http://www.soda-is.com SoDa Service providing data and other information in Solar Radiation

Project name:
Integration and exploitation of networked Solar radiation Databases for environment monitoring
Project acronym:
SoDa
Contract number:
IST-1999-12245

Deliverable title:
How to declare a resource in the SoDa Intelligent System – Draft 2
Deliverable number:
none
Type of deliverable:
Report
Security:

Author:
ENSMP (M. Albuisson, L. Wald) and JRC (E. van der Goot)
Contributors:

Contractual date of delivery to the Commission:
none
Actual date of delivery to the Commission:
April 2002

Table of Contents

Introduction

The SoDa Intelligent System (IS) is a service facilitator. The aim of this document is to provide guidelines to declare resources (databases, applications) within the SoDa IS.

Reading the documentation written by the JRC on the prototype of the SoDa service and especially "**Specification of SoDa XML**", "**Overview of the Prototype, Architecture and Design**" and "**SoDa 2nd prototype description**" would be of high benefit for a complete understanding.

The SoDa IS is an Internet software system that aims to transparently connect and present resources that are of a distributed nature. The problems associated with such a system can be roughly divided in three broad categories:

- Interoperability, i.e. protocol and dataformats
- Service description and location
- Service presentation and invocation

The SoDa IS is designed in such a way that a service provider can publish details of their algorithm to the prototype and become operational within the overall network. Example services are not raw data from databases but rather end product services. These end user services require the execution of an algorithm on demand where a user may input a number of specific parameters. These parameters can be of several types:

- Geographical selection
- Temporal selection
- Physical parameter(s) specifying a calculation

The SoDa IS is based around a web-server that services incoming requests, builds the necessary GUI for the user, invokes the service(s) and presents the results to the user. The current working prototype is based on proven and available technologies and products (HTTP, HGSS, XML).

The software system behind this prototype service consists of three parts. One part deals with bringing together distributed heterogeneous resources and the other part deals with the Web-based access to these combined resources. The last part concerns the presentation of the data to the user. The technology that underlies the user interface is well understood, namely HTML.

Principle of invocation of a service

Each SoDa service is described by a SoDa XML service descriptor. This descriptor contains a call descriptor (i.e. location and interface definition), one or more input parameters and one or more output parameters. It also contains some general information about the resource (or service), i.e. the resource provider and a link to further information about the resource.

Once the user has selected a SoDa service, the IS exploits the XML service description and generates a user interface that contains all the necessary components to obtain the user input.The GUI collects the necessary parameters. The result is an HTML form that will contain all necessary parameters. Upon submit, the necessary resources are invoked. The resource invocation currently consists of an HTTP GET request for the URL as specified in the resource description, augmented with the necessary parameters.

The invocation of the resources currently relies on a simple HTTP based protocol, where the parameter values are specified as (name, value) pairs. The result of the resource invocation is passed back to the

controlling process in the form of an XML document, embedding the results in SoDa XML specific elements.

The resulting XML is either the required output, or forms the basis for the next resource invocation. If the service is a compound service, the service execution module, possibly using this information as new input, will call the next resource in the chain.

The data presentation module is a lightweight postprocessor. Depending on user input, the resulting output from the service invocation, which is in fact an XML document, will be transformed using XSLT. Using available toolkits, the SoDa IS performs stylesheet based transforms for a variety of output formats and devices.

Description of a service

The service descriptor is a text file that describes the resource, using the specifications of the SoDa XML. This file is then uploaded in the SoDa IS (Service Description and Location) and permits the access to the resource.

First example

Hereafter is the XML service descriptor for the resource "MARS Global Radiation Database Point Timeseries".

```
<?xml version="1.0"?>
<resource>
    <info>
        <name>MARS Global Radiation Database Point Timeseries</name> This is the name of the resource that appears in the Service Description
        <producer>JRC Ispra, MARS-SAI and RIT-ISIS</producer>
        <url>http://response.jrc.it:8080/SoDaSample/PointRad</url> This the URL that is invoked on submission of the request. This URL is usually located at the provider's premises
    </info>
    <input parameter="point" type="geopoint" dimension="single"> Here, one geographical site should be defined ...
        <bbox>27.707,-12.942,71.067,65.984</bbox> ... within this bounding box...
        <show>Select the point of interest</show> ... and the following query text
    </input>
    <input parameter="period" type="timeperiod" dimension="single"> A time-series is requested (timeperiod) ...
        <timerange>1975-01-01,2001-03-20</timerange> ... within this time range...
        <show>Enter the period (first day, last day)</show>  ... and the following query text
    </input>
```

<input parameter="maxrows" type="numeric" dimension="single"> This question is to limit the size of the stream to be ...

