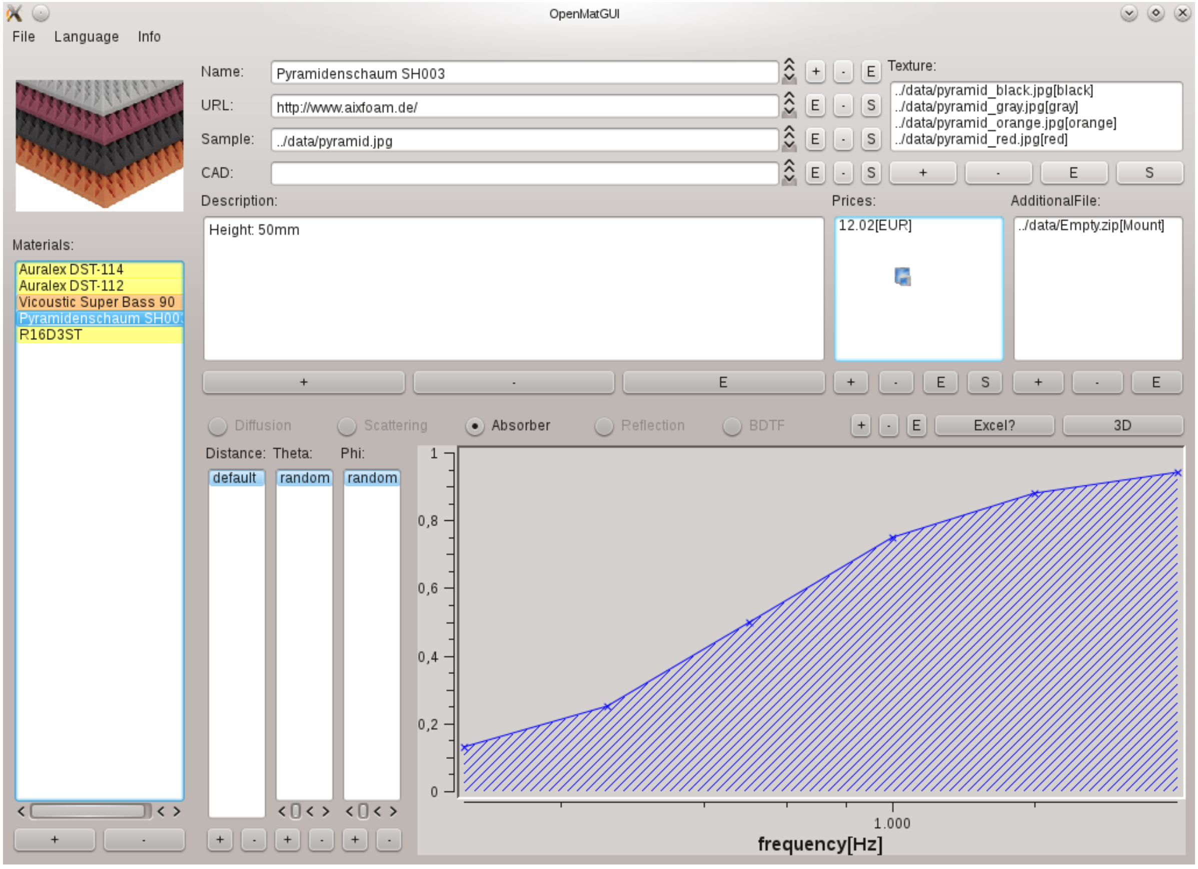
openMat

a XML-database for describing the acoustical properties of materials



OpenMatGUI

Name: Pyramidenschaum SH003

URL: http://www.aixfoam.de/

Sample: ../data/pyramid.jpg

CAD:

Texture:
../data/pyramid_black.jpg[black]
../data/pyramid_gray.jpg[gray]
../data/pyramid_orange.jpg[orange]
../data/pyramid_red.jpg[red]

Description: Height: 50mm

Prices: 12.02[EUR]

AdditionalFile: ../data/Empty.zip[Mount]

Materials:
Auralex DST-114
Auralex DST-112
Vicoustic Super Bass 90
Pyramidenschaum SH00:
R16D3ST

○ Diffusion ○ Scattering ● Absorber ○ Reflection ○ BDTF

Distance: default

Theta: random

Phi: random

The *XML* files described by *OpenMat* (see Sec. 2.1) can be viewed and modified by a graphical user interface (GUI) called *openMatGUI* (see Sec. 1). This interface is written in *C++* using *Qt* libraries, such that it can be built for multiple platforms (Windows, Linux and MacOS were tested). This user manual gives a short overview on the usage of the *OpenMatGUI*.

1 openMatGUI

The graphical user interface is composed by different parts. A menu bar (green) can be used to configure the overall behavior of the GUI. In the left most part (red), a list of materials of the currently loaded database is listed. In the upper part of the window, information on the material is displayed (blue), whereas the lower part is reserved for the numerical data (brown). Finally, in the lower right part of the window, a visualization of the numerical data is shown (purple).

1.1 Menu Bar

In the menu, three entries can be found. In the left most corner, the 'File' selection allows the user to either create a new database, load an existing database, save the

