Seventh Framework Programme
Exploitation of space science and exploration data

IMPEx Simulation Data Model
Version 1.0.1

Title:	**IMPEx Simulation Data Model**
Date:	May 21, 2014
Editor:	**Michel Gangloff**, Université Paul Sabatier Toulouse
Contributors:	Sébastien Hess, Ronan Modolo, Vincent Génot, Riku Jarvinen, Esa Kallio, Lasse Häkkinen, Baptiste Cecconi
Distribution:	Project

Grant Agreement # 262863

Revision History

Version	Date	Released by	Detail
0.0	21/02/12	S. Hess	
1.0	18/02/13	S. Hess	
1.0.0	31/01/14	Michel Gangloff	
1.0.1	21/05/14	Michel Gangloff	

Note: Special thanks to Sébastien Hess who has initiated this work at LATMOS.

INDEX

1 Scope of this document

The purpose of this document is to describe the Data Model for simulations developed in the framework of IMPEx, called "IMPEx Simulation Data Model". The IMPEx Simulation Data Model is an extension of SPASE (DR1), used to describe data coming from observations of the space physics domain. It uses most of the elements defined in the SPASE dictionary, with additions necessary to describe data coming from simulations.

In the following text, element names are written in italics (e.g. *DisplayOutput*).
Elements prefixed with "spase:" are elements belonging to the SPASE dictionary.

2 Reference documents

« A Space and Solar Physics Data Model from the SPASE Consortium », version 2.2.2 (Release Date: 2011-02-27)
http://www.spase-group.org/data/dictionary/spase-2_2_2.pdf

« IMPEx protocol »

3 Metadata related to simulations

3.1 Classification

Metadata within the IMPEx framework are related to three kinds of data. The first kind is the measured data themselves, whose description is not done by the IMPEx project but by the instrument/spacecraft teams. The two others, which are under the responsibility of IMPEx are related to each:

- simulation run, including the code inputs.
- output dataset and fields it contains.

Data generated by a simulation run are grouped into datasets. These datasets come from the post-processing of the simulation results. In particular, different products are generated: 2D cuts of several quantities, interpolation of time series along the trajectories of spacecraft. In all of these products there are different types of data: Magnetic field...

There are four levels of metadata related to simulation results:

- **File**
  Basic information about a data file that is part of a dataset.
- **Dataset**
  Information like:

  - What do the data represent (Magnetic Filed, Density…)
  - Under which form (3DCubes, 2DCuts,TimeSeries…)
  - Are they interpolated on the trajectory of a spacecraft?
  - Detailed contents of fields
  - Identification of the simulation run which generated the dataset
- **Simulation Run**
  Description of the run, including the inputs.
- **Simulation Model**
  Description of the model (MHD, Hybrid…).

3.2 Data mining

Simulation Data may be searched in two ways:

- **From the simulation inputs**, browsing simulation-by-simulation looking for a particular set of input parameters. When an interesting set of parameters is found, look at the catalog of post-processed outputs generated by this simulation, and select a particular output. The metadata necessary to perform the search are in particular the plasma populations, the fields (magnitude and orientation), and the physical processes simulated. Other metadata are useful not for search, but rather for the scientific use of the results (Simulation domain, time steps, boundary conditions). This is typically how users are expected to search.

- **By the simulation outputs**, selecting metadata related to a set of simulation outputs and find both the corresponding data and information about the run which generated it.

4 General description and layout

4.1 General philosophy of SPASE and layout

The highest-level elements of the SPASE and IMPEx data models are resources, corresponding to different datasets, simulation models, simulation runs, instruments, observatories, and catalogs...
Figure 1 shows the IMPEx DM root and top-level elements. All resources are SPASE's, except *Granule* which is a SPASE element with minor modifications and *DisplayOutput, NumericalOutput, SimulationModel, SimulationRun* that are IMPEx elements. A unique *ResourceID* identifies each of these resources. To insure the uniqueness of the resource ID, naming authorities exists both for SPASE and IMPEx. IMPEx IDs take the form impex://NamingAuthority/…

4.2 The schema

The IMPEx Simulation Data Model is implemented in an XSD schema in which several types of elements are merged:

- Original SPASE elements
- SPASE elements modified
- IMPEx defined elements

The IMPEx simulation Data Model uses the “spase” element as root.

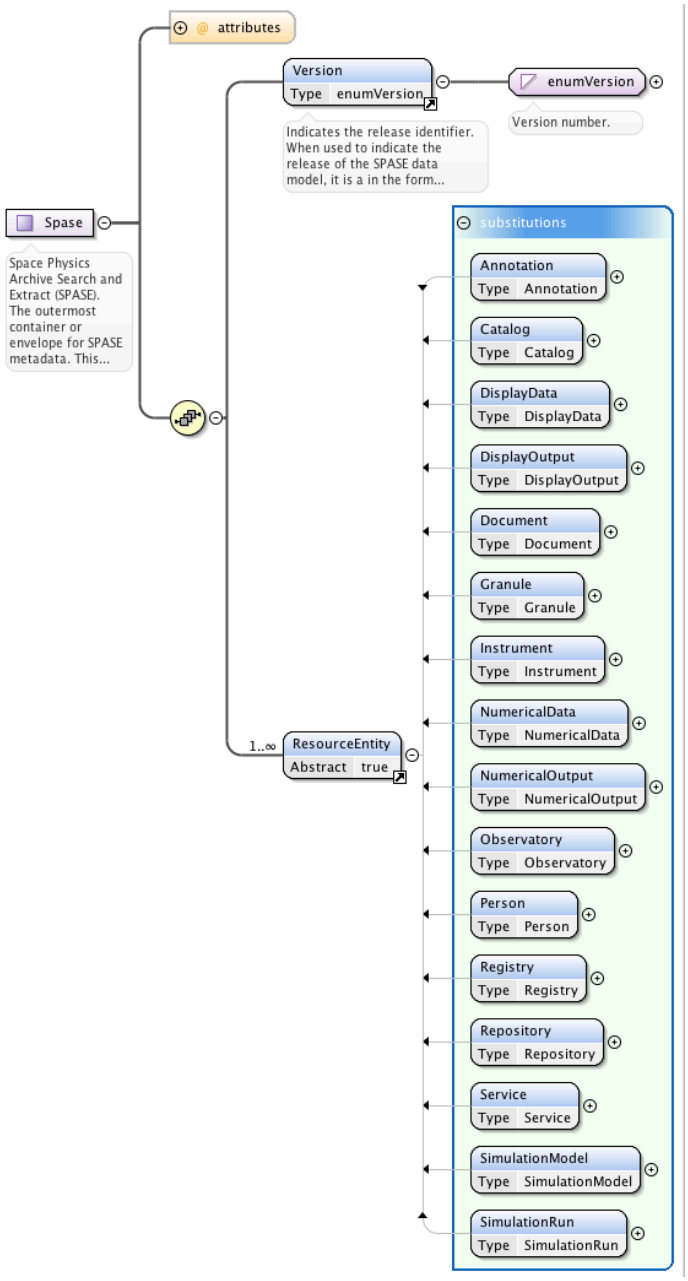


Figure 1- IMPEx Data Model Root

4.3 The Property and Parameter elements

The core elements of the IMPEx Data Model are *Property* and *Parameter*. *Parameter*

is originally a SPASE element that received some extensions. It describes the physical content of the data in a standardized way so that automated tools and end-users can identify the physical parameters attached to it.

Property is a pure IMPEx element whose purpose is to add physical information. The main conceptual difference between *Property* and *Parameter* is that a *Property* should have a defined value, whereas the value of a *Parameter* is the data themselves. Hence, *Parameter* is only used to describe data or simulation outputs, whereas *Property* is mostly used to describe simulation inputs.

4.3.1 Property

Property describes simulation inputs. This is a rather simple but versatile element, which has field to describe the physical meaning of the property, both for humans (*Description*, *Caveats*) and for machines (*PropertyQuantity* and *Qualifier*).