<show>Enter the maximum number of return values</show> ... returned from the service.

</input>

</resource>

An example of the resulting URL is

http://response.jrc.it:8080/SoDaSample/PointRad?point=45,7.78&period=1994-07-15,1994-08-20

if the user selected the geographical site of latitude: 45 N and longitude: 7.78 E and the time period ranging from July 15 1994 to August 20 1994.

In this example,

- "http://response.jrc.it:8080/SoDaSample/" is the name of the server,
- "PointRad" is the name of the script,
- "point=45,7.78&period=1994-07-15,1994-08-20" is the data stream, using the variable names defined in the XML: "point" and "period".

Second example

Hereafter is the XML service descriptor for the resource "Simulation of the irradiation components under clear sky".

<?xml version="1.0"?>

<resource>

<info>

<name>Simulation of the irradiation components under clear sky</name> This is the name of the resource that appears in the Service Description

<producer>Ecole des Mines de Paris</producer>

<producerurl>http://www.helioclim.net</producerurl>

<url>http://www.helioclim.net/com/clearsky.php</url> This the URL that is invoked on submission of the request. This URL is usually located at the provider's premises.

</info>

<input parameter="alt" type="numeric" dimension="single" default="0.0">

<show>Enter a value in meters for the altitude. 0 means database retrieval</show>

</input>

<input parameter="tl" type="numeric" dimension="single" default="3.0">

<show>Enter a value [0..10] for the Linke turbidity coefficient</show>

</input>

```
        <input parameter="tilt" type="numeric" dimension="single" default="0.0">
                <show>Enter a value [0..90] for the tilt angle</show>
        </input>
        <input parameter="azim" type="numeric" dimension="single" default="0.0">
                <show>Enter a value [-180..180] for the azimuth angle</show>
        </input>
        <input parameter="al" type="numeric" dimension="single" default="0.2">
                <show>Enter a value [0..1] for the ground albedo</show>
        </input>
        <input parameter="date" type="timepoint" dimension="single">
                <show>Select the date you want to simulate</show>
        </input>
        <input parameter="latlon" type="geopoint" dimension="single">
                <show>Select on the map the site of the simulation</show>
        </input>
</resource>
```

An example of the resulting URL is

http://www.helioclim.net/com/clearsky.php?alt=0.0&tl=3.2&tilt=45.0&azim=0.0&al=0.2&date=2002-04-22&latlon=56.3,-10.7

if the user selected the geographical site of latitude: 56.3 N and longitude: 10.7 W and the day 22 April 2002.

In this example,

- "http:// www.helioclim.net" is the name of the server,
- "com" is the path to the script,
- "clearsky.php" is the name of the script,
- "alt=0.0&tl=3.2&tilt=45.0&azim=0.0&al=0.2&date=2002-04-22&latlon=56.3,-10.7" is the data stream, using the variable names defined in the XML: "alt", "tl", "tilt", "azim", "date" and "latlon".

What should do the application on the provider side? Sending back to the SoDa IS

The application should preferably be mounted on the same machine that the server of the provider. If the resource is a database, the server should be ODBC compliant.

The invoked application is launched and decodes the variables that are embedded in the encoded URL. The variables should have the same name, of course. E.g., if the XML description uses "point" (example #1), respectively "latlon" (example #2), to denote the geographical co-ordinates, the application should use "point", respectively "latlon".

The application may invoke itself other URLs, for example, the terrain elevation.

Because it was called by the SoDa IS, the application automatically returns outputs to the SoDa IS as data stream. Once the computation performed, the application prints the results in the standard output (e.g., the "printf" function in C or PHP language). The print should be made as to generate a SoDa XML data-stream. This stream is sent back to the SoDa IS.

