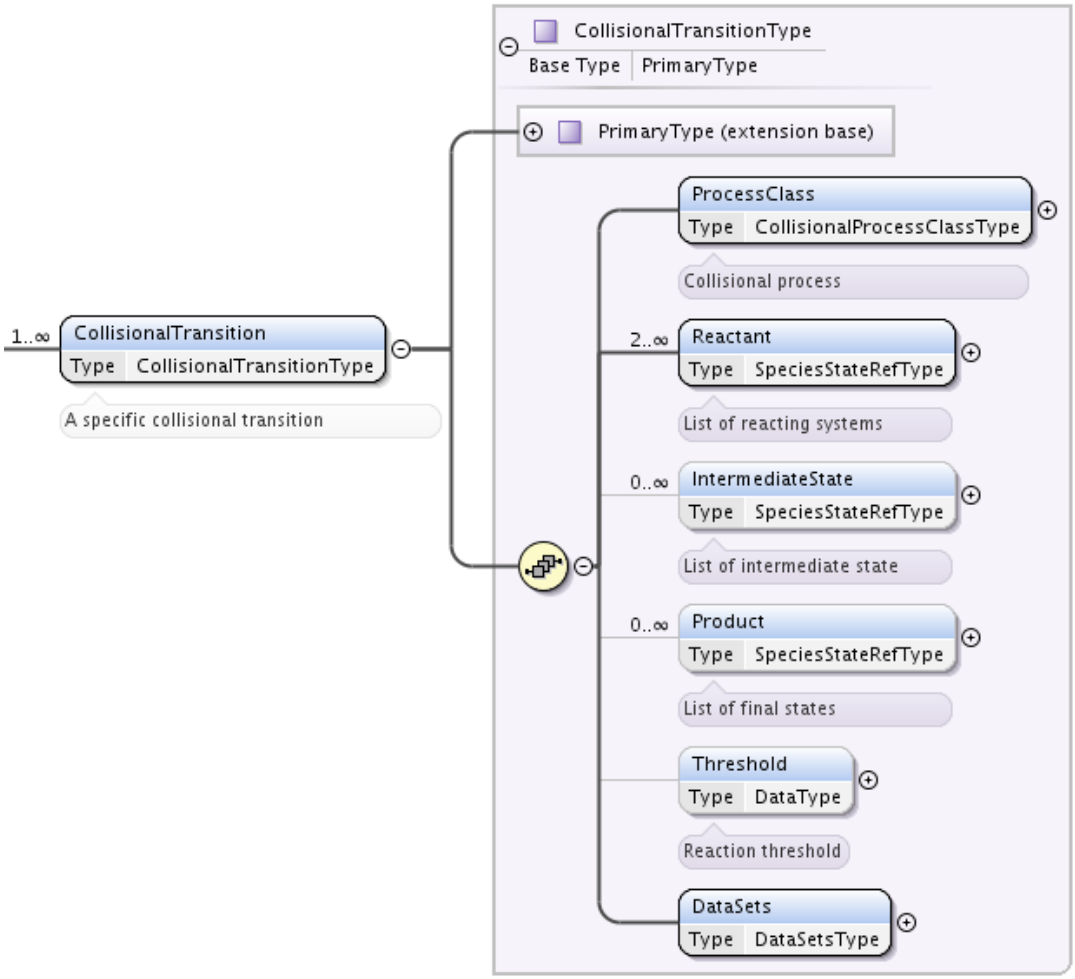
PROCESSES.COLLISIONS

The element **Collisions** of the **Processes** branch, defined by **CollisionsType**, contains one or more **CollisionalTransition** elements.

14.1 CollisionalTransition

The complex element **CollisionalTransition** provides:

- the physical information to fully describe any collision involving photons, atoms, ions, molecules and elementary particles such as electrons or photons, and any particle solid interaction (PSI) - reaction process type - reactants - intermediate states if known - threshold if applicable
- the numerical data, either in a tabulated form or described by a fit function, or both, in *DataSet* elements.