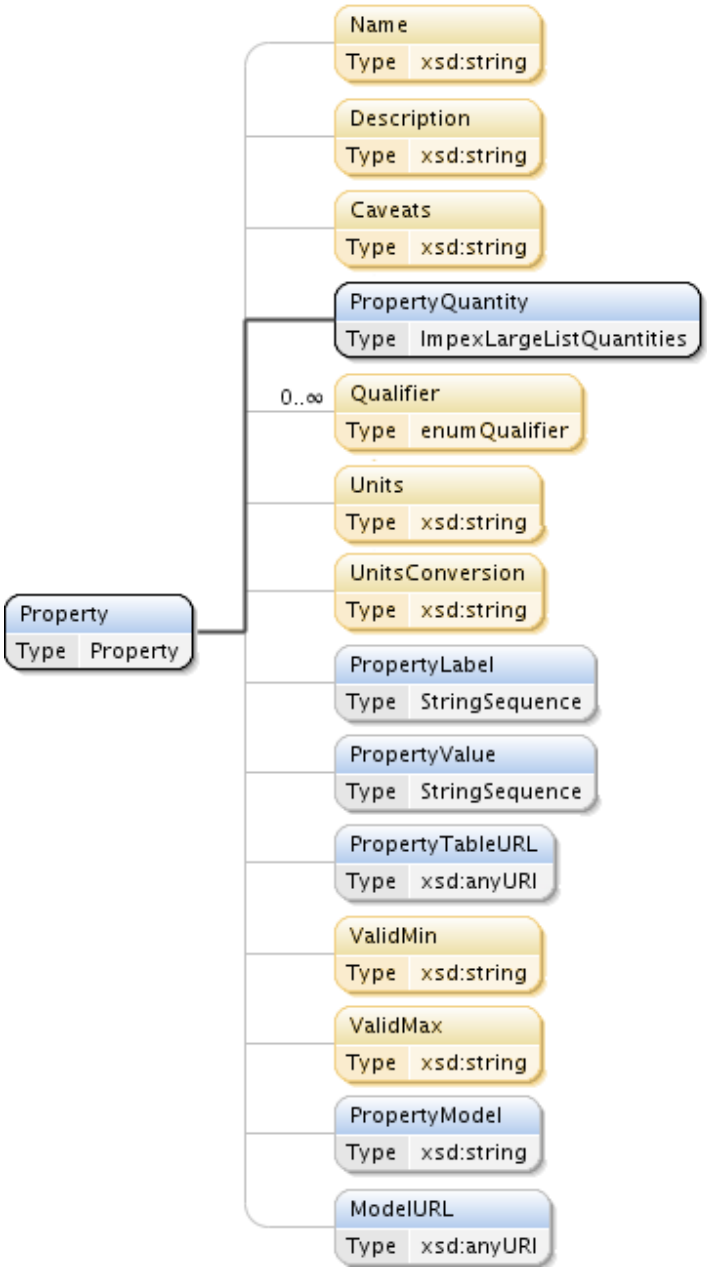


Figure 2 - IMPEx Property element

The list of terms that can be used to describe the *PropertyQuantity* is rather long and eclectic when compared to SPASE lists. This is caused by the facts that (1) simulation codes can use a large variety of inputs and (2) that *Property* elements are not expected to be widely parsed by automated tools (or at least not in a way that requires a maximum accuracy in the definitions). As described in section 3.3, IMPEx provides ways to describe more accurately inputs that are more subject to automatic parsing. The *Property* element is shown in Figure 3. Its primary purpose is to associate a value (*PropertyValue*), a table of values (*PropertyTableURL*), a range of values (*ValidMin, ValidMax*) or a model (*PropertyModel, ModelURL*) to the quantity defined by *PropertyQuantity*.
This property is that of the meta-data corresponding to the element hosting the Property.

Examples:

```
<Property>(Solar wind velocity in km/s,values between 300 and 1000 km/s)
  <Name>SW_Velocity</Name>
  <PropertyQuantity>FlowSpeed</PropertyQuantity>
  <Units>km/s</Units>
  <ValidMin>300</ValidMin>
  <ValidMax>1000</ValidMax>
</Property>

<Property> (Solar UV Activity: given by the F10.7 index)
  <Name>F10.7 index</Name>
  <PropertyQuantity>SolarUVFlux</PropertyQuantity>
  <PropertyValue>150</PropertyValue>
</Property>
```

4.3.2 Parameter

Parameter is a extension of a SPASE element, and describes data. Limited additions made by IMPEx allow giving extra information on a simulated output (as more information is generally accessible for simulated data than for observed data). A few fields are added for particle description allowing for example to differentiate two populations of the same specie – which is possible with models, but not with observations.

Parameter can also contain *Property* to give the value of some quantities relative to the parameter (same reason than for extra particle fields).

The *Parameter* element is depicted in figure 3 (elements added by IMPEx are blue):

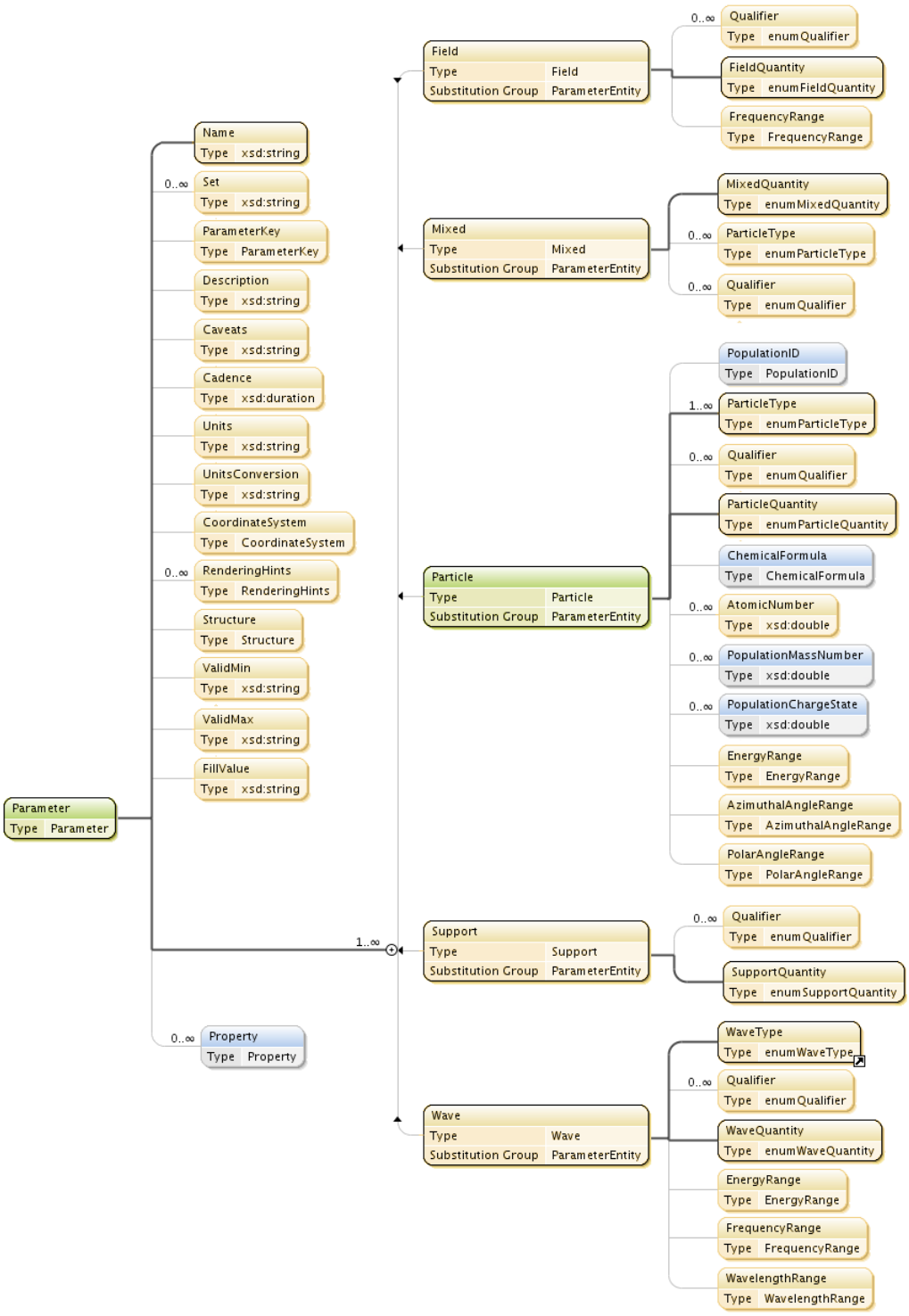


Figure 3 - Parameter element

5 The data

5.1 Resources and general layout

Two resources can be used to describe simulation data depending on their type:

- *NumericalOutput* is a set of data composed of numerical values. It is derived from spase:NumericalData
- *DisplayOutput* is a set of data composed of derived material, like figures or movies. It is derived from spase:DisplayData

The four resources (i.e. *NumericalData*, *DisplayData*, *NumericalOutput*, *DisplayOutput*) have close structures (see Figure 4, IMPEx additions in blue color). Simulation Outputs differ by different names for the *InstrumentID* and targeted Region, and by extra-information that can be brought by *SpatialDescription*, multiple *Property* and by a mandatory *SimulationProduct*. This latter element defines which kind of output is described (3DCubes, TimeSeries...).

Most of the information about the data themselves is given using multiple *Parameter* elements.