First example of a SoDa XML output

Hereafter is an example of a SoDa XML stream, where there is no table.

```
<?xml version="1.0" encoding="UTF-8" ?>
<resource>
    <info> There are given general information about the service
        <name>Surface temperature</name> Provides the label of the service
        <producer>Hungarian Meteorogical Service</producer> Here is the producer of the information ...
        <producerurl>http://www.met.hu</producerurl> ... and its URL
    </info>
    <output parameter="lat" type="numeric" dimension="single"> The variable "lat" is displayed there
        <show>Site latitude</show> ... with its label ...
        <value>46.06</value> ... and its value.
    </output>
    <output parameter="lon" type="numeric" dimension="single">
        <show>Site longitude</show> <value>17.03</value>
    </output>
    <output parameter="vegi" type="numeric" dimension="single">
        <show>Vegetation index</show>
        <value>0.63</value>
    </output>
</resource>
```

Second example of a SoDa XML output with tables

Here is reported the XML output of the resource "Simulation of the irradiation components under clear sky". The XML comprises the description of a table.

The invocation of the service was made by the following URL:

http://www.helioclim.net/com/clearsky.php?alt=0.0&tl=3.2&tilt=45.0&azim=0.0&al=0.2&date=2002-04-22&latlon=56.3,-10.7

The application exploded the variable "latlon" into two variables "lat" and "lon".

```
<?xml version=\"1.0\" encoding=\"UTF-8\" ?>
<resource>
        <info>
                <name>Simulation of the irradiation components under clear sky</name>
                <producer>Ecole des Mines de Paris</producer>
                <producerurl>http://www.helioclim.net</producerurl>
        </info>
        <output parameter="lat" type="numeric" dimension="single">
                <show>Site latitude</show>
                <value>56.30</value>
        </output>
        <output parameter="lon" type="numeric" dimension="single">
                <show>Site longitude</show>
                <value>-10.70</value>
        </output>
        <output parameter="alt" type="numeric" dimension="single"> Display variable "alt"
                <show>Site altitude (meters)</show> ... with its label ...
                <value>0</value> ... and its value.
        </output>
        <output parameter="tl" type="numeric" dimension="single"> Display variable "tl"
                <show>Linke turbidity factor</show> ... with its label ..
                <value>3.2</value> ... and its value.
        </output>
        <output parameter="tilt" type="numeric" dimension="single">
                <show>Tilt angle</show>
                <value>45</value>
        </output>
        <output parameter="azim" type="numeric" dimension="single">
                <show>Azimuth angle</show>
                <value>0</value>
```

</output>

<output parameter="al" type="numeric" dimension="single">

<show>Albedo of the ground</show>

<value>0.20</value>

</output>

<output parameter="date" type="string" dimension="single">

<show>Date of simulation</show>

<value>2002-04-22</value>

</output>

<output parameter="clearsky" type="string" dimension="table"> The table containing the clear sky values will be displayed ...

<show>Irradiation components under clear sky (Wh/m2)</show> ... with this label

In this example, the table is made of 6 columns and 25 rows of values, excluding the labels of each column.

<headrow> Prepare the labelling of the columns. The number of values given below plus one determines the number of columns. Here 6 columns are defined by this combination of tags <headrow>, <value>, </value> and </headrow>.

<value>Beam</value> Label of the second column

<value>Diffuse</value>

<value>Reflected</value>

<value>Global</value>

<value>Global TOA (horizontal)</value>

</headrow>

<row label="[00..24]"> Label of the first row (first column). Here, [00..24] means the duration of the day

<value>6180</value> Value of the daily beam irradiation

<value>1490</value> Value of the daily diffuse irradiation

<value>180</value>

<value>7850</value>

<value>8767</value>

<row label="[00..01]"> Label of the second row (first column). Here, [00..01] means the hour 0-1h.

<value>0</value> Value of the beam irradiation for the hour 0-1

<value>0</value>

<value>0</value>

<value>0</value>

<value>0</value>

<row label="[00..02]">
<value>0</value>
<value>0</value>
<value>0</value>
<value>0</value>
<value>0</value>

etc.