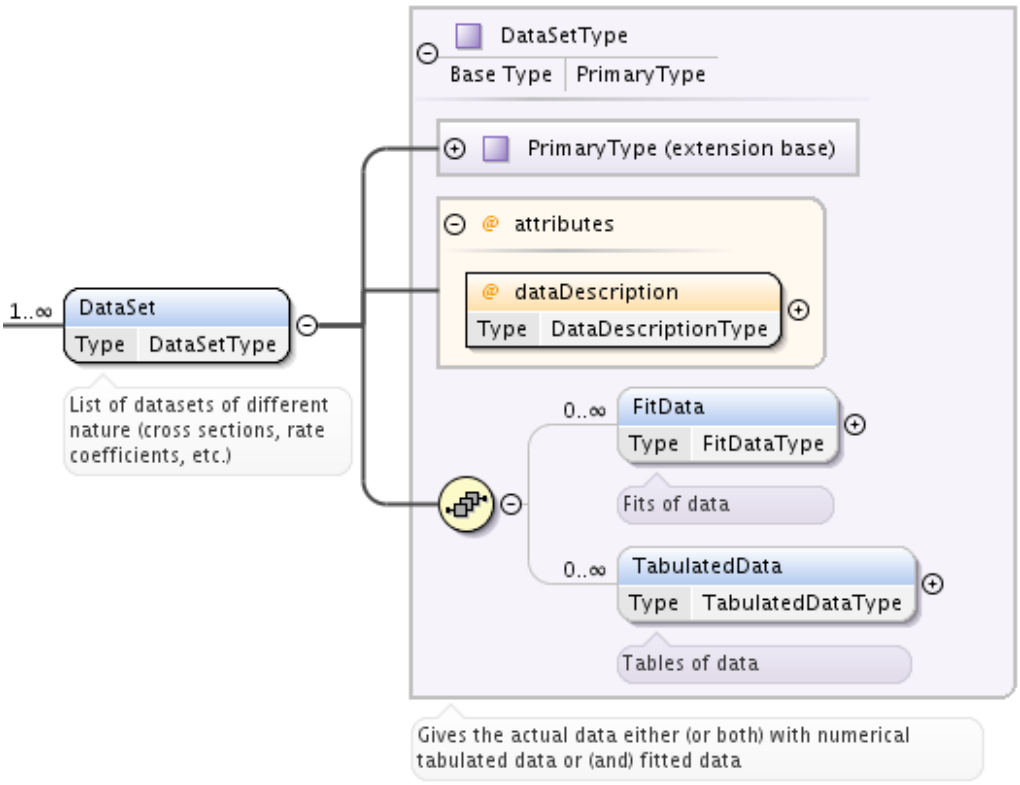


Element **CollisionalTransition**, defined by type **CollisionalTransitionType**, extends *PrimaryType*, adding 6 primary elements:

- optional element **ProcessClass** of type **ProcessClassType**, that describes the collision process or particle surface interaction. It is a combination of three optional elements:
  - **UserDefinition** for arbitrary string description of process,
  - list of **Code** elements, defined by **CodeType**, identifying the process using XSAMS-specific 4-letter *XSAMS Process Codes*. Taking examples for common processes, excitation has one value *exci* and dissociative recombination has two values *diss* and *reco*.
  - **IAEACode** element, defined by **IAEACodeType**, identifies the process using *IAEA DCN codes* from [IAEAProc]. The initial purpose of IAEA codes was the development of search engines for atomic and molecular data. Processes, represented with a three-letter code, are classified in four categories: electron collisions, photon collisions, heavy particle collisions and particle surface interactions
- mandatory list of **Reactant** elements, defined by *SpeciesStateRefType*, having two or more elements,
- optional list of **IntermediateState** elements, defined by *SpeciesStateRefType*.
- optional list of **Product** elements, defined by *SpeciesStateRefType*. Reactants, intermediate states and products are defined by references to species or to states, or both of them, using **SpeciesRef** (*StateRefType*) and **StateRef** (*SpeciesRefType*) elements. See *Species* chapter for a full description of all possible states.
- optional **Threshold** element, of *DataType*, giving the reaction threshold, i.e. the minimum energy required to initiate a reaction. This kind of data is needed in using data such as rate coefficients.

- mandatory **DataSets** element, that provides numerical data for a specific reaction. It is an unbounded sequence of **DataSet** elements, therefore different datasets may be provided for the same reaction. Datasets differ at this level by their **dataDescription** attribute, such as rate coefficients or cross sections for collisions, reflection coefficients or mean penetration depth for PSI data, etc.

14.2 DataSet



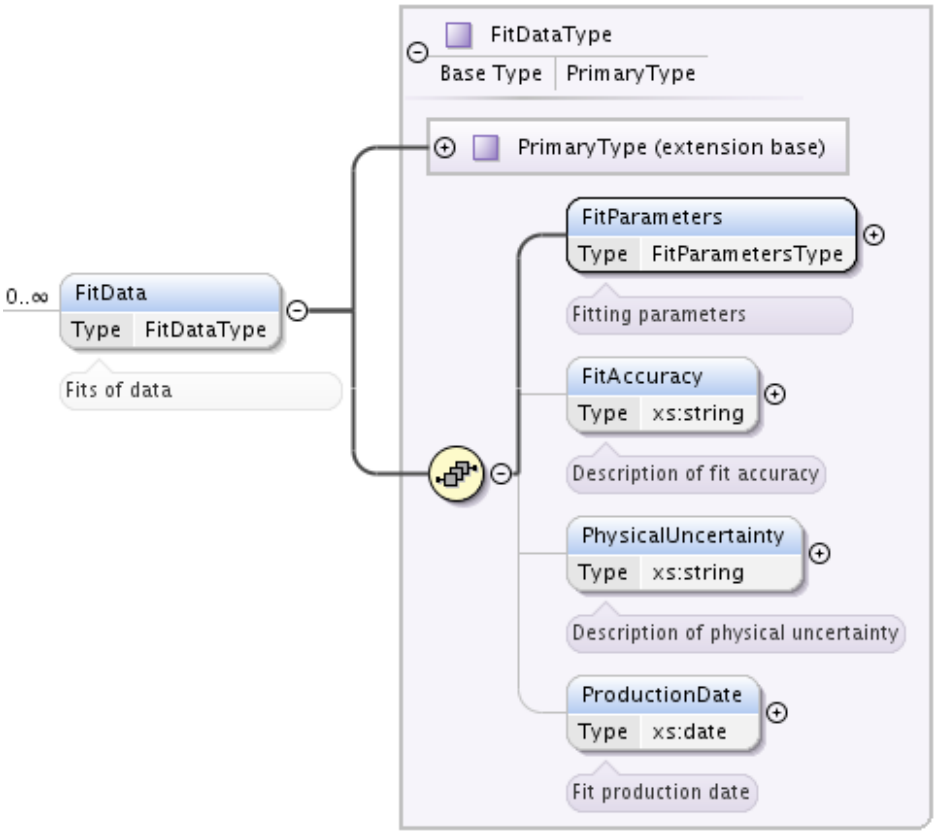
Element **DataSet**, defined by type **DataSetType**, provides numerical data as tabulated data (**TabulatedData** element) or as the parameters values and validity limits of a fit function (**FitData** element). The fit function is defined in the *Functions* branch. Extending *PrimaryType*, it adds following attributes and elements:

- string attribute **dataDescription**, that specifies the type of data. Allowed values are:
  - crossSection
  - collisionStrength
  - rateCoefficient
  - probability
  - effectiveCollisionStrength
  - sputteringYield
  - sputteredEnergyCoefficient
  - particleReflectionCoefficient
  - energyReflectionCoefficient
  - meanPenetrationDepth
- optional list of *FitData* elements

- optional list of *TabulatedData* elements

Data are therefore fit data or tabulated data or both. Data may originate from different sources or be produced using different methods, but must refer to the same data quantity, defined in the **dataDescription**. For different quantities, use separate **DataSet** elements.

14.3 FitData



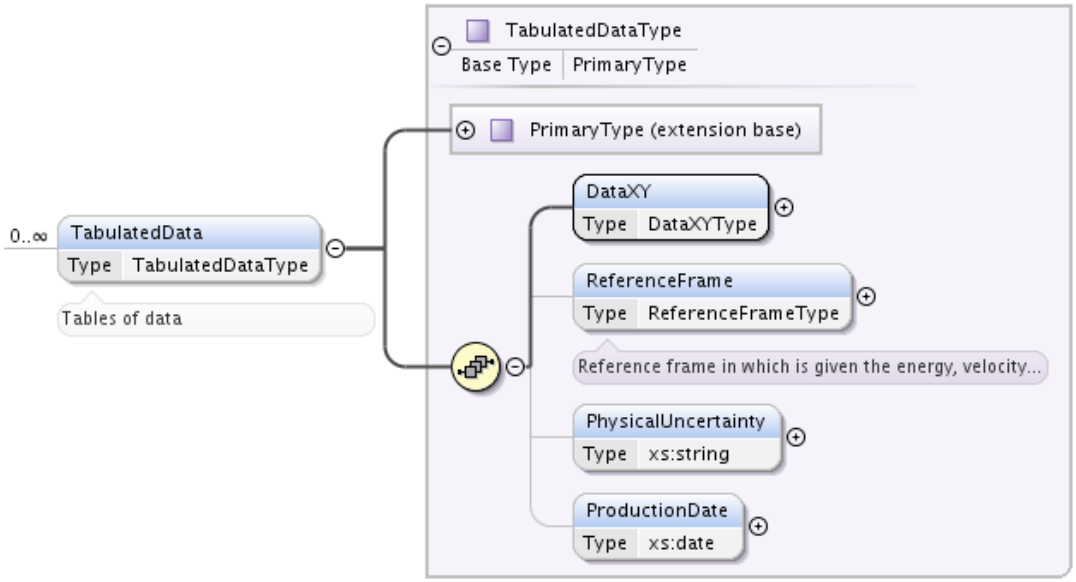
The **FitData** element of type **FitDataType** gives all data necessary to calculate the numerical output using a fit function. Function reference should be given in the **functionRef** attribute of **FitParameters** element. Validity limits of the arguments (x1, x2...) and the fit parameter values are also provided there.

To complete the description, the fit accuracy, the physical uncertainty and the data production date may be reported.

Extending *PrimaryType*, **FitData** defines following elements:

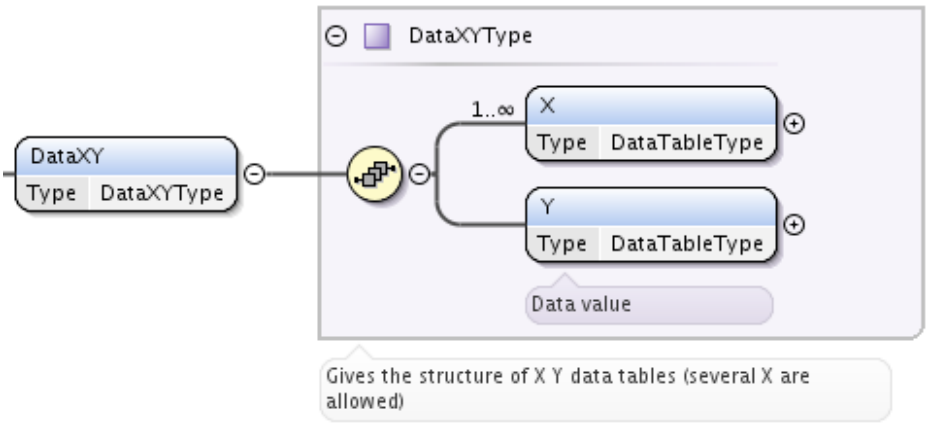
- mandatory **FitParameters** element, defined by *FitParametersType*, fully describing the set of parameters needed to calculate function value;
- optional **FitAccuracy** string element, filled in free format,
- optional **PhysicalUncertainty** string element, filled in free format,
- optional **ProductionDate** element of type **xs:date**, that may keep the production date of the fit.

14.4 TabulatedData



This element of type **TabulatedDataType**, extension of the *PrimaryType*, provides the necessary information to describe the numerical data in a tabulated form. It includes the elements *DataXY*, *ReferenceFrame*, **PhysicalUncertainty** and **ProductionDate** (see description of *FitData* element for definitions).