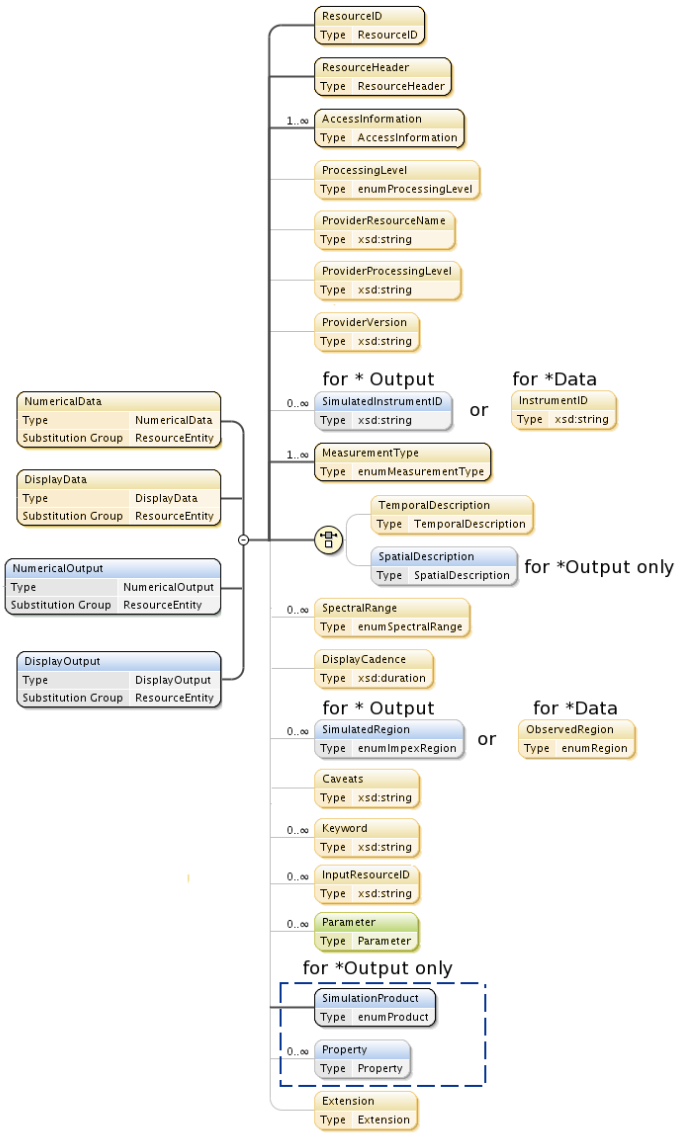


Figure 4 - Resources used to describe data

5.2 Spatial & temporal information and target definition

SimulatedRegion gives the name of the studied region. The list of available values for *SimulatedRegion* is the same list used by SPASE for ObservedRegion.
Other spatial parameters relevant for the data handling, such as the spatial extension of 3D cubes or the plane definition for 2D cuts, are in S*patialDescription*.

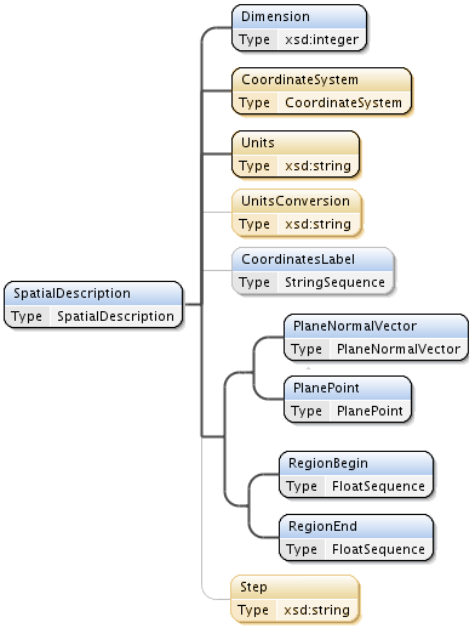


Figure 5- SpatialDescription element

5.3 Relate the data to a simulation run and an instrument

InputResourceID refers to a resource used to generate the dataset. It is the *ResourceID* of the *SimulationRun* at the origin of the dataset. To relate the data to the instrument whose outputs have been simulated, one can specify the ResourceID of the instrument in *SimulatedInstrumentID*.

5.4 Data content

The data content is described by the *Parameter* elements. There must be one such element for every parameter present in the data files. *Parameter* contains: (1) a name to the parameter (*Name*) that tools can display; (2) a reference to the parameter name in the data files (name of the column for files in VOTable format, name of the variable for files in netCDF, CDF...); and (3) description of the physical content of the parameter in a way that can be handled by automated tools (in particular through the use of the **Quantity* and *Qualifier* elements. For particle parameters, *PopulationID* identifies a given population.

5.5 Simulation and post processing information

The kind of data product that composes the dataset is given by *SimulationProduct,* which can have the following values: 3DCubes, 2DCuts, TimeSeries, SpatialSeries, Spectra, Lines.
The simulation datasets can also present supplementary *Property* elements to give the value

of simulation-related parameters.

6 Simulation runs

6.1 General layout

SimulationRun describes each individual simulation run. Its sub-elements are *TemporalDependance* to indicate whether the outputs of the simulation code are time-dependant (can be Yes or No), *SimulatedRegion* to indicate the target of the run, and *LikelihoodRating* to describe the likelihood of the physical event simulated.
The model used to perform the run is defined by the *Model* element, depicted in Figure 6.

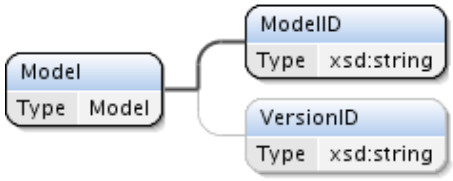


Figure 6 - Model element

The temporal domain in which the simulation run is performed is described in the *SimulationTime* element, depicted in figure 7.

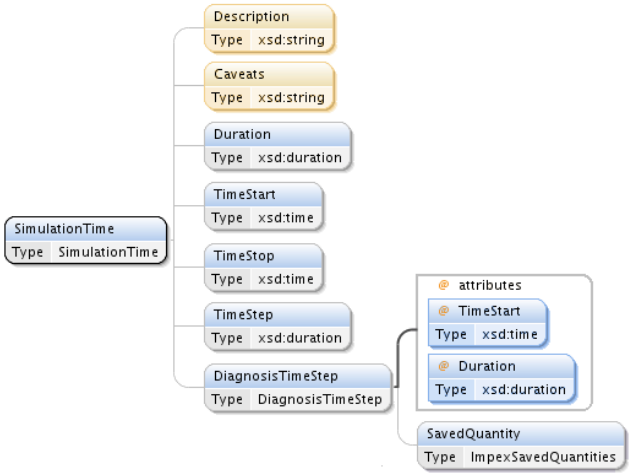


Figure 7 - SimulationTime element

The start and end times of the simulation, its duration and time step can be given here. *DiagnosisTimeStep* gives the times at which diagnoses are performed (for time dependent simulations) and identifies the physical quantities saved at each step.
The spatial domain is indicated in *SimulationDomain*. The coordinate system is given in the SPASE *CoordinateSystem* element. The number of dimensions for the description of positions, velocities and fields are given in *SpatialDimension*, *VelocityDimension*, and *FieldDimension*. The grid structure and resolution are given in *GridStructure* and *GridCellSize*. *Symmetry* allows defining if there are symmetries in the simulation.

In particular, an Axial symmetry allows for rotations of interpolated trajectories in the getDataPointValue method of the IMPEx protocol (DR2).

The boundary conditions for particles and fields can also be specified for the different walls of the simulation domain (i.e. front wall by which a flow is inserted, the opposite wall, the other wall, and the obstacle).

