RÉPUBLIQUE FRANÇAISE
Liberté
Égalité
Fraternité

ONERA
THE FRENCH AEROSPACE LAB

ChaMISEn™ Specifications

Characteristics of Material Interactions with Space Environments

Characteristics of Material Interactions with Space Environments

-

ChaMISEn™

-

Specifications

A software product of the “Office National d’Etudes et de Recherches Aerospatiales”

ONERA - The French Aerospace Lab

ChaMISEn Framework specifications

Authors:

Sébastien Hess
sebastien.hess@onera.fr

ONERA – DPHY
Département Physique, Instrumentation, Environnement, Espace
2 Avenue Edouard Belin
F31055 Toulouse CEDEX
France

Changelog :

Version	Date	Author	Change
0.0	15/06/2018	S. Hess	Creation
0.1	15/05/2020	S. Hess	Update with 0.1 DM version

Content

I Introduction

This document aims at presenting the specification of the Characteristics of Material Interactions with Space Environment (ChaMISEn) data model (DM) and of the related database management system (DBMS). This DM aims at describing the data arising from the measurements of material characteristics linked to the material interactions with the space environment. While the targeted interactions are primarily the electrostatic interactions with low energy plasma, the DM aims at covering the neutral interaction (Erosion, contamination, ATOX,…) and the higher energy interactions (effect of high energy radiations,…). Nonetheless, the targeted interactions are primarily physic interaction rather than chemical ones.

The ChaMISEn DM is targeting research or industrial facilities that perform these measurements and aims at helping them to keep track of all information important to archive and retrieve these data as well as all the related information that are of importance to understand and put in context the measurements.

The ChaMISEn DM thus handles a higher than usual level of polymorphism of the resources stored in the database, so that such data (or metadata) cannot be inserted in a relational database in a way than is simple or even makes sense. Hence, the ChaMISEn DM is primarily though for object oriented databases. Nonetheless, each data provider is free to implement the model in its own way.

The ChaMISEn Query Language (QL) is built to be very similar to the Structured Query Language (SQL) specification to ease integration in larger data handling frameworks. It is similar, although somewhat more targeted and less general than the Object Query Language (OQL), so that the use of the ChaMISEn QL with OQL framework should be simple to implement.

The ChaMISEn Data Model (in any form, in particular the specification document, and XML Schema Definition), the ChaMISEn Query Language, the ChaMISEn JAVA Library and the ChaMISEn JDBC Driver as well as all software cited in the present document are products of the Office National d'Etudes et Recherches Aerospatiales (ONERA). They are distributed under an Apache v2.0 license for the specifications and under CeCILL licenses for software. In particular, the ChaMISEn Data Model inherits from the Space Physics Archive Search and Extract (SPASE) Data Model edited by the SPASE consortium (http://spase-group.org) and conforms to its license terms.

More information and updates can be found on the ChaMISEn web site (http://onera.fr).

The ChaMISEn models and framework were developed at ONERA as part of the SMART-EYES Research Program (2018-2021).

II Data Model Specifications

II.1 SPASE model

The SPASE model is a hierarchical data model that aims at describing and providing access to all of the data relevant for heliophysics. As it describes itself:

"The SPASE (Space Physics Archive Search and Extract) Data Model is a set of terms and values along with the relationships between them that allow describing all the resources in a heliophysics data environment. It is the result of many years of effort by an international collaboration (see http://spase-group.org) to unify and improve on existing Space and Solar Physics data models. The intent of this Data Model is to provide the means to describe resources, most importantly scientifically useful data products, in a uniform way so they may be easily registered, found, accessed, and used.

The SPASE data model divides the heliophysics data environment into a limited set of resources types. A key resource type is Numerical Data. This type of resource typically consists of a set of files containing values of one or more physical variables and that differ from each other only by the time span. To full describe a Numerical Data resource requires other types of Resources, namely Observatory, Instrument, Person, and Repository, whose names are self-explanatory, and each of which has its own set of attributes. Often, numerical data are presented in prepared images (gif or jpeg), and such presentations are referred to as Display Data resources. The other data related resource types are Catalog which are lists of events; Annotation which enable expert comments on data products; and Granule which describe individual files within another resource (i.e., Numerical Data, Display Data or Catalog). Other types of resources include Document which can contain narratives or supporting information; Service that provide software to use data resources; Repository for storage locations; and Registry for metadata collections. Resource descriptions and the links in them are intended to make the Resource useful to scientific users."

Its use as an international standard has led to the creation of derivatives data model for particular uses:

The UIGONET "Inter-university Upper atmosphere Global Observation NETwork" DM (http://iugonet.org) is a SPASE DM derivative developed by a consortium of Japanese Universities which targets the upper atmosphere observations.

The IMPEx "Integrated Medium for Planetary Exploration" DM (http://impex-fp7.oeaw.ac.at) is a SPASE DM extension developed in the Seventh Framework Program of the European Commission. It provides an extension to SPASE allowing describing numerical simulations. This extension was proposed for integration in the SPASE DM and is now provided as the SPASE simDM extension. This extension is included in the ChaMISEn DM.

More information about the SPASE DM can be found on the SPASE consortium website (http://spase-group.org), including detailed information about ChaMISEn DM elements that may not be detailed in the present document.

II.2 ChaMISEn Database structure

The ChaMISEn Database structure is composed of a repository, containing all data files and providing access to them and a registry that provides the location of all each resources, either metadata or data. The metadata may be gathered in a unique document or split between independent ones; in any case each metadata resource must be accessible through a registry, itself described as a resource accessible directly or via inheritance from the top level central registry at (http://onera.fr).

It is possible for entities to set up their own top level registry, but in this case the registry must only contain links toward resources on which this entity owns the Intellectual Property Rights (IPR) with an identification of each resource that prevents conflicts with resources registered in the central registry. This prevents the possibility to redirect links made in third-party resources from an original resource to a modified resource. The practice of deliberately using existing third-party resource identifier to redirect the links between resources without the consent of the third-party resource IPR owner may be punished by law under many jurisdictions as both resource identifier IPR violation and/or unwanted data flow redirection may be forbidden by these jurisdictions.

The ChaMISEn Data Model resources are organized in a hierarchical structure. Although the examples provided in the present document uses the XML format, the format choice is let free and is not part of the specifications. Compliant online database shall however provide resources in XML format to SOAP requests.

The ChaMISEn top level element is always of type “Chamisen”, which contains the DM version number and all top level resources. Hereafter, we describe the descriptive elements common to each resource, followed by a description of each resource individually.

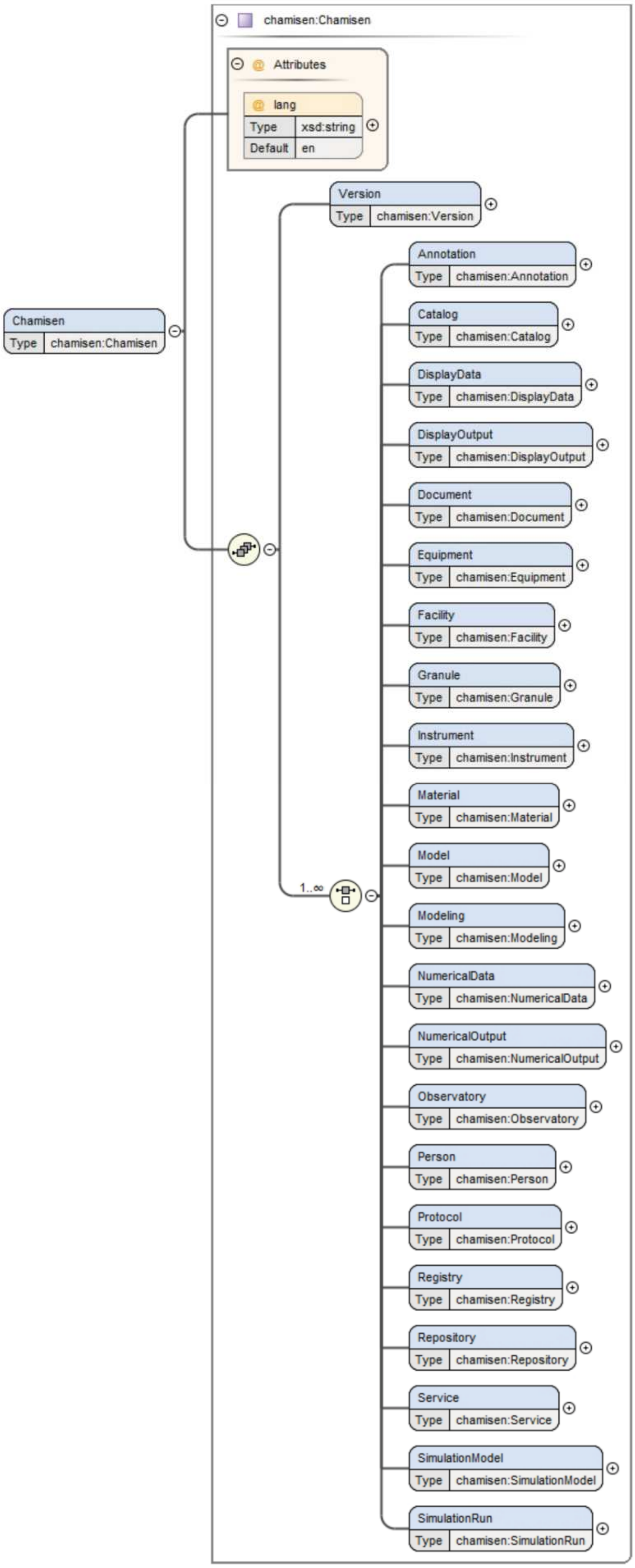

Figure 1: Chamisen top level element structure

II.3 Resource types

II.3.1 Resource structure

(a) Resource ID

Every resource has a unique identifier so that it can be tracked and referenced within a system. This identifier is specified as the text content of the <ResourceID> element of the resource, which must be the first element of any resource. Following the SPASE data model recommendations, each resource identifier is a URI that has the form

scheme://authority/path

where “scheme” is “chamisen” for those resources administered through the ChaMISEn framework and “spase” for those resources administered through the SPASE framework. “authority” is the unique identifier for the naming authority within the data environment: “ONERA” is the unique identifier used by ONERA for the identification of the resource it generates. “path” is the unique local identifier of the resource within the context of the “authority”. The resource ID must be unique within the data environment.

(b) Resource Header

Each resource, except for the “Person” resource, must provide essential information in a header specified in the <ResourceHeader> element. This element directly follows the <ResourceID> element. The <ResourceHeader> element is directly inherited from the SPASE data model.

This header provides in particular the resource name (with possible alternatives) and description, the resource release date and resource contact information. In order to be compliant with the ISO 15836 norm, contacts must include as a minimum list a contact the <Role> “Publisher”, and one (eventually the same) with the <Role> “DataProducer” (which corresponds to the “Creator” field in the ISO 15836 norm).

The header also provides the resource revision history as well as the funding, acknowledgements and associated resources or documents information.

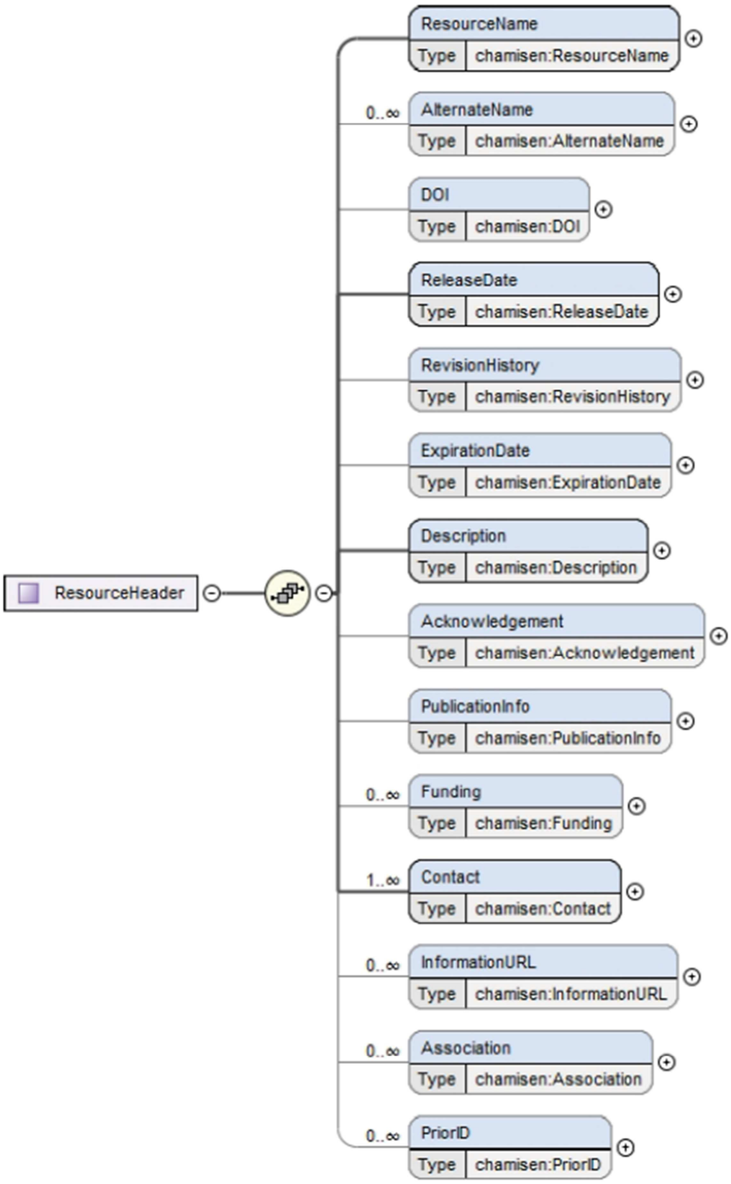

Figure 2: View of the ResourceHeader element (from SPASE v2.3.1)

(c) Access Information

Each resources corresponding to the "Data resource" kind (see section hereafter), i.e. resources describing information stored as data (as opposed to metadata) in the database, must provide access information to the data in the <AccessInformation> element. This element directly follows the <ResourceHeader> element.

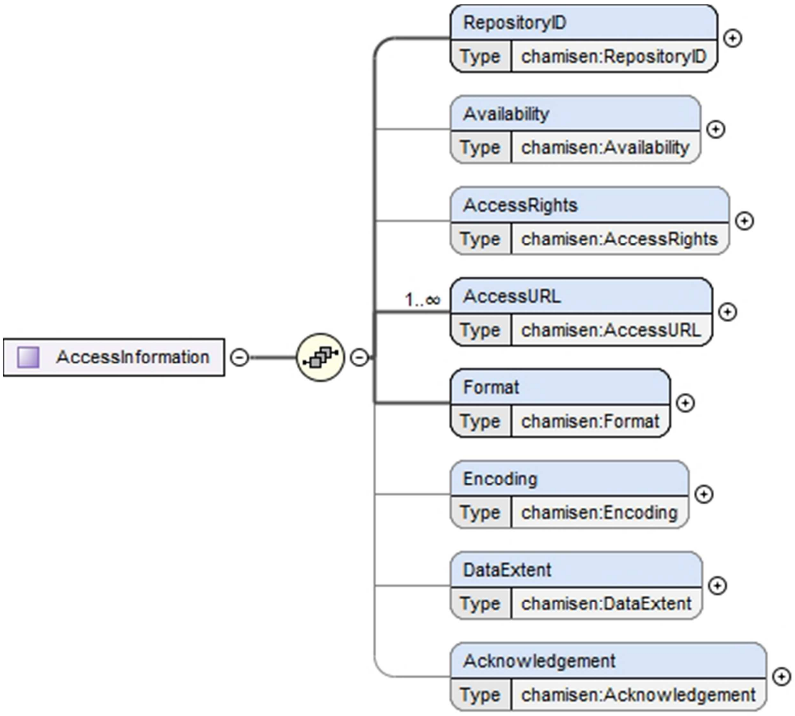

Figure 3 : View of the AccessInformation element (from SPASE v2.3.1)

This element provides in particular the resource location (<AccessURL>) and its format: the document or data file format if the resource describes file content; the resource format (here XML) otherwise. It also describes the access rights.