<row label="[04..05)">
<value>0</value>
<value>1</value>
<value>0</value>
<value>1</value>
<value>6</value>
<row label="[05..06)">
<value>2</value>
<value>15</value>
<value>2</value>
<value>19</value>
<value>143</value>

etc.

<row label="[23..24]">
<value>0</value>
<value>0</value>
<value>0</value>
<value>0</value>
<value>0</value>

The number of rows of the table is defined by the combination of the tags <output>, <row> and </output>. There is no need to provide the size in row and column.

</output>

</resource>

In the case of numerous outputs, it is recommended to design a lightweight library performing the XML ouput. Here is given the example of a part of the code written in PHP for the clear sky resource.

Main

```
        out_info("Simulation of the irradiation components under clear sky");
        out_numeric("lat","Site latitude",$lat,"%.2f");
        out_numeric("lon","Site longitude",$lon,"%.2f");
        out_numeric("alt","Site altitude (meters)",$alt,"%d");
        out_numeric("tl","Linke turbidity factor",$tl,"%.1f");
        out_numeric("tilt","Tilt angle",$tilt,"%d");
        out_numeric("azim","Azimuth angle",$azim,"%d");
        out_numeric("al","Albedo of the ground",$al,"%.2f");
        out_string("date","Date of simulation",$date);
```

Library

```
function out_info($label)
{
        print("<info>\n");
        print("<name>$label</name>\n");
        print("<producer>Ecole des Mines de Paris</producer>\n");
        print("<producerurl>http://www.helioclim.net</producerurl>\n");
        print("</info>\n");
}

function out_numeric($name,$label,$value,$fmt)
{
        print("<output parameter=\"$name\" type=\"numeric\" dimension=\"single\">\n");
        print("<show>$label</show>\n");
        $sval = sprintf($fmt,$value);
        print("<value>$sval</value>\n");
        print("</output>\n");
}

function out_string($name,$label,$value)
{
        print("<output parameter=\"$name\" type=\"string\" dimension=\"single\">\n");
        print("<show>$label</show>\n");
        print("<value>$value</value>\n");
```

```
        print("</output>\n");
}
```