14.4.1 DataXY



The **DataXY** element of type **DataXYType** provides the numerical data in a tabulated form. It has two elements **X** and **Y** of same *DataTableType*. The element **X** may be specified multiple times to allow description of multi-dimensional data, as for example: $y = f(x_1, x_2, x_3)$.

Each numerical list from any **X** or **Y** element must have the same number of values. This is necessary to transfer the numerical data in a consistant way. This way, the n'th value of the **DataList** element of the **Y** element has as argument values the n'th value of the **DataList** element from each **X** elements. It is important to note that no missing values are possible within the **DataList** element, as each **Y** value has always a defined set of **X** values.

The following table:

Differential cross-sections in $10^{-16} cm^2$

$x1(eV)\|x2(deg)$	**0**	**20**	**40**
1.	.1	.2	.3
2.	.4	.5	.6
3.	.7		.9

produces as output:

```
<DataXY>
        <X unit='deg'>
                <Datalist>0 20 40 0 20 40 0 40</Datalist>
        </X>
        <X unit='eV'>
                <Datalist>1. 1. 1. 2. 2. 2. 3. 3.</Datalist>
        </X>
        <Y unit='cm2'>
                <Datalist>.1e-16 .2e-16 .3e-16 .4e-16 .5e-16 .6e-16 .7e-16 .9e-16</Datalist>
        </Y>
</DataXY>
```

14.4.2 ReferenceFrame

For numerical data depending on such parameters as energy or velocity, it is important to know in which frame this dependence is calculated. The possible values of the element **ReferenceFrame** are:

- **CenterOfMass**
- **LaboratoryFrame**
- **TargetFrame**

CHAPTER

FIFTEEN

LIST OF ATOMIC ELEMENTS

15.1 Introduction

List of Elements extracted from the IUAPAC Commission on Atomic Weights and Isotopic Abundances. (http://www.chem.qmul.ac.uk/iupac/). The atomic weights (called as well Atomic Mass) have been removed from the original table as they correspond to the average mass of the atoms of an element. This is a weighted average of the naturally-occurring isotopes and this is not relevant for astrophysical applications.

The quantities described in this table are:

- the atomic number **Z**, corresponding to the number of protons found in the nucleus of an atom. The atomic number uniquely identifies a chemical element.
- the symbol
- the name
- the isotopic masses for light atomic species
- nuclear spins for light atomic species

15.2 Isotopic atomic species

Note that isotopic atomic species of a given element is uniquely identified by the mass number **A**, also called atomic mass number, which is the number of nucleons (protons and neutrons) in an atomic nucleus. The full isotope symbol would also have the atomic number **Z** as a subscript to the left of the element symbol directly below the mass number: ${}^{12}_{6}\mathrm{C}$.

For the special case of Hydrogen, isotopic atomic species have specific symbols and names:

- 1 neutron and 1 proton: Symbol='D' and Name='Deuterium'
- 2 neutrons and 1 proton: Symbol='T' and Name='Tritium'

15.3 List of Elements in Atomic Number Order

The names and symbols for elements 112-118 are under review. The temporary system [IUPACAt] recommended by J Chatt, Pure Appl. Chem., **51**, 381-384 (1979) is used above.

The names of elements 101-109 were agreed in 1997 (see Pure Appl. Chem., **69**, 2471-2473, 1997), for element 110 in 2003 (see Pure Appl. Chem., **75**, 1613-1615, 2003) and for element 111 in 2004 (see Pure Appl. Chem., **76**, 2101-2103, 2004).

Atomic Number	Symbol	Name	Atomic Mass Number	Nuclear Spin

Continued on next page

Table 15.1 – continued from previous page

1	H	Hydrogen	1	0.5
	D	Deuterium	2	1
	T	Tritium	3	0.5
2	He	Helium	4	0
			3	0.5
3	Li	Lithium	7	1.5
			6	1
4	Be	Beryllium	9	1.5
5	B	Boron	11	1.5
			10	3
6	C	Carbon	12	0
			13	0.5
7	N	Nitrogen	14	1
			15	0.5
8	O	Oxygen	16	0
			17	2.5
			18	0
9	F	Fluorine	19	0.5
10	Ne	Neon	20	0
			21	1.5
			22	0
11	Na	Sodium	23	1.5
12	Mg	Magnesium	24	0
			25	2.5
			26	0
13	Al	Aluminium	27	2.5
14	Si	Silicon	28	0
			29	0.5
			30	0
15	P	Phosphorus	31	0.5
16	S	Sulfur	32	0
			33	1.5
			34	0
17	Cl	Chlorine	35	1.5
			37	1.5
18	Ar	Argon	40	0
			36	0
			38	0
19	K	Potassium	39	1.5
			40	4
			41	1.5
20	Ca	Calcium	40	0
			42	0
			43	3.5
			44	0
			46	0
			48	0
21	Sc	Scandium	45	3.5
22	Ti	Titanium	46	0
			47	2.5
			48	0
			49	3.5
			50	0
23	V	Vanadium	51	3.5
			50	6
24	Cr	Chromium	52	0