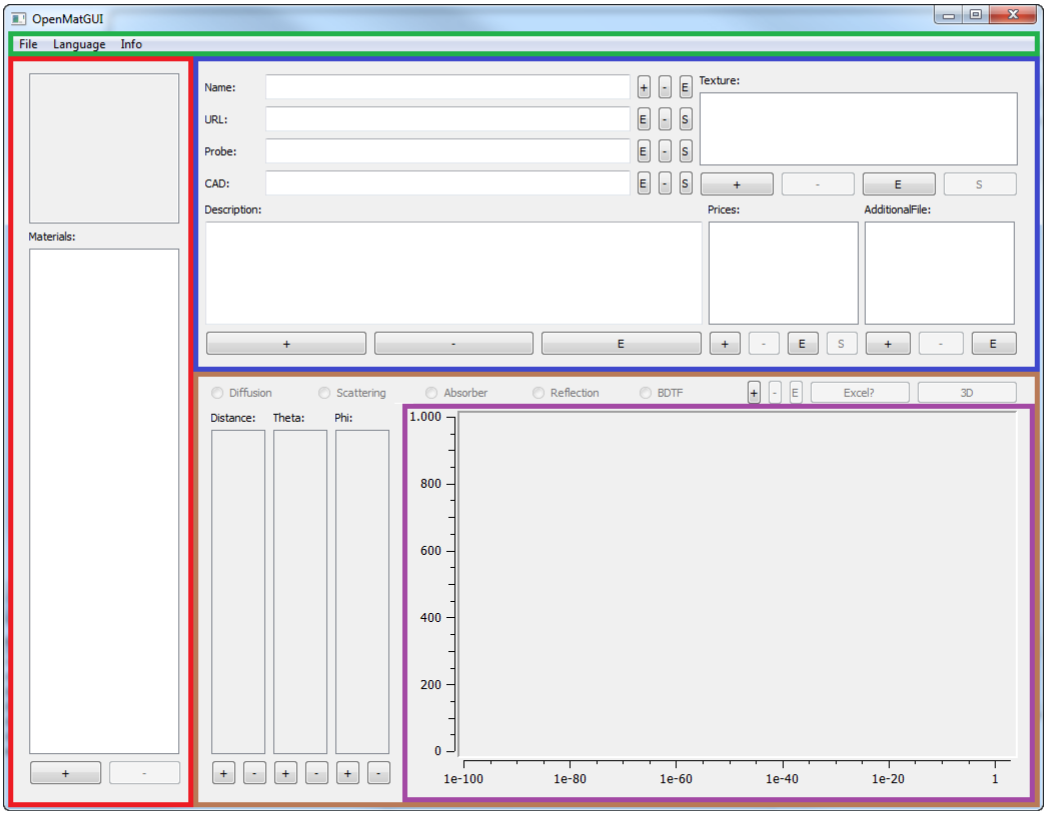
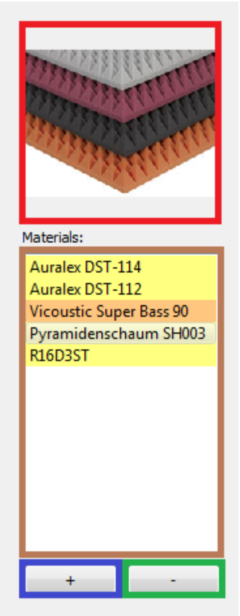


Figure 1.1: Screenshot of the graphical user interface *OpenMatGUI*

current database or quit the program. When a new database is created, an empty database is initialized internally. No *XML* file is created unless the database is *saved*.

If the user wants to load a database, a file dialog is shown for choosing the respective XML-database from the file system. Before the database is loaded, the *XML* file is validated using the *XSD* schema file containing the database restrictions (see Sec. 2.1). Finally, all materials that are encapsulated in the *XML* file are loaded and displayed by means of a *GUI*. Here, the first material is selected as default. The *save*-dialog asks the user to choose a database file. If an existing file is chosen, this file will be overwritten. In case of a non-existing file, a new file will be created. Furthermore, single materials of a given database can be either imported or exported using the respective windows (more details on their usage are given in 1.7). The language can be changed by using the 'Language' tab. This setting does not change the GUI itself, but changes the displayed material values by part. Both name and description entries in the database are available in different languages (optional) and will be display in the selected language. If, however, no entry for that language is available, the (required) English version is displayed.

1.2 Material Selection

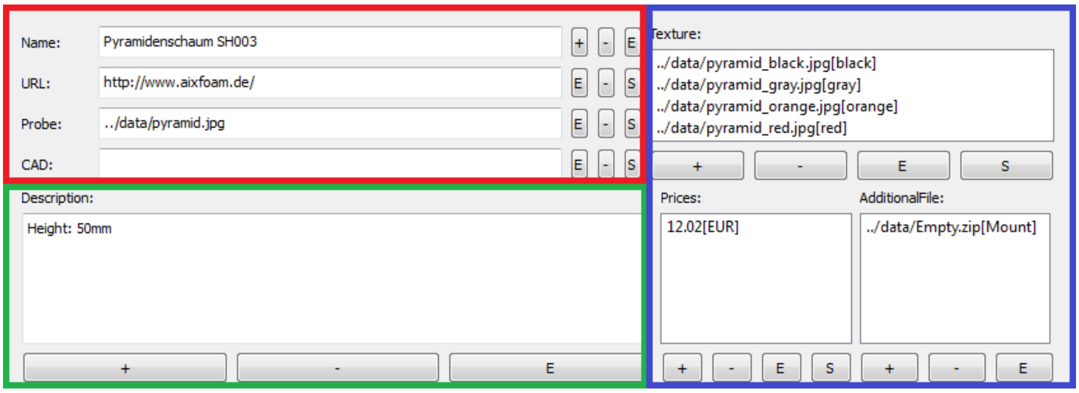


In the upper part of the material widget, the sample image of the material (red) is shown (only if available). All materials of the active database are listed in the material list, where different background colors can be identified: the yellow color indicates that the material has an infinite extent, i.e., acoustic foam (type: material), whereas the orange color indicates an object of finite extent, i.e., bass trap (type: object). The type of the material can be edited by double-clicking on a specific material. Note that the name of the material depends on the chosen language.

The lower buttons '+' and '-' add and remove a material, respectively. If a material is added, both the type and the name (in English) have to entered before the material is created. If a material is selected, it can also be deleted by pressing the '-' button.

1.3 Material Information

The material information widget consists of eight parts. In the upper left corner, the name of the material, the URL, the address of the sample image and the link(s) to (a) CAD file(s) are shown (red). The name is displayed in the selected language. By pressing the 'E' button, the name of the selected language can be edited ('E' for 'Edit'), while '+' and '-' button ask the user for a language entry to add or remove. For the



Name: Pyramidenschaum SH003

URL: http://www.aixfoam.de/

Probe: ../data/pyramid.jpg

CAD:

Description:

Height: 50mm

Texture:

../data/pyramid_black.jpg[black]
../data/pyramid_gray.jpg[gray]
../data/pyramid_orange.jpg[orange]
../data/pyramid_red.jpg[red]

Prices:

12.02[EUR]

AdditionalFile:

../data/Empty.zip[Mount]

Figure 1.2: Screenshot of the material information widget

links to the URL, sample image and CAD file, the 'E' button allows the user to edit the entry (file dialog for sample- and CAD-files), whereas the '-' button resets/deletes these entries. The button 'S' ('S' for 'Show') opens an additional window that browses to the given URL (Internet required), shows the sample image in original size and opens the folder containing the CAD file, respectively.

In the lower left corner, the description of the material is displayed in the selected language. The buttons behave in the same manner as described above. In the right part, the three widgets present the optional texture and price information together with additional files. Both the texture and the additional files are identified by a tag. In the case of textures, this tag is the color, while free key tags can be chosen for additional files. For the prices, more information regarding date and source is available by clicking the 'S' button. All values are editable in a separate window by clicking the 'E' button.

1.4 Material Data

In the first row of the numerical data widget, it can be chosen between the possible data types. Unset data types are disabled. The currently selected data type can be deleted by pressing the '-' button, whereas the button '+' adds a new data type (which can be chosen in an additional widget).