Chaining several resources

The SoDa IS and the SoDa XML permit to offer a piping of resources via the CompoundService.

```
<?xml version="1.0" encoding="UTF-8"?>
<service>
        <!--
    The service level info section. The <name> will appear as the title on the output
    -->
        <info>
                <name>Solar Home System</name>
                <xslt>http://prime.jrc.it/SoDa/data/solarhome.xslt</xslt>
        </info>
        <!--
    The first resource to call. The ID can be any string, although here the call order is being used. The
resources HAVE to appear in the correct order,
    the software does not  perform a dependency analysis. The only relevant element of the <info> section
is the <url>. Also the <show> and
    <bbox> elements are not being used by the service builder. It should be obvious that the only valid
<source> elements in the first resource
    to call can only be refences to the service level input (i.e. no id attribute) or a constant (e.g. the
maxrows parameter)
    -->
        <resource id="0">
                <info>
                        <name>INPUT parameter conversion for Solar Home System</name>
                        <producer>EHF, Dept. of Physics, University of Oldenburg</producer>
                        <producerurl>www.physik.uni-oldenburg.de/ehf/</producerurl>
                        <url>http://soda.ehf.uni-oldenburg.de/cgi-bin/inputshs</url>
                </info>
                <input parameter="year" type="numeric"  dimension="single">
            <show>Year 1997-2001</show>
```

```
                    <source parameter="year"/>
                </input>
                <input parameter="load0_5" type="string" dimension="single">
                    <source parameter="load0-5"/>
                </input>
            <input parameter="load6_11" type="string" dimension="single">
                    <source parameter="load6-11"/>
                </input>
            <input parameter="load12_17" type="string" dimension="single">
                    <source parameter="load12-17"/>
                </input>
            <input parameter="load18_23" type="string" dimension="single">
                    <source parameter="load18-23"/>
                </input>
                <output parameter="period" type="timeperiod" dimension="single"/>
            <output parameter="load profile" type="numeric" dimension="table">
                <show>Daily Load Profile</show>
                    <headrow>
                        <value>Hour</value>
                        <value>Load (W) </value>
                    </headrow>
                </output>
            </resource>
        <!--
        The second resource to call. The ID can be any string, although here the call order is being used.
<source> elements can refer to both user input and
        previous resource output parameters
        -->
        <resource id="1">
                <info>
                    <name>MARS Global Radiation Database Point Timeseries</name>
                    <producer>JRC Ispra, MARS-SAI and RIT-ISIS</producer>
                    <url>http://response.jrc.it:8080/SoDaSample/PointRad</url>
                </info>
                <input parameter="point" type="geopoint" dimension="single">
                    <source parameter="latlon"/>
```

```
                </input>
                <input parameter="period" type="timeperiod" dimension="single">
                        <source id="0" parameter="period"/>
                </input>
                <input parameter="maxrows" type="numeric" dimension="single">
                        <source const="366"/>
                </input>
                <output parameter="timestamp" type="string" dimension="single"/>
                <output parameter="point" type="geopoint" dimension="single"/>
                <output parameter="period" type="timeperiod" dimension="single"/>
        <output parameter="grid" type="geopoint" dimension="single"/>
                <output parameter="distance" type="numeric" dimension="single"/>
                <output parameter="query results" type="numeric" dimension="table">
                        <headrow>
                                <value>Day</value>
                                <value>Global Radiation (Wh/m**2) </value>
                        </headrow>
                </output>
        </resource>
        <!--
    The third resource to call. The ID can be any string, although here the call order is being used.
<source> elements can refer to both user input and
    previous resource output parameters
    -->
        <resource id="2">
                <info>
                        <name>Performance of a Solar Home System given daily global horizontal
irradiance values</name>
                        <producer>EHF, Dept. of Physics, University of Oldenburg</producer>
                        <producerurl>www.physik.uni-oldenburg.de/ehf/</producerurl>
                        <url>http://soda.ehf.uni-oldenburg.de/cgi-bin/shs</url>
                </info>
                <input parameter="latlon" type="geopoint" dimension="single">
                        <show>Enter the geografic coordinates Lat N, Lon (E+,W-
)(xx.xxx,xxx.xxx)</show>
                        <source parameter="latlon"/>
```

```
        </input>
        <input parameter="tilt" type="numeric" dimension="single" default="0">
                <show>Tilt angle of module [degrees]</show>
                <source parameter="tilt"/>
        </input>
        <input parameter="azim" type="numeric" dimension="single" default="0">
                <show>Azimuth angle of module [degrees]</show>
                <source parameter="azim"/>
        </input>
        <input parameter="albedo" type="numeric" dimension="single" default="0.0">
                <show>Albedo of ground [0.00..1.00]</show>
                <source parameter="albedo"/>
        </input>
<input parameter="isoc" type="numeric" dimension="single" default="90">
                <show>Initial state of charge of battery
                [20..100 percent]</show>
                <source parameter="isoc"/>
        </input>
        <input parameter="period" type="timeperiod" dimension="single">
                <show>Enter the period (first day, last day)</show>
                <source id="1" parameter="period"/>
        </input>
        <input parameter="radiation" type="numeric" dimension="table">