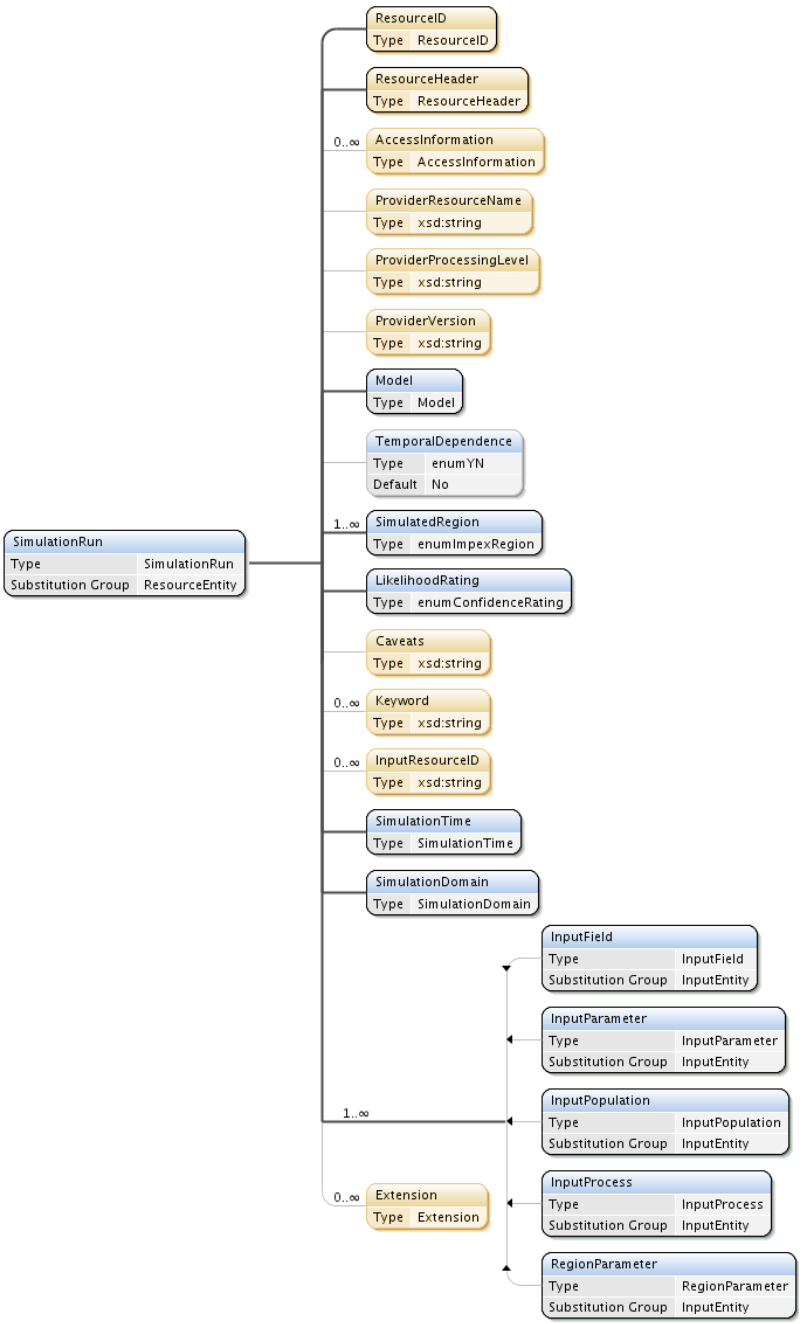


Figure 8 - SimulationRun element

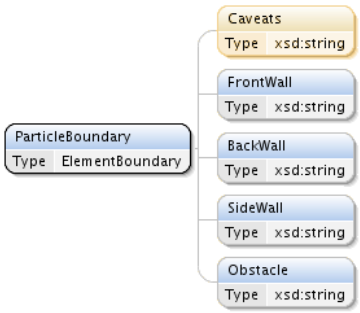


Figure 9 - ParticleBoundary element

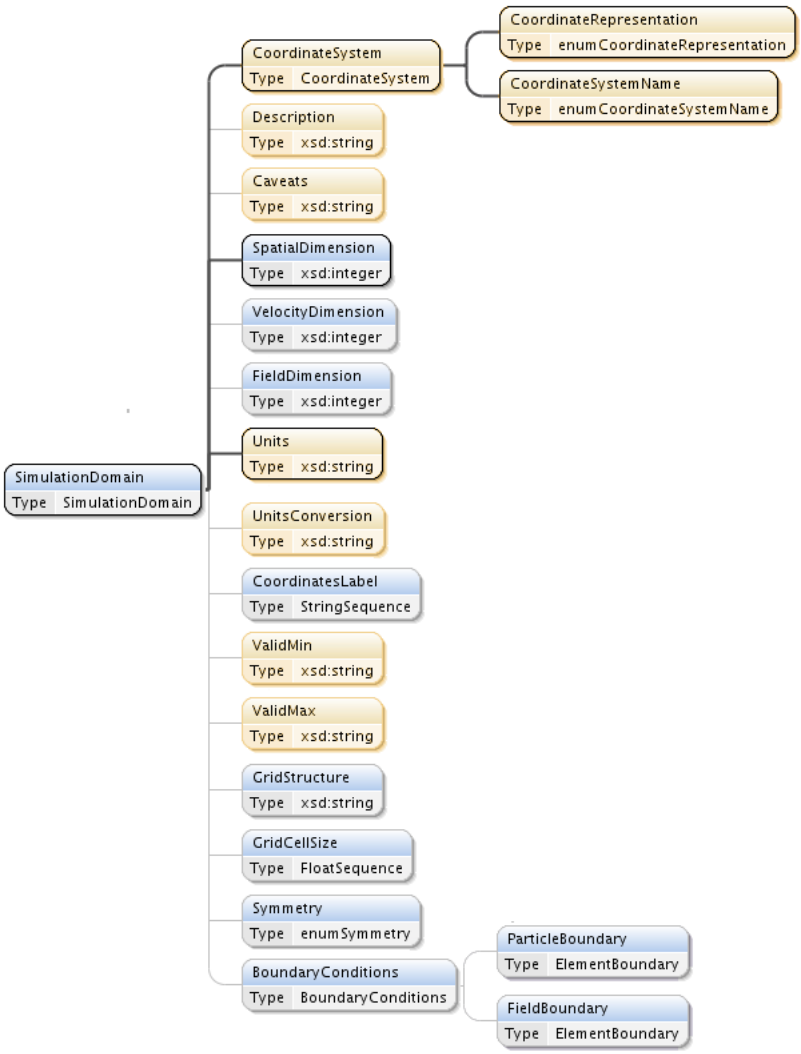


Figure 10 - SimulationDomain element

6.2 Simulation parameters

The parameters of the simulation and their associated value are defined through several elements. The prototype of these elements is *InputParameter*. Its physical meaning is set through *ParameterQuantity*. The list of possible values for this quantity is the same than for the properties. *Qualifier* can refine the *Quantity* meaning. The parameter itself has no value. It is composed of several *Property* elements, which also have physical meaning (within that of the parameter) and may have values.

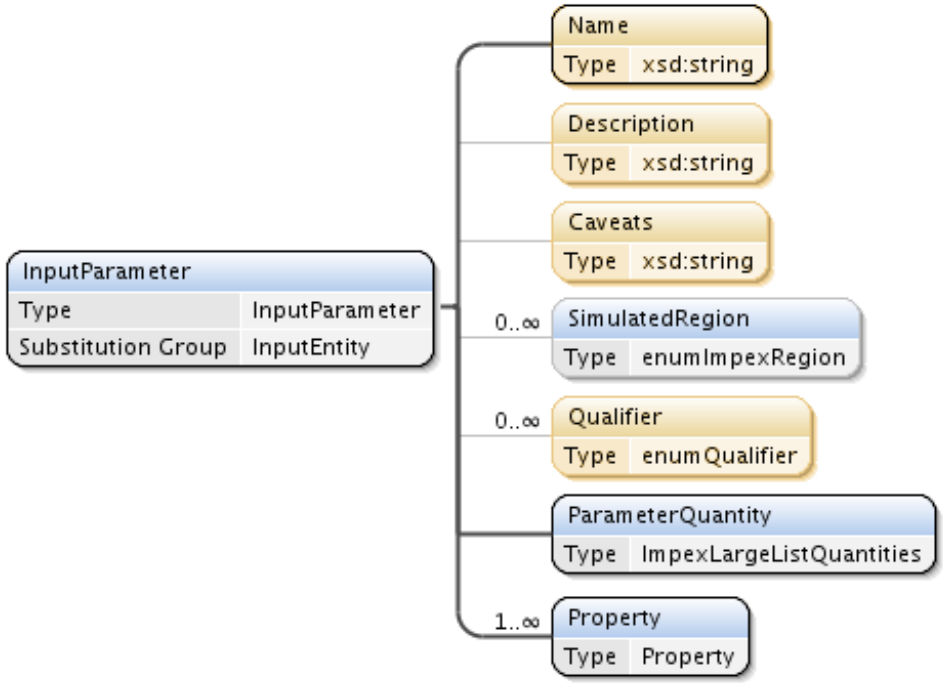


Figure 11 - InputParameter element

6.3 Pre-defined types of simulation parameters.

The prototype of input parameters can virtually describe any possible parameter, and associate to most of them a physical quantity (providing that this physical quantity is related to some extent to the space-physics domain).
This contributes to make the IMPEx DM versatile, and allows automated tools to "understand" the parameters. However, it is not that easy for tools to really get a sense of what exactly the physical parameters are and to perform precise search on them. Hence, we set up various pre-defined types of simulation parameters. These pre-defined types can be understood as InputParameters in which the quantity is predefined, and the properties (and their quantities) also. There is a pre-defined type for:

- particle population
- fields
- chemical processes
- celestial bodies.

6.3.1 RegionParameter

Although the simulated region is related to an astrophysical region through *SimulatedRegion*, it may be necessary to specify or re-define some of the physical properties of the region, which may differ from their actual values.
RegionParameter allows re-defining the celestial body radius, the sub-longitude of its parent body (Sun, or planet), its rotation period, its mass and some other parameters specified by *Property* elements.