(d) SPASE and ChaMISEn mapping to ISO 15836

ISO 15836	Spase/ChaMISEn	registry
Title	ResourceHeader.ResourceName	ResourceName
Creator	ResourceHeader.Contact (.Role=DataProducer)	DataProducer
Subject	Keyword	Keyword
Description	ResourceHeader.Description	Description
Publisher	ResourceHeader.Contact (.Role=Publisher)	Publisher
Contributor	ResourceHeader.Contact (.Role=Contributor)	Contributor
Date	ResourceHeader.ReleaseDate	ReleaseDate
Type	The Resource type may be considered as the ISO 15836 type.	ResourceType
Format	AccessInformation.Format	Format
Identifier	ResourceID	ResourceID
Source	ResourceHeader.Association(.AssociationType=DeriveFrom) ParentID (for <Granule>)	ParentID
Language	AccessInformation.AccessURL.Language	Language
Relation	ResourceHeader.Association	Association
Coverage	*Resource dependant*	*Coverage*
Rights	AccessInformation.AccessRights	AccessRights
-	ResourceHeader.ExpirationDate	ExpirationDate
-	*Resource dependant*	URL

II.3.2 Resource content

The data content of a resource that describes data are either described by a <Property> element or by a <Parameter> element. Bother elements share many characteristics, although a <Property> is designed to be more generic and associated to a particular value, while a parameter is designed to provide with a more precise description of the content of a resource (typically the content of a column of a table). Resources describing measurement or simulation data (e.g. NumericalData, NumericalOutput) thus gather <Parameter> elements, while resources describing model input (e.g. DataSet) gather <Property> elements. Resources describing procedures (Protocol, Model, SimulationModel) gather both (<Property for inputs, <Parameter> for output). Application resources (Experiment, Modeling, SimulationRun) use particular elements inherited from the IMPEx data model that gather properties in more precise structures close to that of <Parameter>.

(a) Property

Propery elements aims at defining a characteristic with a value in a way that is generic and versatile enough so that any king of characteristic can easily be defined. It does not share the much heavier structure of the parameter elements that provides more context (parameter type) and information (Structure,…) about fields.

The main characteristics of the Property element are the possibility to define characteristics name, description and label (which act as a local unique identifier within the scope of the resource).

The property can have a value and or specify a range associated to a unit. If the property value is multidimensional or correspond to a long list, the value can be stored in a file which location is given by PropertyTableURL.

If the property is related to a model, it can be identified in the ModelURL element and described in the property model element. The model relation indicates that the property is either an input or an output of the model (in which case it is recommended to have matching property/parameter identifiers). If the property is extracted from data, the model description may contain <InputDataID> tags pointing toward a resource describing the property value origin.

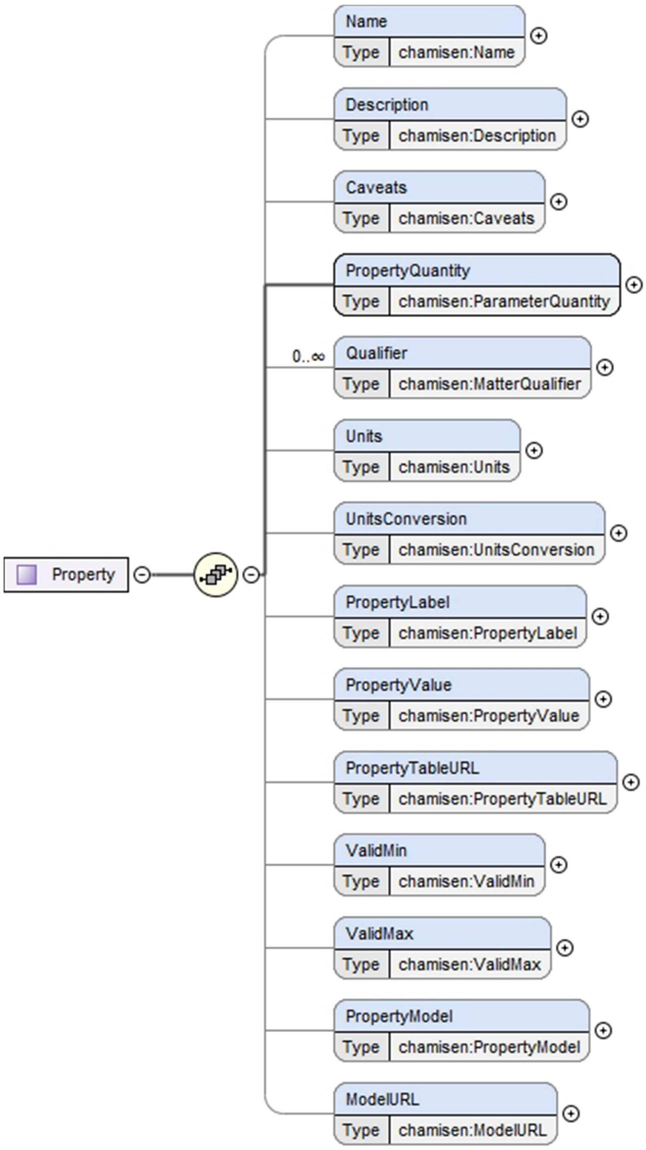

Figure 4 : Property element structure

The Property elements must provide the most possible description of the physic meaning of each property, through the use of the PropertyQuantity and Qualifier element. The Quantity element must correspond to the physics quantity represented by the property, regardless of its context, while the Qualifer(s) precise the quantity characteristics (vector, tensor, exponent) and context (SEEE,…).

(b) Parameter

Parameter elements aims at describing file contents and simulation outputs.

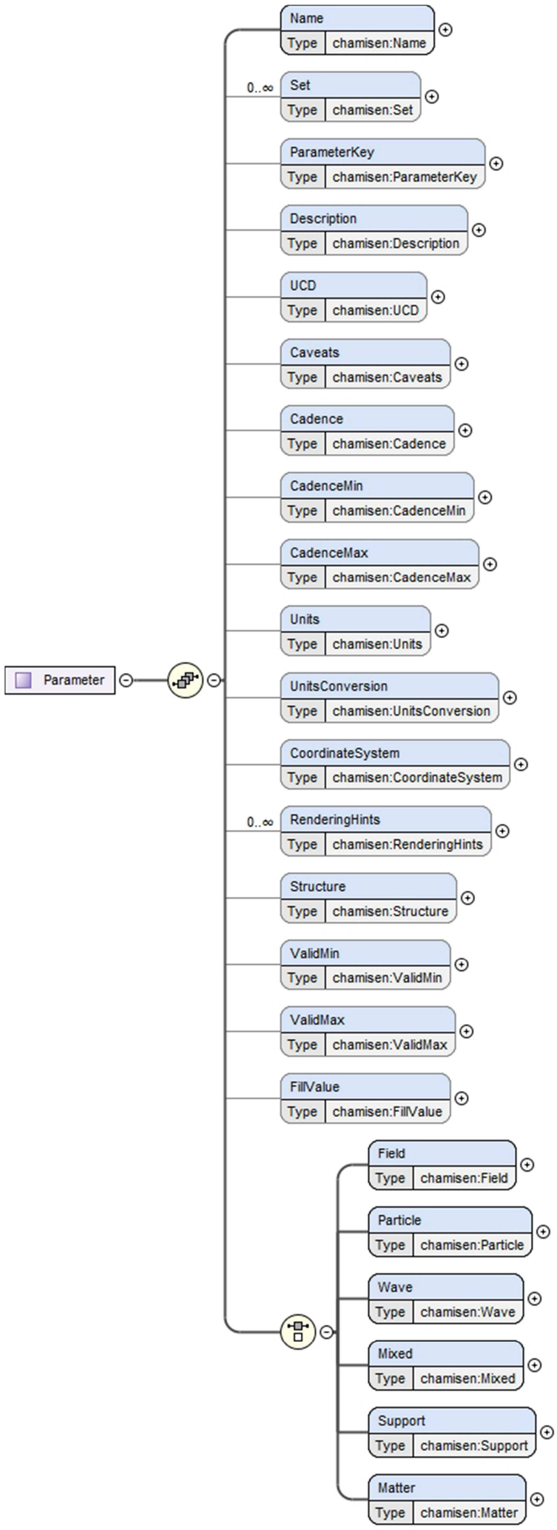

Figure 5: Parameter element structure

They are not intended to contain any value by themselves, which constitute the main difference with the Property elements. As they intend to provide a description of data that may archived for decades, parameter elements provide a more detailed and less versatile description of each parameter, which is translated by the parameter type elements (Field, Particle, Wave, Mixed, Support, Matter) which provides a strong context (and string restrictions) to the parameter. While in most cases these elements only contain a subset of the possible Quantity and Qualifiers, some may provide additional descriptors (Particle type…). The Quantity element must correspond to the physics quantity represented by the property, regardless of its context, while the Qualifier(s) precise the quantity characteristics (vector, tensor, exponent) and context (SEEE…).

In addition parameters allows the description of more complex data structure, through the Structure element which allows one to describe the sub-elements of a composite parameter and to precise the dependence of a table on a quantity in the RenderingHints element. Section III.3 provides an example of indexed parameter, in which an integer codes a reference toward a resource, showing how to declare the index signification (here resources described by their ResourceID).

(c) Input Parameters

The application resources (Experiment, Model, SimulationModel) have their input properties grouped in blocks figured by the InputParameter element. The InputParameter is a named collection of properties to which it is possible to associate a global qualified quantity. This approach is inherited from IMPEx's SimulationRun resource and was adopted because simulations often have as inputs several similar "Objects" (which may or may not be translated as IT objects) defined by similar properties.

For notation simplicity, InputParameter may be replaced by predefined elements to define particle populations (InputPopulation), fields (InputField) and physic process (InputProcess).

II.3.3 Material resources

(a) Material

Material resources are support resources (i.e. containing or pointing towards no data) that are used to define a material or a sample.

The “Material” resource both represent a generic material (ex: “Aluminum”) or a specific sample. To link samples to materials, and specific materials to more generic materials, each material can have a parent material defined by the <ParentID> element that contains the parent material’s identifier.

A “Material” can only have a single parent, but it can be related to other materials through the definition of resource Association (section II.3.1(b)).

Material data are in principle described in one of the “Data Resources” (section II.3.4) or in “Dataset” resources (see hereafter). Hence, material data sheet should be reconstructed “on-the-fly” by the database manager or client by gathering and processing the resource related to a given material data (see database manager specifications).

Nonetheless, database manager implementations are allowed to store “temporary” data in the material resource which corresponds to a pre-processing of these resources. These is achieved through the use of <InheritedProperty> elements that define the properties that are inherited from a parent material, <GatheredProperty> that define properties processed from data and “DataSet” resources and <AggregatedProperty> elements that define properties obtained by the processing of children materials.

A given property cannot be defined more than once, a property cannot be inherited if it can be obtained by gathering material data or aggregated from children materials, a property cannot by aggregated if it can be obtained by gathering material data (see database manager specifications).

While <InheritedProperty> elements are simple <Property> elements, <GatheredProperty> and <AggregatedProperty> elements associate a <Property> element to a processing model <MergingModel>. The <MergingModel> is either a <Model> resource identifier or one of “Single” (meaning that only a single property

RÉPUBLIQUE FRANÇAISE Liberté Égalité Fraternité

ONERA THE FRENCH AEROSPACE LAB

value was found) or "Mean" (Meaning that the property value is an averaged of those found).

An application of material inheritance is given in section III.2.

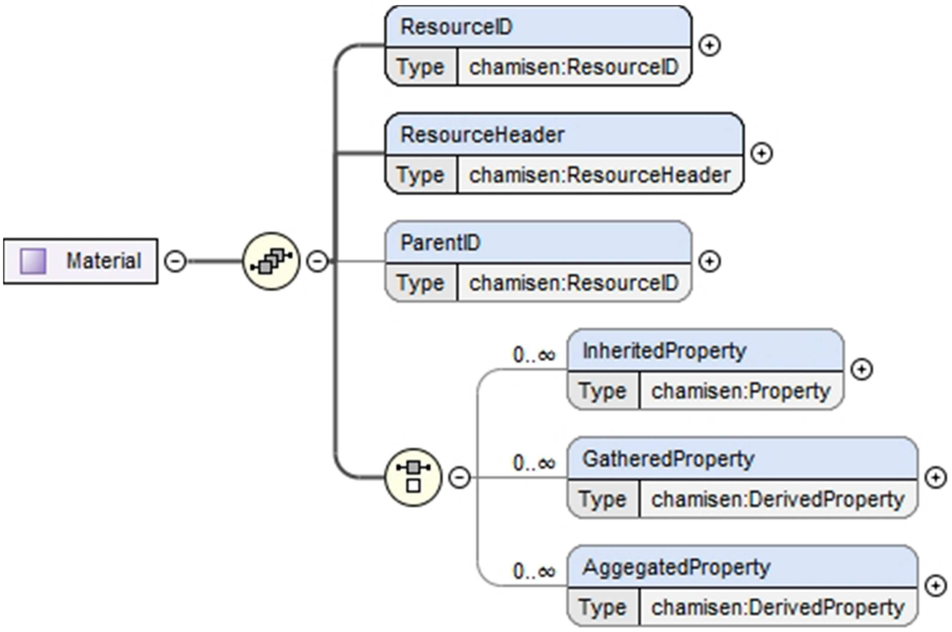

Figure 6: Material Resource structure

(b) DataSet

<DataSet> resources are genuine ChaMISEn resources that intend to contain data, by contrast to the other SPASE inspired resources that only contain metadata. DataSets are used to store material model characteristics that are obtained from measurement, modelling or simulation rather than the measurements or model results themselves. Thus, data stored in datasets should be lightweight, typically scalars and not corresponds to large arrays or tables.

Material characteristics are stored as <Property>.

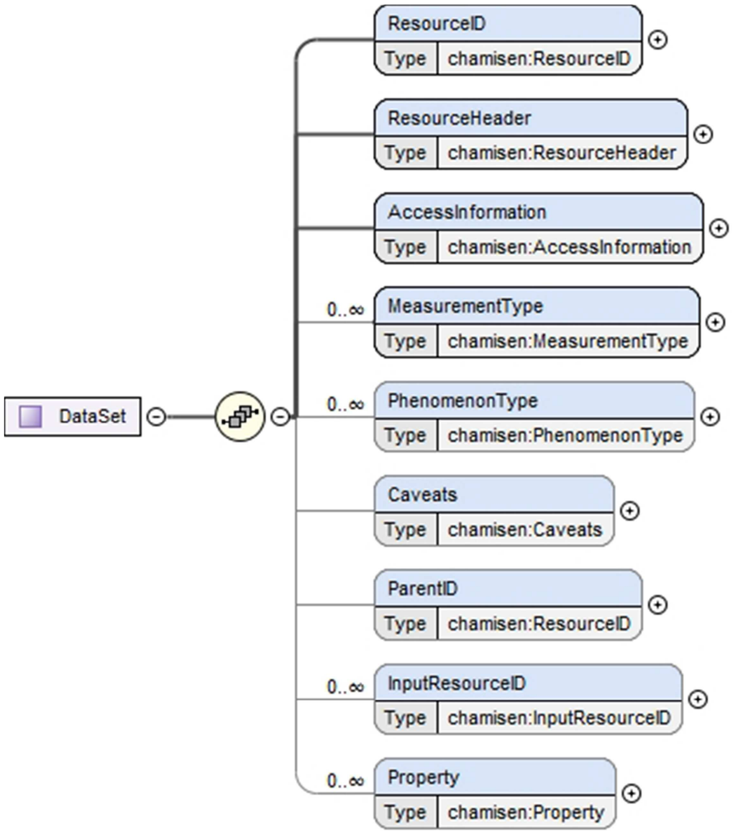

Figure 7: DataSet Resource structure

II.3.4 Data resources

(a) NumericalData

NumericalData resources are used to describe the content of archived files containing data measured by one or several instruments in one or several experiments as numerical values. The instruments used to generate the data and the experiments are defined by setting their identifier in the <InstrumentID> element.

The primary aim of NumericalData is to provide an insight of the measurement type, a context to the measurements and to provide a detailed description of the data file content. The measurement type is defined by the <MeasurementType> element and the studied phenomenon by <PhenomenonType> element(s), while the data fields are described by a <Parameter> element.