                <show>Enter daily sums of global radiation (xxxx,xxxx,xxxx,...)</show>
                <source id="1" parameter="query results[][1]"/>
        </input>
<input parameter="load" type="numeric" dimension="table">
                <show>Enter 24 values for daily load profile (xxxx,xxxx,xxxx,...)</show>
                <source id="0" parameter="load profile[][1]"/>
        </input>
        <output parameter="loss of load probability[][][][]" type="numeric" dimension="table">
                <show>Loss of load probability</show>
    <headrow>
                        <value>Time</value>
                        <value>Load (W)</value>
            <value>Loss of load (hours)</value>
```

```
                <value>Loss of load (%)</value>
                        </headrow>
                </output>
                <output parameter="state of charge distribution" type="numeric" dimension="table">
                        <show>State of charge distribution for requested period</show>
                        <headrow>
                                <value>State of charge</value>
                                <value>Hours within period</value>
                        </headrow>
                </output>
                <output parameter="irradiance distribution" type="numeric" dimension="table">
                        <show>Irradiance distribution for requested period</show>
                        <headrow>
                                <value>Hourly irradiance on module(W/m**2)</value>
                                <value>Hours within period</value>
                        </headrow>
                </output>
        </resource>
        <!--
    The service level input section. These parameters will drive the generation of the user interface. Note
that these elements
    are namespace qualified.
    -->
        <input parameter="year" type="numeric" dimension="single">
                <show>Choose a year</show>
        <list>
            <option>1997</option>
            <option>1998</option>
            <option>1999</option>
            <option>2000</option>
            <option>2001</option>
        </list>
        </input>
        <input parameter="latlon" type="geopoint" dimension="single">
                <show>Select your site on the map</show>
                <bbox>28,-13,71,66</bbox>
```

```
        </input>
        <input parameter="tilt" type="numeric" dimension="single" default="0">
                <show>Tilt angle of photovoltaic modules [degrees]</show>
        </input>
        <input parameter="azim" type="numeric" dimension="single" default="0">
                <show>Azimuth angle of photovoltaic modules [degrees]</show>
        </input>
        <input parameter="albedo" type="numeric" dimension="single" default="0.0">
                <show>Albedo of ground [0.00..1.00]</show>
        </input>
    <input parameter="isoc" type="numeric" dimension="single" default="90">
                <show>Initial state of charge of battery [20..100 percent]</show>
        </input>
    <input parameter="load0-5" type="string" dimension="single" default="0,0,0,0,0,0" >
            <show>Electric load [W] for hours 0 - 5 </show>
        </input>
    <input parameter="load6-11" type="string" dimension="single" default="0,0,0,0,0,0">
            <show>Electric load [W] for hours 6 - 11</show>
        </input>
    <input parameter="load12-17" type="string" dimension="single" default="0,0,0,0,0,0">
            <show>Electric load [W] for hours 12 - 17</show>
        </input>
    <input parameter="load18-23" type="string" dimension="single" default="0,0,0,0,0,0">
            <show>Electric load [W] for hours 18 - 23</show>
        </input>
        <!--
    The service level output section. These parameters will determine the final output. Note that the output
can be constructed from
    user input and any resource output.
    -->
    <output parameter="errormsg0" type="string" dimension="single">
        <source id="0" parameter="errormsg"/>
    </output>
    <output parameter="errormsg2" type="string" dimension="single">
        <source id="2" parameter="errormsg"/>
    </output>
```

```
<output parameter="location" type="geopoint" dimension="single">
        <show>The selected location (lat, lon)</show>
        <source parameter="latlon"/>
</output>
<output parameter="location with data" type="geopoint" dimension="single">
        <show>Simulation performed for (lat, lon)</show>
        <source id="1" parameter="grid"/>
</output>
<output parameter="distance" type="numeric" dimension="single">
        <show>Distance between requested location and location
        in database (km)</show>
        <source id="1" parameter="distance"/>
</output>
<output parameter="period" type="timeperiod" dimension="single">
        <show>Period simulated</show>
        <source id="1" parameter="period"/>
</output>
<output parameter="loss of load probability[][][][]" type="numeric" dimension="table">
        <show>Loss of load probability for requested period</show>
<headrow>
            <value>Time</value>
            <value>Load (W)</value>
    <value>Loss of load (hours)</value>
    <value>Loss of load (%)</value>
        </headrow>
<source id="2" parameter="loss of load probability[][][][]"/>
</output>
<output parameter="state of charge distribution" type="numeric" dimension="table">
        <show>State of charge distribution for requested period</show>
        <headrow>
                <value>State of charge(Ah)</value>
                <value>Hours within period</value>
        </headrow>
        <source id="2" parameter="state of charge distribution[][]"/>
</output>
<output parameter="irradiance distribution" type="numeric" dimension="table">
```

```
                <show>Irradiance distribution for requested period</show>
                <headrow>
                        <value>Hourly irradiance on module(W/m**2)</value>
                        <value>Hours within period</value>
                </headrow>
                <source id="2" parameter="irradiance distribution[][]"/>
        </output>
        <output parameter="radiationOut" type="numeric" dimension="table">
                <show>MARS Radiation values used:</show>
                <headrow>
                        <value>Date</value>
                        <value>Radiation(Wh/m**2) </value>
                </headrow>
                <source id="1" parameter="query results[][]"/>
        </output>
</service>
```