The 'E' button opens the data acquisition widget. Here, the acquisition data location, institute, person, date and a reference URL is shown and can be edited. Furthermore, the acquisition method can be changed between *simulation*, *measurement* and *unknown*. In the description field, a detailed description can be written. Note that this field is different for every selected language.

The actual data for the given data type can either be edited or visualized in the widget or, alternatively, in an excel-like widget available by the 'Excel' button (see 1.6).

For each data type exist the entries a) distance, b) theta (elevation angle) and c) angle phi (azimuth angle). These lists are hierarchically organized. Hence, all available distances are listed in the distance list. If a distance is selected, the neighboring theta list shows all available theta angles **for the selected distance**. In the next level, the

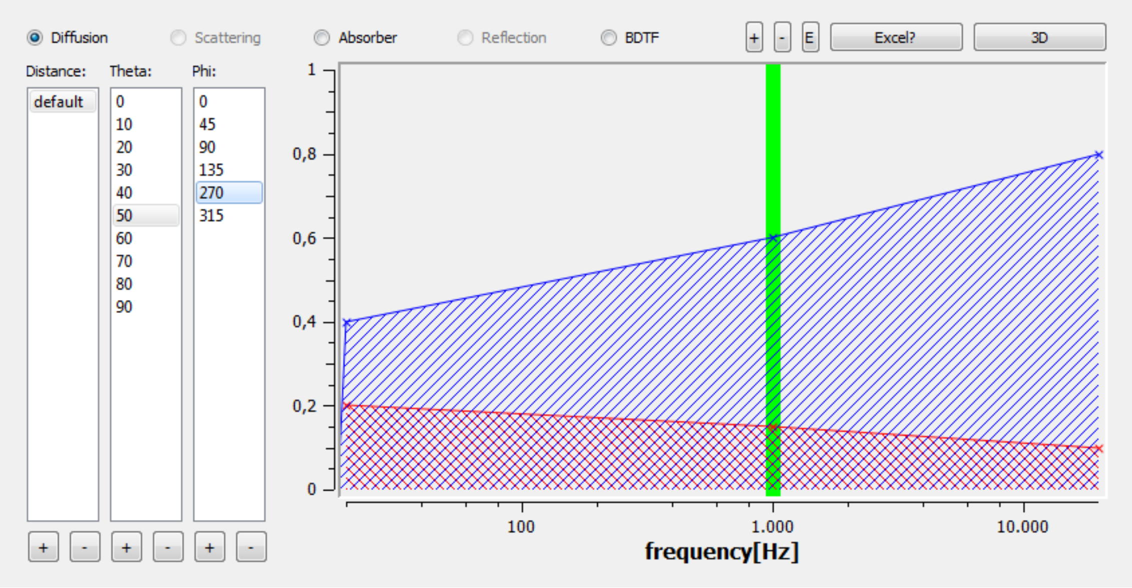


Figure 1.3: Screenshot of the data widget in DIFF mode

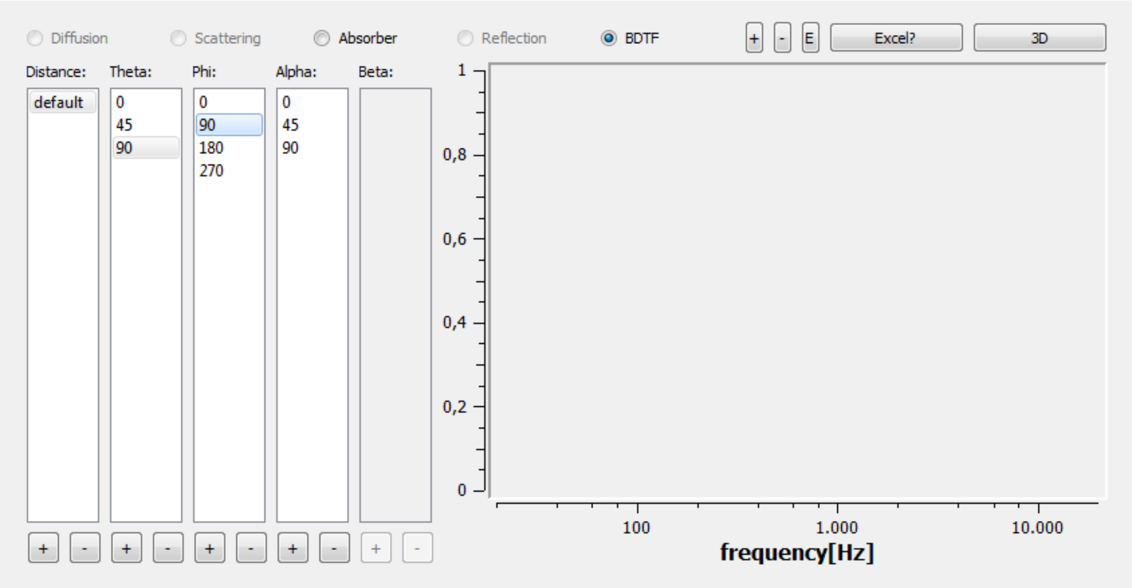


Figure 1.4: Screenshot of the data widget in BDTF mode

phi list contains all available phi angles **for the selected distance and theta**. In case of the BDTF data type, this hierarchical list schema is extended by alpha and beta angles (outgoing elevation and azimuth angle). For any of these lists, additional values can be added by pressing the '+' button and selected entries can be deleted by pressing the '-' key. If an entry is deleted, the whole subtree will be deleted. If a phi angle is selected (or a beta angle for the BDTF case), the data is plotted in the plot widget as a function of frequency.

1.5 Plotter

In the plot widget, the blue curve shows the real part of data, while the red curve belongs to the imaginary part. For non-complex values, only the blue curve is plotted as magnitude. Each data point is shown as a cross. The curve is plotted by a linear interpolation between them. At the moment, the data range is restricted to 0 to 1 and a frequency from 20 Hz to 20000 Hz. Data points that are marked by the interpolated flag are highlighted by a green bar. Each of these points can be selected by simply clicking on them (marked with yellow color).