Continued on next page

Table 15.1 – continued from previous page

			50	0
			53	1.5
			54	0.5
25	Mn	Manganese	55	2.5
26	Fe	Iron	56	0
			54	0
			57	0.5
			58	0
27	Co	Cobalt	59	3.5
28	Ni	Nickel	58	0
29	Cu	Copper	63	1.5
30	Zn	Zinc	64	0
31	Ga	Gallium	69	1.5
32	Ge	Germanium	74	0
33	As	Arsenic	75	1.5
34	Se	Selenium	80	0
35	Br	Bromine	79	1.5
			81	1.5
36	Kr	Krypton	84	0
37	Rb	Rubidium	85	2.5
			87	1.5
38	Sr	Strontium	88	0
39	Y	Yttrium	89	0.5
40	Zr	Zirconium	90	0
41	Nb	Niobium	93	4.5
42	Mo	Molybdenum	98	0
			92	0
			94	0
			95	2.5
			96	0
			100	0
43	Tc	Technetium	99	2.5
44	Ru	Ruthenium	102	0
			99	2.5
			100	0
			101	2.5
			104	0
45	Rh	Rhodium	103	0.5
46	Pd	Palladium	106	0
			104	0
			105	2.5
			108	0
47	Ag	Silver	107	0.5
			109	0.5
48	Cd	Cadmium	114	0
			110	0
			111	0.5
			112	0
			113	0.5
			116	0
49	In	Indium	115	4.5
			113	4.5
50	Sn	Tin	120	0
51	Sb	Antimony	121	2.5
			123	3.5
52	Te	Tellurium	130	0

Continued on next page

Table 15.1 – continued from previous page

53	I	Iodine	127	2.5
54	Xe	Xenon	132	0
			129	0.5
			131	1.5
55	Cs	Caesium	133	3.5
56	Ba	Barium	138	0
57	La	Lanthanum	139	3.5
58	Ce	Cerium	140	0
59	Pr	Praseodymium	141	2.5
60	Nd	Neodymium	142	0
			143	3.5
			144	0
			145	3.5
			146	0
61	Pm	Promethium		
62	Sm	Samarium	144	0
			150	0
			152	0
			154	0
63	Eu	Europium	151	2.5
			153	2.5
64	Gd	Gadolinium	154	0
			155	1.5
			156	0
			157	0
			158	0
			160	0
65	Tb	Terbium	159	1.5
66	Dy	Dysprosium	156	0
			158	0
			160	0
			161	2.5
			162	0
			163	2.5
			164	0
67	Ho	Holmium	165	3.5
68	Er	Erbium	162	0
			164	0
			166	0
			167	3.5
			168	0
			170	0
69	Tm	Thulium	169	0.5
70	Yb	Ytterbium	168	0
			170	0
			171	0.5
			172	0
			173	2.5
			174	0
			176	0
71	Lu	Lutetium	175	3.5
72	Hf	Hafnium	176	0
			177	3.5
			178	0
			179	4.5
			180	0

Continued on next page

Table 15.1 – continued from previous page

73	Ta	Tantalum	181	3.5
74	W	Tungsten	180	0
			182	0
			184	0
			186	0
75	Re	Rhenium	185	2.5
76	Os	Osmium	187	0.5
			188	0
			189	1.5
			190	0
			192	0
77	Ir	Iridium	191	1.5
			193	1.5
78	Pt	Platinum	192	0
			194	0
			195	0.5
			196	0
			198	0
79	Au	Gold	197	1.5
80	Hg	Mercury	196	0
			198	0
			199	0.5
			200	0
			201	1.5
			202	0
			204	0
81	Tl	Thallium	203	0.5
			205	0.5
82	Pb	Lead	206	0
			207	0.5
			208	0
83	Bi	Bismuth	209	4.5
84	Po	Polonium		
85	At	Astatine		
86	Rn	Radon		
87	Fr	Francium		
88	Ra	Radium		
89	Ac	Actinium		
90	Th	Thorium	232	0
91	Pa	Protactinium		
92	U	Uranium	235	3.5
			238	0
93	Np	Neptunium		
94	Pu	Plutonium		
95	Am	Americium		
96	Cm	Curium		
97	Bk	Berkelium		
98	Cf	Californium		
99	Es	Einsteinium		
100	Fm	Fermium		
101	Md	Mendelevium		
102	No	Nobelium		
103	Lr	Lawrencium		
104	Rf	Rutherfordium		
105	Db	Dubnium		
106	Sg	Seaborgium		

Continued on next page

Table 15.1 – continued from previous page

107	Bh	Bohrium		
108	Hs	Hassium		
109	Mt	Meitnerium		
110	Ds	Darmstadtium		
111	Rg	Roentgenium		
112	Uub	Ununbium		
113	Uut	Ununtrium		
114	Uuq	Ununquadium		
115	Uup	Ununpentium		
116	Uuh	Ununhexium		
118	Uuo	Ununoctium		

CHAPTER

SIXTEEN

XSAMS PROCESS CODES

Version 0.2, October 2008

16.1 Introduction

In order to facilitate searching and sorting XSAMS files or blocks of data within a file, a classification scheme is defined to provide standard definitions for the fundamental process that is being described (e.g., the reflection of a particle from a surface or the excitation of an atomic state).

The process codes are "fundamental" in that they do not define the reactions to a great degree of detail that is carried out by specific elements of the overall schema. Rather, they provide a high level identification that can be used to aggregate similar data or provide a simple search point.

16.2 Process Codes

The following table defines the fundamental reactant codes. The process codes embody a high level description and a much more complete definition of the process that data pertain to is contained in the elements of the full schema. Processes are split into two categories: collisions (electron and heavy particle collisions) and particle surface interactions (PSI). Within each category, the codes are intended for use with any reactant. For example, a single process code is used to describe elastic scattering whatever is the projectile (electron, photon, atom, ion or molecule).