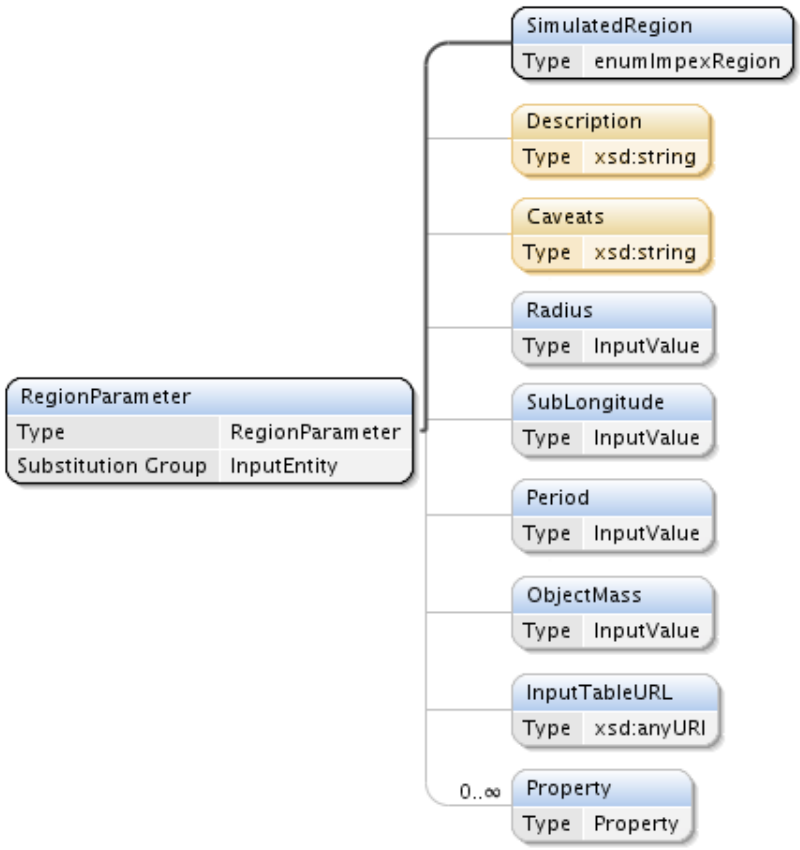


Figure 12 - RegionParameter element

6.3.2 InputField

The fields in the simulation can be specified by *InputField* elements. The input parameter can be related to a region, using *SimulatedRegion*. *CoordinateSystem* specifies the coordinate system in which the field is expressed. *Qualifier* and *FieldQuantity* specify the physical meaning of the parameter. Then, the parameter value(s) can be specified in the same way that in the properties.

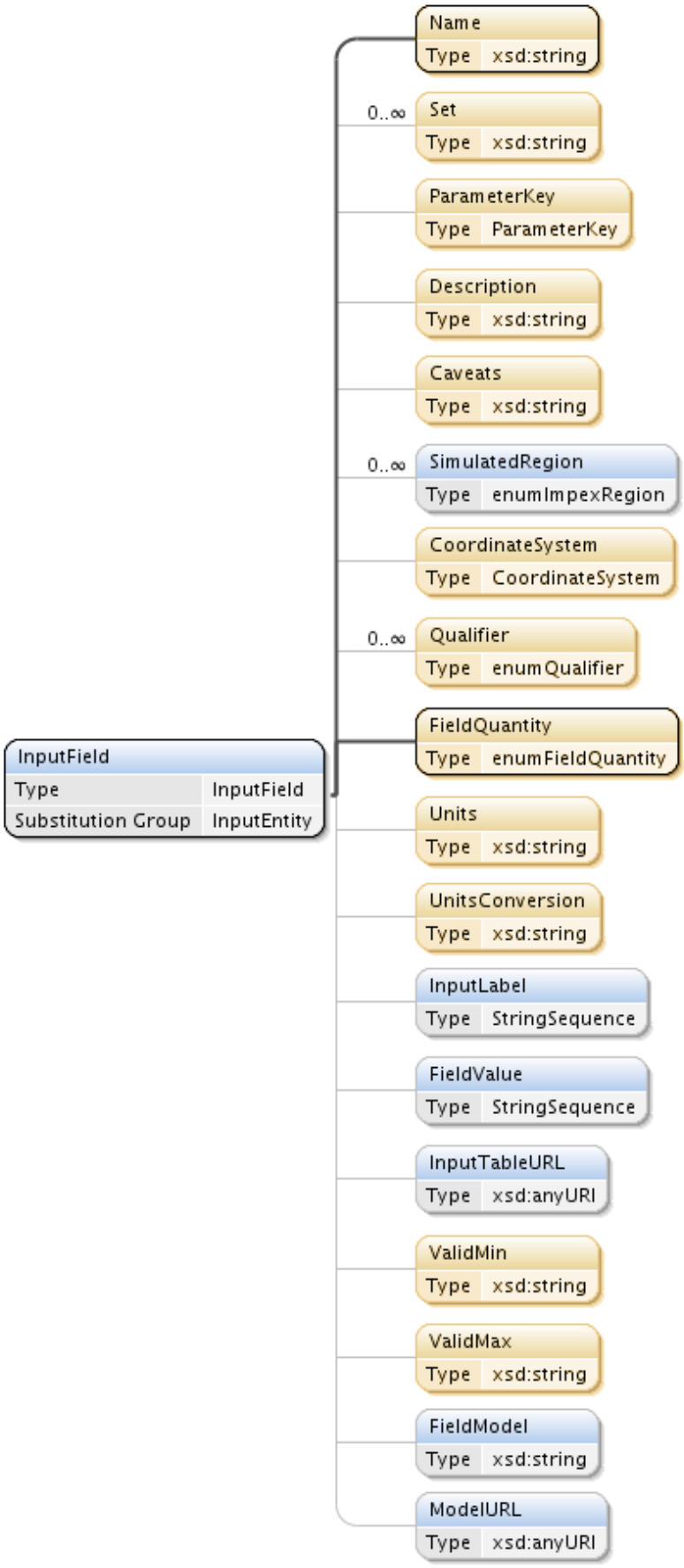


Figure 13 - InputField element

6.3.3 InputProcess

InputProcess describes the chemical processes happening in the simulation. *ProcessType* defines the type of simulated process (ChargeExchange, PhotoIonization, DissociativeRecombination, or ElectronImpact).
ProcessCoeffType specifies the type of coefficient associated to the process (Frequency, CrossSection, or Rate).

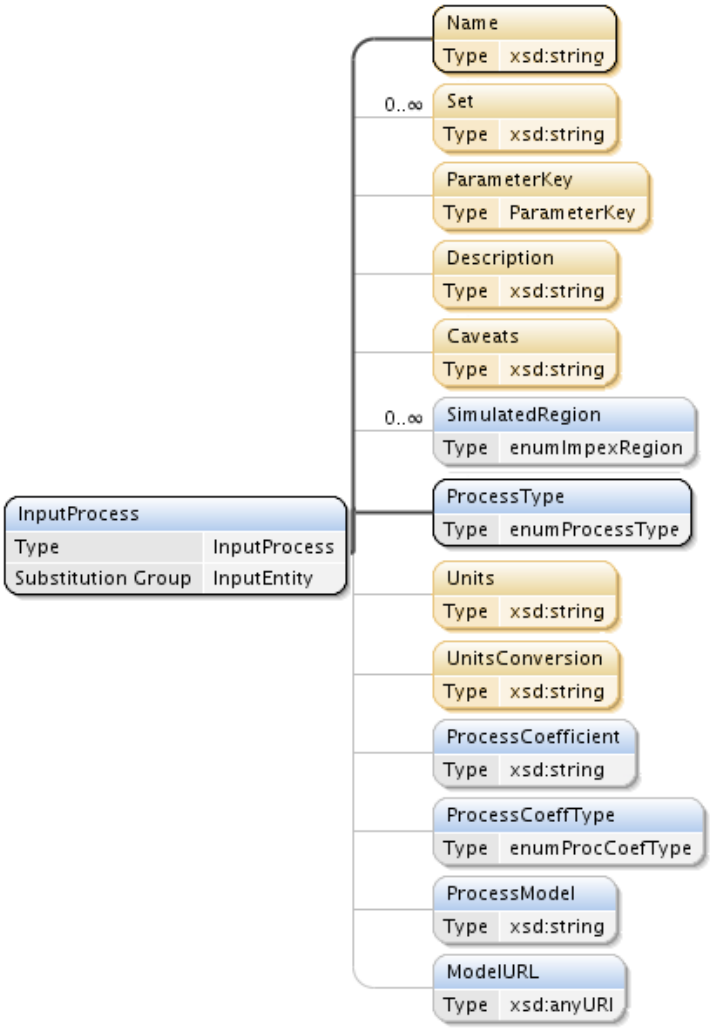


Figure 14 - InputProcess element

6.3.4 InputPopulation

The particle population can be described using *InputPopulation* to specify the particles type, chemical formula, atomic number (for atoms) or mass number, and the particle charge.