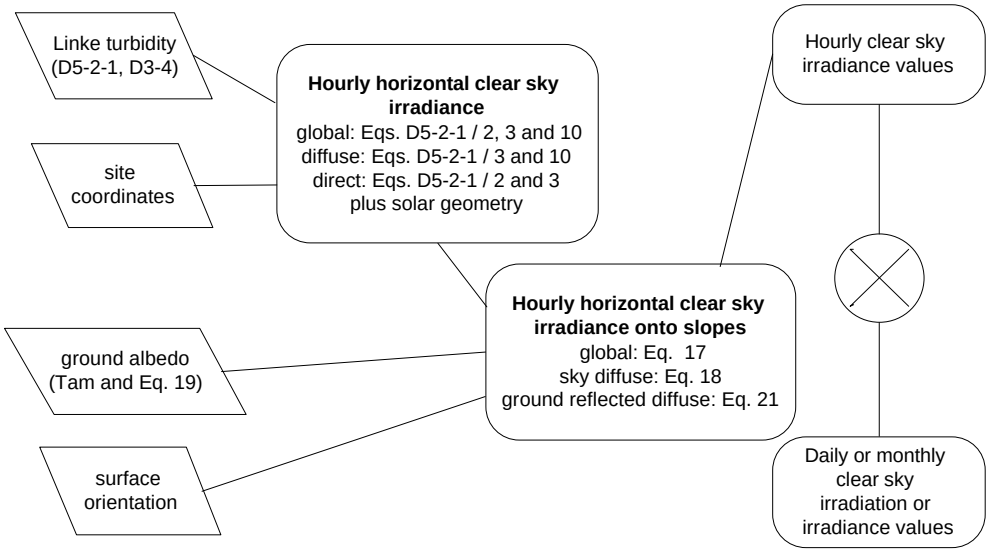

Fig. 3: Flowchart of SODA algorithmic chain no. 1

```
<?xml version="1.0" encoding="UTF-8"?>
<service>
	<!--
	The service level info section. The <name> will appear as the title on the output
	-->
	<info>
		<name>Chain 1: Clear sky global radiation</name>
	</info>
	<!--
	The first resource to call.
	-->
	<resource id="0">
		<info>
			<name>Linke turbidity</name>
			<producer>METEOTEST</producer>
			<producerurl>http://www.meteotest.ch/</producerurl>
			<url>http://stratus.meteotest.ch/soda_tl/soda_tl.asp</url>
		</info>
		<input parameter="latlon" type="geopoint" dimension="single">
			<source parameter="latlon"/>
		</input>
		<input parameter="month" type="numeric" dimension="single">
			<source const="0"/>
		</input>
		<input parameter="type" type="string" dimension="single">
			<source const="TL"/>
		</input>
		<input parameter="altitude" type="numeric" dimension="single">
			<source parameter="altitude"/>
		</input>
		<output parameter="latlon" type="geopoint"/>
		<output parameter="altitude" type="numeric" dimension="single"/>
		<output parameter="TL" type="numeric" dimension="table"/>
	</resource>
	<!--
	The second resource to call.
	-->
	<resource id="1">
		<info>
			<name>SODA chain of algorithms</name>
			<producer>meteotest</producer>
			<producerurl>http://www.meteotest.ch/</producerurl>
			<url>http://stratus.meteotest.ch/soda_ch/soda_ch.asp</url>
		</info>
		<input parameter="latlon" type="geopoint" dimension="single">
			<source parameter="latlon"/>
		</input>
		<input parameter="month" type="numeric" dimension="single">
			<source parameter="month"/>
		</input>
		<input parameter="altitude" type="numeric" dimension="single">
			<source parameter="altitude"/>
		</input>
		<input parameter="method" type="string" dimension="single">
			<source const="clearsky"/>
		</input>
		<input parameter="resolution" type="string" dimension="single">
			<source parameter="resolution"/>
		</input>
```

```
<input parameter="Inclination" type="numeric" dimension="single">
        <source parameter="Inclination"/>
</input>
```