This approach has been taken not only for economy but because an exhaustive list of processes involving elementary particles, atoms, molecules, and solids is not likely possible. Even if it were, using it would be cumbersome in that some non-intuitive coding would be necessary, for example, to encode inverse bremstrahlung, sublimation of water from a dust grain, a production of an atomic inner shell vacancy with a subsequent Coster-Kronig decay, etc.

Multiple process codes can be given in order to build more complex descriptions from the fundamental processes, e.g., charge transfer + ionization in ion-atom collisions could be described by the code for ionization (liberation of an electron from the target or projectile to the continuum) and the code for charge transfer (the transfer of an electron from one collision partner to another). This combining of individual codes avoids the need for a code for all possible combinations of processes that are at least relatively common, e.g., dissociative recombination simply has the codes for dissociation and for recombination. Other examples of combinations of codes are given in the final table.

16.3 Atomic and Molecular Collisions

Code	Name	Description
phem	Photon emission	Emission of a photon or photons from a reactant (e.g.,. atom, molecule, surface), fluorescence
phab	Photon absorption	Absorption of a photon or photons from a reactant (e.g., atom, molecule, surface)
phsc	Photon scattering	Scattering of a photon or photons by a reactant
elas	Elastic scattering	Scattering of one reactant from another without change of state or energy, including related processes such as momentum transfer
inel	Inelastic scattering	Scattering of one reactant from another with change of state or energy. This code is provided in case none of the other specific inelastic codes are applicable or appropriate (e.g., energy or spin transfer reactions, projectile energy loss)
exci	Excitation	Excitation from a lower to higher state of any fundamental reactant, e.g., electron-impact excitation of an atom, photoexcitation of a molecule to a higher ro-vibrational state
deex	De-excitation	Induced or spontaneous transition from a higher state to a lower state, e.g., vibrational de-excitation in atom-diatom scattering
ioni	Ionization	Removal of an electron from any reactant
tran	Charge transfer	Transfer of an electron from one “center” (atomic ion, atom, molecule, etc.) to another
exch	Electron exchange	The exchange of an electron with another electron (most commonly in electron-impact processes) reco & Recombination & Capture of an electron by an atomic or molecular ion, e.g., in dissociative recombination, dielectronic recombination, or radiative recombination
elat	Electron attachment	The formation of a negative ion by electron attachment
eldt	Electron detachment	The removal of the weakly bound electron of a negative ion by photon impact or collision with another particle such as an electron or surface
asso	Association	Association of two (or more) reactants, typically neutrals, collisionally, or radiatively
diss	Dissociation	The splitting of two (or more) reactants e.g., via electron-impact of a molecule, photodissociation, molecular break-up on a surface
intr	Interchange	The exchange of a heavy particle (atom, ion) in a reaction, e.g., $D + H2 \rightarrow DH + H$
chem	Chemical reaction	The exchange of atoms or groups of atoms in chemical reactions, e.g., $C + 2O \rightarrow CO2 + heat$

16.4 Particle Solid Interactions

Code	Name	Description
sure	Reflection	The elastic or inelastic reflection of a reactant from a surface
suem	Emission or erosion	Any form of erosion of a surface, e.g., physical or chemical sputtering, etching, sublimation, emission of particle or macroscopic pieces, desorption, secondary electron emission
sudp	Deposition	Absorption of particles by a surface, sticking, surface implantation
such	Change	Change of the composition or properties of a surface induced by any reactant
sope	Penetration	The penetration of a reactant into a solid, characterized by the change of energy, e.g., stopping, straggling, energy loss, range, charge state equilibrium, or change of structure, e.g., trapping, diffusion, deep implantation

16.5 Combination of Processes

The following table gives examples of the use of the fundamental process codes to describe more complex but still common processes. Some simply fall within the broad scope of one of the fundamental codes and others can be described by use of multiple codes.

Common Name	Codes	Description
Penning ionization	asso + ioni	Association of atoms with ionization
Transfer ionization	tran + ioni	Charge transfer between and ion and an atom, for example, with ionization
Transfer excitation	tran + exec	Charge transfer with excitation
Stripping	ioni	Ionization of the projectile in a collision of an ion or atom with an atom, molecule, or solid
Dissociative recombination	diss + reco	
Dielectronic recombination	reco	
Auger ionization	exci + ioni	
Spin-flip photoionization	inel ioni	
Three-body recombination	reco	
Superelastic scattering	inel	
Surface catalysis	sure + chem	
Stark shift		
Line broadening		
Bremsstrahlung		
Compton scattering		

CHAPTER

SEVENTEEN

IAEA DCN CODES

17.1 Structure and Spectra

CODE	SUBCATEGORY
SGN	General
SLS	Line Broadening, Shapes and Shifts
SIA	Interatomic Potentials
SPM	Polarizabilities, Electric Moments
SEW	Energy Levels and Wavelengths
STP	Transition Probabilities and Oscillator Strengths
SSM	Potential Curves and Structure of Molecules
SDP	Dynamic Polarizability
SIR	Infrared Spectra
SVS	Visible Spectra
SUV	UV/VUV/XUV Spectra
SXR	X-Ray Spectra
SRS	Rotational Spectra
SVB	Vibrational Spectra
SAI	Autoionization
SAD	Autodetachment
SDS	Autodissociation
SMM	Magnetic Moments
SHF	Hyperfine Structure
SIE	Isoelectronic Sequences
SFT	Forbidden Transitions
SQE	QED Effects
SRP	Relaxation Processes
SIP	Ionization Potentials
SRY	Rydberg States