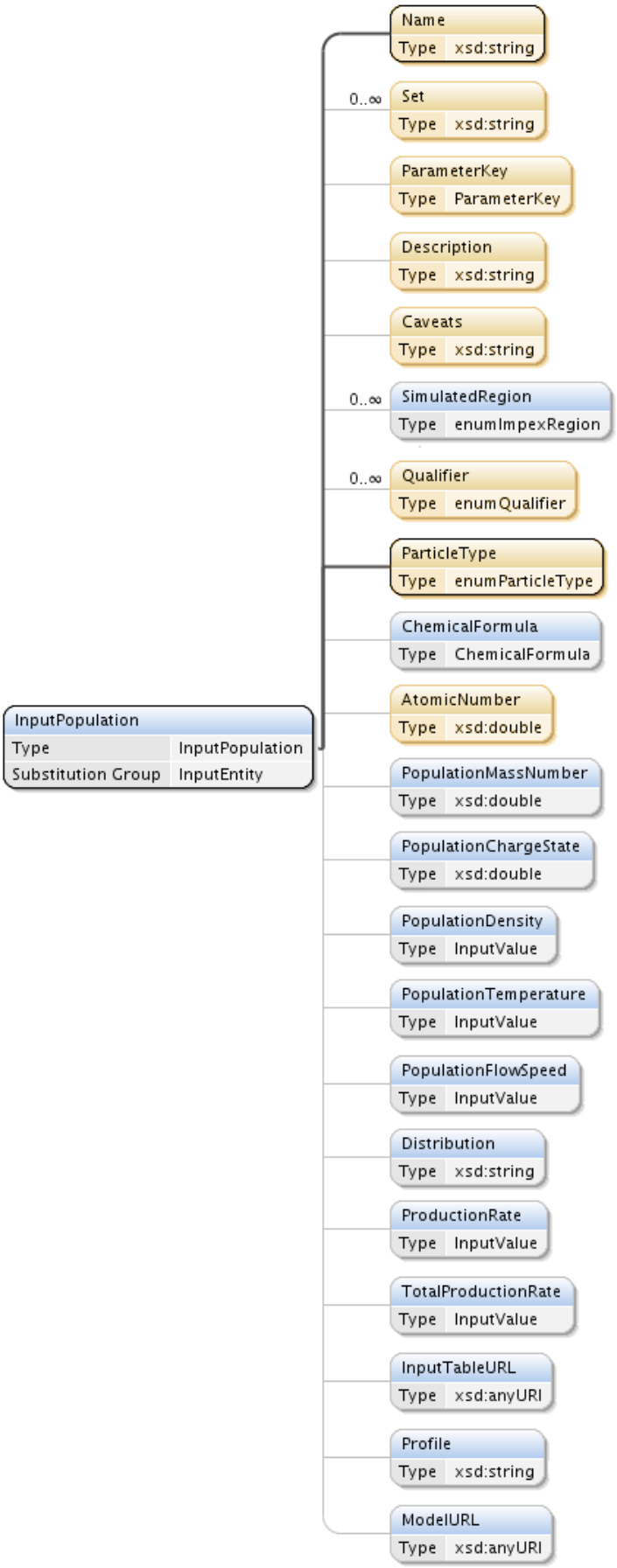


Figure 15 – InputPopulation element

7 Simulation models

7.1 General layout

Models are described in *SimulationModel*. The primary purpose of this element is to keep track of the model and model version used to perform a run. The model can be further documented through the use of *ModelURL* that gives the URL of a webpage or paper describing the code in more details.
Models are described in a very succinct way, as the algorithm of the model itself is not described other than by the simulation type (*SimulationType* can be Particles, Hybrid, MHD, Paraboloid...). One may also define in which language the code is written (*CodeLanguage*) and whether the code results depend on physical time (*TemporalDependence*). The default value is 'No'.
However, in order for tools to be able to ask for simulations on demand, some additional elements can be defined. *SpatialDescription* is similar to that of *NumericalOutput* and allows documenting the spatial information in the run results. *SimulatedRegion* refers to the physical target of the run. *InputProperties* and *OutputParameters* are list of *Property* and *Parameter* tags, listing respectively the input arguments to the method and the outputs of the run.
The general layout of the *SimulationModel* element is depicted in Figure 16:

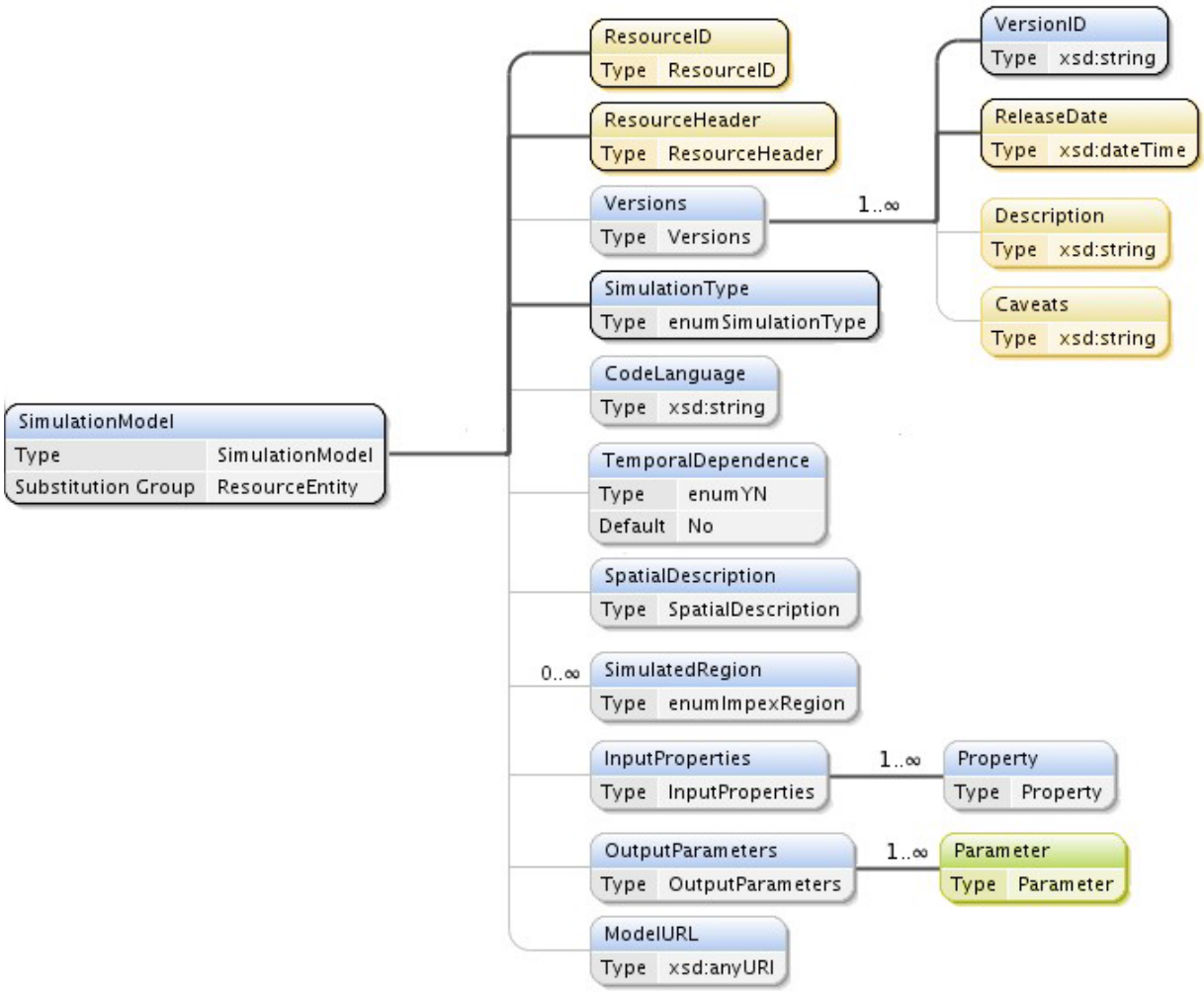
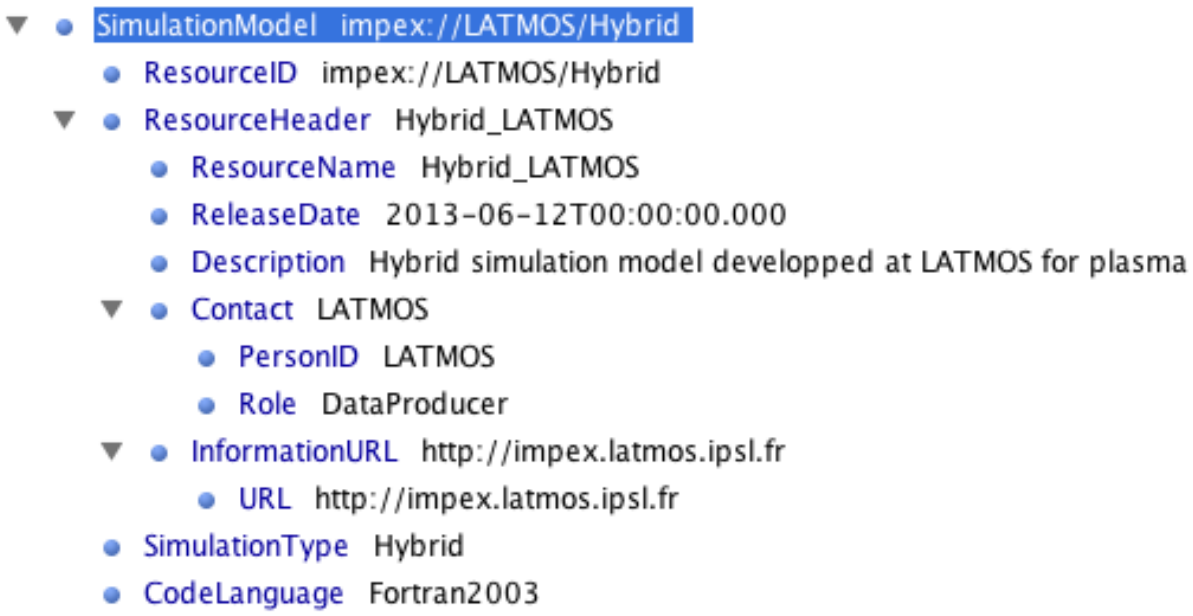