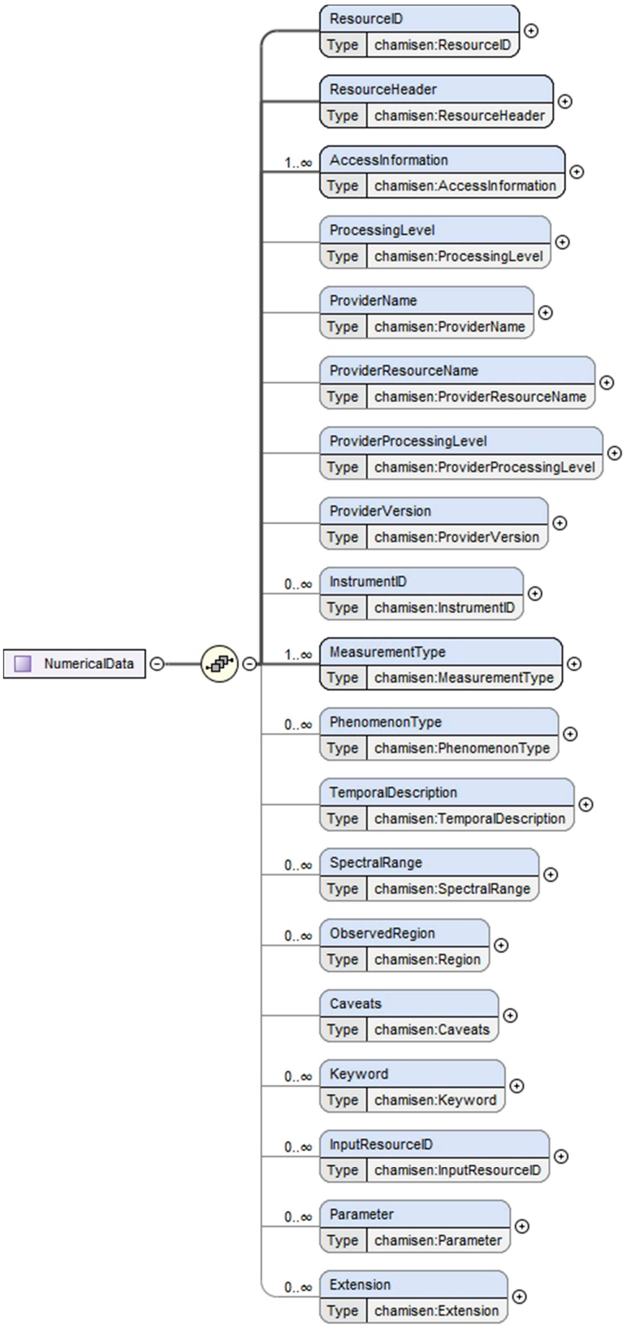

Figure 8: NumericalData Resource Structure

(b) NumericalOutput

NumericalOutput resources are used to describe the content of archived files containing data as numerical values originating from simulations or models. They differ from the NumericalData resources by the possibility of providing a special description and to specify the dimensionality of the results. Simulation or modeling resources used to generate the data are specified in the InputResourceID.

The primary aim of NumericalOutput resources is to provide an insight of the measurement type, a context to the measurements and to provide a detailed description of the data file content. The measurement type is defined by the <MeasurementType> element and the studied phenomenon by <PhenomenonType> element(s), while the data fields are described by a <Parameter> element.

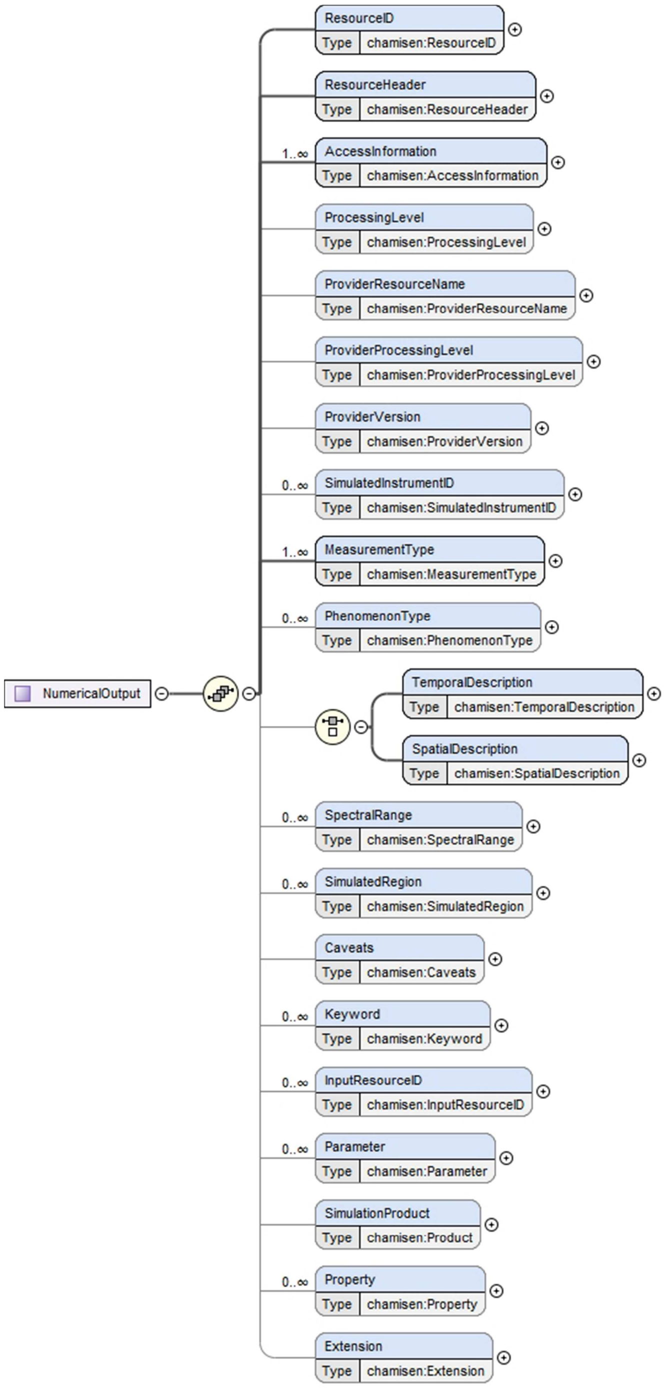

Figure 9: NumericalOutput Resource structure

(c) DisplayData

DisplayData resources are used to describe the content of archived files containing data measured by one or several instruments in one or several experiments in any form except numerical values. The instruments used to generate the data and the experiments are defined by setting their identifier in the <InstrumentID> element.

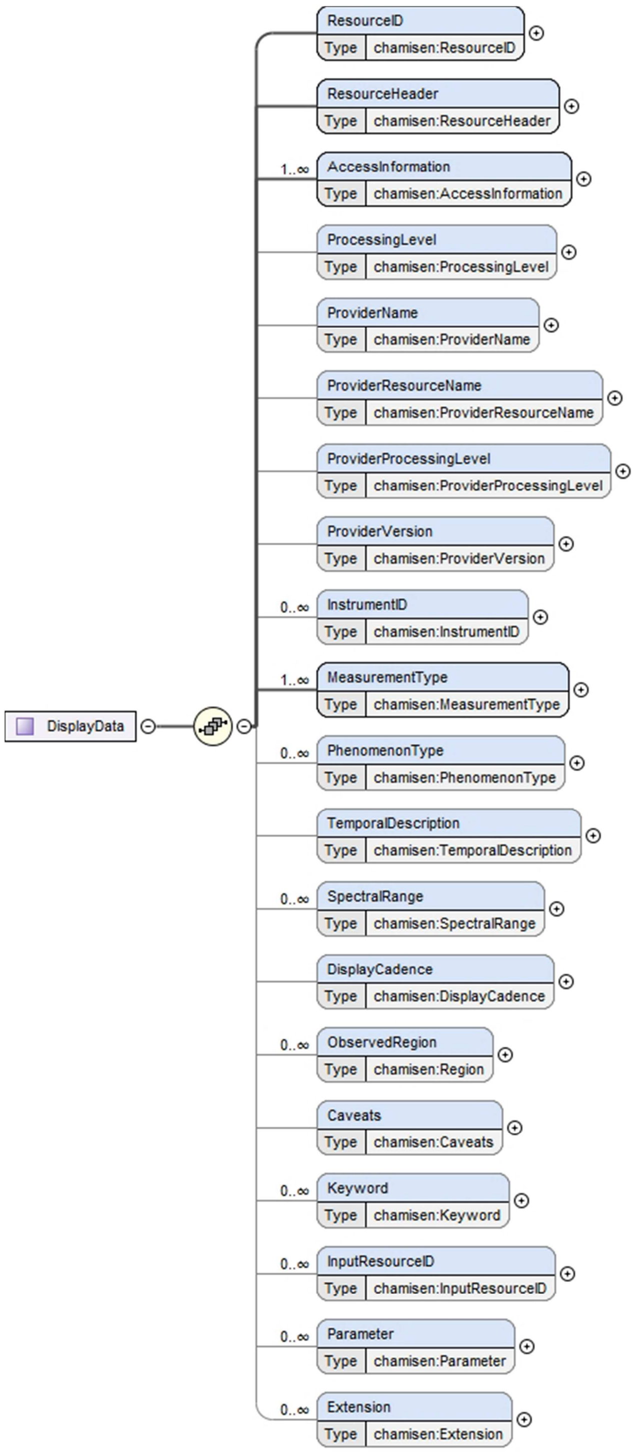

Figure 10: DisplayData resource structure

The primary aim of DisplayData is to provide an insight of the measurement type, a context to the measurements and to provide a detailed description of the data file content. The measurement type is defined by the <MeasurementType> element and the studied phenomenon by <PhenomenonType> element(s), while the data fields are described by a <Parameter> element.

(d) DisplayOutput

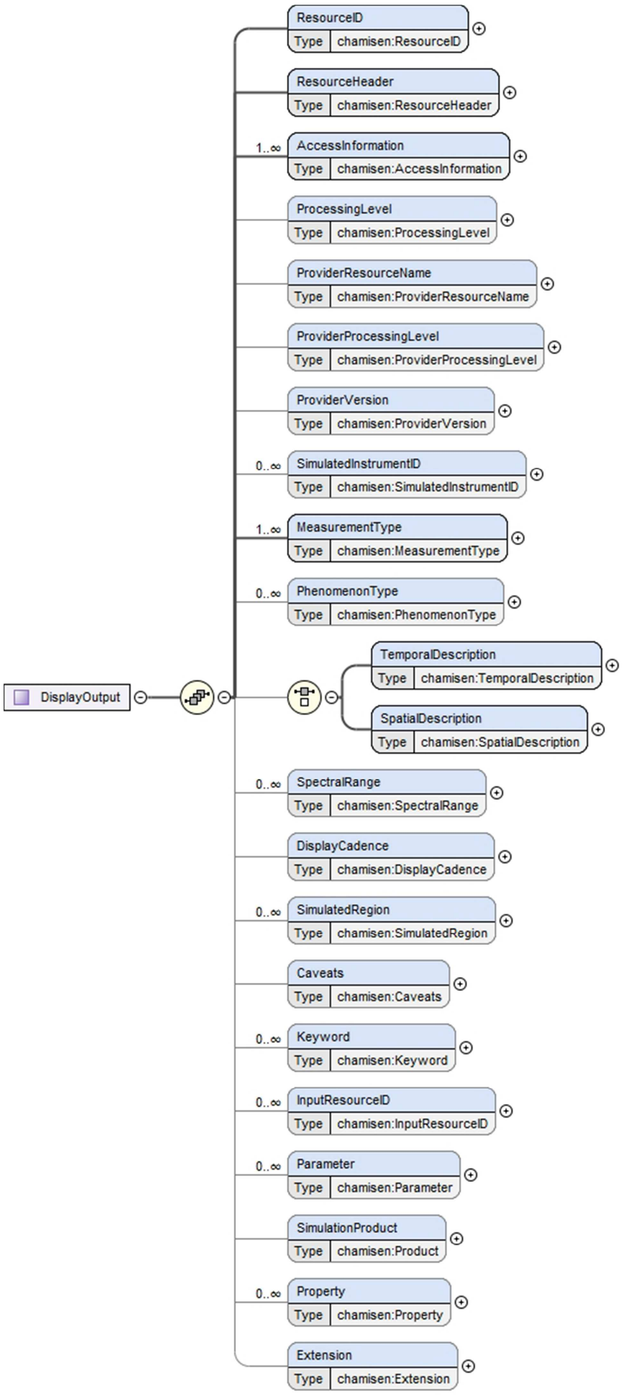

Figure 11: DisplayOutput Resource Structure

DisplayOutput resources are used to describe the content of archived files containing data as numerical values originating from simulations or models experiments in any form except numerical values. They differ from the DisplayData resources by the possibility of providing a special description and to specify the dimensionality of the results. Simulation or modeling resources used to generate the data are specified in the InputResourceID.

The primary aim of DisplayOutput resources is to provide an insight of the measurement type, a context to the measurements and to provide a detailed description of the data file content. The measurement type is defined by the <MeasurementType> element and the studied phenomenon by <PhenomenonType> element(s), while the data fields are described by a <Parameter> element.

(e) Catalog

The Catalog resources are used to define catalogs of events stored in archived files. The type of the studied phenomenon causing the event is mandatory for catalogs and is specified by <PhenomenonType> element(s).

As for previous resources, the primary aim of Catalog resources is to provide an insight of the measurement type, a context to the measurements and to provide a detailed description of the data file content. The measurement type is defined by the <MeasurementType> element and the instrument(s) performing the measurements by <InstrumentID> element(s), while the data fields are described by a <Parameter> element.

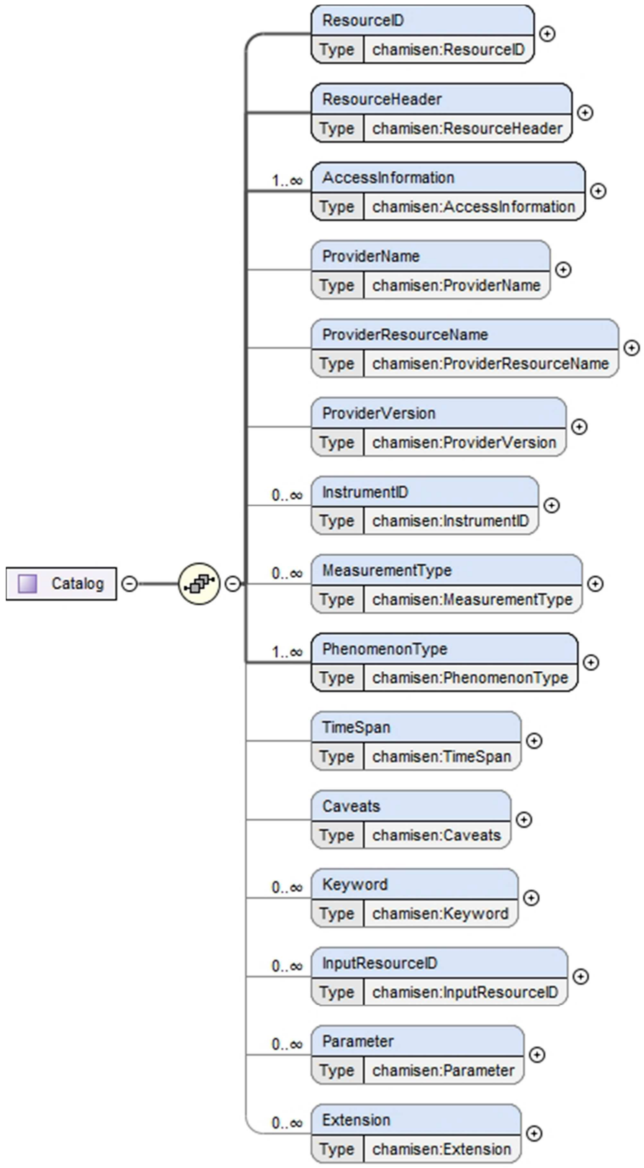

Figure 12: Catalog Resource structure

(f) Document

The Document resource is used to link toward a document or communication of any kind: article, report, presentation, poster….

Although a Document resource does not aim at describing in detail the document content, it may contain Parameter elements if the document contains data tables.

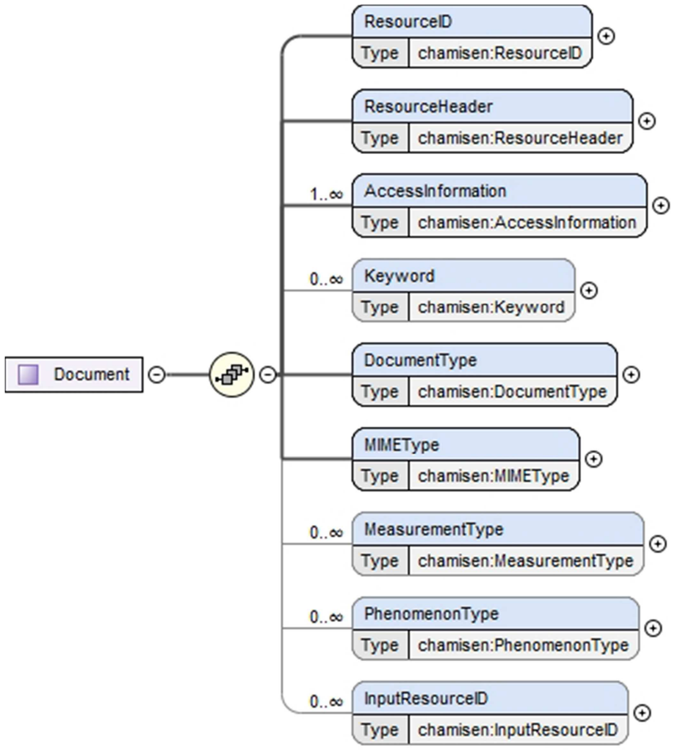

Figure 13: Document Resource structure

II.3.5 Application resources

(a) Experiment

The Experiment resource aims at describing an actual experiment. The facility in which the experiment was performed, as well as the protocol that was followed, shall be referenced through their respective identifiers by the ProtocolID and FacilityID elements. If extra equipment and instrument were used in addition to those given by the facility or protocol resources, they should be referenced by their identifiers in the InputResourceID elements.