17.2 Electron-Heavy-Particle Interactions

CODE	SUBCATEGORY	PROCESS
EGN	General	
EAS	Angular Scattering	
EBS	Bremsstrahlung	$e + A \rightarrow e + A + h\nu$
EDX	Deexcitation	$e + A^* \rightarrow e + A$
EEL	Elastic Scattering	$e + A \rightarrow e + A$
ELB	Line Broadening, Shapes and Shifts	
ETS	Total Scattering	
EDT	Detachment	$e + A^- \rightarrow A + 2e$

Continued on next page

Table 17.1 – continued from previous page

EFL	Fluorescence	
EEX	Excitation	$e + A \rightarrow e + A^*$
EEX	Change of Excitation	
EIN	Ionization	$e + A \rightarrow e + A^+ + e$
EMI	Multiple Ionization	$e + A \rightarrow A^{+n} + (n+1)e$
ENI	Negative Ion Formation	$e + A \rightarrow A^-$
EMT	Momentum Transfer	
EMT	Transport CS's (momentum,...)	
EUP	Unknown Products	
EDP	Depolarization, Change of Polarization	
EIP	Creation of an ion pair (positive-negative)	$e + AB^+ \rightarrow A^- + B^-$
ERC	Recombination (general)	$A^{+q} + e \rightarrow A^{+(q-1)}$
ERR	Radiative Recombination	$e + A^+ \rightarrow A + h\nu$
ERD	Dielectronic Recombination	$e + A^+ \rightarrow A^{**} \rightarrow A^* + h\nu + e$
ERT	3-body Recombination	$e + e + A^+ \rightarrow A + e$
ERO	e-i-o Recombination	$e + A^+ + B \rightarrow A + B$
EDC	Dielectronic Capture	$e + A^+ \rightarrow A^{**}$
EDS	Dissociation	$e + AB \rightarrow e + A + B$
EDR	Dissociative Recombination	$e + AB^+ \rightarrow A + B$
EDA	Dissociative Attachment	$e + AB \rightarrow A + B^-$
EDE	Dissociative Excitation	$e + AB \rightarrow A^* + B + e$
EDI	Dissociative Ionization	$e + AB \rightarrow A^+ + B + 2e$

17.3 Photon-Particle and Field-Particle Interactions

CODE	SUBCATEGORY	PROCESS
PGN	General	
PTS	Total Absorption, Scattering	
PDS	Photodissociation	$h\nu + AB \rightarrow A + B$
PES	Elastic Scattering	$h\nu + A \rightarrow h \rightarrow +A$
PMA	Multiphoton Absorption (excitation and ionization)	$nh\nu + A \rightarrow A^*(A^+)$
PDT	Photodetachment	$A + B \rightarrow AB + h\nu$
PFL	Fluorescence	
PEX	Photoexcitation	$h\nu + A \rightarrow A^*$
PIN	Photoionization	$h\nu + A \rightarrow A^+ + e$
PFF	Free-Free Absorption or Inverse Bremsstrahlung	$h\nu + e + A \rightarrow e + A$
PEA	Effective Absorption, Total Diffusion	
PTA	True Absorption	
PAD	Angular Diffusion (scattering)	
PED	Elastic Diffusion (Thomson, Rayleigh)	
PNL	Non-linear Effects	
PLE	Emission of Line	
PZE	Zeeman Effect	
PSE	Stark Effect	
PGF	General Electromagnetic Field	
PTF	Interaction with Time-Varying Fields	

17.4 Heavy-Particle Interactions

CODE	SUBCATEGORY	PROCESS
HGN	General	
HAS	Association	$A + B \rightarrow AB$
HLB	Line Broadening, Shapes and Shifts	
HDS	Dissociation	$A + BC \rightarrow A + B + C$
HDX	Deexcitation	$A^* + B \rightarrow A + B$
HES	Elastic Scattering	$A + B \rightarrow A + B$
HCX	Charge Transfer	$A^+ + B \rightarrow A + B^+, A^- + B \rightarrow A + B^-$
HUP	Unknown Products	
HAS	Angular Scattering	
HIR	Interchange Reactions	$A + BC \rightarrow AB + C$
HEL	Inelastic Energy Losses	
HET	Energy Transfer	
HIP	Interaction Potentials	
HRC	Recombination	
HTS	Total Scattering	
HDT	Detachment	$A + B^- \rightarrow A + B + e$
HFL	Fluorescence	
HEX	Excitation	$A + B \rightarrow A^* + B$
HIN	Ionization	$A + B \rightarrow A + B^+ + e$
HPN	Penning Ionization	$A^* + B \rightarrow A + B^+ + e$
HST	Stripping (of projectile)	$A + B \rightarrow A^+ + B + e$
HAT	Attenuation	
HXT	Excitation Transfer	$A^* + B \rightarrow A + B*$
HAI	Associative Interchange Reactions	
HDI	Dissociative Interchange Reactions	
HDC	Dissociative Charge Transfer	$A^+ + BC \rightarrow A + B^+ + C$
HMN	Mutual Ion-Ion Neutralization	$A^+ + B^- \rightarrow A + B$