By either double clicking on the data point or pressing the 'E' key on the keyboard, the current (yellow) data point can be edited by an upcoming window. Alternatively, the data point can be moved by the *drag-and-drop* method in a vertical scale. While dragging the data point, a cross-hair on the current value is shown. With both modification methods, however, a variation of the frequency is not possible. In addition, the selected data point can also be deleted by pressing the '-' key on the keyboard. Finally, with the

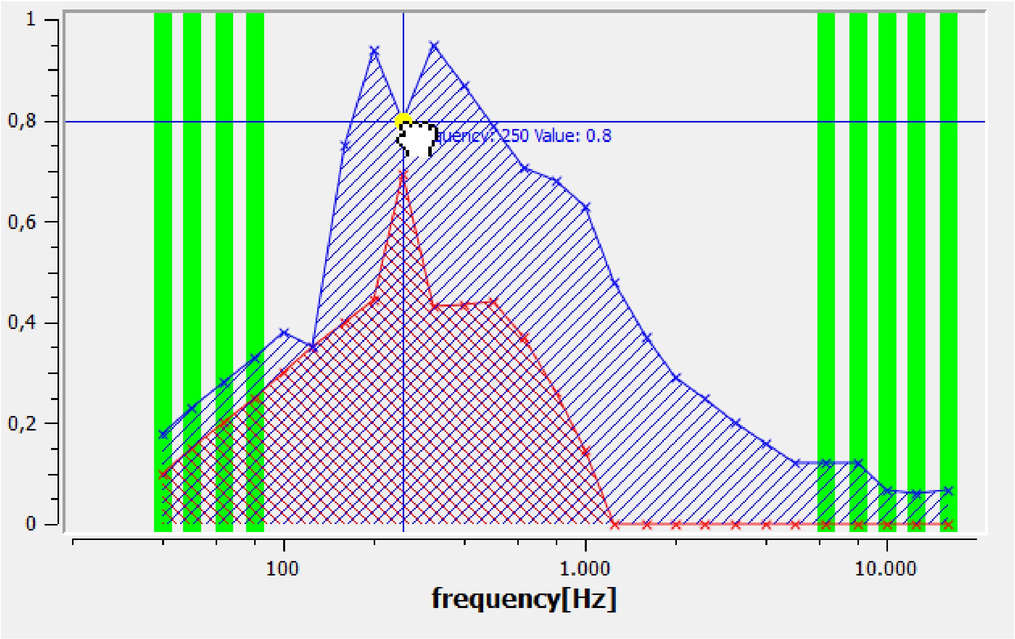
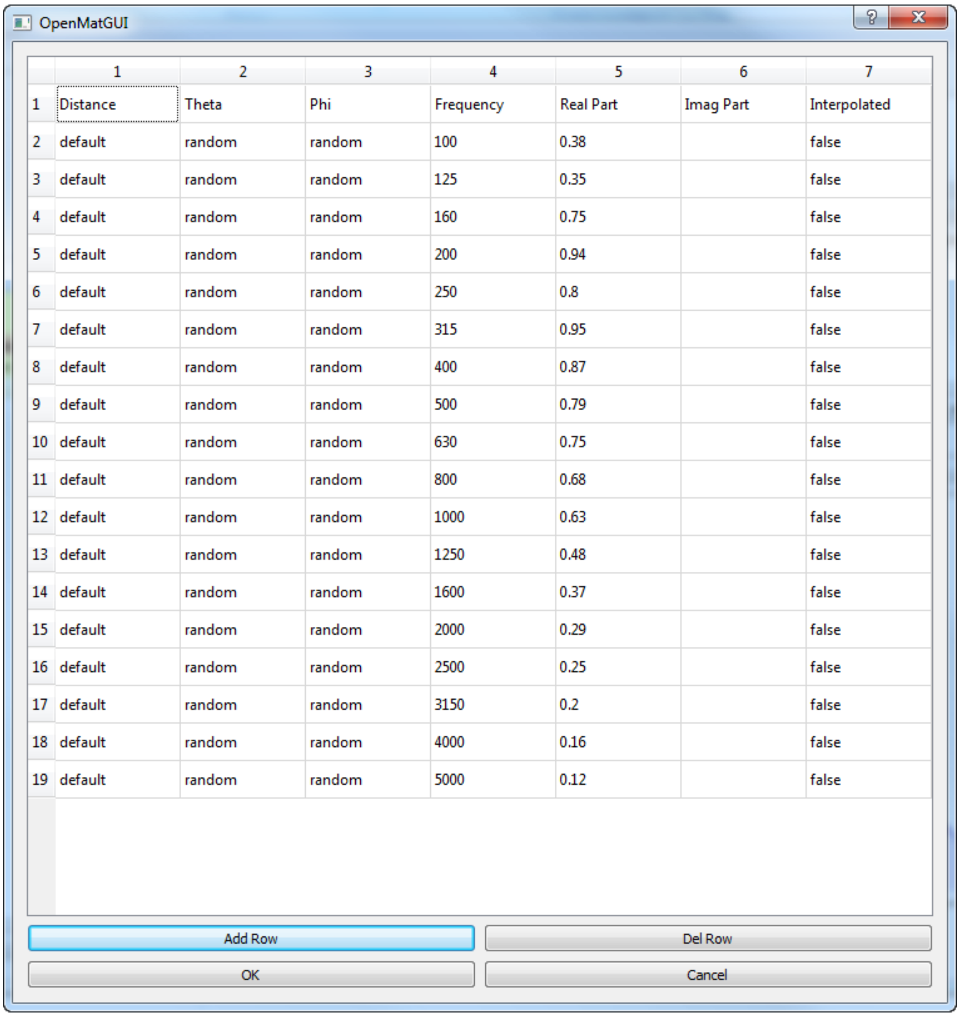


Figure 1.5: Screenshot of the data widget in BDTF mode

'+' key on the keyboard or double clicking into the plot's white region, a new value can be added, where also the frequency can be defined by the user.

1.6 Excel Widget



	1	2	3	4	5	6	7
1	Distance	Theta	Phi	Frequency	Real Part	Imag Part	Interpolated
2	default	random	random	100	0.38		false
3	default	random	random	125	0.35		false
4	default	random	random	160	0.75		false
5	default	random	random	200	0.94		false
6	default	random	random	250	0.8		false
7	default	random	random	315	0.95		false
8	default	random	random	400	0.87		false
9	default	random	random	500	0.79		false
10	default	random	random	630	0.75		false
11	default	random	random	800	0.68		false
12	default	random	random	1000	0.63		false
13	default	random	random	1250	0.48		false
14	default	random	random	1600	0.37		false
15	default	random	random	2000	0.29		false
16	default	random	random	2500	0.25		false
17	default	random	random	3150	0.2		false
18	default	random	random	4000	0.16		false
19	default	random	random	5000	0.12		false

Figure 1.6: Screenshot of the excel widget

The Excel widget lists all available data points for both the selected material and data type. In different columns, the frequency, theta and phi angle, real and imaginary part, and the state of interpolation are shown (for the BDTF data type, also alpha and beta columns are inserted). In the initial states, all rows are sorted in order of frequency and angles. This sorting does not have to be fulfilled by the user after the editing is finished - all these values will be sorted internally after closing the widget. Values can be edited by the user by double clicking on a cell. Furthermore, new cells can be added by pressing the 'add row' button, and the current row can be deleted by pressing the 'remove row' button. Multiple cells can be selected and their content can be interchanged with excel like office programs in both ways by copy and paste using 'Ctrl + C' and 'Ctrl + V' commands. After finishing the modification of the data, the new data is transferred to the database by pressing the 'OK' button, while all changes will be discarded by pressing the 'Cancel' button.