Figure 16 - SimulationModel element

7.2 Runs on Request

To perform runs on request we need: information about the inputs, and information about the outputs. For outputs, information is globally the same than those required for static pre-computed data that are a Spatial Description and a list of Parameters.
Describing the inputs for run on-request is a bit more complex and is achieved by attributing a *Property* element for each input argument to the method. The name of the Property corresponds to the method argument. Its meaning is defined by Property Quantity, eventually a Property Model and a Model URL can be added for user's information (but are not used to define the argument value).
The value that can be taken by the input argument can be either defined as a list in *PropertyValue* or be defined between a minimum and a maximum with *ValidMin* and *ValidMax*. The units of the input value, if any, is given by *Units*, and may be completed by *UnitsConversion* if the units are not SI (such as planet radii for example).

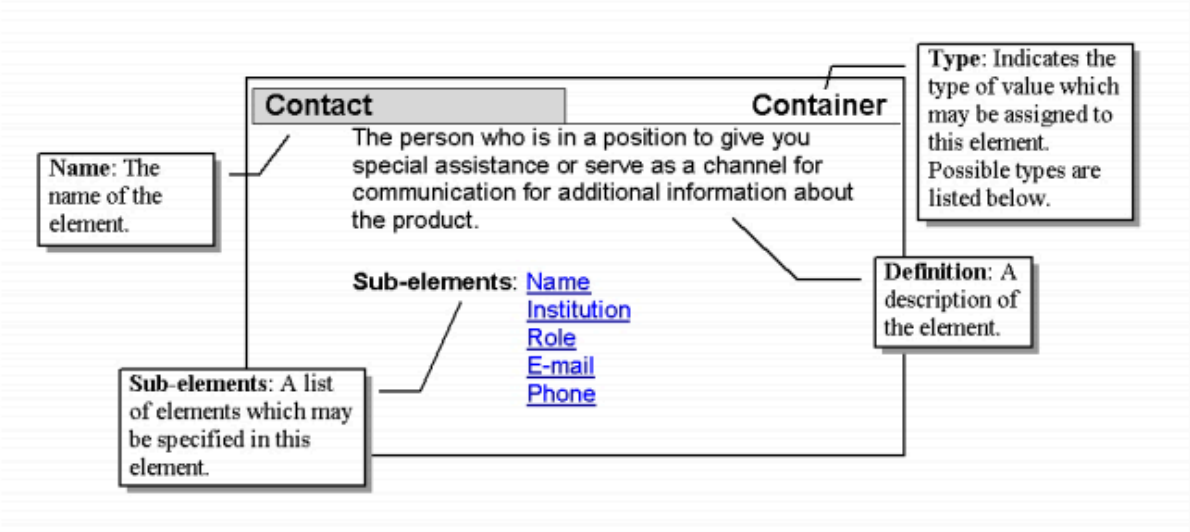
8 Example of SimulationModel Resource



9 Definition of the Data Model terms

How to Read a Definition

Each element has certain attributes and context for use. The details for each element are presented in the following form:



The value associated with an element must be one of the following:

Container: An element that is a container of other elements. If "Container" is specified the element must have sub-elements specified. When a container element is used no value is assigned to the element. All values are contained within the sub-elements.

Enumeration: An element that has a value selected from a list of values. The list to use is indicated in the definition. For example, "Enumerates - see Project List" indicates only values found in the "Project List" may be assigned to this element. **Elements in bold characters are those added by IMPEx to the SPASE original enumeration, e.g. "GphiO" in CoordinateSystemName list .**

Item: An element, which indicates a state or existence of an attribute. An item is valueless. An item may not contain other elements, but could have attributes. An item may also be a member of an enumerated list.

BackWall — Item

Back wall of the simulation domain by which the plasma flow may exit the simulation.

BoundaryConditions — Container

Parameters associated to the simulation boundaries.

Sub-elements:

ParticleBoundary
FieldBoundary

ChemicalFormula — Item

Chemical formula representing a population of particle. Should only contain Chemical Symbols of the elements, numbers and Parenthesis: e.g. O, O2, and CO2. Charge should be specified elsewhere.
Note: use small case x, for undefined number of a given atom in a molecule (e.g. NOx), and W for water group if needed (H2O, OH, H3O...)

CodeLanguage — Item

Language in which a numerical code is written.

CoordinatesLabel — Item

A string list of the labels of each dimension of the spatial domain.

Coordinate System Name — Enumeration

Identifies the coordinate system in which the position, direction or observation has been expressed.

Allowed Values:

CGM
Carrington
CSO
DM
GEI
GEO
GPHIO
GSE
GSEQ
GSM
HAE
HCC
HCI
HCR
HEE
HEEQ
HG
HGI
HPC
HPR
HSM

J2000
JSM
JSO
KSM
KSO
LGM
MAG
MFA
MSO
RTN
SC
SE
SM
SR
SR2
SSE
SSE_L
Spacecraft Orbit Plane
TIIS
VSO
WGS84

DiagnosisTimeStep — Item

The times at which a diagnosis is performed and quantity values saved.

DisplayOutput — Container

A graphical representation of data wherein the underlying numeric values are not (readily) accessible for analysis. Examples are line plots and spectrograms. A Display Output resource is a type of "data product" which is a set of data that is uniformly processed and formatted. A data product may consist of a collection of granules of successive time or spatial spans, but may be a single high-level entity.

Sub-elements:

spase:ResourceID
spase:ResourceHeader
spase:AccessInformation
spase:ProcessingLevel
spase:ProviderResourceName
spase:ProviderProcessingLevel
spase:ProviderVersion
SimulatedInstrumentID
spase:MeasurementType
spase:TemporalDescription or SpatialDescription
spase:SpectralRange
spase:DisplayCadence
SimulatedRegion
spase:Caveats
spase:Keyword
spase:InputResourceID
spase:Parameter
SimulationProduct

Property
spase:Extension

Distribution — Item

Velocity distribution of the particles in a population.

Duration — Item

Duration of a simulation.

FieldBoundary — Container

Boundaries of a field.

Sub-elements:

Caveats
FrontWall
BackWall
SideWall
Obstacle

FieldDimension — Item

Number of field dimensions in the simulation domain.

FieldModel — Item

Field model imposed in the simulation run.

FieldValue — Item

A string list of the values of an InputField.

FrontWall — Item

Front wall of the simulation domain by which the plasma flow may be injected.

Granule — Container

An accessible portion of another resource. A Granule may be composed of one or more physical pieces (files), which are considered inseparable. For example, a data storage format that maintains metadata and binary data in separate, but tightly coupled files. Granules should not be used to group files that have simple relationships or which are associated through a parent resource. For example, each file containing a time interval data for a Numerical Output resource would each be considered a Granule. The ParentID of a Granule resource must be a NumericalOutput resource. The attributes of a Granule supersede the corresponding attributes in the NumericalOutput resource.

Sub-elements:

spase:ResourceID
spase:ReleaseDate
spase:ExpirationDate
spase:ParentID
spase:PriorID
spase:StartDate or RegionBegin
spase:StopDate or RegionEnd

spase:Source

ImpexQuantity — Enumeration

Part of PropertyQuanty List.

Allowed values:

IMFClockAngle
SolarUVFlux

InputField — Container

A field associated to a the simulation.

Sub-elements:

spase:Name
spase:Set
spase:ParameterKey
spase:Description
spase:Caveats
SimulatedRegion
spase:CoordinateSystem
spase:Qualifier
spase:FieldQuantity
spase:Units
spase:UnitsConversion
InputLabel
FieldValue
InputTableURL
spase:ValidMin
spase:ValidMax
FieldModel
ModelURL

InputParameter — Container

A parameter that is an input of a simulation run.

Sub-elements:

spase:Name
spase:Description
spase:Caveats
SimulatedRegion
spase:Qualifier
ParameterQuantity
Property

InputPopulation — Container

A simulated specie.

Sub-elements:

spase:Name
spase:Set

spase:ParameterKey
spase:Description
spase:Caveats
SimulatedRegion
spase:Qualifier
spase:ParticleType
ChemicalFormula
spase:AtomicNumber
PopulationMassNumber
PopulationChargeState
PopulationDensity
PopulationTemperature
PopulationFlowSpeed
Distribution
ProductionRate
TotalProductionRate
InputTableURL
Profile
ModelURL

InputProcess Container

A chemical process happening in a simulation.