The experimental parameters can be given as input parameters (II.3.2(c)).

The material, or samples, on which the experiments were conducted, must be specified in the MaterialID elements.

As for all application resource, it is possible to specify the type of phenomenon and the type of measurement corresponding to the resource, as well as the degree of likelihood, or representativity of the experiment.

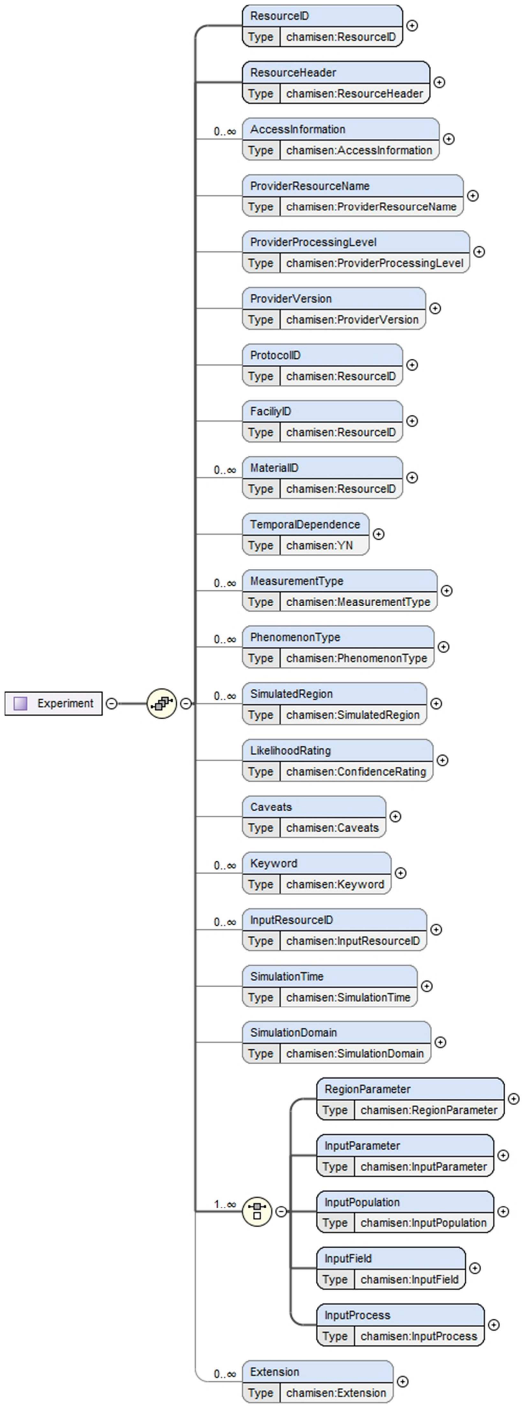

Figure 14: Experiment Resource structure

(b) Modeling

The Modeling resource aims at describing an application of a model. The model that is used must be specified by the InputResourceID elements. If the model is applied on a set of data, they should be referenced in the InputResourceID elements as well, while if sets of data are used as supporting resource they should be referenced in an input parameter property.

The experimental parameters can be given as input parameters (II.3.2(c)).The material, or samples, on which the experiments were conducted, must be specified in the MaterialID elements. As for all application resource, it is possible to specify the type of phenomenon and the type of measurement corresponding to the resource, as well as the degree of likelihood, or representativity of the experiment.

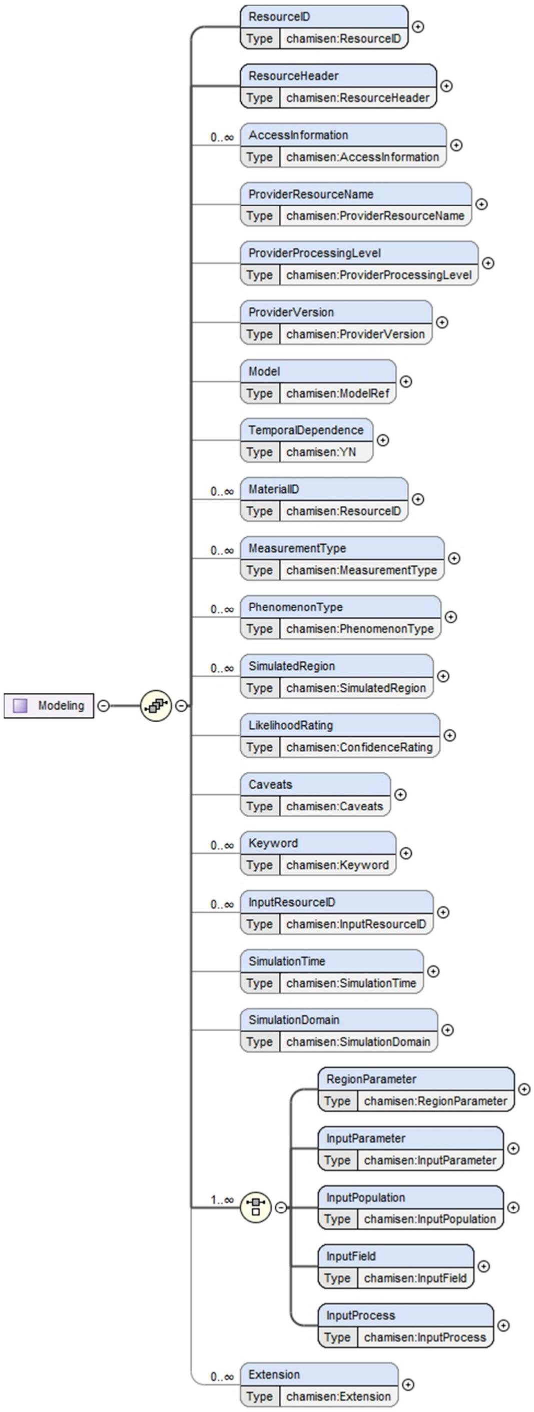

Figure 15: Modeling Resource structure

(c) SimulationRun

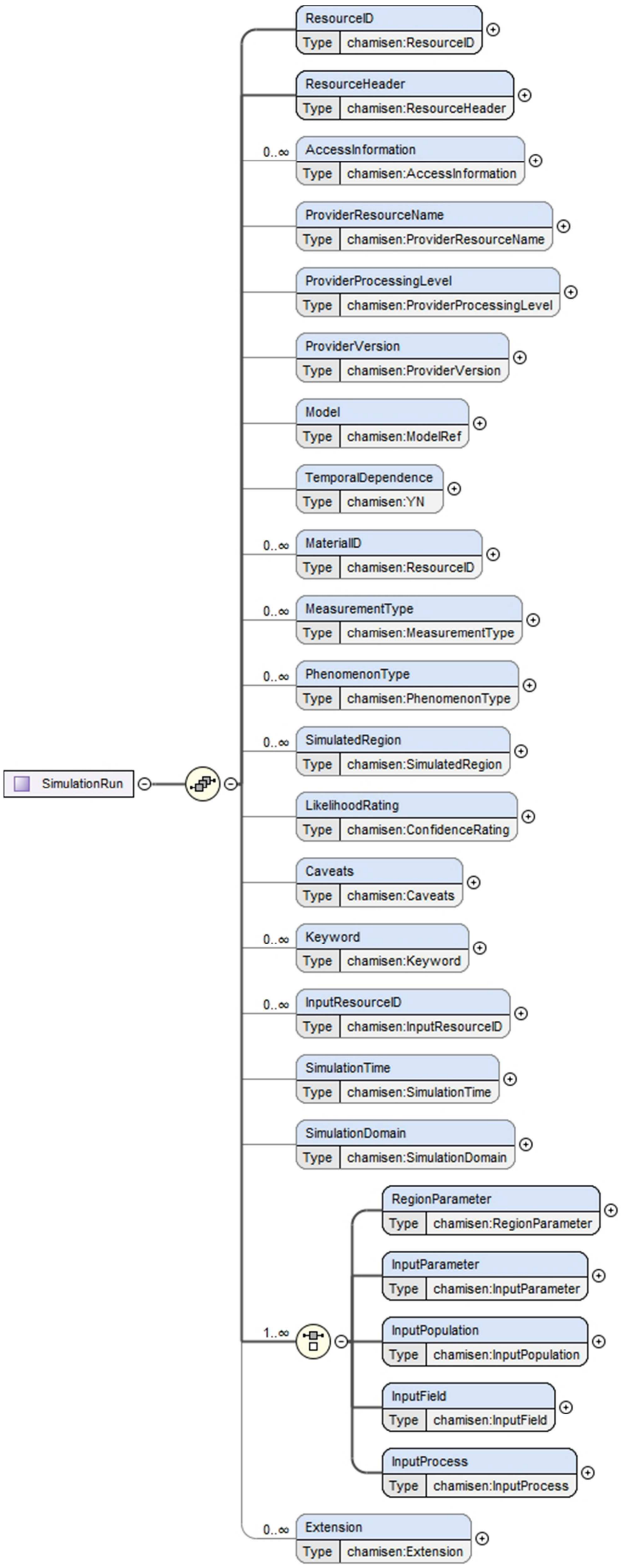

Figure 16: SimulationRun Resource structure

The simulation run resource aims at describing a simulation performance, it allows to describe the inputs of the simulation run and links towards the simulation model used to perform the simulation. The model that is used must be specified by the InputResourceID elements. If the model is applied on a set of data, they should be referenced in the InputResourceID elements as well, while if sets of data are used as supporting resource they should be referenced in an input parameter property.

The experimental parameters can be given as input parameters (II.3.2(c)).The material, or samples, on which the experiments were conducted, must be specified in the MaterialID elements. As for all application resource, it is possible to specify the type of phenomenon and the type of measurement corresponding to the resource, as well as the degree of likelihood, or representativity of the experiment.

II.3.6 Origin resources

(a) Facility

The Facility resource allows to define a ground facility (Vacuum chamber, accelerator,…) in which an experiment is performed.

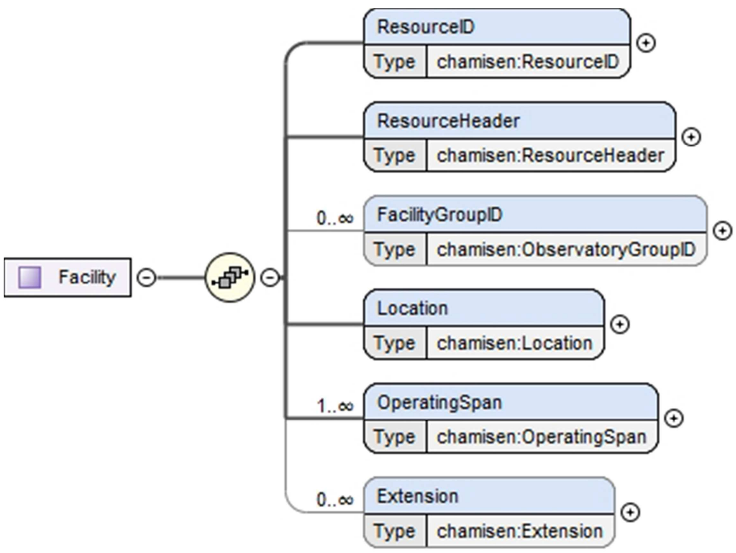

Figure 17: Facility Resource structure

(b) Observatory

The Observatory resource is a legacy resource from SPASE that may not be of any use for ChaMISEn resources, except in the case the measurements are performed in situ on a spacecraft.

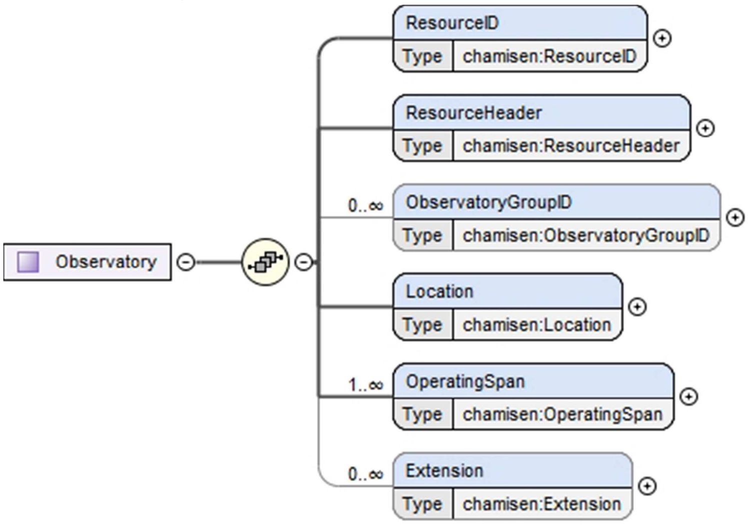

Figure 18: Observatory Resource structure

(c) Equipment

The equipment resource allows defining experiment equipment that is used to perform the experiment but does not aim at performing measurements.

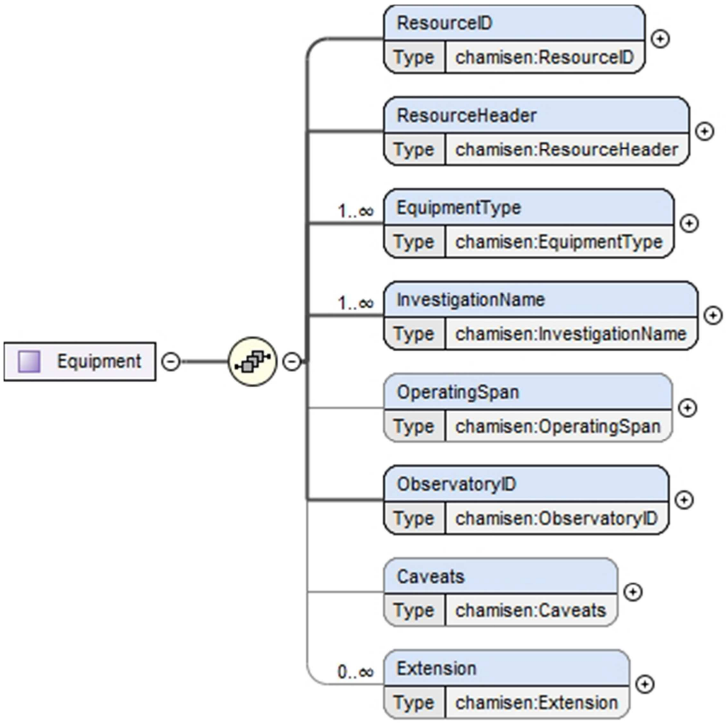

Figure 19: Equipment Resource structure

(d) Instrument

The equipment resource allows defining experiment equipment that is used to perform measurements.

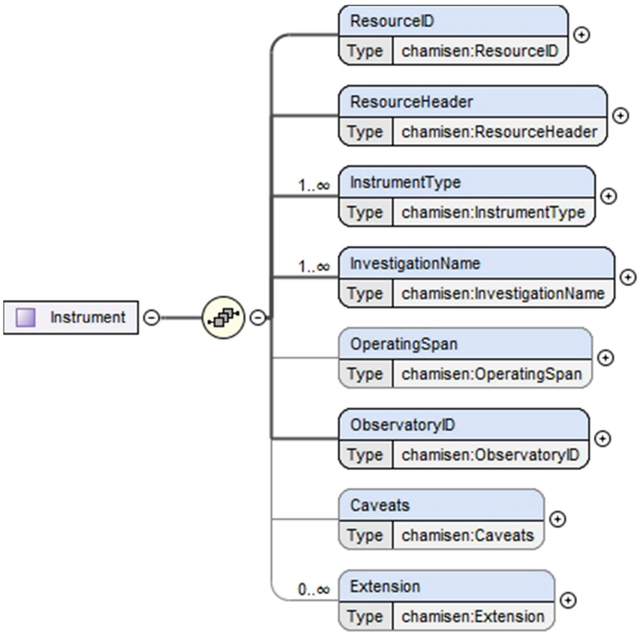

Figure 20: Instrument Resource structure

(e) Protocol

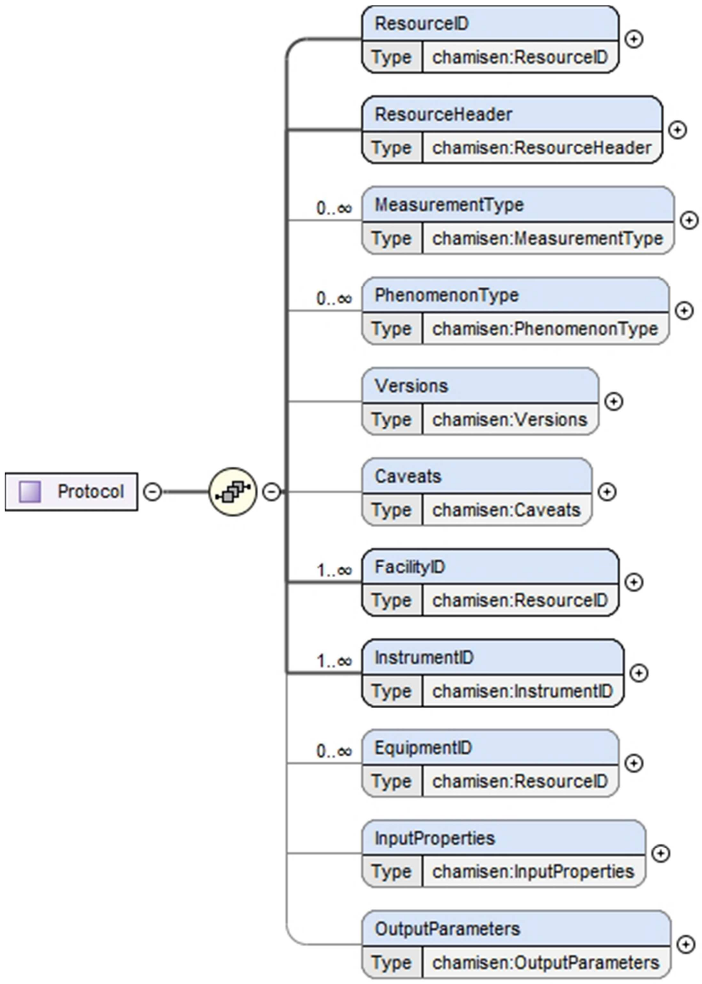

Figure 21: Protocol Resource Description

The <Protocol> resource is used to define experimental protocol that is applied to perform the measurement of a material characteristic. In addition to the standard resource description, the <Model> resource is divided in three sections: the model description, the model inputs description and the model output description.