1.7 Import and Export

In order to use multiple databases for different purposes, the import/export functionality allows the exchange of material data between different databases. The import widget contains a list of all material of the current database (red, on the left-hand side) and a second, empty database on the right-hand side (green). By pressing the 'Load Import Database' button, a file selection menu allows the user to load in a second database. From this second database (green), multiple materials can be selected by holding the 'Shift' or 'Ctrl' key. The selected materials can either be *moved* or *copied* (blue) to the initial material list on the left hand side (red). Note, if materials are moved, their respective entries are also removed from the originating material list. However, the respective database, stored on the hard disk, is not modified during this process. If the databases contain mutual entries that represent the same physical material, these entries can be *merged* (blue) to a single material in the left-hand database. Here, empty entries will be overwritten with existing values of the imported material, while identical entries stay unchanged. For entries with different content, an additional widget pops up in order to allow the user to either choose the correct content or add up both entries, separated with an underscore.

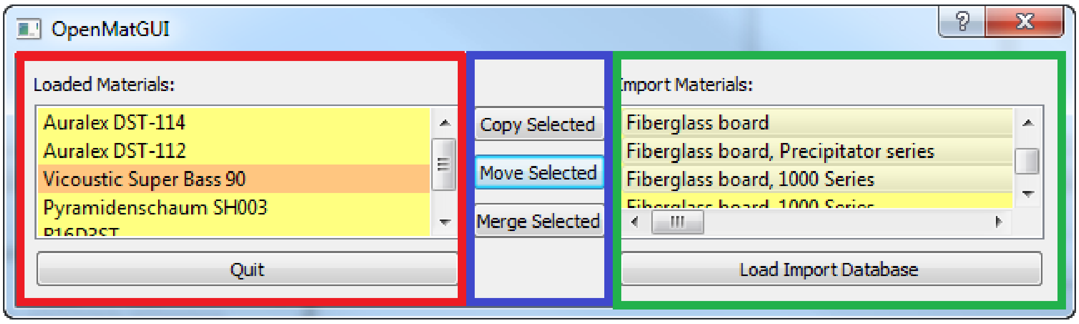


Figure 1.7: Screenshot of the material import widget

The respective export widget functions in the same manner. It shows on the left-hand side (red) the currently loaded database. As described above, multiple materials are selected by holding the 'Shift' or 'Ctrl' key. These materials can then either be *moved* or *copied* (blue) to a new database (green). The final list of materials on the right-hand side can be *saved* into a new database by pressing the 'Save Export Database' button. Materials that are moved to a new database (green) are also removed from

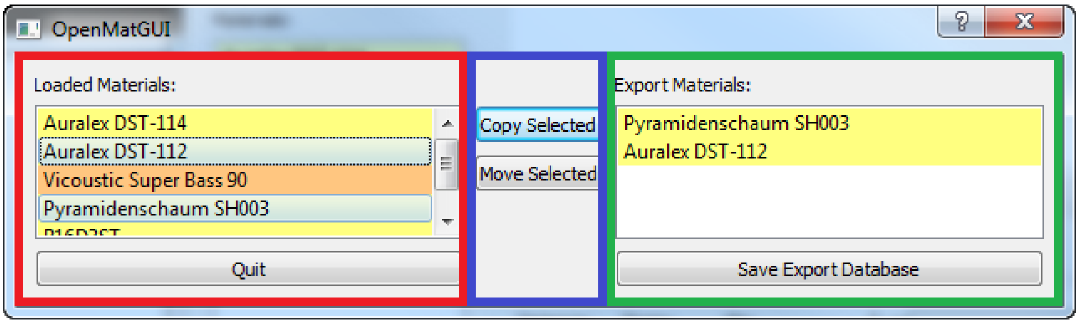


Figure 1.8: Screenshot of the material export widget

the originating file list, i.e. database (red), and are not accessible anymore unless the genuine database is loaded-in again.

In general, the move functionality has to be understood as a special feature to make the merging of database more convenient. Most likely the copy functions are sufficient in most other cases.

2 Implementation Details

2.1 XML Description

The XML schema applied in OpenMat is defined by a XML Schema Definition (XSD) file in compliance with the W3C recommendations. The XSD is – according to the W3C – a formal description pattern[1] that

- provides a list of elements and attributes in a vocabulary.
- associates types, such as integer, string, etc., or more specifically such as hatsize, sock_color, etc., with values found in the documents.
- constrains where elements and attributes can appear, and what can appear inside those elements, such as saying that a chapter title occurs inside a chapter, and that a chapter must consist of a chapter title followed by one or more paragraphs of text.
- provides documentation that is both human-readable and machine-processable.

In the following, the most important parts of the openMat schema are briefly described. Please go through the complete XSD for a comprehensive description of the openMat schema (it is self-explaining even if you know just a little bit about XML).

2.1.1 The Database Schema

The database contains xsd:elements that have a name and are of type *materialElement*. The database is allowed to be empty (minOccurs = ”0”) and there is no restriction on the maximum number of materials (maxOccurs = ”unbounded”).

```
<xsd:element          name       = ”material_database”>
  <xsd:complexType>
    <xsd:choice       maxOccurs = ”unbounded”>
      <xsd:element    name       = ”material”
                      type       = ”materialElement”
                      minOccurs  = ”0”
                      maxOccurs  = ”unbounded”/>
    </xsd:choice>
  </xsd:complexType>
</xsd:element>
```

[1] http://www.w3.org/standards/xml/schema

2.1.2 The materialElement Schema

A *materialElement* consists of up to fourteen sub-elements of different data types. Only three of these fourteen elements are mandatory, while the rest is optional. However, a material entry should contain as much information as possible.

name

Data-Type: *textElement*
Description: Name (and only the name) of the material. Multiple languages are supported.
Mandatory: Yes (English entry per definition, this is not part of the XSD schema).

description:

Data-Type: *textElement*
Description: Detailed description of the material. Multiple languages are supported.
Mandatory: Yes (English entry per definition, this is not part of the XSD schema).

type:

Data-Type: *typeElement*
Description: Each material has a specific type, either it is defined as *material* or *object*.
Mandatory: Yes.

diffData:

Data-Type: *dataElement*
Description: Contains diffusion coefficients and information on data acquisition. Angle- and distance dependent data is supported, as are complex values.
Mandatory: No.

scatData:

Data-Type: *dataElement*
Description: Contains scattering coefficients and information on data acquisition. Angle- and distance dependent data is supported, as are complex values.
Mandatory: No.

absoData:

Data-Type: *dataElement*
Description: Contains absorption coefficients and information on data acquisition. Angle- and distance dependent data is supported, as are complex values.
Mandatory: No.

reflData:
Data-Type: *dataElement*
Description: Contains reflection factors and information on data acquisition. Angle- and distance dependent data is supported, as are complex values.
Mandatory: No.