17.5 Particle-Matter Interactions

CODE	SUBCATEGORY
MGN	General
MAC	Accomodation
MAD	Adsorption
MCR	Chemical Reactions
MDE	Desorption
MRE	Reemission
MRF	Reflection
MSD	Surface Damage
MSE	Secondary Electron Emission
MIR	Radiation Induced by Particle Impact on Surfaces
MNE	Neutralization, Ionization, Dissociation
MSP	Sputtering
MRS	Radiation-Enhanced Sublimation
MTD	Trapping, Detrapping
MPE	Photoelectric Ejection of Electrons
MEL	Energy Loss and Stopping Power
MPR	Particle Range
MMS	Multiple Scattering
MCP	Charge State Population
MEP	Excited State Population
MRH	Reflection of Heavy Particles from Surfaces
MRL	Reflection of Electrons from Surfaces

17.6 Data Compilations

CODE	SUBCATEGORY
DGN	General
DEH	Electron-Heavy Particle Interactions
DHH	Heavy Particle-Heavy Particle Interactions
DPF	Photon-Particle and Field-Particle Interactions
DSS	Structure and Spectra
DTP	Transport Properties
DPM	Particle-Matter Interactions

17.7 Bibliography

CODE	SUBCATEGORY
BGN	General
BEH	Electron-Heavy Particle Interactions
BHH	Heavy Particle-Heavy Particle Interactions
BPF	Photon-Particle and Field-Particle Interactions
BSS	Structure and Spectra
BTP	Transport Properties
BPM	Particle-Matter Interactions

CHAPTER

EIGHTEEN

LINESHAPES DICTIONARY

Possible broadening and shifting names, lineshapes, parameters are listed below. The list is incomplete, it will be expanded before the release and extended later based on user requests and feedback.

18.1 Broadening names

- **pressure** for collisional broadening processes
- **doppler** for Doppler broadening
- **instrument** for instrument-specific broadening
- **natural** for line broadening caused by finite lifetime of initial and final states. Usually, Lorentzian line profile should be used.

18.2 Shifting names

- **pressure** for linear pressure shifting in hi-res spectroscopy

18.3 Lineshape names and parameters

- Lorentzian
  - gammaL
- Voigt
  - gamma
  - sigma

to be continued...

BIBLIOGRAPHY

[Hanisch] Hanisch, R. (Editor), Resource Metadata for the Virtual Observatory, 2002, see website: http://www.ivoa.net/Documents/.

[Conventions1] Schutte, C. J. H., Bertie, E., Bunker, P. R., Hougen, T., Mills, I., Watson, J. K. G. and Winnewisser, B. P. (1997)

Notations and conventions in molecular spectroscopy: part 1. General spectroscopic notation.

Pure and Applied Chemistry, 69 (8). pp. 1633-1639. ISSN 0033-4545

[Conventions2] Schutte, C. J. H., Bertie, J. E., Bunker, P. R., Hougen , J. T., Mills, I., Watson, J. K. G. and Winnewisser, B. P. (1997).

Notations and conventions in molecular spectroscopy: part 2. Symmetry notation.

Pure and Applied Chemistry, 69 (8). pp. 1641-1649. ISSN 0033-4545

[Conventions3] Bunker, P. R., Schutte, C. J. H., Hougen, J. T., Mills, I., Watson, J. K. G. and Winnewisser, B. P. (1997)

Notations and conventions in molecular spectroscopy: part 3. Permutation and permutation-inversion symmetry notation.

Pure and Applied Chemistry, 69 (8). pp. 1651-1657. ISSN 0033-4545

[Janev] Janev, R.K. (Editor), Summary Report of an IAEA Consultants' Meeting on Atomic Data Base and Fusion Applications Interface, Vienna, 9-13 May 1988, IAEA report INDC(NDS)-211/GA (1988).

[Lorenz] Lorenz, A. and Seamon, R.E. (Editors), Summary Report on the First Meeting of the Atomic and Molecular Data Centre Network, 9-13 May 1977, IAEA report INDC(NDS)-88/GB (1977).

[Mulliken55] Mulliken, R. 1955, J. Chem. Phys., 23, 1997

[ADAS] http://www.adas.ac.uk/

[XSAMS] IAEA version of XSAMS: http://www-amdis.iaea.org/xsams/

[VAMDC-XSAMS-PDF] PDF of VAMDC-XSAMS reference guide: http://www.vamdc.org/documents/vamdc-xsams-guide_v0.2.pdf

[VAMDC-XSAMS-SCHEMA] Tarball of VAMDC-XSAMS schema files (and examples): http://www.vamdc.org/downloads/vamdc-xsams-v0.2.tar.bz2

[VAMDC-XSAMS-DOC] http://www.vamdc.org/documents/vamdc-xsams-doc-v0.2/ Autogenerated HTML documentation for XSAMS.

[VAMDC-XSAMS-LOG] Change Log Document : http://www.vamdc.org/documents/vamdc-xsams-changelog_v0.2.pdf

[case-by-case] Reference Document for case-by-case description of molecular states: http://www.vamdc.org/documents/cbc_v0.2.pdf

[UnitsML] http://unitsml.nist.gov/

[DOI] http://www.doi.org/

[IUPAC] http://www.chem.qmul.ac.uk/iupac/

[IUPACAt] Recommendations for the Naming of Elements of Atomic Numbers Greater than 100, http://www.chem.qmul.ac.uk/iupac/AtWt/element.html

[InChI] http://www.iupac.org/inchi/

[CML] Chemical Markup Language, http://www.xml-cml.org/

[IAEAProc] HUMBERT Denis, RALCHENKO Yuri, KRSTIC Predrag, CLARK Robert E.H., Classification of Processes, Version 1.2, 7 October 2003. http://www-amdis.iaea.org/DCN/ProcessClassification.pdf

[IVOA] International Virtual Observatory Alliance (IVOA), http://www.ivoa.net/