The experiment settings are specified as a collection of <Property> in the <InputProperties> element. If a value is attributed to a property, it defines its defaults value, while <ValidMin> and <ValidMax> define the range of valid input values.

The experiment outputs are specified as a collection of <Parameter> in the <OutputParameters> element.

(f) Model

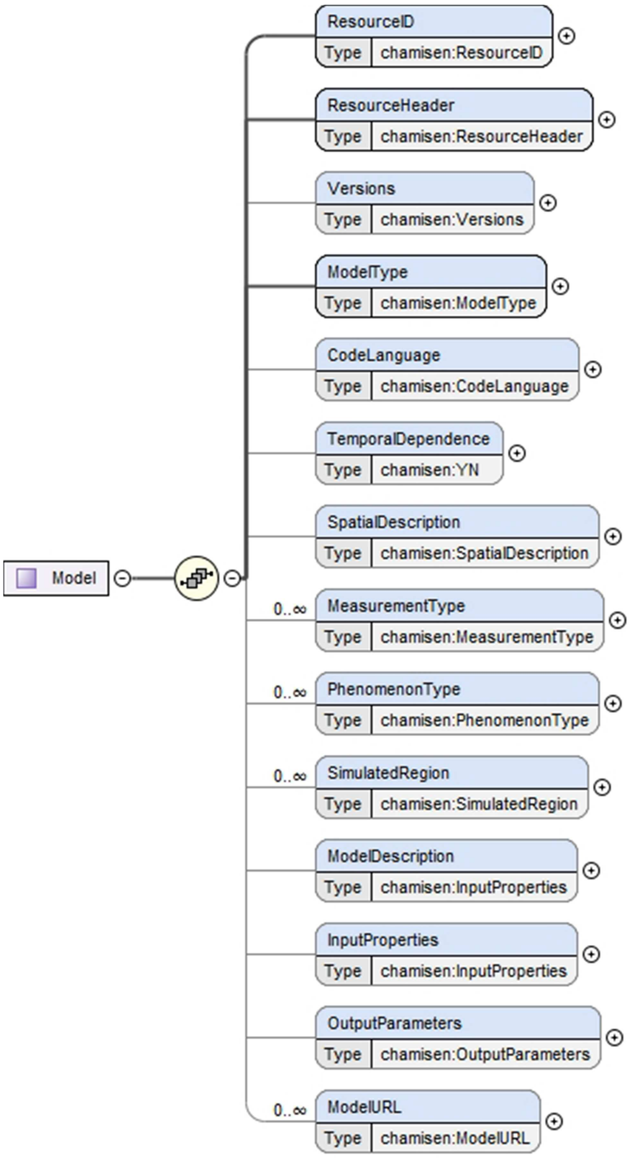

Figure 22: Model Resource structure

The <Model> resource is used to define a numeric, mathematic or physic model that is not used to perform simulations but rather to process data or, in the case of physic model, allows defining the scope of a material property. In addition to the standard resource description, the <Model> resource is divided in three sections: the model description, the model inputs description and the model output description.

```xml
<Model>
    <ResourceID>chamisen://ONERA/DPHY/CSE/SEEY_SimpleModel</ResourceID>
    <ResourceHeader>
        <ResourceName>SEEY Simple Model</ResourceName>
        <ReleaseDate>10/03/2019</ReleaseDate>
        <Description>Simple Model for the fit of the secondary electron
            emission yield curves
        </Description>
        <Contact>
            <PersonID>chamisen://ONERA/DPHY/CSE/SHESS</PersonID>
            <Role>Publisher</Role>
        </Contact>
        <InformationURL />
        <AccessRight>Open</AccessRight>
    </ResourceHeader>
    <ModelDescription>
        <ModelComponent>
            <Name>SEE Emitted electron yield equation</Name>
            <AnalyticFormula>
                EEY=MSEY*(SEE_S*E/PEE)/(SEE_S-1+(E/PEE)^2)
            </AnalyticFormula>
            <SolverType>LeastSquareFit</SolverType>
            <SolverID />
        </ModelComponent>
        <ModelComponent>
            <Name>SEE first cross-over energy expression</Name>
            <AnalyticFormula>EC1=PEE*MSEY*SEE_S/2*(1-\SQRT{1-4*(SEE_S-1)/(MSEY*SEE_S)^2})
            </AnalyticFormula>
            <SolverType>DirectComputation</SolverType>
            <SolverID />
        </ModelComponent>
        <ModelComponent>
            <Name>SEE second cross-over energy expression</Name>
            <AnalyticFormula>
                EC2=PEE*MSEY*SEE_S/2*(1+\SQRT{1-4*(SEE_S-1)/(MSEY*SEE_S)^2})
            </AnalyticFormula><!-- <ModelType>ProcessType/PhenomenonType/ExperimentType -->
            <SolverType>DirectComputation</SolverType>
            <!-- DirectComputation/Fit/LeastSquareFit/DataInversion/Interpolation/SimulationResultInterpolation/SimulationSetInterpolation -->
            <SolverID />
        </ModelComponent>
    </ModelDescription>
```

The model description part allows defining several basic components of the model, defined by their name, analytic equations or model type, and a solver type and identifier if any. The solver type is one of : *DirectComputation* (a parameter value is given simply by solving the model or equation: Input x Model=Output), *Fit* (a parameter value is obtained from a data fit), *LeastSquareFit* (a parameter value is given by the least square fit of data), *DataInversion* (a parameter value is obtained from the data inversion applied to the model: Output x Model = Input), *Interpolation* (the parameter value is obtained through interpolation between existing data), *SimulationResultInterpolation* (the parameter value is obtained through interpolation between existing simulation results), *SimulationSetInterpolation* (the parameter value is obtained through interpolation between inputs of existing simulations by comparing the outputs). The solver used to solve the model can be specified through the <SolverID> element.

The simulation model inputs are specified as a collection of <Property> in the <InputProperties> element. If a value is attributed to a property, it defines its defaults value, while <ValidMin> and <ValidMax> define the range of valid input values.

```
<InputProperties>
    <Property>
        <Name>Secondary Electron Emission Yield</Name>
        <Description>Secondary emission yield curve</Description>
        <PropertyQuantity>NumberFlux</PropertyQuantity>
        <Qualifier>Ratio</Qualifier>
        <Units />
        <PropertyLabel>EEY</PropertyLabel>
    </Property>
    <Property>
        <Name>Incident Electron Energy</Name>
        <Description>Table of the incident electron energy</Description>
        <PropertyQuantity>Energy</PropertyQuantity>
        <Units>eV</Units>
        <PropertyLabel>E</PropertyLabel>
    </Property>
</InputProperties>
```

The simulation model outputs are specified as a collection of <Parameter> in the <OutputParameters> element.

```
<OutputParameters>
    <Parameter>
        <Name>Secondary Electron Emission Spectra</Name>
        <Description>Secondary emission yield curve</Description>
        <ParameterKey>SEEYETab</ParameterKey>
        <Units />
        <Particle>
            <ParticleType>Electron</ParticleType>
            <Qualifier>Ratio</Qualifier>
            <Qualifier>Fit</Qualifier>
            <ParticleQuantity>NumberFlux</ParticleQuantity>
        </Particle>
    </Parameter>
    <Parameter>
        <Name>Incident Electron Energy</Name>
        <Description>Table of the incident electron energy</Description>
        <ParameterKey>E</ParameterKey>
        <Units>eV</Units>
        <Particle>
            <ParticleType>Electron</ParticleType>
            <ParticleQuantity>Energy</ParticleQuantity>
        </Particle>
    </Parameter>
    <Parameter>
        <Name>Integrated secondary emission yield</Name>
        <Description>Secondary emission yield integrated over the secondary
            electron energy
        </Description>
        <ParameterKey>SEEY</ParameterKey>
        <Units />
        <Particle>
            <ParticleType>Electron</ParticleType>
            <Qualifier>Integral</Qualifier>
            <Qualifier>Ratio</Qualifier>
            <Qualifier>Fit</Qualifier>
            <ParticleQuantity>NumberFlux</ParticleQuantity>
            <ValidMin>0</ValidMin>
        </Particle>
    </Parameter>
    <Parameter>
        <Name>Material Work Function</Name>
        <ParameterKey>W</ParameterKey>
        <Units>eV</Units>
        <Matter>
            <MatterQuantityType>Electrical</MatterQuantityType><!-- Electrical /
                Thermal / Chemical /Structural -->
            <Qualifier>Characteristic</Qualifier>
            <Qualifier>Fit</Qualifier>
            <MatterQuantity>BindingEnergy</MatterQuantity>
            <ValidMin>0</ValidMin>
        </Matter>
    </Parameter>
</OutputParameters>
</Model>
```

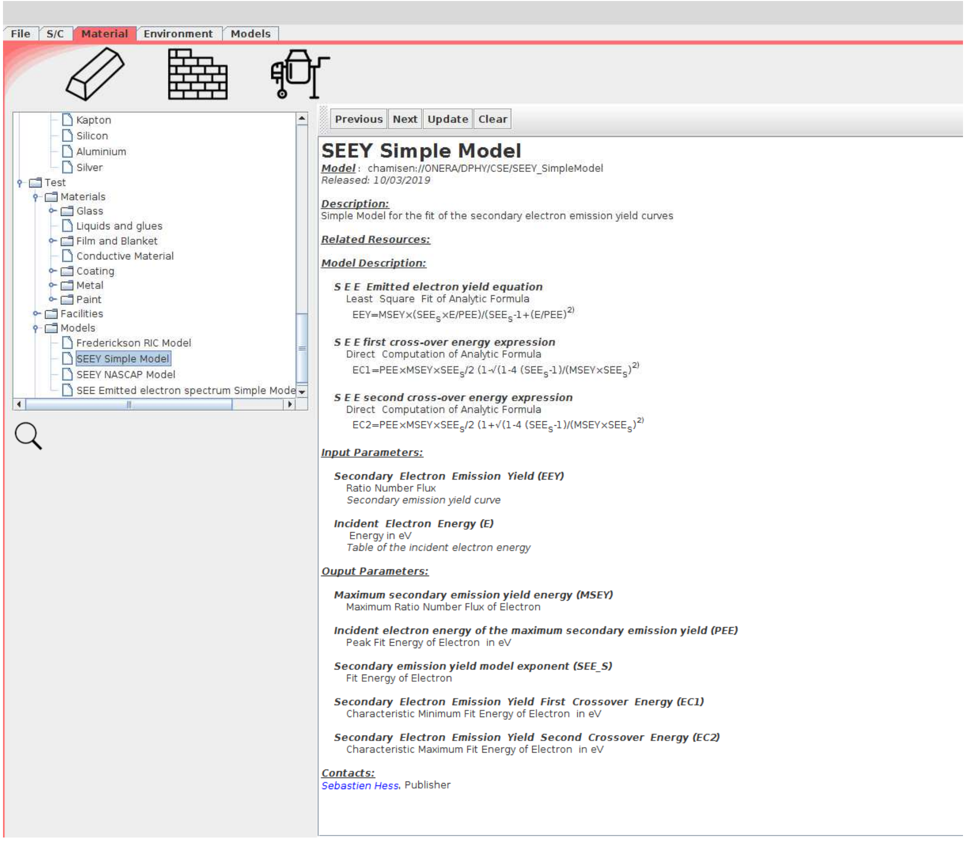

Figure 23: View of a Model resource in in ONERA‘s SoCCER (Solar Cell Charging and Electrostatic Risks) software which includes the Chamisen Library.

(g) SimulationModel

The SimulationModel resources are used to describe a numerical simulation model. In addition the human language description of the model in the <Description> element, the simulation type and the type of measurements and phenomena simulated can be described be setting the corresponding keywords in the <SimulationType>, <MeasurementType> and <PhenomenonType>, respectively.

The code language as well as a description of its spatial and temporal dependence can be provided too.

The simulation model inputs are specified as a collection of <Property> in the <InputProperties> element. If a value is attributed to a property, it defines its defaults value, while <ValidMin> and <ValidMax> define the range of valid input values.

The simulation model outputs are specified as a collection of <Parameter> in the <OutputParameters> element.

Finally, it is possible to provide an URL toward the model.

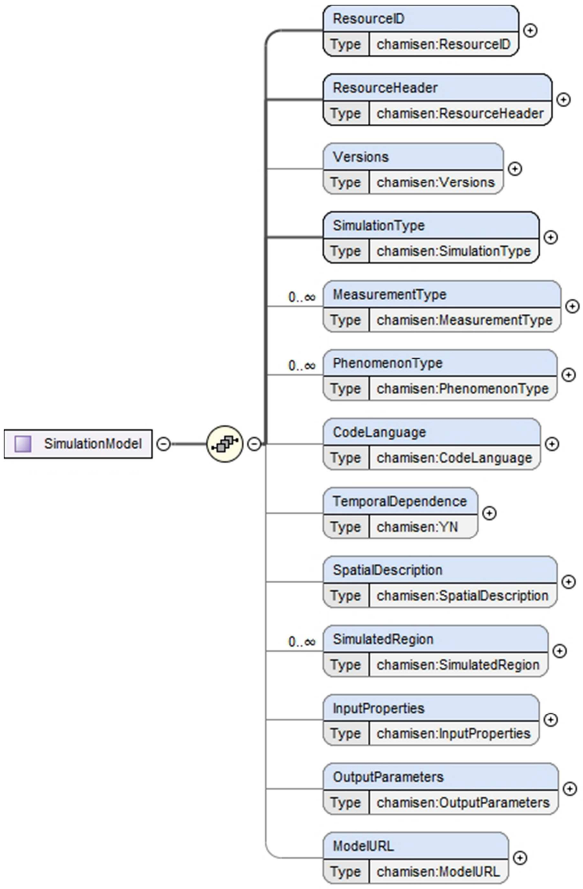

Figure 24: SimulationModel Resource structure

(h) Person

The Person resource is used to describe a physical person or organization. It is reminded the importance of preserving personal privacy and in particular European Commission's General Data Protection Regulation (GDPR) in collecting personal data. Although it is not this document purpose to expose the GDPR rules, it can be seen as the general policy that:

- Only essential information are to be stored *(is it necessary to have more than a name?)*
- The person must be informed that she/he appears in the database. Best practice, persons should be added by themselves or their employer: This has for side effect to help preventing the person to appear twice within the ChaMISEn system.
- A Data Protection Officer must be designated to handle data privacy.