BDTFData:
Data-Type: *dataElementBD*
Description: Contains bidirectional transfer distribution function (BDTF) values and information on data acquisition. Angle- and distance dependent data is supported, as are complex values.
Mandatory: No.

URL:
Data-Type: *xsd:string*
Description: Contains one URL to further information on the material (manufacturer, etc.).
Mandatory: No.

sampleImage:
Data-Type: *xsd:string*
Description: Contains one file link to one photo (jpg or png format) of the material sample that was processed (measurement, simulation).
Mandatory: No.

CADModel:
Data-Type: *xsd:string*
Description: Contains one file link to one CAD file (dxf format) of the material. This makes sense if the material was defined as an *object*, e.g., chairs. See **type**.
Mandatory: No.

texture:
Data-Type: *fileElement*
Description: Contains one or more file links to one or more (seamless) textures of the material for the usage in 3D modelers. Since multiple file links are allowed, textures can be used that vary, e.g., in color or resolution.
Mandatory: No.

additionalFile:
Data-Type: *fileElement*
Description: Contains one or more file links to one or more additional files that relate to the material, e.g., a brochure.
Mandatory: No.

price:

Data-Type: *priceElement*

Description: Contains price information of the material. Several currencies are accepted (at the moment USD, EUR and NOK, but this can be extended easily). This information is helpful if decisions on pricing have to be made.

Mandatory: No.

XML Schema of a single *materialElement*:

```
<xsd:complexType      name = ”materialElement”>
  <xsd:sequence>
    <xsd:element      name = ”name”             type      = ”textElement”
                                                minOccurs = ”1”
                                                maxOccurs = ”unbounded”/>

    <xsd:element      name = ”description”      type      = ”textElement”
                                                minOccurs = ”1”
                                                maxOccurs = ”unbounded”/>

    <xsd:element      name = ”diffData”         type      = ”dataElement”
                                                minOccurs = ”0”
                                                maxOccurs = ”1”/>

    <xsd:element      name = ”scatData”         type      = ”dataElement”
                                                minOccurs = ”0”
                                                maxOccurs = ”1”/>

    <xsd:element      name = ”absoData”         type      = ”dataElement”
                                                minOccurs = ”0”
                                                maxOccurs = ”1”/>

    <xsd:element      name = ”reflData”         type      = ”dataElement”
                                                minOccurs = ”0”
                                                maxOccurs = ”1”/>

    <xsd:element      name = ”BDTFData”         type      = ”dataElementBD”
                                                minOccurs = ”0”
                                                maxOccurs = ”1”/>

    <xsd:element      name = ”URL”              type      = ”xsd:string”
                                                minOccurs = ”0”
                                                maxOccurs = ”1”/>

    <xsd:element      name = ”sampleImage”      type      = ”xsd:string”
                                                minOccurs = ”0”
                                                maxOccurs = ”1”/>

    <xsd:element      name = ”CADModel”         type      = ”xsd:string”
                                                minOccurs = ”0”
                                                maxOccurs = ”1”/>

    <xsd:element      name = ”texture”          type      = ”textureElement”
                                                minOccurs = ”0”
                                                maxOccurs = ”unbounded”/>

    <xsd:element      name = ”additionalFile”   type      = ”fileElement”
                                                minOccurs = ”0”
                                                maxOccurs = ”unbounded”/>

    <xsd:element      name = ”price”            type      = ”priceElement”
                                                minOccurs = ”0”
                                                maxOccurs = ”1”/>
  </xsd:sequence>
  <xsd:attribute      name = ”type”             type      = ”typeElement”
                                                use       = ”required”/>
</xsd:complexType>
```

The data type *dataElement*

contains the actual data, or say it more precisely, it is a container for the actual data. It consists of three sub-elements: *acquisition*, *distance* and *method. Acquisition* is **mandatory** and describes the method of data acquisition - this is either a detailed description of the measurements including the setup, applied DIN norms, and description of the setup, or a detailed description of the simulation methods (more details can be found under *acquiElement*). The type of the acquisition method is also **mandatory** and defined by *method*, with 0 = *measurement*, 1 = *simulation*, 2 = *unknown*, -1 = *not valid. Distance* contains **at least one** *distanceElement* – a set of data (see below) – for a specific measurement distance. This discrimination was applied to take into account both near-field and far-field measurements (far-field = -1, near-field = 0, else: distance in [cm], default = -1 (far-field)).

```
<xsd:complexType     name = ”dataElement”>
  <xsd:sequence>
    <xsd:element     name = ”acquisition”     type      = ”acquiElement”
                                              minOccurs = ”1”
                                              maxOccurs = ”1”/>
    <xsd:element     name = ”distance”        type      = ”distanceElement”
                                              minOccurs = ”1”
                                              maxOccurs = ”unbounded”/>
  </xsd:sequence>
  <xsd:attribute     name = ”method”          type      = ”xsd:integer”
                                              use       = ”required”/>
</xsd:complexType>
```

The data type *distanceElement*

contains **at least one** frequencyElement which relates to one single frequency, i.e., there are 31 frequencyElement for one-third-octave resolution. In general, the number of frequencyElements is unlimited. Distances are stored as float numbers (in [m], default = -1 (far-field)).

```
<xsd:complexType     name = ”distanceElement”>
  <xsd:sequence>
    <xsd:element     name = ”frequency”       type      = ”frequencyElement”
                                              minOccurs = ”1”
                                              maxOccurs = ”unbounded”/>
  </xsd:sequence>
  <xsd:attribute     name = ”d”               type      = ”xsd:float”
                                              default   = ”−1”/>
</xsd:complexType>
```

The data type *frequencyElement*

contains **at least one** angleElement that relates to **one** angle combination (*phi*, *theta*). The number of *angleElement*s is unlimited. Frequencies are stored as float numbers.

```
<xsd:complexType      name = ”frequencyElement”>
  <xsd:sequence>
    <xsd:element      name = ”angle”            type      = ”angleElement”
                                                minOccurs = ”1”
                                                maxOccurs = ”unbounded”/>
  </xsd:sequence>
  <xsd:attribute      name = ”f”                type      = ”xsd:float”
                                                use       = ”required”/>
</xsd:complexType>
```