Fig. 16a: Service description of SODA algorithmic chain no. 1. 2 resources are accessed: first the atmospheric resource to get TL values, second a part of the soda chains resource.

```
                <input parameter="Azimuth" type="numeric" dimension="single">
                        <source parameter="Azimuth"/>
                </input>
                <input parameter="Radunit" type="string" dimension="single">
                        <source parameter="Radunit"/>
                </input>
                <input parameter="groundtype" type="string" dimension="single">
                        <source parameter="groundtype"/>
                </input>
                <input parameter="tl" type="numeric" dimension="single">
                        <source id="0" parameter="TL[][1]"/>
                </input>
                <output parameter="latlon" type="geopoint"/>
                <output parameter="altitude" type="numeric" dimension="single"/>
                <output parameter="Gc" type="numeric" dimension="table"/>
                <output parameter="Gcd" type="numeric" dimension="table"/>
                <output parameter="Gcm" type="numeric" dimension="table"/>
</resource>
<!--
The service level input section. These parameters will drive the generation of the user interface.
-->
<input parameter="latlon" type="geopoint" dimension="single">
        <show>Select Latitude/Longitude</show>
        <bbox> -90, -180, 90, 180</bbox>
</input>
<input parameter="altitude" type="numeric" dimension="single">
        <show>altitude</show>
</input>
<input parameter="Inclination" type="numeric" dimension="single">
        <show>Inclination</show>
</input>
<input parameter="Azimuth" type="numeric" dimension="single">
        <show>Azimuth (90Â°=E, 270Â°=W)</show>
</input>
<input parameter="Radunit" type="string" dimension="single">
        <show>Radiation unit (daily and monthly values)</show>
        <list>
                <option default="true">irradiation</option>
                <option>irradiance</option>
        </list>
</input>
<input parameter ="resolution" type="string" dimension="single">
        <show>Select time resolution</show>
        <list>
                <option default="true">monthly</option>
                <option>daily</option>
                <option>hourly</option>
        </list>
</input>
<input parameter="month" type="numeric" dimension="single">
        <show>month (0=all, 1=Jan, 13=year)</show>
        <list>
                <option>0</option>
                <option>1</option>
                <option>2</option>
                <option>3</option>
                <option>4</option>
                <option>5</option>
                <option>6</option>
                <option>7</option>
```

Fig. 16b: Continuation: Service description of SODA algorithmic chain no. 1. 2 resources are accessed: first the atmospheric resource to get TL values, second a part of the soda chains resource.

```
                        <option>8</option>
                        <option>9</option>
                        <option>10</option>
                        <option>11</option>
                        <option>12</option>
                        <option>13</option>
                </list>
        </input>
        <input parameter="groundtype" type="string" dimension="single">
                <show>type of ground (for albedo)</show>
                <list>
                        <option default="true">grass</option>
                        <option>sand</option>
                        <option>street</option>
                        <option>concrete</option>
                        <option>snow</option>
                </list>
        </input>
        <output parameter="latlon" type="geopoint"  dimension="single">
                <source parameter="latlon"/>
        </output>
        <output parameter="altitude" type="numeric" dimension="single">
                <source parameter="altitude"/>
        </output>
        <output parameter="Gc" type="numeric" dimension="table">
                <show>=Gc: global clear sky radiation</show>
                <headrow>
                        <value>year</value>
                        <value>month</value>
                        <value>day</value>
                        <value>hour</value>
                        <value>Gc</value>
                </headrow>
                <source id="1" parameter="Gc[][]"/>
        </output>
        <output parameter="Gcd" type="numeric" dimension="table">
                <show>=Gcd: global clear sky radiation daily</show>
                <headrow>
                        <value>year</value>
                        <value>month</value>
                        <value>day</value>
                        <value>Gcd</value>
                </headrow>
                <source id="1" parameter="Gcd[][]"/>
        </output>
        <output parameter="Gcm" type="numeric" dimension="table">
                <show>=Gcm: global clear sky radiation monthly</show>
                <headrow>
                        <value>year</value>
                        <value>month</value>
                        <value>Gcm</value>
                </headrow>
                <source id="1" parameter="Gcm[][]"/>
        </output>
</service>
```

Fig. 16c: Continuation: Service description of SODA algorithmic chain no. 1. 2 resources are accessed: first the atmospheric resource to get TL values, second a part of the soda chains resource.