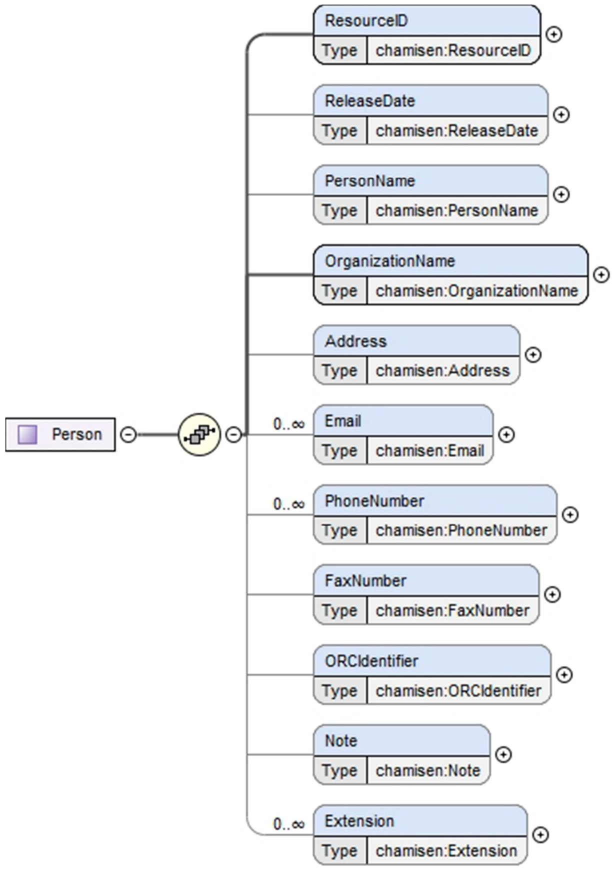

Figure 25: Person Resource structure

II.3.7 Infrastructure resources

(a) Registry

The Registry Resource is meant to specify the "registry" and "chamisen" service URLs.

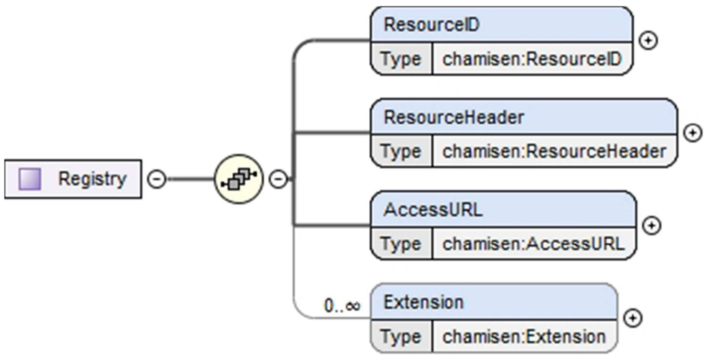

Figure 26: Registry Resource structure

(b) Repository

A Repository resource provides a description and a location for the repository archiving the data files.

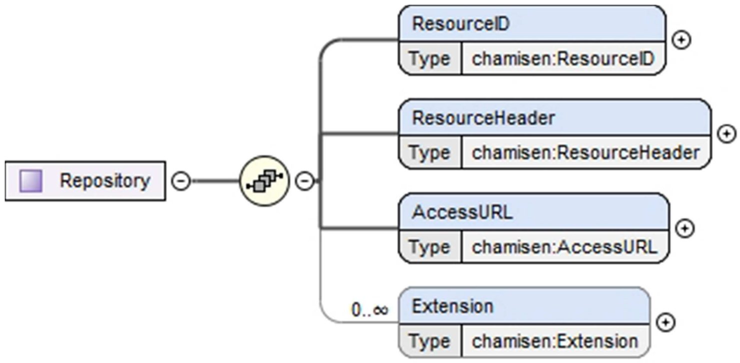

Figure 27: Repository Resource structure

(c) Service

A Service resource provides a description and an access to a web-service. In particular, access to the "mascat" table access shall be specified as a Service.

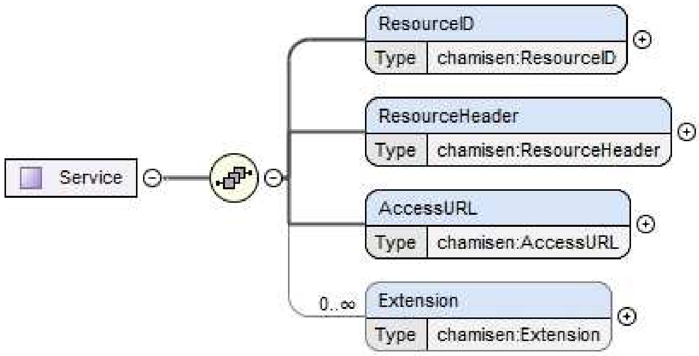

Figure 28: Service Resource structure

III Use cases

III.1 Published secondary measurements

This case takes the example of the archiving of published measurements performed by M. Belhaj from ONERA. These measurements concern secondary electron under electrons impact yield measurements performed in the DEESSE facility at ONERA.

The publication is described by a <Document> resource that provide a short description (or the abstract) of the publication, the list of authors, a link toward the publication (<AccessURL>) and a link toward the content (<Association>) which is here an experiment.


```
<Document>
    <ResourceID>chamisen://ONERA/DPHY/CSE/DESP14036.1408452594</ResourceID>
    <ResourceHeader>
        <ResourceName>Electron Emission Properties of Space Used Dielectric
            Materials
        </ResourceName>
        <ReleaseDate>10/03/2019</ReleaseDate>
        <Description>
            An experimental investigation of electron emission proprieties of spacecraft representative materia
            Onera under ESA contract N° 4000104742. Some of the results are reported. The measurements include
            emission yields as well as the energy distribution of the secondary electrons. The effect
            of the incident current flux on the EEY is also investigated.
        </Description>
        <Acknowledgement>
            The authors would like to thank ESA for RT funding to investigate material charging properties and
            Thales Alenia Space with Qioptiq that provided some of the studied samples.
        </Acknowledgement>
        <PublicationInfo>
            <Authors>M. Belhaj, T. Paulmier, D.Rodgers</Authors>
            <PublicationDate>2014-11-07</PublicationDate>
            <PublishedBy>HAL</PublishedBy>
        </PublicationInfo>
        <Funding>
            <Agency>European Space Agency</Agency>
            <Project>Project</Project>
            <AwardNumber>4000104742</AwardNumber>
        </Funding>
        <Contact>
            <PersonID>chamisen://ONERA/DPHY/CSE/MBELHAJ</PersonID>
            <Role>PrincipalInvestigator</Role>
        </Contact>
        <Contact>
            <PersonID>chamisen://ONERA/DPHY/CSE/SHESS</PersonID>
            <Role>Publisher</Role>
        </Contact>
        <InformationURL />
        <PriorID>HAL Id:hal-01081400</PriorID>
        <AccessRight>Open</AccessRight>
        <Association>
        <AssociationID>chamisen://ONERA/DPHY/CSE/ESA4000104742/CMX100UVS</AssociationID>
        <AssociationType>Other</AssociationType>
        </Association>
    </ResourceHeader>
    <AccessInformation>
        <RepositoryID>url:hal.archives-ouvertes.fr</RepositoryID>
        <Availability>Online</Availability>
        <AccessRight>Restricted</AccessRight>
        <AccessURL>
            <URL>https://hal.archives-ouvertes.fr/hal-01081400</URL>
            <Language>en></Language>
        </AccessURL>
        <Format>PDF</Format>
    </AccessInformation>
    <DocumentType>Report</DocumentType>
    <MIMEType>application/pdf</MIMEType>
</Document>
```

The experiment is described by an <Experiment> resource type. It provides the value of the experiment setting (<InputParameter>), the type of experiment and the representativity or degree of confidence in the experiment (<LikelihoodRating>).


```xml
<Experiment>
    <ResourceID>chamisen://ONERA/DPHY/CSE/ESA4000104742/CMX100UVS
    </ResourceID>
    <ResourceHeader>
        <ResourceName>Measurement of EEY properties of CMX100UVS (ESA #4000104742)
        </ResourceName>
        <ReleaseDate>10/03/2019</ReleaseDate>
        <Description>Experiment in conducted in DEESSE regarding the measurement of EEY properties of
        CMX-ITO performed by ONERA under ESA contract #4000104742 </Description>
        <Funding>
            <Agency>European Space Agency</Agency>
            <Project>?</Project>
            <AwardNumber>4000104742</AwardNumber>
        </Funding>
        <Contact>
            <PersonID>chamisen://ONERA/DPHY/CSE/MBELHAJ</PersonID>
            <Role>PrincipalInvestigator</Role>
        </Contact>
        <Contact>
            <PersonID>chamisen://ONERA/DPHY/CSE/SHESS</PersonID>
            <Role>Publisher</Role>
        </Contact>
        <InformationURL />
        <AccessRight>Open</AccessRight>
    </ResourceHeader>
    <ProtocolId></ProtocolId>
    <SampleID>chamisen://Qioptiq/Material/CMX100UVS</SampleID>
    <LikelihoodRating>Strong</LikelihoodRating>
    <FacilityID>chamisen://ONERA/DPHY/CSE/DEESSE</FacilityID>
    <InstrumentID>chamisen://ONERA/DPHY/CSE/DEESSE/RFA</InstrumentID>
    <ExperimentType>SecondaryElectronSpectroscopy</ExperimentType>
    <InputResourceID>chamisen://ONERA/DPHY/CSE/DEESSE/ELG2
    </InputResourceID>
    <InputResourceID>chamisen://ONERA/DPHY/CSE/DEESSE/ELG2022B
    </InputResourceID>
    <InputParameter>
        <Name>Electron beam </Name>
        <Description>Characteristics of the electron gun beams used to  measure the secondary electron yield
        </Description>
        <ParameterQuantity>Electron</ParameterQuantity>
        <Property>
            <Name>Pulse duration</Name>
            <Description>duration of the pulse from the electron beam
            </Description>
            <PropertyQuantity>Temporal</PropertyQuantity>
            <Units>s</Units>
            <ValidMin>5E-6</ValidMin>
            <ValidMax>20E-6</ValidMax>
        </Property>
        <Property>
            <Name>Beam incidence</Name>
            <Description>duration of the pulse from the electron beam
            </Description>
            <PropertyQuantity>ElevationAngle</PropertyQuantity>
            <Units>degrees</Units>
            <PropertyValue>0</PropertyValue>
        </Property>
    </InputParameter>
</Experiment>
```


It also provides the sample on which the experiment is performed (<MaterialID>). The materialID corresponds to the description of a particular CMX sample, which derives from a more generic CMX material, itself deriving from the more general Glass material.

```
<Material>
    <ResourceID>chamisen://ONERA/Material/Glass</ResourceID>
    <ResourceHeader>
        <ResourceName>Glass</ResourceName>
        <ReleaseDate>10/03/2019</ReleaseDate>
        <Description>Generic material for glasses</Description>
        <Contact>
            <PersonID>chamisen://ONERA/DPHY/CSE/SHESS</PersonID>
            <Role>Publisher</Role>
        </Contact>
        <InformationURL />
        <AccessRight>Open</AccessRight>
    </ResourceHeader>
    <ParentID />
    <Properties />
</Material>
<Material>
    <ResourceID>chamisen://ONERA/Material/CMX</ResourceID>
    <ResourceHeader>
        <ResourceName>CMX</ResourceName>
        <ReleaseDate>10/03/2019</ReleaseDate>
        <Description>Cerium doped glass mainly used for solar cell cover
            glasses
        </Description>
        <Contact>
            <PersonID>chamisen://ONERA/DPHY/CSE/SHESS</PersonID>
            <Role>Publisher</Role>
        </Contact>
        <InformationURL />
        <AccessRight>Open</AccessRight>
    </ResourceHeader>
    <ParentID>chamisen://ONERA/Material/Glass</ParentID>
    <Properties />
</Material>
<Material>
    <ResourceID>chamisen://Qioptiq/Material/CMX100UVS</ResourceID>
    <ResourceHeader>
        <ResourceName>CMX-100 UVS (Qioptic)</ResourceName>
        <ReleaseDate>10/03/2019</ReleaseDate>
        <Description>Cerium doped glass mainly used for solar cell cover
            glasses manufactured by Qioptic under the reference CMX-100 UVS used
            as solar reflector OSR
        </Description>
        <Contact>
            <PersonID>chamisen://ONERA/DPHY/CSE/SHESS</PersonID>
            <Role>Publisher</Role>
        </Contact>
        <InformationURL />
        <AccessRight>Open</AccessRight>
    </ResourceHeader>
    <ParentID>chamisen://ONERA/Material/CMX</ParentID>
    <Properties />
</Material>
```

The inheritance from sample to most generic materials is indicated in the <Material> resource by the <ParentID>.

The <Experiment> resource also points toward the facility in which it was performed (<FacilityID>)as well as the instruments and equipment used.

```xml
<Facility>
    <ResourceID>chamisen://ONERA/DPHY/CSE/DEESSE</ResourceID>
    <ResourceHeader>
        <ResourceName>DEESSE</ResourceName>
        <ReleaseDate>10/03/2019</ReleaseDate>
        <Description>Vacuum chamber for the precise characterization of the
            secondary electron emission (yield, spectrum)
        </Description>
        <Contact>
            <PersonID>chamisen://ONERA/DPHY/CSE/MBELHAJ</PersonID>
            <Role>PrincipalInvestigator</Role>
        </Contact>
        <Contact>
            <PersonID>chamisen://ONERA/DPHY/CSE/SDADOUCH</PersonID>
            <Role>TechnicalContact</Role>
        </Contact>
        <Contact>
            <PersonID>chamisen://ONERA/DPHY/CSE/SHESS</PersonID>
            <Role>Publisher</Role>
        </Contact>
        <InformationURL />
        <AccessRight>Open</AccessRight>
    </ResourceHeader>
</Facility>
<Instrument>
    <ResourceID>chamisen://ONERA/DPHY/CSE/DEESSE/RFA</ResourceID>
    <ResourceHeader>
        <ResourceName>DEESSE Retarding Field Analyzer</ResourceName>
        <ReleaseDate>10/03/2019</ReleaseDate>
        <Description>The retarding field analyzer can measure the energy
            spectrum and integrated flux
            of secondary electrons emitted by an
            irradiated target in DEESSE
        </Description>
        <Contact>
            <PersonID>chamisen://ONERA/DPHY/CSE/MBELHAJ</PersonID>
            <Role>PrincipalInvestigator</Role>
        </Contact>
        <Contact>
            <PersonID>chamisen://ONERA/DPHY/CSE/SDADOUCH</PersonID>
            <Role>TechnicalContact</Role>
        </Contact>
        <Contact>
            <PersonID>chamisen://ONERA/DPHY/CSE/SHESS</PersonID>
            <Role>Publisher</Role>
        </Contact>
        <InformationURL />
        <AccessRight>Open</AccessRight>
    </ResourceHeader>
    <InstrumentType>RetardingPotentialAnalyzer</InstrumentType>
    <InvestigationName>DEESSE</InvestigationName>
    <FacilityID>chamisen://ONERA/DPHY/CSE/DEESSE</FacilityID>
</Instrument>
<Equipment>
    <ResourceID>chamisen://ONERA/DPHY/CSE/DEESSE/ELG2</ResourceID>
    <ResourceHeader>
        <ResourceName>Kimpball ELG-2</ResourceName>
        <ReleaseDate>10/03/2019</ReleaseDate>
        <Description>Kimpball physics electron gun ELG-2 irradiating target
            between 1eV and 2keV with electrons in DEESSE
        </Description>
        <Contact>
            <PersonID>chamisen://ONERA/DPHY/CSE/MBELHAJ</PersonID>
            <Role>PrincipalInvestigator</Role>
        </Contact>
        <Contact>
            <PersonID>chamisen://ONERA/DPHY/CSE/SDADOUCH</PersonID>
            <Role>TechnicalContact</Role>
        </Contact>
        <Contact>
            <PersonID>chamisen://ONERA/DPHY/CSE/SHESS</PersonID>
            <Role>Publisher</Role>
        </Contact>
        <InformationURL />
        <AccessRight>Open</AccessRight>
    </ResourceHeader>
    <EquimentType>ElectronGun</EquimentType>
    <InvestigationName>DEESSE</InvestigationName>
    <FacilityID>chamisen://ONERA/DPHY/CSE/DEESSE</FacilityID>
</Equipment>
```

III.2 Material Characterization over several Samples

This case defines a material which characteristics are defined from measurements performed over several samples or derived materials. We take as an example the definition of a generic "Paint" material whose characteristics are derived from those of two of the conductive black paints characterized and distributed in the SPIS datasets (Aquadag and Electrodag).