The data type *angleElement*

contains the actual value (coefficient) for a specific angle combination of *phi* (azimuth) and *theta* (elevation) (default angle is 0.0° for both). These float values can be stored either as real (one entry) or complex (two entries). The value type is also indicated by the boolean flag *complex*. The boolean flag *wasInterpolated* indicates if the value was either measured/simulated or was interpolated.

```
<xsd:complexType      name = ”angleElement”>
  <xsd:sequence>
    <xsd:element      name = ”value”             type      = ”xsd:float”
                                                 minOccurs = ”1”
                                                 maxOccurs = ”2”/>
  </xsd:sequence>
  <xsd:attribute      name = ”theta”             type      = ”xsd:float”
                                                 default   = ”0.0”/>
  <xsd:attribute      name = ”phi”               type      = ”xsd:float”
                                                 default   = ”0.0”/>
  <xsd:attribute      name = ”wasInterpolated”   type      = ”xsd:boolean”
                                                 default   = ”false”/>
  <xsd:attribute      name = ”complex”           type      = ”xsd:boolean”
                                                 default   = ”false”/>
</xsd:complexType>
```

The data type *dataElementBD*

is similar to *dataElement* (see above). The only difference is that it refers to *distanceElementBD* instead of *distanceElement* since this element type is designed for **bidirectional** data.

```
<xsd:complexType      name = ”dataElementBD”>
  <xsd:sequence>
    <xsd:element      name = ”acquisition”     type      = ”acquiElement”
                                               minOccurs = ”1”
                                               maxOccurs = ”1”/>
    <xsd:element      name = ”distance”        type      = ”distanceElementBD”
                                               minOccurs = ”1”
                                               maxOccurs = ”unbounded”/>
  </xsd:sequence>
  <xsd:attribute      name = ”method”          type      = ”xsd:integer”
                                               use       = ”required”/>
</xsd:complexType>
```

The data type *distanceElementBD*

is similar to *distanceElement* (see above). The only difference is that it refers to *frequencyElementBD* instead of *frequencyElement* since this element type is designed for **bidirectional** data.

```
<xsd:complexType      name=”distanceElementBD”>
  <xsd:sequence>
    <xsd:element      name=”frequency”         type      = ”frequencyElementBD”
                                               minOccurs = ”1”
                                               maxOccurs = ”unbounded”/>
  </xsd:sequence>
  <xsd:attribute      name=”d”                 type      = ”xsd:integer”
                                               default   = ”−1”/>
</xsd:complexType>
```

The data type *frequencyElementBD*

is similar to *frequencyElement* (see above). The only difference is that it refers to *angleElementBD* instead of *angleElement* since this element type is designed for **bidirectional** data.

```
<xsd:complexType      name = ”frequencyElementBD”>
  <xsd:sequence>
    <xsd:element      name = ”angle”           type      = ”angleElementBD”
                                               minOccurs = ”1”
                                               maxOccurs = ”unbounded”/>
  </xsd:sequence>
  <xsd:attribute      name = ”f”               type      = ”xsd:float”
                                               use       = ”required”/>
</xsd:complexType>
```

The data type *angleElementBD*

contains the actual value (coefficient) for a specific bi-directional angle combination of *phi* (azimuth, incidence), *theta* (elevation, incidence), *alpha* (azimuth, reflection) and *beta* (elevation, reflection) (default is 0.0° for all four angle types). This float value can be stored either as real (one entry) or complex (two entries). The value type is also indicated by the boolean flag *complex*. The boolean flag *wasInterpolated* indicates if the value was either measured/simulated or was interpolated.

```
<xsd:complexType      name = ”angleElementBD”>
  <xsd:sequence>
    <xsd:element      name = ”value”           type      = ”xsd:float”
                                               minOccurs = ”1”
                                               maxOccurs = ”2”/>
  </xsd:sequence>
  <xsd:attribute      name = ”theta”           type      = ”xsd:float”
                                               default   = ”0.0”/>
  <xsd:attribute      name = ”phi”             type      = ”xsd:float”
                                               default   = ”0.0”/>
  <xsd:attribute      name = ”alpha”           type      = ”xsd:float”
                                               default   = ”0.0”/>
  <xsd:attribute      name = ”beta”            type      = ”xsd:float”
                                               default   = ”0.0”/>
  <xsd:attribute      name = ”wasInterpolated” type      = ”xsd:boolean”
                                               default   = ”false”/>
  <xsd:attribute      name = ”complex”         type      = ”xsd:boolean”
                                               use       = ”required”/>
</xsd:complexType>
```

The data type *acquiElement*

contains detailed information on the actual data acquisition. It consists of the following elements: *location*, *institute*, *person*, *date*, *description* and *URL*. The elements *location*, *institute* (or company) and *person* (or group) give details on the performer of the measurements/simulations. *Date* contains the date of the measurements/simulations (format after DIN 8601 yyyy-mm-dd). *URL* gives a link to a website with further information (if available). *Description* contains detailed information on how these measurements/simulations were carried out (DIN, simulation tools, etc.). The *description* element supports multiple languages. None of these elements is mandatory, but it is highly recommended to give here as much information as possible.

```
<xsd:complexType       name = ”acquiElement”>
  <xsd:sequence>
    <xsd:element       name = ”location”         type      = ”xsd:string”
                                                 minOccurs = ”0”
                                                 maxOccurs = ”1”/>
    <xsd:element       name = ”institute”        type      = ”xsd:string”
                                                 minOccurs = ”0”
                                                 maxOccurs = ”1”/>
    <xsd:element       name = ”person”           type      = ”xsd:string”
                                                 minOccurs = ”0”
                                                 maxOccurs = ”1”/>
    <xsd:element       name = ”date”             type      = ”xsd:date”
                                                 minOccurs = ”0”
                                                 maxOccurs = ”1”/>
    <xsd:element       name = ”description”      type      = ”textElement”
                                                 minOccurs = ”0”
                                                 maxOccurs = ”unbounded”/>
    <xsd:element       name = ”url”              type      = ”xsd:string”
                                                 minOccurs = ”0”
                                                 maxOccurs = ”1”/>
  </xsd:sequence>
</xsd:complexType>
```

The data type *typeElement*

describes the *type* of the material, which is either a *material (mat)* or an *object*. This discrimination is important since material data don’t necessarily refer to a ‘flat’ surface but to whole structures, in particular, when thinking of diffusion and scattering coefficients.

```
<xsd:simpleType         name  = ”typeElement”>
  <xsd:restriction      base  = ”xsd:string”>
    <xsd:enumeration    value = ”object”/>
    <xsd:enumeration    value = ”mat”/>
  </xsd:restriction>
</xsd:simpleType>
```

The data type *textElement*

is a data container that contains text that is linked to a language tag. The default language is *English*.

```
<xsd:complexType      name = ”textElement”>
  <xsd:simpleContent>
    <xsd:extension    base = ”xsd:string”>
      <xsd:attribute  name = ”lang”           type     = ”langElement”
                                              default  = ”en”/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
```