Sub-elements:

spase:Name
spase:Set
spase:ParameterKey
spase:Description
spase:Caveats
SimulatedRegion
ProcessType
spase:Units
spase:UnitsConversion
ProcessCoefficient
ProcessCoeffType
ProcessModel
ModelURL

InputProperties Container

A sequence of Property elements.

Sub-elements:

Property

Model Container

The Simulation Model corresponding to a Simulation Run.

Sub-elements:

ModelID
VersionID

ModelVersion Container

Version of a simulation model.

Sub-elements:

VersionID
spase:ReleaseDate
spase:Description
spase:Caveats

NumericalOutput Container

Data stored as numerical values in a specified format. A Numerical Output resource is a type of "data product" which is a set of data that is uniformly processed and formatted, from one or more instruments, typically spanning the full duration of the observations of the relevant instrument(s). A data product may consist of a collection of granules of successive time spans, but may be a single high-level entity.

Sub-elements:

spase:ResourceID
spase:ResourceHeader
spase:AccessInformation
spase:ProcessingLevel
spase:ProviderResourceName
spase:ProviderProcessingLevel
spase:ProviderVersion
SimulatedInstrumentID
spase:MeasurementType
spase:TemporalDescription or SpatialDescription
spase:SpectralRange
SimulatedRegion
spase:Caveats
spase:Keyword
spase:InputResourceID
Parameter
SimulationProduct
Property
spase:Extension

OutputParameters Container

A sequence of Parameter elements.

Sub-elements:

Parameter

Parameter Container

A container of information regarding a parameter whose values are part of the product. Every product contains or can be related to one or more parameters.

Sub-elements:

spase:Name
spase:Set

spase:ParameterKey
spase:Description
spase:Caveats
spase:Cadence
spase:Units
spase:UnitsConversion
spase:CoordinateSystem
spase:RenderingHints
spase:Structure
spase:ValidMin
spase:ValidMax
spase:FillValue
spase:Field
spase:Mixed
spase:Particle
spase:Support
spase:Wave
Property

Particle Container

A description of the types of particles observed in the measurement. This includes both direct observations and inferred observations.

Sub-elements:

PopulationID
spase:ParticleType
spase:Qualifier
spase:ParticleQuantity
ChemicalFormula
spase:AtomicNumber
PopulationMassNumber
PopulationChargeState
spase:EnergyRange
spase:AzimuthalAngleRange
spase:PolarAngleRange

PlaneNormalVector Item

A string list of the component in each dimension of the vector normal to a plane.

PlanePoint Item

Coordinates of a point in a plane.

PopulationID Item

Unique Name of a particle population, for references.

ProcessCoeffType Enumeration

Defines whether the simulation results are obtained from a stationnary solution or are dynamically computed.

Allowed values :

Frequency
CrossSection
Rate
Other

ProcessType — Enumeration

Type of chemical process.
Allowed values :
ChargeExchange
DissociativeRecombination
ElectronImpact
PhotoIonization

Property — Container

A container of information regarding a property of an input parameter.
Sub-elements:
spase:Name
spase:Description
spase:Caveats
PropertyQuantity
spase:Qualifier
spase:Units
spase:UnitsConversion
PropertyLabel
PropertyValue
PropertyTableURL
spase:ValidMin
spase:ValidMax
PropertyModel
ModelURL

PropertyQuantity — Enumeration

A quantity characterizing a Property.
Allowed values:
All elements belonging to the following lists:
spase:DirectionAngle
spase:FieldQuantity
spase:InstrumentType
spase:MeasurementType
spase:MixedQuantity
spase:ParticleQuantity
spase:ParticleType
spase:SpectralRange
spase:WaveQuantity
spase:WaveType
ProcessType
ProcessCoefType
ImpexQuantity
SimulationProduct

RegionParameter — Container

A sequence of element characterizing a region of space.

Sub-elements:

SimulatedRegion
spase:Description
spase:Caveats
Radius
SubLongitude
Period
ObjectMass
InputTableURL
Property

SavedQuantity — Enumeration

Quantity saved during a diagnosis.

Allowed values :

All elements belonging to the following lists :

spase:FieldQuantity
spase:ParticleQuantity
spase:MixedQuantity
spase:WaveQuantity
SimulationProduct

SimulatedRegion — Enumeration

The portion of space on which the simulation is focused.

Allowed values :

Asteroid
Callisto
Comet
Earth
Earth.Magnetosheath
Earth.Magnetosphere
Earth.Magnetosphere.Magnetotail
Earth.Magnetosphere.Main
Earth.Magnetosphere.Polar
Earth.Magnetosphere.Radiation Belt
Earth.Near Surface
Earth.Near Surface.Atmosphere
Earth.Near Surface.Auroral Region
Earth.Near Surface.Equatorial Region
Earth.Near Surface.Ionosphere
Earth.Near Surface.Ionosphere.D-Region
Earth.Near Surface.Ionosphere.E-Region
Earth.Near Surface.Ionosphere.F-Region
Earth.Near Surface.Ionosphere.Topside
Earth.Near Surface.Mesosphere
Earth.Near Surface.Plasmasphere
Earth.Near Surface.Polar Cap
Earth.Near Surface.SouthAtlanticAnomalyRegion
Earth.Near Surface.Stratosphere

Earth.Near Surface.Thermosphere
Earth.Near Surface.Troposphere
Earth.Surface
Enceladus
Europa
Ganymede
Heliosphere
Heliosphere.Heliosheath
Heliosphere.Inner
Heliosphere.Near Earth
Heliosphere.Outer
Heliosphere.Remote 1AU
Incident
Interstellar
Io
Jupiter
Jupiter.Magnetosphere
Mars
Mercury
Mercury.Magnetosphere
Neptune
Planet
Planet.Magnetosphere
Pluto
Rhea
Saturn
Saturn.Magnetosphere
Sun
Sun.Chromosphere
Sun.Corona
Sun.Interior
Sun.Photosphere
Sun.Transition Region
Titan
Uranus
Venus

SimulationDomain Container

Parameters associated to the simulation spatial domain.

Sub-elements:

spase:CoordinateSystem
spase:Description
spase:Caveats
SpatialDimension
VelocityDimension
FieldDimension
Spase:Units
spase:UnitsConversion
CoordinatesLabel

spase:ValidMin
spase:ValidMax
GridStructure
GridCellSize
Symmetry
BoundaryConditions

SimulationModel Container

Descriptor of a simulation model: type of numerical scheme, versions...

Sub-elements:

spase:ResourceID
spase:ResourceHeader
Versions
SimulationType
CodeLanguage
TemporalDependence
SpatialDescription
SimulatedRegion
InputProperties
OutputParameters
ModelURL

SimulationProduct Enumeration

Type of simulation results

Allowed values:

3DCubes
2DCuts
SpatialSeries
TimeSeries
Spectra
Lines

SimulationRun Container

Description of a simulation run, including the code ID, the run spatial and temporal description, and all the relevant inputs.

Sub-elements:

spase:ResourceID
spase:ResourceHeader
spase:AccessInformation
spase:ProviderResourceName
spase:ProviderProcessingLevel
spase:ProviderVersion
Model
TemporalDependence
SimulatedRegion
LikelihoodRating
spase:Caveats
spase:Keyword
spase:InputResourceID

SimulationTime
SimulationDomain
InputEntity
spase:Extension

SimulationTime Container

A container of parameters associated to the simulation time.

Sub-elements:

spase:Description
spase:Caveats
Duration
TimeStart
TimeStop
TimeStep
DiagnosisTimeStep

SimulationType Container

A container of parameters associated to the simulation time.

Sub-elements:

Analytic
Hybrid
MHD
Paraboloid
PIC
Test-Particle

SpatialDescription Container

A characterization of the spatial extent over which the measurement was taken.

Sub-elements:

Dimension
spase:CoordinateSystem
spase:Units
spase:UnitsConversion
CoordinatesLabel
CutsDescription
CubesDescription
Step

Versions Container

A container for the description of multiple versions of a simulation model

Sub-elements:

ModelVersion

10 History of versions

10.1 Version 1.0.1

- Addition of « CSO » in CoordinateSystemName enumeration
- Addition of « HSM » in CoordinateSystemName enumeration
- Addition of « TIIS » in CoordinateSystemName enumeration
- Addition of « Mercury.Magnetosphere » in SimulatedRegion enumeration
- Addition of « Jupiter.Magnetosphere » in SimulatedRegion enumeration
- Addition of « Saturn.Magnetosphere » in SimulatedRegion enumeration
- Addition of « Planet» in SimulatedRegion enumeration
- Addition of « Planet.Magnetosphere » in SimulatedRegion enumeration
- Addition of « Analytic » in SimulationType enumeration
- Modification of the definition of MSO and VSO in CoordinateSystemName