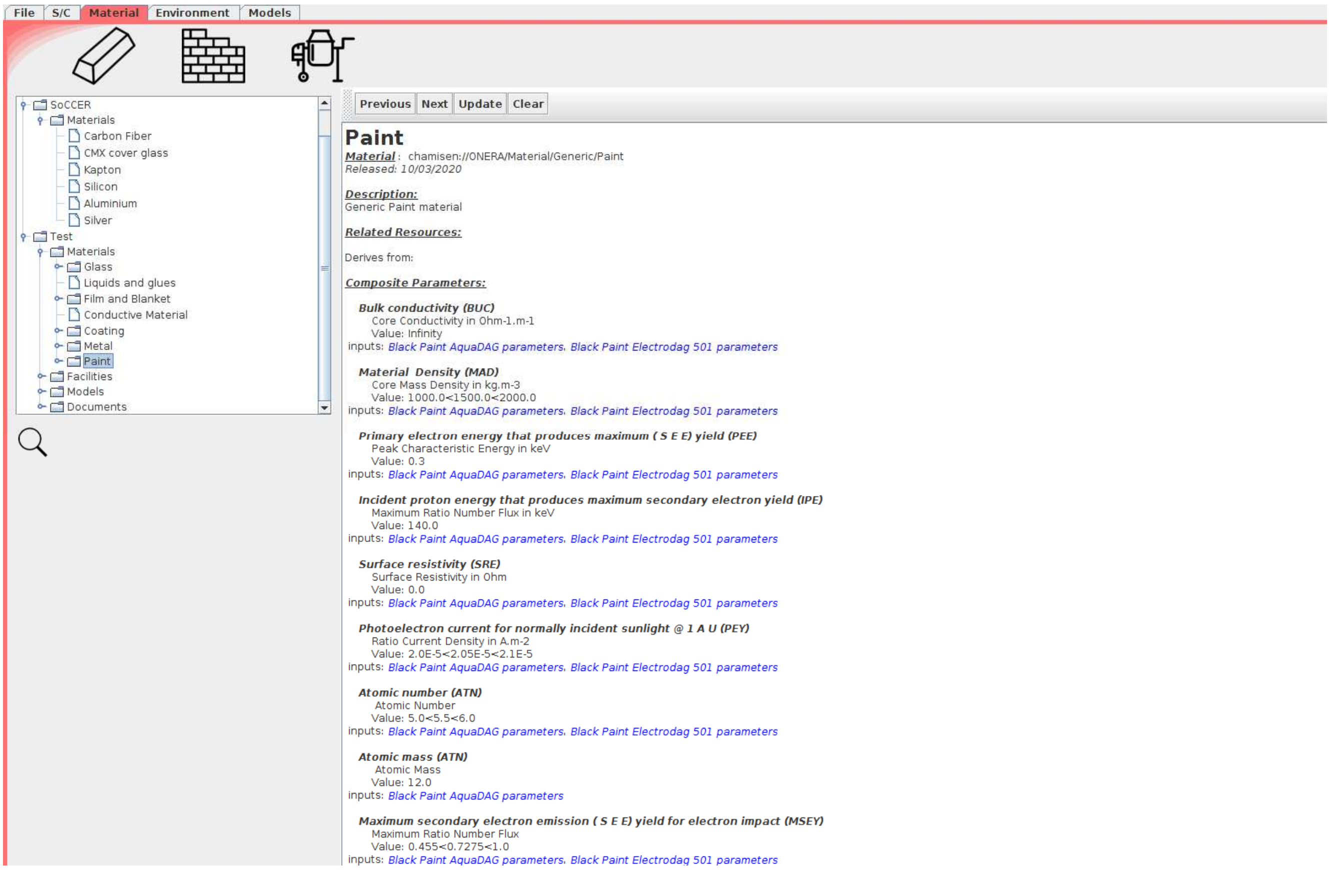

Figure 29: View of the Paint material datasets in ONERA's SoCCER (Solar Cell Charging and Electrostatic Risks) software which includes the Chamisen Library. The description of the material characteristics indicates the origin of the data. When several data sources are used, the minimum, averaged and maximum values are indicated.

III.3 ESD test measurements

This use case assumes that ESD triggering tests have been conducted over samples of different materials. The experiment protocol consists in gradually increasing the potential difference between the dielectric samples and the metallic substrate and to observe and count ESD triggered on each samples. The data file lists: the ESD occurrence time, the metallic plate potential bias, the ESD peak current, the id (number) of the sample.

This data file should be described in the database using a <Catalog> resource:

```
<Catalog>
  <ResourceID>chamisen://ONERA/Catalog/ESD/testxxx<ResourceID>
  <ResourceHeader>
       <ResourceName>ESD triggering test run XXX<ResourceName>
       …
       <Association>
         <AssociationID>chamisen://ONERA/Experiment/testxxx</AssociationID>
         <AssociationType>ObservedBy</AssociationType>
       </Association>
  </ResourceHeader>
  <AccessInformation>
       … (link toward the data file)
  </AccessInformation>
  <PhenomenonType>ESD<PhenomenonType>
  <InputResourceID>chamisen://ONERA/Material/Kapton/SampleXXX</InputResourceID>
  <InputResourceID>chamisen://ONERA/Material/Kapton/SampleXXY</InputResourceID>
  <InputResourceID>chamisen://ONERA/Material/Teflon/SampleXXZ</InputResourceID>
  <Parameter>
    <Name>Time</Name>
    <ParameterKey>time</ParameterKey>
    <Units>s</Units>
    <ValidMin>10</ValidMin> (assuming the first ESD happened after 10 seconds)
    <ValidMax>3600</ValidMax> (assuming the last ESD happened after 1 hour)
    <Support>
      <SupportQuantity>Temporal</SupportQuantity>
    </Support>
  </Parameter>
  <Parameter>
    <Name>Potential</Name>
```

```
      <ParameterKey>V</ParameterKey>
      <Units>V</Units>
      <ValidMin>1000</ValidMin> (assuming the first ESD happened at 1000V)
      <ValidMax>2600</ValidMax> (assuming the last ESD happened at 2600V)
      <Field>
        <FieldQuantity>Potential</ FieldQuantity >
      </ Field>
    </Parameter>
    <Parameter>
      <Name>Current</Name>
      <ParameterKey>I</ParameterKey>
      <Units>A</Units>
      <ValidMin>0<ValidMin> (assuming the first ESD happened at 1000V)
      <ValidMax>10<ValidMax> (assuming the last ESD happened at 2600V)
      <Field>
        <Qualifer>Peak</Qualifer>
        <FieldQuantity>Current</ FieldQuantity >
      </ Field>
    </Parameter>
    <Parameter>
      <Name>Sample</Name>
      <ParameterKey>id</ParameterKey>
      </Units>
      <RenderingHints>
          <Index>  (Material samples indexed by sample number 1, 2 and 3 in the data)
              chamisen://ONERA/Material/Kapton/SampleXXX;
              chamisen://ONERA/Material/Kapton/SampleXXY;
              chamisen://ONERA/Material/Teflon/SampleXXZ
        </Index>
      </RenderingHints>
      <ValidMin>1</ValidMin>
      <ValidMax>3</ValidMax>
      <Support>
        <Qualifer>Indexed</Qualifer>
        <SupportQuantity>Identifier</SupportQuantity >
      </Support>
    </Parameter>
</Catalog>
```

III.4 Secondary electron emission parameter extraction from measurements in a facility calibrated through simulation.

This use case consists in a more complex, though more realistic, case where a publication is made of the measurement of some material parameters of the Vaughan's model of SEEE secondary electron emission under electron impact. The experiment is conducted in a dedicated instrumented vacuum chamber (e.g. ONERA's DEESSE chamber) following a well stablished protocol involving a post-processing of the raw measurement to mitigate the sample charging based on an estimates of such effect obtained by numerical simulation using the SPIS software. The calibrated dataset is then fitted by a Vaughan's model using the COMPEX data extraction code to obtain the Vaughan's model parameters.

For archiving and further analysis, all the relevant information about how the publication results were obtained, including all raw, process and support data must be archived and accessible. This is permitted by the ChaMISEn data model as shown on Figure 30: Resources describing the Use Case 4.Figure 30.

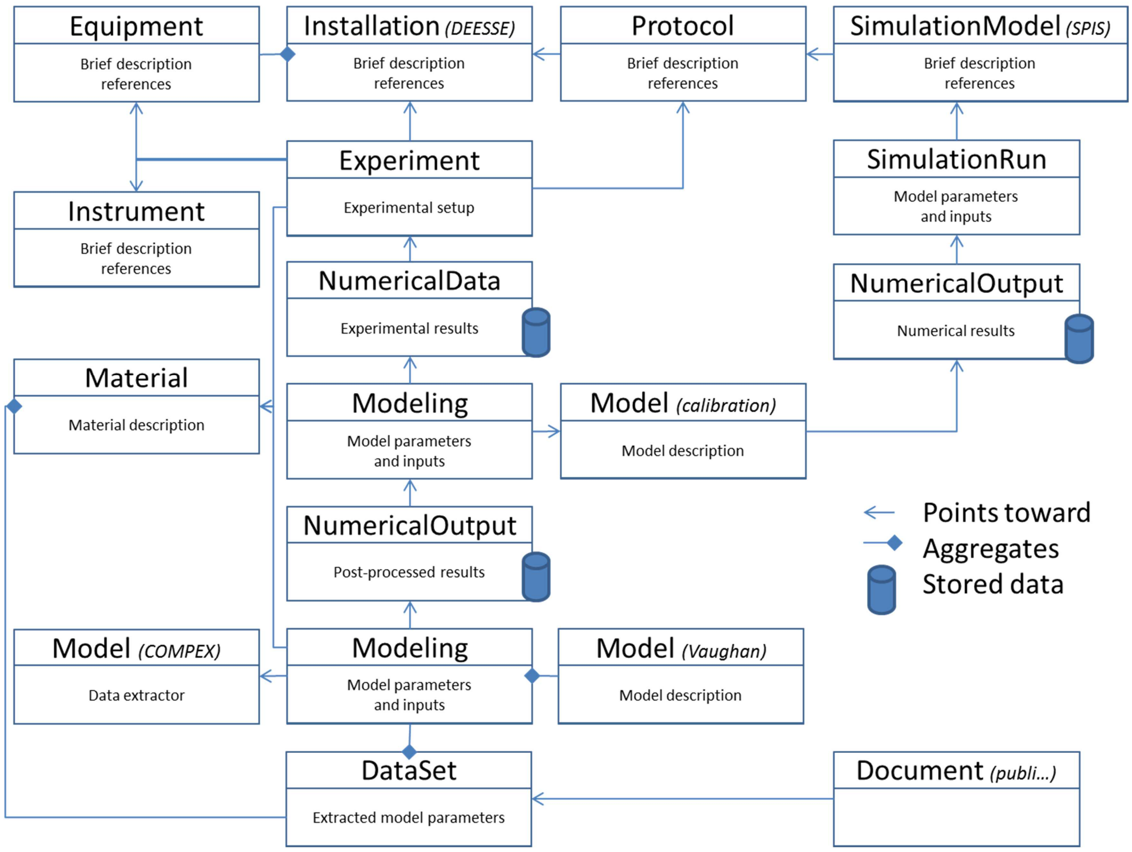

Figure 30: Resources describing the Use Case 4.

IV Database manager specifications

In order to fulfill most of the common needs expressed by users of a material database, while keeping a global coherence in the query language, the database manager is required to offer different levels of access to the data and metadata through a single SQL-like request format. The three levels can be selected as table names in the SQL request and are:

- registry: a relational database that lists all resources and data files in the database indexed by a unique identifier whose attributes corresponds to the ISO 15836 fields with the addition of two fields: a URL targeting either the data file or the metadata xml description of the resource and the expiration date for deprecated resources.

- mascat: (material simple catalog) a relational database that list all materials indexed by their unique identifier with a selection of some of their scalar characteristics. The list of these characteristics corresponds to the NASCAP/SPIS/Maptis properties and the attribute name must corresponds to the Maptis specification to ease interoperability

- chamisen: the full object database. This object is a pseudo table: it should appear in place of the table name in SQL request, but the object nature of the database prevents it to be effectively represented as a table.

V Query language specification

V.1 Database modification

V.1.1 *registry* and *mascat* Tables

Within the ChaMISEn framework, the "registry" and "mascat" tables cannot be modified directly as they are extracted subsets of the information contained in the "chamisen" pseudo-table. As the global coherence of the database must be preserved, the "registry" and "mascat" tables must be updated by the database manager consistently with the "chamisen" modifications.

V.1.2 *chamisen* Table

A specificity of the ChaMISEn database is that it is composed of polymorph objects rather than standardized records. These objects are immutable: they can be created, but not deleted nor modified. This addresses the need of persistence of the ChaMISEn database: if some uses are made of the data or metadata at a given time, it must be possible to retrieve this exact data/metadata anytime later, even if correction/deletion were made in between. Instead of deletion, the data is marked as expired by filling the Resource Header’s <ExpirationDate> field (which is the sole modification that is accepted for a resource). Updated resource should point toward the deprecated resource they replace by filling the Resource Header’s <PriorID> field.

(a) INSERT INTO

To add a resource in a ChaMISEn database, use the INSERT INTO statement.

Syntax: INSERT INTO chamisen (*Resource*) {AS *newID*};

The *Resource* field is an XML object (coded as a UTF-8 string). The AS keyword allows specifying the updated resource ID. Otherwise, the resource ID is that given in the XML resource. The database manager must check that:

- the XML is well formed,
- that it complies with the ChaMISEn specifications
- and that the resource ID is unique,
- define the new resource ID and check that it is unique,
- add the new resource in the database,
- update the *registry* and eventually the *mascat* tables (the latter only if the resource is a <Material> or a <DataSet> resource). These updates may not be instantaneous, and the mascat update may not be automatic.

(b) UPDATE

To update a resource in a ChaMISEn database, use the UPDATE statement.

Syntax: UPDATE chamisen SET *oldID=(Resource)* {AS *newID};*

The *Resource* field is an XML object (coded as a UTF-8 string). The AS keyword allows to specify the updated resource ID. Otherwise, the resource ID is that given in the XML resource (if different from oldID) or the oldID augmented by an incremental version number (if the new resource ID it the same than oldID).

The database manager must check:

- check that the XML is well formed,
- check that it complies with the ChaMISEn specifications,
- check that a resource identified by the oldID exists,
- check that the new and old resources are of the same kind,
- attribute the current time to the old resource <ExpirationDate>,
- define the new resource ID and check that it is unique,
- attribute the old resource ID to the new resource <PriorID>,
- add the new resource in the database,
- update the registry and eventually the mascat registry (the latter only if the resource is a <Material> or a <DataSet> resource). These update may not be instantaneous, and the mascat update may not be automatic.

(c) DELETE FROM

In is impossible to deleted a resource in ChaMISEN, instead the resource is deprecated. To deprecate a resource, use the DELETE FROM statement.

Syntax: DELETE FROM chamisen {WHERE condition}*;*

This statement deprecates all resources that match the condition. If the WHERE statement is omitted, all resources from the database are deprecated. The DELETE without WHERE statement may be forbidden by the database manager in order to prevent from errors.

The database manager must check:

- filter the database resource following the WHERE statement,
- attribute the current time to the deprecated resource <ExpirationDate>,
- update the registry and eventually the mascat registry (the latter only if the resource is a <Material> or a <DataSet> resource). These update may not be instantaneous, and the mascat update may not be automatic.

V.2 Database interrogation

V.2.1 Registry and mascat databases

The “registry” and “mascat” databases are simple relational database, which should be interrogated using standard SQL (nonetheless excluding to possibility of table modification, see previous section). The specification hereafter concerns the minimal SQL language elements to be implemented to handle the “chamisen” pseudo-table.

V.2.2 SELECT … FROM chamisen

This statement is used to select data from the metadata. The selected data is stored in a temporary pseudo-table.

Syntax: SELECT {TOP N} [*|field1, field2,…] FROM chamisen {[WHERE|ORDER BY]}

If the * symbol is used, SELECT returns resources in a ChaMISEn compliant format, while if fields are specified it only returns the requested fields content XML elements in a VOTable <TABLE> element format. The TOP keyword allows returning only the N first results. In the case where fields are specified, it is possible to specify aliases for the columns and table with the AS keyword.

Examples:

The request:

```
SELECT * FROM chamisen WHERE Contact.Publisher = “chamisen://PERSON/John.Smith”
```

will return:

```
<Chamisen>
 <NumericalData>
  <ResourceID> chamisen://AUTHORITY/SomeResourceID1</ResourceID>
  <ResourceHeader>
               …
    <Contact>
        <PersonID> chamisen://PERSON/John.Smith </PersonID>
        <Role>Publisher</Role>
    <Contact>
  </ResourceHeader>
        …
 </NumericalData >
 <Experiment>
```

```
    <ResourceID> chamisen://AUTHORITY/SomeResourceID2<ResourceID>
    <ResourceHeader>
                ...
      <Contact>
          <PersonID> chamisen://PERSON/John.Smith </PersonID>
          <Role>Publisher</Role>
      <Contact>
    </ResourceHeader>
          ...
  </ Experiment>
</Chamisen>
```

The request:

SELECT ResourceID AS id FROM chamisen WHERE Contact.Publisher="chamisen://PERSON/John.Smith"

will return :

```
<VOTABLE>
  <RESOURCE name="ResultSet">
   <TABLE name="request-012">
     <DESCRIPTION>
          SELECT ResourceID AS id FROM chamisen WHERE Contact.Publisher =
          "chamisen://PERSON/John.Smith"
     </DESCRIPTION>
     <FIELD ID="id" datatype="char"/>
      <DATA>
       <TABLEDATA>
         <TR><TD>chamisen://AUTHORITY/SomeResourceID1</TD</TR>
         <TR><TD>chamisen://AUTHORITY/SomeResourceID2</TD></TR>
       </TABLEDATA>
     </DATA>
    </TABLE>
  </RESOURCE>
</VOTABLE>
```

Note:

The latest request example should rather be performed on "registry" instead of "chamisen" for faster results.