The data type *langElement*

contains all available language tags in order to limit the number of input languages. This, however, can be extended anytime.

```
<xsd:simpleType       name  = ”langElement”>
  <xsd:restriction    base  = ”xsd:string”>
    <xsd:enumeration  value = ”en”/>    // English
    <xsd:enumeration  value = ”da”/>    // Danish
    <xsd:enumeration  value = ”de”/>    // German
    <xsd:enumeration  value = ”es”/>    // Spanish
    <xsd:enumeration  value = ”fi”/>    // Finish
    <xsd:enumeration  value = ”fr”/>    // French
    <xsd:enumeration  value = ”it”/>    // Italian
    <xsd:enumeration  value = ”nl”/>    // Dutch
    <xsd:enumeration  value = ”no”/>    // Norwegian
    <xsd:enumeration  value = ”sv”/>    // Swedish
    <xsd:enumeration  value = ”pt”/>    // Portuguese
  </xsd:restriction>
</xsd:simpleType>
```

The data type *priceElement*

stores pricing information on the material. Price information is linked to a currency tag. In addition, information on date (price quoted is correct only as of this date) and the source of pricing information, typically a URL. Date format after ISO 8601 (yyyy-mm-dd).

```
<xsd:complexType      name = ”priceElement”>
  <xsd:simpleContent>
    <xsd:extension    base = ”xsd:float”>
      <xsd:attribute  name = ”currency”       type     = ”currencyElement”
                                              use      = ”required”/>
      <xsd:attribute  name = ”date”           type     = ”xsd:date”
                                              use      = ”required”/>
      <xsd:attribute  name = ”source”         type     = ”xsd:string”
                                              use      = ”required”/>
    </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
```

The data type *currencyElement*

contains all available currency tags in order to limit the number of input currencies. This, however, can be extended anytime.

```
<xsd:simpleType         name  = ”currencyElement”>
  <xsd:restriction      base  = ”xsd:string”>
    <xsd:enumeration    value = ”EUR”/>   // EURO
    <xsd:enumeration    value = ”USD”/>   // US Dollar
    <xsd:enumeration    value = ”NOK”/>   // Norwegian Crowns
  </xsd:restriction>
</xsd:simpleType>
```

The data type *fileElement*

is a data container that contains a file path that is linked to a user-defined tag.

```
<xsd:complexType        name = ”fileElement”>
  <xsd:simpleContent>
    <xsd:extension      base = ”xsd:string”>
      <xsd:attribute    name = ”keyTag”            type       = ”xsd:string”
                                                   use        = ”required”/>
  </xsd:extension>
  </xsd:simpleContent>
</xsd:complexType>
```

2.2 openMat XML Example

```
<?xml version="1.0" encoding="UTF-8" ?>
<material_database xmlns="http://www.openmat.info" xmlns:xsi="http://www.w3.org/2001/
    XMLSchema-instance" xsi:schemaLocation="http://www.openmat.info␣MaterialDatabase.
    xsd">
  <material type="mat">
      <name lang="en">Mineral Wool</name>
      <name lang="de">Faserabsorber</name>
    <description lang="en"> not flamable, 5 mm thick, white color</description>
    <description lang="de"> nicht brennbar, 5mm dick, weiss</description>
    <diffData method="1">
      <acquisition>
        <location>North Pole</location>
        <institute>Institute for Measurement Techniques</institute>
        <person>Dr. Santa Clause</person>
        <date>2011-12-24</date>
      </acquisition>
      <distance>
        <frequency f="31.5" > <angle>
                              <value>0.25   </value>
                              <value>0.5    </value>
                              </angle> </frequency>
        <frequency f="63"   > <angle>
                              <value>0.25   </value>
                              <value>0.5    </value>
                              </angle> </frequency>
        <frequency f="125"  > <angle>
                              <value>0.25   </value>
                              <value>0.5    </value>
                              </angle></frequency>
        <frequency f="250"  > <angle wasInterpolated="true">
                              <value>0.59   </value>
                              <value>0.5    </value>
                              </angle> </frequency>
        <frequency f="500"  > <angle>
                              <value>0.81   </value>
                              <value>0.5    </value>
                              </angle> </frequency>
        <frequency f="1000" > <angle>
                              <value>0.64   </value>
                              <value>0.5    </value>
                              </angle> </frequency>
        <frequency f="2000" > <angle>
                              <value>0.26   </value>
                              <value>0.5    </value>
                              </angle> </frequency>
        <frequency f="4000" > <angle>
                              <value>0.17   </value>
                              <value>0.5    </value>
                              </angle> </frequency>
        <frequency f="8000" > <angle wasInterpolated="true">
                              <value>0.17   </value>
                              <value>0.5    </value>
                              </angle> </frequency>
        <frequency f="16000"> <angle wasInterpolated="true">
                              <value>0.17   </value>
                              <value>0.5    </value>
                              </angle> </frequency>
        </distance>
    </diffData>
    <absoData method="1">
      <acquisition>
        <location>North Pole</location>
        <institute>Institute for Measurement Techniques</institute>
        <person>Dr. Santa Clause</person>
        <date>2011−12−24</date>
      </acquisition>
      <distance>
        <frequency f=”31.5” > <angle>
                              <value>0.25    </value>
                              </angle> </frequency>
        <frequency f=”63”    > <angle>
                              <value>0.25    </value>
                              </angle> </frequency>
        <frequency f=”125”   > <angle>
                              <value>0.25    </value>
                              </angle> </frequency>
        <frequency f=”250”   > <angle>
                              <value>0.59    </value>
                              </angle> </frequency>
        <frequency f=”500”   > <angle>
                              <value>0.81    </value>
                              </angle> </frequency>
        <frequency f=”1000”  > <angle>
                              <value>0.64    </value>
                              </angle> </frequency>
        <frequency f=”2000”  > <angle>
                              <value>0.26    </value>
                              </angle> </frequency>
        <frequency f=”4000”  > <angle>
                              <value>0.17    </value>
                              </angle> </frequency>
        <frequency f=”8000”  > <angle>
                              <value>0.17    </value>
                              </angle> </frequency>
        <frequency f=”16000”> <angle>
                              <value>0.17    </value>
                              </angle> </frequency>
      </distance>
    </absoData>
    <url>           http://de.wikipedia.org/wiki/Mineralwolle/url>
    <sampleImage>   ../data/SantasMineralWool.png</sampleImage>
    <cadModel>      ../data/santa.dxf</cadModel>
    <texture        color    = ”red”>../data/blau.png</texture>
    <additionalfile keyTag   = ”Rudolf”>../data/reindeer.zip</additionalfile>
    <price          currency = ”EUR”
                    source   = ”www.santa.com”
                    date     = ”2012−12−24”>12.00</price>
    </material>
</material_database>
```