V.2.3 WHERE

The WHERE statement is used to filter a pseudo-table, it is followed by a condition.

Syntax: WHERE *condition*

The condition is generally expressed under the form: Field <operator> Value

Field corresponds to the field name in the ChaMISEn datamodel, with eventual subfields separated with dots (ex: AccessInformation.AccessURL.Language).

The resource type can be expressed as the ResourceType field (ex: ResourceType="NumericalData").

When the targeted field is in the <ResourceHeader>, ResourceHeader can be omitted (ex: ResourceName instead of ResourceHeader.ResourceName).

If the target field is an object with a value (e.g. parameters, properties…) the operator is applied to the object value. Else, if the field has an identifier of any kind, the operator tests the identifier as a string (ex: Contact=XXX is the same than Contact.PersonID=XXX).

When the targeted field is an object associating a value with an enumeration, it is possible to test both the value and the type by setting Field.type <operator> value (ex: Contact.Publisher=XXX matches contact with PersonID=XXX and Role=Publisher).

Parameters and properties can both be addressed with the Parameter field. This only addresses "Output" parameters (from NumericalData/Ouput and DisplayData/Output) or Dataset properties, not "Input" Parameter or properties. For "Input" parameters or properties, replace "Parameter" by "InputParameter". The possible subfields are:

- Parameter.Name : points toward the parameter or property <Name>
- Parameter.QualifiedQuantity: points toward the concatenation of the parameter or property <Qualifer> and <*Quantity>, the qualifiers coming first.
- Parameter.Key : points toward the <ParameterKey> or <PropertyLabel>
- Parameter.Type: cannot be used for properties. Corresponds to the parameters type (see section II.3.2(b)). If the parameter type has a subtype defined (ex: Particle as a ParticleType), the type is the concatenation of the parameter type and the subtype (in this order).
- Parameter.Value: The parameter or property value
- Parameter.*fieldName*: any other field named fieldname that exists for the parameter or property
- Parameter.'*identifier*': points toward the value of a parameter or property which key, qualified name or name (in this order) matches the 'identifier' string. It is possible to specify whether the key (Parameter.Key.'*identifier*'), the name (Parameter.Name.'*identifier*') or the qualified name (Parameter.QualifiedQuantity.'*identifier*') must be targeted.

(a) =,<>,!=

To test whether a field has a given value, the "=" operator must be used. To test whether a field does not have a given value, the "!=" or the "<>" operator must be used. These operators can be applied on fields of any type.

If the field type is a string, the operator test for an exact match, disregarding leading and trailing blank characters as well as carriage return, tabulations and new line characters.

If the field is a numerical value, the operators test the value equality with the precision of the specified requested Value.

If the field is an object with a value (e.g. parameters, properties…) the operator is applied to the object value. Else, if the field has an identifier of any kind, the operator tests the identifier as a string (see examples in section V.2.2).

(b) <,>,>=,<=

When the field is a numerical value or is an object with a value (e.g. parameters, properties,…), it is possible to test for inequalities using the following operators:

“<” test if the field value is strictly lower than the specified Value

“>” test if the field value is strictly larger than the specified Value

“<=” test if the field value is lower than or equal to the specified Value

“>=” test if the field value is larger than or equal to the specified Value

If the field is a numerical value, the operator tests the inequality with the precision of the specified requested Value.

If the field is an object with a value (e.g. parameters, properties…) the operator is applied to the object value.

(c) IN

The IN keyword allows one to test whether the field value equals a value in the list following the value keyword.

Syntax: WHERE Field IN (value1, value2,…)

If the field type is a string, the operator test for an exact match, disregarding leading and trailing blank characters as well as carriage return, tabulations and new line characters.

If the field is a numerical value, the operators test the value equality with the precision of the specified requested Value.

If the field is an object with a value (e.g. parameters, properties…) the operator is applied to the object value. Else, if the field has an identifier of any kind, the operator tests the identifier as a string (see examples in section V.2.2).

(d) BETWEEN

The IN keyword allows one to test whether the field numerical value is comprise between the specified min value and max value.

Syntax: WHERE Field BETWEEN min AND max

The operators test the value inequality with the precision of the specified requested min and max values.

If the field is an object with a value (e.g. parameters, properties…) the operator is applied to the object value.

(e) LIKE

Syntax: Field LIKE string

The LIKE keyword is used to test whether a field string content matches or not the specified string.

If the field is an object with a value (e.g. parameters, properties…) the operator is applied to the object value. Else, if the field has an identifier of any kind, the operator tests the identifier as a string (see examples in section V.2.2).

(f) AND, OR, NOT

Several conditions may be combined using simple Boolean logic operators AND, OR and NOT.

(g) EXISTS

The EXISTS keywork is used to test whether parameters or properties are defined in a resource. It is possible to test is parameters are absent with NOT EXISTS

Syntax: EXISTS(Field,…)

SELECT * FROM chamisen WHERE EXISTS (Parameter."Bulk Conductivity")

Returns all resources that have a parameter which name or label is "Bulk Conductivity" or which qualified description is "Bulk Conductity" (Qualifer:Bulk and Quantity:Conductivity)

V.2.4 ORDER BY

Syntax: ORDER BY field {ASC|DESC}

The ORDER BY keywork is used to organize request results based of a field value.

If the field type is a string, the operator test for an alphabetical order, disregarding leading and trailing blank characters as well as carriage return, tabulations and new line characters.

If the field is a numerical value, the operators test the value inequality with the precision of the specified requested Value.

If the field is an object with a value (e.g. parameters, properties…) the operator is applied to the object value. Else, if the field has an identifier of any kind, the operator tests the identifier as a string (see examples in section V.2.2).

VI – Software Suite

VI.1 ChaMISEn Library™

The ChaMISEn libray is a software library that provides tools to access and parse ChaMISEn and SPASE database and display information on resources. It depends on the ONERA toolkit library, both available under a CeCILL-C licence on the ChaMISEn website (http://onera.fr).

VI.2 ChaMISEn Desktop™

The ChaMISEn Desktop is a simple Graphic User Interface (GUI) that uses the ChaMISEn Library to explore ChaMISEn and SPASE databases. It is based on the ONERA Graphical Environment (OGrE) which is a customizable application that can easy be extended thanks to plugins. This software is under CeCILL license and can be downloaded from the ChaMISEn website (http://onera.fr).

It is possible to navigate between resources either by exploring the resource tree in the side bar, or by following hypertext links in each resource descriptions.

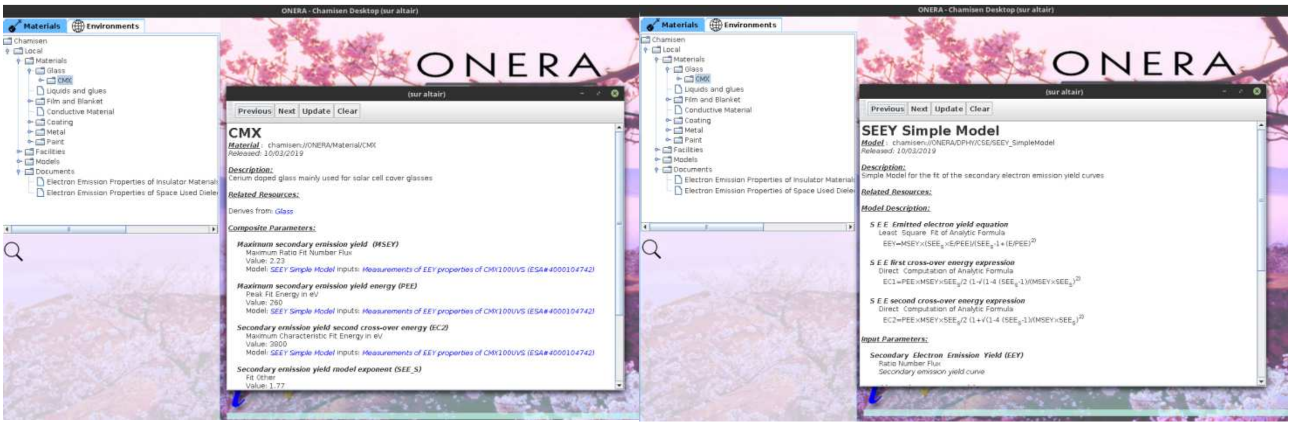

Figure 31: Screenshots of the ChaMISEn Desktop

VI.3 JAVA Data Base Connectivity Driver

The JAVA Data Base Connectivity (JDBC) Driver for ChaMISEn is a software component that allows a standardized connectivity between different DataBase Management Systems. It allows many SQL clients (Microsoft® SQL Server®, MariaDB® Server, SQuirreL Client®…) to connect to a ChaMISEn database and to interrogate it through SQL queries. The JDBC Driver is available on the ChaMISEn website (http://onera.fr) under CeCILL-C licence.

VII ChaMISEn Mapping to other data formats

VII.1 Mapping to Artenum's FRIDA format for SPIS

The Spacecraft-Plasma Interaction Software (SPIS) is a simulation software that allows to compute many of the interactions that a spacecraft (or more generally speaking, a surface) may have with the space environment (http://spis.org). Although primarily targeting the computation of the surface charging due to the exchange of electrical charges with the plasma, it also handles ESD triggering, contamination, erosion, internal charging,…

The ChaMISEn Library includes a Frida to ChaMISEn mapper that allows translating SPIS datasets into ChaMISEn resources (Figure 32).

We provide hereafter a mapping between the SPIS material characteristics defined in Artenum's Frida 0.1 format and their description as properties in ChaMISEn.

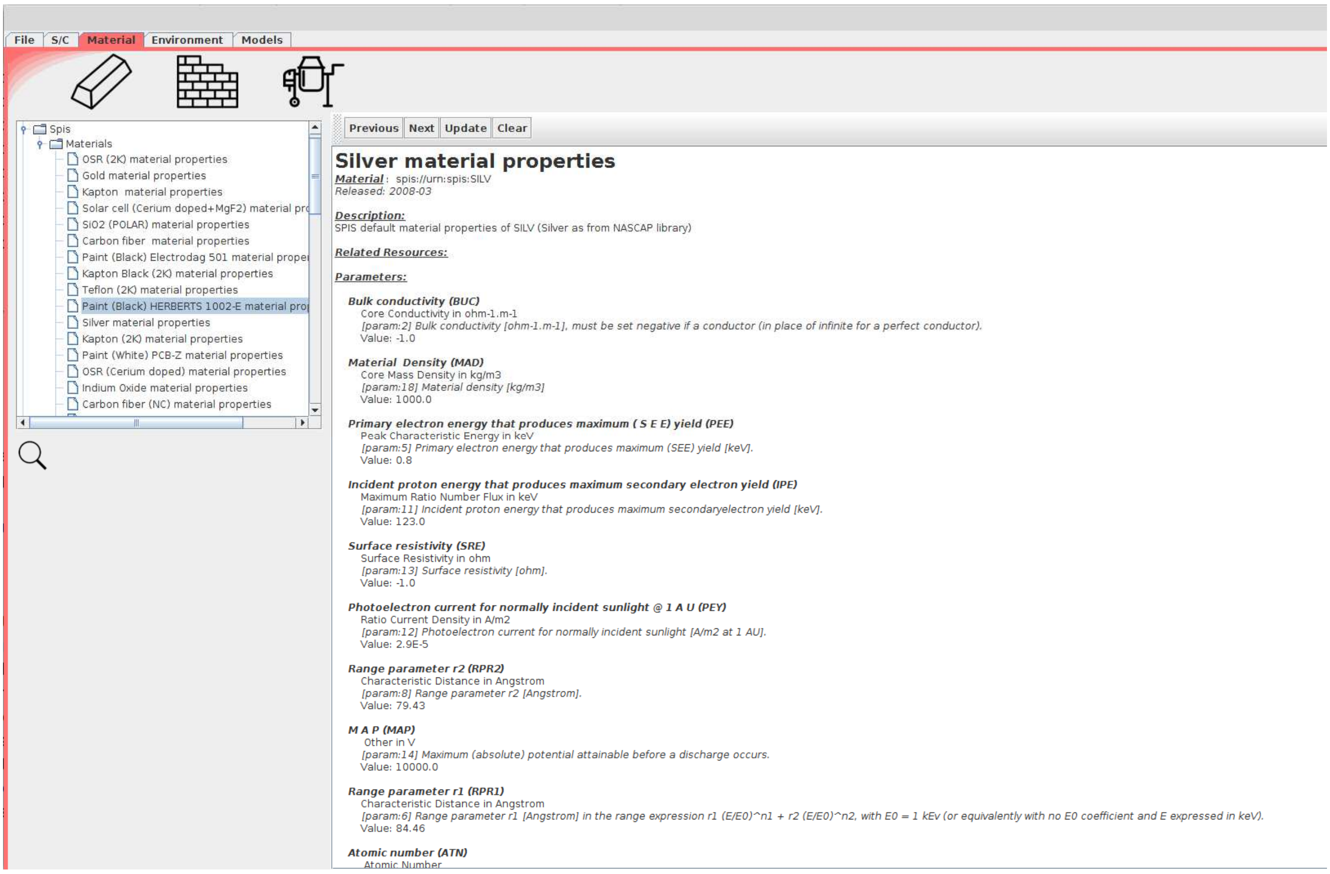

Figure 32: View of SPIS material datasets in ONERA's SoCCER (Solar Cell Charging and Electrostatic Risks) software which includes the Chamisen Library.

VII.1.1 Basic SPIS material characteristics

ChaMISEN			FRIDA – SPIS	
Key	**Quantity**	**Qualifier(s)**	**Type**	**Name**
BUC	Conductivity	Core	Double	BUC
MAD	MassDensity	Core	Double	MAD
PEE	Energy	SEEE; Characteristic; Peak	Double	PEE
IPE	NumberFlux	SEEEI; Maximum; Ratio	Double	IPE
SRE	Resistivity	Surface	Double	SRE
WORK	Energy	Surface; Binding	Double	WORK
PEY	CurrentDensity	PhotoEmission; Ratio	Double	PEY
ATN	AtomicNumber		Double	ATN
MSEY	NumberFlux	SEEE; Maximum; Ratio	Double	MSEY
RCP	Other	RIC; Exponent	Double	RCP
RCC	Conductivity	RIC;	Double	RCC
SEY	NumberFlux	SEEI; Ratio	Double	SEY
RDC	Permittivity	Ratio;	Double	RDC

RPR1	Distance	SEEE; Characteristic	Double	RPR1
RPN1	Other	SEEE; Exponent	Double	RPN1
RPR2	Distance	SEEE; Characteristic	Double	RPR2
RPN2	Other	SEEE; Exponent	Double	RPN2
ATOM	MassNumber		Double	ATOM
SUBE	Energy	Sublimation	Double	SUBE
VAPE	Energy	Vaporization	Double	VAPE
FUSE	Energy	Melting	Double	FUSE
MELT	Temperature	Melting	Double	MELT
SFTN	Energy	Surface; Characteristic	Double	SFTN
TEMI	Emissivity	Temperature; Characteristic	Double	TEMI
THCO	ThermalConductivity		Double	THCO
HEAT	SpecificHeat		Double	HEAT
MAP	Potential	ESD;Maximum	Double	MAP
MPD	Potential	ESD;Maximum; Differential	Double	MPD
ACEN	Energy	Characteristic	Double	ACEN
ALPHA	Rate	Recombination	Double	ALPHA

TREF	Temperature	Reference	Double	TREF
PREF	Pressure	Reference	Double	PREF
EGAP	Energy	Binding	Double	EGAP
TAUN	Temporal	Characteristic; Binding	Double	TAUN
TAUP	Temporal	Characteristic; Binding	Double	TAUP
TAUNT	Temporal	Characteristic; Binding	Double	TAUNT
TAUPT	Temporal	Characteristic; Binding	Double	TAUPT
R0	Distance	Characteristic; Binding	Double	R0

VII.1.2 Model-dependent SPIS characteristics

The SPIS software is quite versatile and allows using different models to describe the interaction of materials with the space environment. The different models and their entries are specified using the SPASE format in the "spase" plugin for SPIS numerical core (available under CeCILL License). Specific conventions that are use to apply the SPASE standard to SPIS are provided in the technical documentation provided with the plugin.

It is possible to provide the most adapted material characteristic sets to SPIS based on the description of the model it accepts (which may be dependent on the available plugin-dependent models). In this case, it is recommended for each model of SPIS to check whether all required entries are available in the ChaMISEn database:

1. Giving preference to properties that explicitly refer to the model identifier (see section II.3.2(a))

2. Relying on qualified quantity description and label that match that of the model inputs rather than on the names.
3. Giving preference to properties that belong to the material rather than inherited/aggregated properties